Blajinschi Dan

dan.blajinschi8@gmail.com

Technical Skills

Distributed Systems Engineer. I design **microservice** backends (**gRPC**, **Kafka/RabbitMQ**, **PostgreSQL/MongoDB/Redis**), build the **frontends** on top, and own the complete delivery path — **CI/CD**, containers, ingress, and observability — end to end. I also build AI agent systems as a first-class discipline: **my own tool-calling harness**, **multi-step agent loops**, and **knowledge-graph retrieval, tuned for cost and efficiency.**

Programming Languages

- C#
- Java (Spring ecosystem concepts)
- C
- TypeScript (NestJs & React)
- JavaScript

Backend Development & Others

C# / .NET / ASP.NET & xUnit(8 & 9) – Web APIs, worker services, high-performance services, multiple microservices for email processing, focused on high load

Node.js, NestJS & Jest – scalable server-side applications, distributed systems with microservices

gRPC & Protobuf – high-performance inter-service communication (.net services & node.js)

SMTP Integrations – email parsing, relays, and gateway logic

Java / Spring - REST APIs, layered architecture, dependency injection concepts, JUnit testing

---

Build & Dependency Management

Maven – project setup, dependency management for Java applications

npm / pnpm – managing dependencies for Node.js / TypeScript projects

---

AI Engineering & Agent Systems

own tool-calling agent harness (sandboxed execution, persistent sessions, agent loops); **agent-loop** + **knowledge-graph** retrieval for cost/efficiency; **OpenAI API** (function calling / Responses); **AI-driven development** with Claude Code, Cursor, MCP, GitHub Copilot. Integrated AI harness in our projects, wrote/improved skills, agents, tools and deployed a claude agent for automated workflows

---

Microservices & Architecture

Microservices Architecture,

Event-Driven Systems with Apache Kafka

Distributed systems & service orchestration

Databases & Storage

MongoDB / DocumentDB – advanced querying, performance optimization, large-scale datasets

PostgreSQL – relational data modeling, indexing, transactional systems

Redis – caching strategies, performance optimization

InfluxDB – time-series data for metrics, statistics dashboard

AWS S3 – object storage integration

DevOps, Cloud & Environments

AWS - S3, cloud integration patterns, distributed system design principles

Docker & Docker Compose – containerized microservices environments

CI/CD: Jenkins, Git-based workflows

Linux environments & Bash scripting

API documentation: OpenAPI / Swagger

Observability & Logging

Grafana(dashboards with logs and metrics), Loki, Promtail, Prometheus - logging and server resources usage

Serilog – structured logging in distributed systems

Coralogix/ElasticSearch - used to analyze logs in production/qa/staging environments

---

Front-End Development

HTML, CSS, SASS, JavaScript, React(basics)

Soft Skills

Analytical Problem-Solving – debugging complex data flows & distributed systems

Communication – explaining technical concepts to non-technical stakeholders

Collaboration – working across backend, frontend, DevOps, and data-focused teams

AI code pairing - skilled in using efficiently Copilot for a faster

Experience

Middle Software Engineer, OPSWAT TECHNOLOGIES SRL, 07/2023 - Current:

- Designed and developed scalable microservices-based backend systems using .**NET** and **Node.js**, following principles commonly used in **Java Spring** architectures (layered design, dependency injection, modular services)
- Optimized **MongoDB** query performance by designing a custom tokenization strategy for the emails repository(biggest collection), enabling instant search capabilities over massive datasets without performance degradation.
- Implemented secure Microsoft 365 integration using **Graph API** for automated tenant onboarding and mail flow configuration.
- Built a real-time statistics engine using **InfluxDB** and **gRPC**, providing granular visibility into threat detection and email traffic.
- Analyzing production logs on **Coralogix** to debug production issues
- Covered entire project with test to pass security audit for Global Availability release. Used **Jest & xUnit** frameworks for testing

Experience

SoftwareDeveloper, Continental Automotive Romania SRL, 12/2022 - 07/2023(8m):
Java (Spring-style backend concepts, Maven, JUnit) – production experience in backend services development, testing, and CI pipelines, Jira (Windows + Linux)

- Developed backend processing tools using Java, Maven, and JUnit
- Built data aggregation pipelines and report generation systems based on configurable inputs
- Applied object-oriented design principles and testing strategies in a production environment
- Worked with CI/CD pipelines (Jenkins) and version control (Git) in a Linux-based environment

Experience

TEACHER 08/2021-04/2024 ImpactAcademies&Camps, Chisinau, Moldova Teaching Romanian & Russian to children from 7 to 17 years old. Material divided into three sections, depending on age.

IT MASTER(8-10 years) :

- Visual programming, no-code web design, 2D game creation, basic motion design

SENIOR IT (11-14 years old) :

- No-code app creation, Unity 2D/3D + basic C#, advanced no-code web design, Python game development

IT EXPERT(14-17 years old) :

- FrontendDev with React
- C# programming
- App Dev using Java
- Data Science using python language (data analysis with different libraries, graphing, decision trees, data analysis from different CSV, Excel etc.)
- 3D Modeling using Blender
- DataBases(SQL & NoSQL)
- QA testing

TEACHER 05/2023-09/2023 Academia micilor developeri, Bucharest, Romania The courses are addressed to children and young people between 8 and 18 years old and are developed by IT specialists and pedagogues with international experience.
The mission of the Little Developers Academy is to transform children and teenagers from technology consumers into technology creators.

