Arthur Kharkivskyi

Senior Java & AI-Native Software Engineer

(writing also Python/ReactJS/Flutter/Swift pet-projects)

13 years of commercial software engineering

Our superheroes — Ukrainian engineers →

arthurdnu@gmail.com
+380 67 690 690 0

Telegram @MeetUpps Viber WhatsApp LinkedIn

Location: Dnipro, Ukraine (relocation or business trips — after the war)

Languages: Ukrainian (native), English — Upper-Intermediate (B2)

Compensation: 6000 USD (from my previous job, open to discussion)

01 ●●●

Spec-Driven Agentic Workflows & Reusable Skills

I guide agentic AI systems through versioned engineering rules, clear technical specifications, compact session summaries, and reusable skill playbooks rather than ad-hoc chat prompts.

02 ●●●

Custom MCP Tools & Organized Tool Catalogs

I build and organize domain-scoped Model Context Protocol (MCP) tool catalogs that connect agents cleanly to file operations, file comparison and diffing, database business data search, data validation, report generation, and workflow automation.

03 ●●●

Cost-Saving & Fast Voice Control Architecture

I designed a cost-saving, low-latency `Voice Control` pipeline for hands-free desktop and web workflows—routing microphone capture through local speech recognition (`Whisper`), instant bilingual dictionary and vector-embedding matching, and local model fallback before dispatching `MCP` commands over native OS accessibility interfaces.

04 ●●●

Multi-Agent Orchestration & Independent Code Review

I split complex coding, testing, and refactoring tasks across specialized parallel worker agents under a coordinator, pairing them with independent review agents that audit diffs and reach consensus before changes are accepted.

05 ●●●

Tensormesh & Redis Semantic Code Indexing

I accelerate repository navigation and structured code lookup using `Tensormesh` and `Redis` caching with vector embeddings, retrieving targeted outlines and code symbols without flooding model context windows.

06 ●●●

Hybrid Local & Cloud Model Routing

I route frequent intent parsing, semantic embeddings, grammar checking, classification, and routine checks to local open-weight models while reserving frontier cloud models for architecture design and complex reasoning.

07 ●●●

Test-Driven AI Coding & Visual Verification

I enforce test-first development across AI-assisted changes—writing failing automated tests before implementation, running regression checks, and capturing headless browser screenshots to verify layout and UI copy.

08 ●●●

Governed Terminal Execution & Safe Guardrails

I keep automated terminal command execution, container operations, and repository hygiene predictable by running agent actions through policy-governed proxies and explicit approval gates on sensitive changes.

09 ●●●

Context Compression & Cost-Aware Monitoring

I control token usage and runtime cost through structured cache reads, compact command output filtering, automated log redirection, and event-driven background status tracking instead of repetitive polling loops.

10 ●●●

Human-in-the-Loop Architecture & Core Ownership

I use AI agents as force multipliers for daily coding, testing, code review, and automated engineering workflows while keeping system architecture, difficult debugging, and high-risk production logic under direct senior engineering control.

Technology stack

Languages & platforms — Java Ecosystem

- Java 25, Java 21, Java 17, Java 8, Java 6
- Groovy, SQL, bash automation scripts

Java 25 | Java 21 | Java 17 | Groovy | SQL

Frameworks & backend — 8+ years Spring Boot

- Spring Boot, Spring Core, Spring Data JPA, Spring Security, Hibernate
- RESTful Web Services, pipelined API architecture

Spring Boot | Spring Security | Hibernate | REST API

Multithreading & concurrency — High-load optimization

- thread pools, concurrent collections, atomic references, transaction locking strategies
- race condition resolution when calculating coordinates for millions of vehicles
- non-blocking batch processing on tables with 100 million records
- Redis distributed locks for critical sections

Concurrency | Redis Locks | Batch Processing | Zero Downtime

Databases & persistence — Relational & Key-Value

- PostgreSQL (indexing, query optimization)
- MySQL, SybaseSQL
- Redis (caching and distributed resource locking)

PostgreSQL | Redis | MySQL | SybaseSQL

Containerization & CI/CD — DevOps

- Docker (writing Dockerfiles, multi-stage image optimization)
- GitLab CI, Jenkins

Docker | GitLab CI | Jenkins

Data pipelines & messaging — Kafka, Pulsar, Airflow

- Apache Kafka, Apache Pulsar
- Apache Airflow: Directed Acyclic Graph (DAG) workflows for collision-free synchronization of 3–6M records

