Maxime Blais

me@maximeblais.dev | +1 819-960-8381
github | linkedin

Education

Bachelor of Science - Computer Science **2016-2019**

University of Sherbrooke *Sherbrooke, QC*

- Courses: Data Structures and Algorithms, Operating Systems, Compiler Design, Security and Cryptography, Distributed Systems, Real-time Systems, Web and Mobile Applications

Diploma of College Studies - Systems and Network Administration **2014 - 2016**

CEGEP Sherbrooke *Sherbrooke, QC*

Skills

Programming Languages	Scala | Java | Kotlin | Python | C# | JavaScript | Bash | PowerShell
Technologies	GCP | AWS | Kubernetes | Spring Boot | Finagle | PostgreSQL | MySQL | Redis
Tools	Terraform | Github Actions | DataDog | ELK Stack | Jenkins

Experience

Hopper **2022 - Present**

Senior Software Developer, Fraud Prevention *Remote, QC*

- Designed a workflow engine to enable the creation of complex fraud prevention rules.
- Created an automated chargeback deflection and dispute system.
- Integrated various 3rd-party providers to reduce false positives and improve real fraud detection.

CEGEP Victoriaville **2021**

Teacher, Android App Development *Victoriaville, QC*

- Taught Android app development to students in the Computer Science program.
- Created a course plan and a series of exercises to help students learn the basics of Android development.
- Evaluated students' work and provided feedback.

Coveo **2020-2022**

Software Developer, Sources *Remote, QC*

- Developed a feature to allow seamless promotion and synchronization of configuration changes between environments.
- Owned and maintained multiple backend services part of the document ingestion pipeline.

Notable Projects

ReaQtor.info **2019 (Capstone Project)**

Full Stack Developer *Sherbrooke, QC*

- Collaborated with a team in building a web application for the collection of audience reactions in real-time. The app would be used by various organizations during conferences and events, and would allow analysts to analyze the audience sentiment and engagement.
- Front-end development using Razor with ASP.NET and TypeScript. The front-end offered a real-time view of the audience reactions and allowed the user to create and manage events, perform analysis and export data.
- Back-end development of a C++ server using the µWebSockets library to handle the real-time communication with the front-end. Events and reactions were stored in a PostgreSQL database.
- Containerized the application using Docker and deployed it on a Kubernetes cluster on Google Cloud Platform.
- Automated the testing and deployment of the application using Gitlab CI, Terraform and various bash scripts.