# Oleg Dmytriienko

ARCHITECT · AGENTIC & DISTRIBUTED SYSTEMS

20 YEARS OF EXPERIENCE

// Location: Osaka, Japan · Remote
// Focus: Agentic & Distributed Systems
// Akka SDK · Effect TS · Fintech

uakennysoft@gmail.com
linkedin ↗ · github ↗

## CORE EXPERTISE

### Angular

Full ecosystem expert (TypeScript, NgRx, RxJS)

### Spring Boot

Backend services, REST APIs, microservices (Java, Hibernate, JPA)

## LANGUAGES PROFICIENCY

Java · JavaScript · TypeScript · Effect TS · Scala (FP · ZIO)

## ARCHITECTURE PATTERNS

Outbox Pattern · Saga Pattern · State Machines / Workflow Orchestration · CQRS · Event Sourcing · Microservices · Hexagonal Architecture · Clean Architecture · DDD · Event-Driven · Serverless · Layered (N-tier) · Actor Model

## CURRENTLY BUILDING WITH

### Spring Boot

Production backends on Java 21 — REST & reactive APIs (WebFlux), Spring Data JPA, Spring Security (OAuth2 / JWT), Spring Cloud microservices, Kafka, PostgreSQL, Flyway, Testcontainers, observability with Actuator, Micrometer and OpenTelemetry, Docker & Kubernetes

### Effect TS

Type-safe TypeScript end to end — typed errors, dependency injection with Layers, structured concurrency with fibers, Schema validation, Streams, retries & scheduling, resource safety, and built-in OpenTelemetry tracing

### Akka SDK (Java)

The Agentic Systems Platform — distributed, event-driven services at scale on the JVM I know best (Lightbend certified)

*"Architect and hands-on engineer with 20 years across frontend and backend. AI agents are distributed systems: they hold state, call unreliable tools, run for minutes or days and must survive crashes. That's the problem space I've worked in for two decades: event sourcing, sagas, durable workflows, delivery guarantees. Today I build agentic systems on both sides: Akka SDK on the JVM (Lightbend certified) and Effect TS in TypeScript. I take a product from idea to production and own it there, using AI to move fast without trading away reliability or maintainability."*

**MENTORSHIP**

Passionate mentor with a track record of leveling up engineering teams.

## ENGINEERING PRINCIPLES

- **Reliability over cleverness**
- **Maintainability as a first-class feature**
- **Event-driven consistency**
- **Functional purity where it counts**

## EMPLOYMENT HISTORY

### Shift4 (formerly Credorax, Finaro)

SENIOR FULLSTACK ENGINEER — 2021 – Present

- Reverse-engineered undocumented legacy applications with zero knowledge transfer, then rearchitected them around NgRx and RxJS — making implicit state explicit and eliminating a class of hard-to-reproduce state-related bugs.
- Refactored the codebase to a reactive, modular architecture, making state transitions explicit and traceable — which simplified feature work and made regressions easier to isolate.
- Designed and built the Billing domain frontend from the ground up, establishing the architecture, component library, and data flow patterns that the team continues to build on today.

**Scala · FP**
Functional programming and reactive patterns — Scala, ZIO, and the actor model (Coursera, Udemy, Rock the JVM certified)

- Expanded scope to fullstack, delivering end-to-end features across Spring Boot microservices and Angular — shipping complete features without waiting on separate backend availability.

`Angular` `TypeScript` `NgRx` `RxJS` `Reactive Programming` `State Management` `REST APIs` `HTML5` `SCSS` `Spring Boot` `Java` `Hibernate` `JPA` `Microservices` `Docker` `CI/CD` `Git`

## BrainRocket (formerly Soft2Bet)

FRONTEND ARCHITECT

2020 – 2021

- Designed and implemented a framework for rapid casino integrations and seamless addition of gaming features into the betting flow.
- Defined and delivered a scalable solution architecture based on headless components and custom directives, abstracting complex state management and backend communication.
- Acted as Technical Lead for a team of two senior engineers, guiding architecture decisions and ensuring high development standards.
- Enabled developers to focus on UI/UX and feature delivery by encapsulating core logic within reusable building blocks.
- Reduced dependency on highly specialized engineers by enabling HTML/CSS-focused developers to contribute effectively with minimal programming knowledge.
- Conducted knowledge transfer sessions and established best practices that accelerated team onboarding and got new contributors productive quickly.

