

CV – Jelle De Vleminck

Personal Information



10/06/1997

Ternat

Belgian



Bachelor Computer Science, Cum Laude



Profile



Medior Software Engineer

Java	Expert
AWS	Advanced
DevOps	Advanced
Kafka	Advanced

Key Certificates

Issuer	Certificate	Acquired
Oracle	OCP, Java SE 8 Programmer	2019
Amazon	AWS Certified Developer – Associate	2019
Amazon	AWS Certified Solutions Architect - Associate	2019
Confluent	Confluent Certified Developer for Apache Kafka	2019
CNCF	Certified Kubernetes Application Developer	2020

Languages

	Speaking	Understanding	Writing
Dutch	Mother tongue	Mother tongue	Mother tongue
English	Fluent	Fluent	Fluent
French	Intermediate	Intermediate	Intermediate



Career History

Axxes

08/2018 - Present

Client	Axxes - Brussels Airport Company
Period	10/2018 - Present
Location	Zaventem
Description	<p>Software Engineer</p> <p>Environment:</p> <p>During my time as a developer for Brussels Airport company I initially had to maintain existing REST APIs and make new REST APIs that use our core framework written with the Akka framework. The quality was very important here because a bug in an api can cause the airport to go down. In addition to developing, I also built support dashboards in Kibana to thoroughly monitor our APIs.</p> <p>As time progressed and my Java knowledge improved (obtaining OCP certificate) I was allowed to work more and more on changes in the core framework. This allowed me to implement new features, as well as make proposals to make the life of the developer and / or support easier. With all changes to the core framework, code generation was taken into account in order to be able to develop new APIs faster. Once the core framework was general enough, we built a code generation tool with Spring Boot and VueJS to save us the boilerplate work (about 3 hours) when creating a new API.</p> <p>After about 5 months we started making AWS services. This was necessary because we wanted to process IOT data and because management decided to have a "Cloud first" approach. Because I knew that AWS was relevant in the labor market, I definitely wanted to work with this. I then obtained 2 associate AWS certificates so that I had a good basic knowledge to start with. I subsequently worked on a project in which sensor data and historical data are processed to make a prediction of the waiting times at the security check. Hereby I have written some AWS Lambda functions that process data from Kinesis and enrich it with a search in DymanoDB. I also participated in a project where the metal detector data is processed to know how many times an alarm has been triggered in a given time and how many people have passed through a location. I made this project with the Serverless framework (infrastructure as code framework).</p> <p>Afterwards it was decided to abandon AWS Kinesis for real time applications and to use Apache Kafka. The first projects where Kafka was used is the BLIP case where the waiting times of security are predicted, and the PAX forecast where the amount of passengers that the airport can expect is predicted. I worked on the PAX forecast project, which later became a separate feature team. We decided to use Apache Spark, together with a machine learning model to predict the expected amount of passengers for the upcoming 6 months. We connected everything through apache Kafka to do the integration. Once we had the passengers forecast finished, we built features on top of this. The main feature built on top of this was calculating the amount of passengers per location per timeslot. For example: X amount of passengers at security screening</p>



	<p>between 15.00 and 15.30. This improved the efficiency and the planning of G4S Security officers a lot.</p> <p>Tools: Java, Akka, AWS, RabbitMQ, Sybase, MariaDB, Redhat 3Scale, Keycloak, CentOS Linux, ELK stack, Atlassian Tools (Jira, Bitbucket, Bamboo, Confluence), Kafka, Kubernetes, Terraform, Serverless</p>
--	--

Client	Axxes – JAVA Traineeship
Period	08/2018 – 09/2018
Location	Antwerp
Description	<p>Internal Training</p> <p>General & Methodologies</p> <ul style="list-style-type: none"> • Java • Developing Enterprise Applications • Version Control with GIT • Scrum • SOLID • Clean Code • Monitoring & Logging • Continuous Integration • Linux • Docker • Amazon Web Services • Communication skills <p>Back end & databases</p> <ul style="list-style-type: none"> • Spring: Spring Data, Spring Security • JPA • Hibernate • Big Data • MongoDB <p>Front end</p> <ul style="list-style-type: none"> • JavaScript • Angular • React <p>Testing</p> <ul style="list-style-type: none"> • Software Testing • Mocking & Unit Testing