Apache Kafka | Apache Pulsar | Apache Airflow | DAG

Testing & code quality — TDD, Volume Tests, PoC

- JUnit 5 (Jupiter), Mockito, Postman, Spock/Groovy
- TDD (test-first), volume testing, PoC benchmarks

JUnit 5 | Mockito | TDD | Volume Tests

Skills & engineering competencies

Architectural competencies — System Design

- microservices design and breaking down business requirements
- Event-Driven Architecture and Event Sourcing for asynchronous event processing
- high-load ETL pipeline design between different databases
- high availability and zero-downtime operations

Transaction reliability & security — Fintech & PCI DSS 3.0

- idempotency for business operations: unique idempotency keys, database constraints
- idempotent webhook handlers: protection against redelivery and duplicate charges
- regular automated reconciliation of internal records with payment providers and banks
- PCI DSS 3.0 compliance: 512-bit card hash recalculation and secure archiving of 100M records

Technical leadership & engineering practices — Mentorship & Delivery

- Java Chapter Lead role (6 months): mentoring engineers, architectural guidance, PR checks
- regular detailed code reviews, supporting clean code and SOLID principles
- Continuous Delivery and daily production deployments with Trunk-Based Development
- monthly planning and backlog management (2–4 weeks ahead) to adapt quickly to business needs

Commercial work experience

Ardas • Senior Java Software Engineer / Java Chapter Lead

09.2020 – Present

domain: high-load logistics platform, real-time fleet tracking.

- developed microservices architecture using Java 17 / Java 21, Spring Boot, PostgreSQL, Docker, Apache Pulsar, and Apache Airflow;
- optimized parallel data processing: built parallel DAG workflows in Apache Airflow for collision-free and deadlock-free synchronization of 3–6 million records;
- resolved race conditions in coordinate calculations for millions of vehicles using lock-free concurrency structures and Redis distributed locks;
- worked as Java Chapter Lead for 6 months: mentored engineers, reviewed architecture decisions, did code reviews, and planned technical tasks.

Java 21 | Spring Boot | PostgreSQL | Apache Pulsar | Apache Airflow | Redis | Docker

EPAM Systems • Senior Java Software Engineer

10.2018 – 09.2020 (2 years)

domain: high-load e-commerce platform for European clients (consumer online services).

- developed backend services using Java 8 / 11, Spring Boot, PostgreSQL, and Apache Kafka;
- built order processing pipelines and integrated external APIs with strict low-latency requirements;
- covered business logic with integration and unit tests (JUnit, Mockito), took part in technical design reviews.

Java 11 | Spring Boot | PostgreSQL | Apache Kafka | JUnit

Luxoft • Senior Java Software Engineer

04.2016 – 08.2018 (2 years 5 mos)

domain: Big Data and banking analytics for global investment bank Deutsche Bank.

- organized and processed large financial datasets using the Apache Hadoop ecosystem;
- developed and optimized data transformations using Java, XSLT, Groovy, and Spock;
- ensured accurate regulatory reporting reconciliation and bank transaction stream checks.

Java | Hadoop | Groovy | Spock | XSLT

Scandinavian Financial Platform • Java Software Engineer

04.2015 – 03.2016 (1 year)

domain: payment system with strong data security and audit compliance.

- developed non-blocking batch processing jobs with controlled intervals to safely clean up 100M+ records with zero downtime;
- updated card hashes to 512-bit to meet PCI DSS 3.0 encryption standards;
- kept a high-security payment gateway running reliably with high availability.

Java | PCI DSS 3.0 | Batch Processing | SHA-512

PrivatBank • Java Software Engineer

04.02.2013 – 04.2015 (2 years 2 mos)

domain: banking daily operations automation, SWIFT payments, regular payments, and card products.

- developed new features and optimizations across SWIFT payment processing, recurring transfers, and interbank transactions;
- developed pipelined services for the credit card department and card products in a cross-functional team;
- created APIs and multithreaded processing interfaces for bank payment processing and validation.

Java 6 | SybaseSQL | SWIFT | Regular Payments | Swing

Education

Oles Honchar Dnipro National University

2006 – 2011

Dnipropetrovsk National University (DNU)

- degree: Master of Science (MSc)
- major: mechanical metal processing (Mechanical Metal Processing)