`Angular` `TypeScript` `RxJS` `NgRx` `Component Architecture` `Headless UI` `State Management` `REST APIs` `HTML5` `SCSS` `Agile` `Architecture Design`

## Thomson Reuters / Refinitiv (via EPAM Systems)

SENIOR SOFTWARE ENGINEER

2019 – 2020

- Diagnosed and resolved critical Angular performance bottlenecks in complex table views within the first sprint — introducing NgRx state management and targeted rendering optimizations that restored smooth interaction on large data tables.
- Improved overall application responsiveness and scalability through structured state handling and change detection optimizations.
- Designed and implemented a sequential recursive algorithm to enable isomorphic data transformation between Excel (XLS) files and Angular Forms.

`Angular` `TypeScript` `NgRx` `RxJS` `Performance Optimization` `Change Detection` `State Management` `Algorithms` `Data Transformation` `Excel (XLS)` `Reactive Forms`

## Slated

SENIOR SOFTWARE ENGINEER · B2B CONTRACT

Dec 2018 – Aug 2020

- Built the Angular frontend for Slated — a Hollywood-based platform and the world's leading marketplace for packaging and financing independent films — delivering a polished, professional UI for filmmakers, talent, crew, and investors.
- Designed and implemented the Messaging domain from the ground up: a role-based, multi-account system enabling users to communicate as themselves, as a named producer, or on behalf of a project — reflecting how dealmaking in the film industry actually works.

- Managed complex multi-identity session state with NgRx and Firebase as the reactive data layer — real-time updates flowed through the store, keeping messaging state consistent across producer, project, and personal identities within a single session.

`Angular` `TypeScript` `NgRx` `RxJS` `Firebase` `Material Design` `Messaging` `Role-Based Architecture` `Multi-Account` `State Management` `REST APIs` `HTML5` `SCSS`

### 28 Stone `client: Tullet Prebon / TP ICAP`

SENIOR FULLSTACK ENGINEER → FRONTEND LEAD

`Apr 2016 – Dec 2018`

- Joined as a Senior Fullstack Engineer (Java/GWT), then took ownership of the entire frontend for a greenfield project built for Tullet Prebon — a leading global interdealer broker in the TP ICAP group, facilitating wholesale trading across fixed income, rates, equities, energy, and commodities markets.
- Designed and implemented a fully reactive trading application frontend: WebSocket-driven real-time market data, live order book state, and a composable strategy framework supporting basic call/put options through to advanced structures — butterfly, bespoke, and custom multi-leg strategies. Adopted Angular 2 from beta and deployed via OpenFin for professional desktop trading.
- Contributed to backend development using Spring Integration patterns and Solace messaging broker, gaining full-stack visibility across the event-driven trading pipeline and enabling tighter coordination between frontend state and backend message flows.

`Angular` `TypeScript` `RxJS` `Java` `GWT` `Spring Integration` `Solace` `WebSocket` `OpenFin` `Real-Time` `Financial Markets` `Event-Driven` `HTML5` `SCSS` `REST APIs`

### C.T.Co `client: Swiss Re`

SENIOR FULLSTACK SOFTWARE ENGINEER

`Mar 2014 – Mar 2016`

- Designed and implemented a re-insurance portal for Swiss Re — one of the world's largest reinsurers — end-to-end: backend in Java with Spring Framework and Spring Integration flows, frontend in GWT (Java-to-JavaScript compiler and UI framework).
- Engineered a resumable web-based file uploader capable of handling gigabytes of data and recovering seamlessly after a browser refresh. At a time when the HTML file API lacked low-level access, built an ActionScript plugin to handle file chunking and heavy client-side hashing operations — solving a problem pure JavaScript simply could not.
- Earned an internal 'Exceptional' engineer rating — the highest performance classification at C.T.Co — recognised for technical quality, ownership across the full stack, and delivering solutions to problems that had no off-the-shelf answer.
- Conducted internal GWT training courses for engineers on the team — covering architecture fundamentals, data binding patterns, and the nuances of compiling Java to JavaScript at production scale.

`Java` `Spring Framework` `Spring Integration` `GWT` `ActionScript` `Re-Insurance` `File Upload` `Resumable Upload` `Event-Driven` `Portal Development` `REST APIs` `Fullstack`

### Amber Games

SOFTWARE ENGINEER

Apr 2013 – Apr 2014

- Part of the core team that built Strategoria — a real-time strategy browser game in the spirit of StarCraft — entirely from scratch. No frameworks, no libraries, no scaffolding. Every layer hand-crafted in plain Java: an in-memory database, a clustering and sharding solution, and a full dependency injection framework built from first principles. Demos are still live on YouTube.
- Integrated LuaJ as a dynamic scripting engine, enabling game logic and unit behaviours to be authored and hot-swapped at runtime without recompilation — cleanly separating engine concerns from game design in a system with zero framework support.
- Operated deep in the fundamentals: demanding multithreading, custom data structures tuned for real-time simulation, algorithm-heavy game state management, and applied mathematics for game mechanics — solving problems at a level where off-the-shelf solutions simply do not exist.

