Mohamed Bana's Curriculum Vitae

Software Engineer

Contact	Web
+44-07960-045-281	https://github.com/mbana
jobs@bana.io	https://linkedin.com/in/mbana
https://stackoverflow.com/users/241993/mohamed-bana	https://bana.io/blog

I am a highly skilled Software Engineer with over a decades worth of job experience and a proven record. I love to build and maintain high-availability rock-solid systems that support successful businesses. I am looking for an interesting **remote-only** job, solo or in a small team of professionals to share my knowledge and to learn from. Part-time occupation is negotiable. Please read my cover letter at **https://bana.io/resume/cover-letter**.

Work Experience

Software Engineer and Director, BanaIO Ltd., London, United Kingdom – 03/2017–Present

Software Engineer and Director at BanaIO Ltd. Worked at several companies and projects:

- Cynergy Bank
- Vitrifi
- IBM
- Open Banking
- Ninety Percent of Everything
- Root Capital LLP
- Lloyds Banking Group

Tech: Golang, Kubernetes, Docker, Docker Compose, Linux, Shell scripting/Bash, Google Cloud Platform (GCP) and Google Kubernetes Engine (GKE).

Senior Software Engineer, Kubeshop, Greenwich, United States – 05/2022–01/2023

Worked on kusk-gateway, a OpenAPI-driven API Gateway for Kubernetes. Links:

- **https://kusk.kubeshop.io/** or **https://kusk.io**
- **https://github.com/kubeshop/kusk-gateway**

Kusk is a Kubernetes API gateway powered by Envoy. The main difference with other API Gateways is that Kusk has native support for OpenAPI.

Modern REST APIs are developed using OpenAPI specification that is then used to generate API documentation, tests, server stubs and clients all from the OpenAPI definition. Kusk enables the use of OpenAPI definitions to configure the Ingress Controller of your Kubernetes clusters.

The commits I made against the repository is available to view at: **https://github.com/kubeshop/kusk-gateway/commits?author=mbana**.

Tech: Golang, Kubernetes, Kubernetes Control Plane, go-control-plane, Envoy Proxy, Docker, Docker Compose, Minikube, Shell scripting/Bash, gRPC, Protocol Buffers, GitHub Workflows, Linux.

Software Engineer, Paymentsense Limited, London, UK – 07/2021–10/2021

I worked with Golang on Connect-E (**https://docs.connect.paymentsense.cloud/ConnectE/GettingStarted**).

Tech: Golang with modules, Docker, Docker Compose, TypeScript, GCP, Google Cloud Datastore, Google Cloud Big Query, Google Cloud Pub / Sub.

Senior Front-End Software Engineer, Synthace Ltd., London, UK – 04/2016–11/2016

Did a fair amount of architectural UI work:

- JWT-based authentication: Implemented most, if not all, of the authentication related UI features. Polymer didn't have an authentication module as it's fairly new requiring me to re-implement this feature.
- API interactions: I introduced Swagger JS and did the conversion from plain XHR to Promises, and ensured API was in-sync with the state of the authentication.
- Updates via the Web Socket for notifications and async task updates: STOMP Over WebSocket.
- Bootstrapped the testing using Web Component Tester.
- Deciding on the build, test and hosting strategy, e.g., hosting our own CDN using Azure.
- Performance: 1) pushed to have HTTP/2 enabled, and prototyped, on our custom server written in Go, 2) implemented lazy-loading of our Web Components which are included using HTML Imports, 3) Significantly improved UI build scripts; went from a somewhat un-deterministic build to one that almost always runs.
- Introduced ES6 to the code-base, and moving to defining Polymer elements using ES6 classes.
- Misc: libraries/utils to ease UI development.

We deploy Docker images to our Kubernetes cluster running in Azure using:

- Docker: Fairly comfortable using this.
- Kubernetes: I've done deployments of dev branches, so I understand the deployment model, navigating the Kubernetes dashboard and crude command line interactions, e.g., port-forwarding of the service the pod is running from the cluster to the local machine.

Added support our language called Antha to Monaco Editor, the editor that powers Visual Studio Code.