TEACHER 08/2022-08/2022 Girls Go IT (NGO), Chisinau, Moldova Camp for 25 girls between 15 and 25 years old. The event took place in the framework of the Polytechnic University of Chisinau (UTM) for 10 days. Theme Frontend of an App. Tools such as HTML, CSS were used.

Studies

Bachelor Degree Computer and Information Technology (CTI) Polytechnic University of Timisoara

Liceum (08/2022) **IPLT "Spiru Haret",** Chisinau

Languages

Romanian: First Language

English: C1 (IELTS certificated) — Advanced

Russian: C2 — Fluent

Personal Projects Overview

LioBlaja Bookings

GitLab: Link

NOTE! Still in development stage, consider that this is a per university license project

Tech: .NET 8 (C#), gRPC, REST APIs, PostgreSQL, Redis, Apache Kafka, Docker, GitLab CI/CD, Grafana

Architecture: Microservices (API Gateway, Booking Service, Notification Service)

Backend services implemented with .NET and designed following Java Spring-style architecture principles
Designed and developed a scalable restaurant table-booking platform using a microservices architecture. Implemented synchronous inter-service communication with gRPC and asynchronous workflows with Kafka for booking events and SMS notifications(or Emails). Built the Booking Service with PostgreSQL for schedules and reservations and integrated Redis for fast retrieval of recent restaurants and cities. Containerized all services with Docker and set up monitoring with Grafana. Supported anonymous bookings via SMS verification and automated reminder notifications.

Network Monitor – Real-Time Linux Network Monitoring Tool

GitHub: Link

Tech: C, ncurses, Linux /proc filesystem, multithreading (pthreads)

Developed a real-time network monitoring application in C with a ncurses interactive interface. Implemented multithreaded data collection to monitor active TCP/UDP/ICMP connections, listening sockets, and live network traffic statistics. Parsed system information from /proc/net and system utilities to compute per-connection transfer rates and overall network throughput. Designed a responsive terminal UI with continuous updates, ensuring smooth interaction even under high network load.

IMDB Search Optimization & Isolation Case Study

GitHub: Link

Tech: MySQL, PostgreSQL, Docker / Docker Compose, SQL, Transaction Isolation, Indexing & Query Optimization

Designed and implemented a comparative case study using the IMDB title.basics dataset loaded into MySQL and PostgreSQL containers orchestrated with Docker. Focused on optimizing text search queries on movie titles (suffix and substring search), reducing execution time to under 1 second through indexing strategies and query tuning. Implemented and analyzed transaction isolation violations by configuring isolation levels and crafting concurrent read/write scenarios, documenting expected vs. actual behavior. Collected and compared performance metrics between inefficient and optimized variants, alongside full command history and configuration steps for reproducibility.

Dense Matrix Multiplication – High Performance Matrix Multiplication

GitHub: Link

Tech: C, OpenMP, Algorithm Optimization, Performance Benchmarking, Blocked Matrix Multiplication

Implemented a full suite of high-performance matrix multiplication algorithms, including all six possible loop-ordering variants of the standard triple-nested loop. Developed both serial and parallel (OpenMP) versions, benchmarking each to determine the fastest implementation. Additionally implemented blocked matrix multiplication (serial and parallel), handling arbitrary matrix sizes and tuning block sizes to maximize cache locality and speedup. Performed extensive performance analysis across large N×N matrices (1000–3000), validating correctness against the classical i-j-k algorithm and identifying the most efficient parallel and serial execution strategies.

TextFileUploadServer – Node.js File Upload & Real-Time Client Management

GitHub: Link

https://github.com/LioBlaja/LioBlajaPad.git

Technical Stack: Node.js, Express.js, Socket.io, JavaScript (ES6+), HTML5, CSS3, Multer, async-mutex, WebSockets, RESTful API

Developed a real-time collaborative text and file-sharing web application using Node.js, Express.js, and Socket.io for bidirectional WebSocket communication. Implemented a RESTful API with multiple endpoints for text synchronization and file management operations. Utilized Multer middleware for handling multipart file uploads and async-mutex for thread-safe concurrent operations, preventing race conditions. Built a responsive, mobile-friendly frontend using vanilla JavaScript, HTML5, and CSS3 with Flexbox and Grid layouts. Integrated Socket.io client for real-time event handling, enabling instant synchronization of text and file updates across up to 2 concurrent connected clients. Features include bulk file upload/download, atomic delete operations, and connection limiting for optimal server performance. Demonstrates expertise in event-driven architecture, asynchronous JavaScript, middleware implementation, and real-time communication protocols.

DACSS – Review Moderator System (Pipes-and-Filters & Blackboard Architectures)

GitHub: Link

Tech: C, Modular Architecture, Pipes-and-Filters, Blackboard Architecture, Concurrency, Reusable Components

Designed and implemented two complete versions of an automatic review moderation system using two contrasting architectural styles: Pipes-and-Filters and Blackboard. Built a reusable set of C components for message filtering, transformation, and sentiment analysis, supporting features such as profanity removal, buyer verification, image resizing, competitor-link filtering, and sentiment detection. Demonstrated component reusability by assembling multiple client-specific configurations for both architectures.

Additionally implemented concurrent versions of both architectures, enabling parallel execution of filters/knowledge sources with thread synchronization. Performed experiments on large input sets to compare throughput, scalability, and architectural trade-offs, highlighting performance differences between linear pipelines and shared-state blackboard processing.