Java · LuaJ · Multithreading · Algorithms · Data Structures · Game Development · Real-Time Systems · In-Memory Database · Clustering · Sharding · Dependency Injection · Low-Level Engineering · Math

### Playtech

SOFTWARE ENGINEER

Feb 2012 – Jan 2013

- Contributed to a team at Playtech — one of the world's largest gambling software companies — building a framework for rapid gambling application development, enabling consistent game delivery across platforms with reduced per-integration overhead.
- Built a plugin system enabling a single shared portal to host independently deployable game modules — each game loaded on demand with a defined communication contract between the portal shell and the game runtime. Conceptually equivalent to what Module Federation later formalised: runtime composition of independently built artifacts over a shared interface.
- Deep R&D into cross-platform development: adopted PlayN (Google's cross-platform game framework) to author a single Java codebase that compiled to JavaScript via GWT for web, and ran natively on Android and iOS — a single logic layer shared across all three platforms without duplication.

Java · GWT · PlayN · Android · iOS · Cross-Platform · Gambling · Framework Development · R&D · JavaScript

### Luxoft

client: Deutsche Bank ODC

SOFTWARE ENGINEER

Jun 2010 – Feb 2012

- Worked on a Portfolio Management application for Deutsche Bank through Luxoft's Offshore Development Center — at a time before frontend/backend separation was standard, owning the full stack: Spring Framework for business logic, MyBatis and SQL stored procedures for data access, and GWT for the UI.
- On a second portal project, built interactive data presentation widgets (portlets) using the Flex framework — delivering charting components and visualisations that surfaced financial data from multiple departments in a unified banking interface.
- Conducted internal GWT training courses for colleagues — sharing practical knowledge of the Java-to-JavaScript compilation model, component architecture, and UI patterns that the team relied on daily.

Java · Spring Framework · GWT · MyBatis · SQL · Stored Procedures · Flex · Portlets · Charting · Banking · Portfolio Management · Enterprise

### DioSoft `client: State Street Bank`

SOFTWARE ENGINEER

Nov 2009 – Jun 2010

- Implemented a Rich Internet Application (RIA) for State Street Bank — one of the world's largest custody and asset servicing banks — through DioSoft, using Java EE (J2EE) and Oracle WebLogic Application Server.
- Delivered interactive data views for custody and fund administration workflows — presenting portfolio positions, transaction status, and asset data across accounts in a structured, client-facing interface.

Java · J2EE · WebLogic · RIA · Banking · Enterprise

### EPAM Systems `client: B&Q`

JUNIOR → SOFTWARE ENGINEER

Sep 2006 – Nov 2009

- Joined as a Junior Software Engineer, working as part of a team delivering e-commerce features for B&Q — the UK's largest home improvement retailer — on the ATG platform (now Oracle ATG Web Commerce, part of Oracle Commerce).
- Gained hands-on experience across the full software development lifecycle — requirements, design, implementation, testing, and production support — in a large-scale, real-traffic e-commerce environment.

Java · ATG · Oracle Commerce · E-Commerce · J2EE · Fullstack

## EDUCATION

### National Technical University of Ukraine 'Kyiv Polytechnic Institute'

BACHELOR'S DEGREE · COMPUTER SYSTEMS, AUTOMATION AND CONTROL

Mar 2007

## CERTIFICATIONS (SINCE 2020)

*Note: Courses taken prior to 2020 are considered outdated and are omitted.*

| | | |
|---|---|---|
| 2020 | **Functional programming Haskell**<br>STEPIK | VERIFY_CERT |
| 2020 | **Functional Programming Principles in Scala**<br>COURSERA | VERIFY_CERT |
| 2020 | **Advanced Scala And Functional Programming**<br>UDEMY | VERIFY_CERT |
| 2021 | **Supercharge Scala Future \| FP-Tower**<br>UDEMY | VERIFY_CERT |
| 2022 | **Akka Essentials & Persistence**<br>ROCK THE JVM | |
| 2023 | **ZIO 2**<br>ROCK THE JVM | |

| | | |
|---|---|---|
| 2023-24 | **Japanese Proficiency (A1 & A2, ongoing)**<br>BUSUU | VERIFY_CERT |
| 2025 | **Reactive Architecture: Introduction to Reactive Systems**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Reactive Architecture: Domain Driven Design**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Reactive Architecture: Reactive Microservices**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Reactive Architecture: Building Scalable Systems**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Reactive Architecture: Distributed Messaging Patterns**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Reactive Architecture: CQRS and Event Sourcing**<br>LIGHTBEND | VERIFY_CERT |
| 2025 | **Akka Cluster - Sharding (Scala)**<br>LIGHTBEND | VERIFY_CERT |