

Ratomir Vukadin

Senior Software Engineer/Architect, Public Speaker, Bachelor at Faculty of Electrical Engineering

🏠Address	East Sarajevo, Bosnia and Herzegovina
📞Phone	00387-65/211-581
🎂Birthday	07/09/1992
✉Emails	ratomir@live.com
Education	Computer Science, University of East Sarajevo (Bosnia and Herzegovina), 2011–2016 (graduated date: 09/05/2016), GPA-8.28, U.S. Grade B
Certifications	https://www.youracclaim.com/users/ratomir-vukadin/badges
Portfolio	https://ratomirvukadin.dev/



1 EXPERIENCE

1.1 IEHSOFTLABS, MARCH 2024 – PRESENT

I worked as a *Solution and Software Architect*. My primary task was designing the system from an overall standpoint. We developed systems and services for laboratories that process samples from food, industrial plants, rivers, and animals. The system tracks the entire lifecycle of a sample: its entry, the testing process (including methods and specific pathogens/diseases), and the generation of final status reports (positive, negative, invalid sample, or insufficient quantity).

One of my biggest challenges was designing a system capable of working in a two-level offline mode:

- Technician level: When the technician is far from Wi-Fi, data is saved locally on the mobile device. We solved this using a web browser database, RxDB.
- Laboratory level: When the entire laboratory is offline and cannot communicate with the cloud. We solved this by connecting to in-house servers and synchronizing data later.

We designed the system so laboratories can either use an in-house solution or migrate to the cloud. The software architecture is primarily a modular monolith with clear boundaries between modules. As an architect, I decided to distribute the Identity module as an independent service to support future Single-Sign-On processes.

Modules communicate with each other via a service bus using RabbitMQ. Certain parts of the system use bi-directional communication, established through federation between the local RabbitMQ in the laboratory and cloud RabbitMQ. All modules communicate with the service-bus and use the Inbox-Outbox pattern, both in the laboratories and in the cloud.

Also, as part of the *leadership team*, we planned and executed the total migration of the organization from on-premises systems to managed services on AWS and Digital Ocean. For adaptive scaling according to business needs, we use Kubernetes (k8s) clusters with multiple namespaces per project.

In early 2025, as a member of the leadership team, I performed a detailed analysis of AI tools like GitHub Copilot and Cursor. We then successfully introduced Cursor to the entire development department in phases.

My responsibilities also included tracking the professional development of team members through PDP (Personal Goals) plans. As a *Team Lead*, I conduct periodic 1-on-1 meetings, track progress, and define tasks that help team members exceed expectations.

1.2 DEVTECH, SOFTWARE ENGINEER

[HTTPS://DEVTECHGROUP.COM/](https://devtechgroup.com/)

- Jun 2022 – February 2024, Sarajevo, BiH

As a backend developer in C#, .NET Core. Utilizing Azure DevOps for deployment and Bicep for infrastructure modifications. The biggest challenge was to implement different types of payments, B2B, customer-Business, loans, and credits, following European financial laws with Stripe as the primary payment provider.

1.3 MISTRAL, SOFTWARE/DEVOPS ENGINEER/TECHNICAL LEAD

- June 2019 – May 2022

As a *Technical Lead*, I managed teams of up to ten members. My main responsibilities included the initial setup of technical solutions, system architecture, and resolving critical issues. I was also responsible for managing sprints and preparing tasks for the team. Beyond project work, I monitored the Personal Development Plans (PDPs) of each team member, helped them achieve their goals, and conducted monthly 1-on-1 meetings.

- While working on a Big Data project, we developed a platform that consolidated data from multiple sources, including Microsoft Dynamics, Salesforce, and internal systems. The platform processed data daily on Azure, with Azure Data Factory acting as the main orchestrator. The final product was a Power BI application that consumed data from Analysis Services.

- I also worked on implementing systems entirely based on Serverless architecture. We used the Serverless Framework alongside AWS Lambda or Azure Functions. I implemented a serverless data processing platform based on AWS Lambda and Step Functions.

- As a *DevOps Engineer*, I worked on complex systems and architectures based on AWS or Azure. I handled microservices orchestration via AWS ECS/EKS or Azure AKS managed Kubernetes platforms. I was responsible for maintaining and writing IaC code using Terraform, Azure ARM, or AWS CDK. My previous programming experience helped me out here, as I was the only member of the DevOps team with a

developer background. I also worked on platform implementation for SFF 2020 and 2021. One of my tasks was the deployment and distribution of mobile applications for iOS and Android via Microsoft App Center, Apple TestFlight, and Google Play Console.