Since this is a startup I have done a fair amount of work and I have been given a fair share of responsibility, more so than any of my prior roles.

Tech: JavaScript, ES6, Polymer Project, Web Components, TypeScript, VSCode, Azure, Docker, Kubernetes, Go, Polymer Web Component Tester, Git, Swagger.

Software Engineer, IG Index Ltd. London, UK – 09-2014-02/2016

Price & Indicator Alerts My main responsibility was handling the UI aspect—setting, configuring, triggering—of the various Alerts we support, from the basic Price Alert to Technical Indicators such as RSI and Moving Average. **Tech New:** Modern UI powered by ES2015 (Babel), Ember.js, Handlebars and Less. The UI was then composed of isolated and reusable Ember components. Runs on a NodeJS server, managed with npm and bower, and version controlled using Git. Integration and unit tests written in Mocha then run using Karma. **Tech Old:** Vanilla JavaScript using an in-house framework when changing the previous UI.

Charts Assisted in the conversion of the Adobe Flex Real Time Charts to an SVG-based version. Tech: d3 and tested like above.

Deeplinking A hashed action link, like typical deeplinks, that we send to our Clients which then launches the mobile IG app, or directs them to the app store for the device with the IG app pre-selected. Upon login the action is carried out automatically. I did the bulk of the work with the team lead overseeing it. Tech: Java 8, Spring and acceptance tested using Cucumber. Redirecting and launching of the IG app was done using vanilla JavaScript.

Software Engineer, ITRS (International Trading Room Software) Group Ltd. London, UK – 02-2010-09-2014

JavaScript (UI) Software Engineer - 01/2014-09/2014 UI for the next generation of the product which is built around, loosely speaking, a real-time distributed analytics store. The aim is for the old system to stream data to the new system so we can provide all the great visualizations available in the HTML ecosystem, which was not achievable in a reasonable amount of time in Swing.

The code is entirely modularized using RequireJS so we can test each viewmodel without creating a view, we then test the entire UI (end-to-end tests) using WebDriverJS.

Some Java/C coding required to write NodeJS bindings to interact with the store. We evaluated several frameworks, Angular, Batman and KnockoutJS, by writing prototype applications that connected to our backend for a period of roughly 4-6 months before we chose to settle on Knockout.js.

Tech: JavaScript, NodeJS, Node-WebKit, Durandal, KnockoutJS, RequireJS, Git, Jasmine, Protractor (WebDriverJS), Jenkins, Bower, HTML5, CSS, LoopBack.io.

Java (UI) Software Engineer - 07/2012—12/2013 Worked with two software engineers and one QA member on a new Swing UI, ACLite, that uses our new streaming-based API to access Geneos data, for more info. see **https://resources.itrsgroup.com/OpenAccess**. We access data from a fault-tolerant Cluster that is built on a set of distributed nodes. The system we coded against is somewhat similar to the Amazon distributed key/value store, DynamoDB, except with support for streaming, so I am familiar with dealing with distributed systems.

Tech: Java, Maven, Swing, Eclipse, Jenkins, Git, Vagrant.

C++ Software Engineer - 02/2010—06/2012 Spent one year with Run The Business (RTB) team, a team set-up to fix critical bugs that Customers encounter. A very challenging role which requires all-around product knowledge, good debugging skills and being able to liaise with our Support staff in dealing with the Customers. Prior to this I was one of three software engineers working in the Transactions and Latency Monitoring team (part of the backend team) doing core C++. We wrote and maintained the following plug-ins that are part of the Geneos suite:

1. Geneos FIX plug-in: Monitors FIX (protocol) messages.
2. Feed Latency Monitoring Plug-In: Monitors feeds and calculates latency of instruments and fields across the monitored feeds.
3. Latency Monitoring – Message Tracker FIX adapter: Tracks, generally, FIX messages across several checkpoints.
4. Market Data Monitor: See below. And bespoke plug-ins written for specific firms.

