Mehdi Akiki – Software Engineer

Email: **mehdi.akiki@gmail.com** Website: **mediak.net** LinkedIn: **linkedin.com/in/mehdiakiki** GitHub: **github.com/randomicon00** Location: **NY**

Profile

I bring several years of full-stack software engineering experience across diverse technologies including **JavaScript/TypeScript**, **Python**, **Java**, **Go**, and **Rust**. I also have extensive experience building, testing, and deploying **RESTful APIs** and **microservices** using **Django REST Framework**, **Node.js' Express** and lately **Gin Framework**, and **AWS services**. I develop robust, end-to-end solutions, hold a **Master's in Computer Engineering**, and actively contribute to major open-source projects, including the **Rust Programming Language**.

Skills

Languages & Frameworks: Django/Flask , RESTful API, gRPC, Go, Rust, Python, Java, JavaScript/TypeScript, React, NextJS, Angular, React Native, Spring, Pytest, Node.js, GraphQL, OpenAI, LLMs, LangChain, OAuth

Cloud/DevOps & Data Tools: AWS (EC2, Lambda, DynamoDB, API Gateway, Bedrock), GCP (Cloud Run, BigQuery), Azure (Functions), Kubernetes, Docker, Git, Terraform, PostgreSQL, MongoDB, ClickHouse, Redis, Memcached, Kafka, Spark, Jest, Cypress, Playwright, WebSockets

Experience

Founding Software Engineer | *MonitorMe* | *New York, New York* 12/2024 – Present

MonitorMe (monitorme.xyz) is an open-source observability tool combining **OpenTelemetry** for back-end tracing and front-end session replay to help developers efficiently debug and optimise distributed systems

- Architected high-performance, low-latency trace agents in **Go, Node.js,** and **TypeScript** achieving 40% latency reduction, 10,000+ traces/hour, enhancing real-time observability
- Engineered AI-powered data formatting **microservice** in **Python**, leveraging **Django REST Framework**, transforming JSON trace data to clear text for improved data readability and accelerated issue identification
- Designed scalable **PostgreSQL** database for 100M+ daily data points, optimizing **SQL** queries for 65% faster data retrieval in dashboards & analytics
- Spearheaded the development of a user-centric session replay feature using **rrweb**, directly addressing developer debugging pain points and reducing bug resolution time by 50%
- Implemented a resilient, decoupled, asynchronous data ingestion pipeline leveraging **Apache Kafka**, eliminating data bottlenecks under high concurrency and improving system's fault tolerance
- Developed an interactive, real-time dashboard using **NextJS**, **ReactJS**, **TailwindCSS**, and optimized **RESTful APIs** with **Gin Framework (Go)** and **Django REST Framework** backend services to provide users with insights for bottlenecks and errors
- Integrated AI-driven microservice in **Python** using **Django REST Framework** that aggregates responses from popular AI endpoints, interfacing with the **Golang** backend via **webhooks** to deliver rapid, precise summaries of bottlenecks with actionable resolution paths
- Set up a reliable **Kubernetes** system that automatically scales for 200+ users and stays up 99.95%

Full-stack Software Engineer | *Self-Employed* | *Casablanca, Morocco* 02/2022 – 11/2024

- **AttijariWafa Bank,** 2022 – 2023, *Morocco***:** Architected a highly available **Python Django** and **Django REST Framework** platform for credit advisors and field workers, and **ReactJS**/**JavaScript** front-ends, reducing system latency through asynchronous queue processing with **RabbitMQ.** Built a **low-latency** trading application for a bank's credit clients, leveraging **Java**, **Spring Boot**, microservices, asynchronous messaging (**Kafka/JMS**), **multithreading**, and **advanced caching** for high throughput and resilience
- **Cheikh Zaid Hospital**, 2022, *Morocco*: Developed "MedTrack"—an integrated hospital management platform using **Go**, **Java**, and **Node.js**. Designed and implemented a robust **AWS-based architecture**, extensively utilizing **Lambda functions** to securely retrieve, transform, and anonymize data from legacy databases for

public reporting. Processed thousands of GBs of data monthly, demonstrating deep **AWS** expertise while reducing infrastructure costs by 35% through scalable, cloud-native optimizations

- **Various Startups and Software Consultancy Firms (SQLi, CGI)**, 2022 – 2024, *Morocco*: I developed features, resolved bugs, and optimized performance using **ReactJS**, **Python Django**, **PostgreSQL** and **AWS**; additionally, I built Python scripts leveraging **BeautifulSoup**, **Scrapy**, and background processing to automate data collection in **ClickHouse DB.** Implemented **AWS DynamoDB** to store and rapidly retrieve structured data, significantly enhancing application responsiveness and scalability. Accelerated development of an elearning school catalog by leveraging AI-driven code generation, **React** and **RESTful API** services with **Node.js** and **MongoDB**

Education

Master's Degree in Computer Engineering, GPA: 3.7 — 08/2007 - 01/2010
Concordia University | *Montreal, Canada*

Bachelor's Degree in Electrial Engineering — 09/2000 - 09/2005
Polytechnique Montreal | *Montreal, Canada*

Open Source Contributions

Rust Programming Language | *Rust* — 2021 - Present

- Contributed to the **Rust** language, including source code, compiler, and bootstrap code

Rust Analyzer | *Rust, TypeScript* — 2023 - Present

- Contributed to IDE assists using the **Rust** source code parser and the **VSCode** plugin (**TypeScript**)

Deno | *Rust, TypeScript* — 2022 - 2023

- Actively fixed bugs and enhanced performance on both the **TypeScript** and **Rust** codebases

Mozilla Corporation | *Python, C++* — 2024 - Present

- Contributing to the mozdownload and browser, fixing bugs and enhancing functionality

Publications

Seventh International Conference on Information Technology | *Las Vegas, Nevada* — 04/2010
An Integrated Framework for Automated Firewall Testing and Validation

- Developed a novel firewall testing approach by combining two state-of-the-art techniques
- Automated the identification and correction of common firewall mistakes using **JavaCC**