- Upon my proposal, we introduced the KEDA framework to an AWS EKS-based system for processing film frames and parallelizing workloads. This reduced the processing time from over 7 hours to approximately 2 hours. We achieved the ability to determine the minimum number of pods required to process all frames in the shortest possible time. During this process, we even reached the limits of the AWS cloud provider.

- As a member of the Discovery Team, I worked in the early phases of projects, focusing on understanding client problems and defining solutions. My tasks included preparing architectural blueprints and proposals, analyzing requirements, finding optimal solutions, setting up initial architecture, also creation of C4 diagrams, database schemas and structures, infrastructure diagrams, responsibility matrices, and delivery phases.

1.4 KLIKA, DEVOPS ENGINEER

[HTTPS://WWW.KLIKA.BA](https://www.klika.ba)

- May 2018 – May 2019, Sarajevo, BiH

As a DevOps Engineer, I worked on a project primarily focused on Big Data. The system was developed on the Azure Cloud. The total data portfolio amounted to approximately 2 TB, with about a quarter of that data being active and processed daily. Initially, my task was to establish private virtual networks and isolate secure entry points into the system. Following that, I performed a complete migration of ELT processes from Azure Data Factory v1 to v2. I provided support during deployment sessions and was responsible for the end-to-end deployment process. We performed constant system monitoring, with the highest load being on data services such as Azure EventHub, Azure HDInsight, Azure Data Factory, Azure Cosmos DB, and Azure Table Storage. After implementing specific improvements and stabilizing the system, I organized the migration of data processing from Azure HDInsight to Azure Databricks, specifically moving from SQL to Python.

Technologies: Azure, Azure DevOps, Azure Data Factory, Azure DataLake, Azure Storage, Azure Table Storage, Azure EventHub, Azure Stream Analytics, Azure Databricks, NodeJS, C#, Python, Azure AppService, Azure Cosmos DB.

1.5 LANACO DOO, SOFTWARE ENGINEER

[HTTPS://WWW.LANACO.COM](https://www.lanaco.com)

- October 2016 – April 2018, Banja Luka, BiH

I was a member of the team that developed a complete solution for the Indirect Taxation Authority in Bosnia and Herzegovina. The system included taxpayer registration, accounting, tax card management, submission of tax returns, enforced collection, etc. It covered VAT, fees, freight forwarding, and more tax types. I worked on developing the taxpayer registration and deregistration modules. The system was built on classic monolithic architecture with a strong reliance on processes written in MS-SQL database stored procedures.

My biggest achievement was leading the development of a parser for importing process diagrams based on MS Visio workflows (XML structure) used in tax units. Based on these diagrams, and during requirements gathering with tax officials, we generated steps and connections, while the steps themselves were developed as web pages. The diagram controlled the workflow execution.

Technologies: ASP.NET v4.6, C#, MS-SQL Server, IIS Server, TFS.

- **Internship mentor, Lanaco Software Summer Internship 2017**

From July 31st to August 25th, I assumed the role of Internship Coordinator, leading the development of a software application for the MeNeRaLi association. I oversaw an 18-member group organized into four distinct teams. As the internship mentor, I established software architecture and orchestrated the development process across four sprints. I conducted daily Scrum meetings to enhance communication and track progress, while methodically organizing and prioritizing backlog items to ensure an efficient workflow throughout the development lifecycle.

1.6 AUTHORITY PARTNERS, SOFTWARE ENGINEER

[HTTP://WWW.AUTHORITYPARTNERS.COM](http://www.authoritypartners.com)

- Sep 2015 – Feb 2016, Sarajevo, BiH

As a full-stack engineer, I worked on the development of large-scale enterprise systems. Shortly after starting, I became a team member responsible for providing support during intensive deployments that began and ended outside of regular working hours.

Technologies: ASP.NET v4.6, C#, MS-SQL, IIS Server, TFS, Redis, AngularJS, jQuery.

2 CONTRACT ENGAGEMENTS

2.1 SOLAR-PREDICTION

I worked as a Solution Architect on a project focused on predicting solar power plant production. The system had the following main functionalities: day-ahead and intra-day prediction and loading data in near real-time from available API services provided by inverter manufacturers (Huawei FusionSolar, Solar Cloud, SMB, Fronious).