We also maintained several non-finance specific plug-ins. I ported another bespoke plug-in called Price Latency Monitor (provides latency figures for bonds) to MS VC++ when I worked on this team. Projects:

- I converted the Windows version of the entire product suite from Visual Studio 2005 to 2010.
- I wrote the Market Data Reliability plug-in. This plug-in connects to the Patsystem's Trading API to monitor commodity prices, using their C API, to determine if prices are 'stale'.
- I ported a significant part of our product to Solaris x86-64 (64-bit non-sparc architecture).

Tech: C++, STL, Boost, Visual Studio, Linux/Unix, GDB, DBX, Make, Configure, XML, XPath, CPPUnit, Financial Information eXchange (FIX) protocol, Geneos FIX plug-in, Feed Latency Monitoring Plug-In, Latency Monitoring – Message Tracker FIX adapter, Market Data Monitor, Geneos PATS-STATUS Plug-In, Patsystem's Trading API.

Software Engineer Internship, then Tester, Thomson Reuters. London, UK – 05-2009-11-2009

C# Software Engineer One of four software engineers working on a search and navigation interface to Global Product Search. Full life-cycle of development; requirements engineering, analysis and design to implementation, testing and deployment.

Tech: C#, Silverlight 3.0, MS SQL Server 2005, LINQ, Web Services (WCF), XML and Visual Studio 2008. I handled deployment using CruiseControl.NET.

Tester User Acceptance Testing of the latest release of Thomson Reuter's 3000 Extra, then called UTAH, now called Eikon. UTAH combines the data from Thomson and Reuters. My primary responsibilities were to validate the end product against pre-defined requirements/workflows. 1. Worked on Thomson Reuters project UTAH as part of a large team. 2. Tasks included testing, observing, documenting software bugs, issues and errors before final release of Utah. 3. Testing was done over multiple iterations.

Education

- **2005-2008:** BSc Computer Science, City, University of London.
- **2008-2009:** MSc Software Systems Engineering, University College London, and Trading & Financial Market Structure module, London Business School.

Additional Information

Misc

- **Passport/Nationality:** I am a British citizen with a British passport.
- **Drivers Licence:** Full UK Driving Licence.

Languages

- **English and Swahili:** Native.
- **Arabic:** Intermediate. I have lived in Marrakech, Morocco for almost two years. I have also lived in Cairo, Egypt and have travelled several times to the UAE.

Mohamed Bana's Cover Letter

Software Engineer

Introduction and background

I am a well-rounded Software Engineer, and I understand what it takes to release a product having worked on the back-end, front-end, testing and finally the deployment aspect of several products.

A role involving web technologies hosted on a cloud-based backend. I've worked on all the tiers of a software product, so I can appreciate the different concerns expressed at each level. What interests me: Go/Golang backends, building frontends in JavaScript, complete end-to-end testing of the final project, CI builds on the cloud, deployment on the cloud using Google Cloud Platform (GCP), containerising software using Docker, and orchestrating the built containers using Kubernetes.

Key skills and preferred job role

I'm flexible in terms of the choice of framework used, however, my ideal role will involve the following:

- Go/Golang as I have good experience with this. I have been using Golang since 2017.
- A role where Linux, Google Cloud Platform/AWS/Azure, Docker/Podman and Kubernetes are being used. Open Banking would be a plus.

I'm up for learning more of Rust but it's just a hobby at the moment.

Summary: Senior Software Engineer with expertise in Golang, Linux, Docker, Docker-Compose, Kubernetes, Cloud Computing, Google Cloud Computing (GCP) and Linux.

Notice period and salary

I'm available to start after one week. I'm willing to negotiate on this as it depends on a few things, that said:

- If it's a contracting role, minimum of £400 per day.
- If it's a permanent position, minimum of £65,000 per year.

The figures I mentioned above are not set in stone and I am willing to listen to counter offers.

Work locations considered

A fully remote role with a UK or US company.

Any visa restrictions – if applicable (ILR, Tier 2, Dependant visa etc)

- **Passport/Nationality:** I am a British citizen with a British passport, right to work, national Insurance number etc.
- **Drivers Licence:** Full UK Driving Licence.