The system collected weather forecast data daily. Based on this, using a custom prediction algorithm, it calculated solar energy production for several days in advance. The system also loaded data from energy exchanges (CroPex, SepExx, NoS BH). It performed adaptive calculations based on weather parameters and market prices.

The system was developed on the Azure cloud platform using a combination of technologies. We used Azure Functions in combination with Container Apps. Since production plans were calculated only once or twice a day, the service did not need to run constantly.

We stored data using multiple database platforms:

- MongoDB: For deep weather models, production plans, and imbalance calculations based on exchange prices.

- Redis: As a key-value cache storage for generated plans and data that did not change often (plants, traders, producers).
- PostgreSQL (Supabase): For static data about power plants, traders, locations, organizations, and users.

The system was based on L1+L2 caching (in-memory and distributed levels).

My tasks were both broad and deep. They included setting up infrastructure (IaC) using Terraform, maintaining and monitoring all services, managing deployments with GitHub Actions, and writing Epics and Stories using GitHub Projects. One of the biggest challenges was developing the prediction algorithm. This required understanding the physics of power plants combined with weather forecast parameters. The algorithm was developed in Python. We used Python Notebooks for analytics and to compare new versions against previous ones, looking at both absolute error and financial impact.

As the Lead Architect, I realized that a standard algorithmic engine could not achieve the best prediction results. Therefore, we moved to creating Machine Learning models. I selected the CatBoost library as the best candidate, as it learns how the power plant behaves based on historical data collected by the system.

The web-portal used by traders was built using React NextJS. The biggest challenge in this part was displaying multiple selected parameters on a single chart, which we solved using the Plotly.js library. The entire system was developed without using commercial libraries, which was another challenge. The energy prediction engine will be released as open source.

Technologies: C#, React, TypeScript, Azure, Azure Functions, Azure Container Apps, PostgreSQL (Supabase), Redis, Python, MongoDB (Atlas), Azure Service Bus, Terraform, GitHub Actions.

2.2 2AM, 2 AMIGOS, SOFTWARE ENGINEER

[WE BUILD CUSTOM SOFTWARE | AGILE SOFTWARE DEVELOPMENT | 2AM.TECH](#)

I worked on algorithms to parse multiple CSV file formats into a single consolidated final model. The CSV Parser processed data from various insurance companies specializing in health insurance.

Technologies: C#, .NET 6, AWS Lambda.

2.3 QUANTUM SOLUTIONS, DEVOPS LEAD

[Q360 | HOME \(QUANTUM360.NO\)](#)

As the Lead DevOps Engineer for a Norwegian startup, I redesigned and deployed the infrastructure for the entire project several times to align with business requirements. As business grew, the infrastructure on Azure transitioned from App Services to a Kubernetes (k8s) environment to enable orchestration and adaptive scaling. One of my key responsibilities, shared with a team member, was writing and maintaining Infrastructure as Code (IaC) using Terraform. I organized and implemented the complete CI/CD process using the Azure DevOps platform across multiple environments (DEV, QA, STG, PROD, Post-Prod). Additionally, established services for 24/7 system monitoring and observability, providing insights into metrics and logs via Azure Application Insights and Log Analytics, including alerting configurations.

The system was based on a microservices architecture (covering 10-12 business contexts). We performed constant cost analysis across multiple Azure subscriptions, organized into PoC, Lower, and Production environments. The central components of the solution were AKS and Azure Service Bus for inter-service communication. Upon my proposal, the system was migrated to Azure Container Apps. These moves significantly reduced costs and simplified platform management, while the platform retained its original structure due to the compatibility between AKS and Azure Container Apps.

Technologies: Azure, Azure DevOps, Azure SQL Database, Azure Cosmos DB, Azure Service Bus, Azure KeyVault, Azure DataLake, Azure Storage, Azure Application Insights/Log Analytics, AKS, Azure Monitor, Azure Billing, Azure Active Directory.

3 PERSONAL SKILLS

3.1 LANGUAGES

- Native: Serbian, Bosnian

	Understanding		Speaking		Writing
	Listening	Reading	Spoken interaction	Spoken production	
English	C1	C1	B2	B2	B2

Levels: A1/2: Basic user – B1/2: Independent user – C1/2 Proficient user