College:

Client	Mobco (internship)
Period	03/2018 – 06/2018
Location	Dilbeek
Description	<p>Developer</p> <p><u>Project 1:</u> Environment: The Mobile Monitoring Service is a platform to which several of the customers are connected and which will carry out a permanent series of tests (each consisting of a script with various elements) on the customer's infrastructure. Based on the results of the tests (and therefore also the results of each of these elements) and the frequency of these results, the customer should be proactively informed via an 'alarm' or 'alert'.</p> <p>Developing the Alerting Module for the Mobile Monitoring Service:</p> <ul style="list-style-type: none"> • interpretation of the information to verify if the customer must be notified • expanding the 'connectors' to the customer's systems (email, sms, SCCM, ...) • to set up a visualization via the portal that allows the customer to discover the root cause as quickly as possible <p>This project was developed within the software development team and was in collaboration with an external software company called Quamotion</p> <p>Tools: C#: .NET Core 2.0, Javascript (Vue.js), HTML/CSS, SmsEagle, Visual Studio, Git</p> <p><u>Project 2:</u> Environment: The enterprise contacts application is an application made for the European Court of Auditors that runs on both Android and iOS and provides the following features:</p> <ul style="list-style-type: none"> • Full listing in alphabetical order of all contacts found in Active Directory, within the predefined search filters • Detail view of the contact details • Detail view of the presence information retrieved in Skype • Detail view of the contact picture provided via an HTTP connection • Tasks view (tasks retrieved from Oracle database) • Useful numbers view • My colleagues view (grouped on an AD property) • My contacts view (individual list of contacts) • Continued functionalities when offline (caching) except for presence <p>The Enterprise Contacts Solution exists out of 3 main components:</p> <ul style="list-style-type: none"> • Contact Gateway Service (Skype for Business, Picture, Contact/Task/Useful Numbers get and post endpoints) • Data extractor (AD, Oracle, Useful numbers JSON) • Native Mobile Application (Xamarin Forms Ios/Android)



	<p>My task in this project was to develop the data extractor and the contact gateway.</p> <p>The data extractor is a console application that runs once, does the job and quits. Therefore, the console application should be scheduled using Windows scheduled tasks. The data that will be extracted depends on the argument passed with the executable:</p> <ul style="list-style-type: none"> • “contacts” argument will extract the Active Directory contacts • “tasks” argument will extract the tasks from the Oracle database • “usefulnumbers” argument will extract the useful numbers from the appsettings.json file <p>Tools: C#, .NET Core 2.0, Xamarin Forms, Visual Studio, Git</p>
--	--

Client	Comprosoft (Final Work)
Period	2017-2018
Location	Brussels
Description	<p>Developer</p> <p>Environment: It often happens that employees incur costs that are reimbursed by their employer. Some typical examples of this are parking costs and restaurant costs. The employee must submit a proof to his company. This means that all paper receipts must be kept and then issued. Once the costs have been issued, they must be approved. If the costs are approved, the employee will see these costs paid back at his subsequent monthly wage. This is a lengthy process that entails some problems. It is not easy for the employee to keep an overview of the costs to be recovered. This makes it difficult for 5/7 the employee to check whether all his costs have been reimbursed. The employer is at the end of the month with a stack of costs that he has to enter manually with the accompanying employee. Mistakes can also easily be made here. Or what happens if an employee loses his proof?</p> <p>The objective of the system is to simplify the life of both the employee and the employer. We want to achieve this by digitizing the process. Employees must be able to easily register costs. These costs must immediately be passed on to the employer. The employer must be able to process these costs easily. Our central research question is: How can one manage the costs of employees in an efficient and well-organized way?</p> <p>This final work is made by 2 people. My tasks in this project were:</p> <ul style="list-style-type: none"> • Research backend technologies • Research & implementation text recognition (possibility to scan receipts) • Saving & viewing pictures • Full implementation Alexa with security features (possibility to enter costs with voice) • Statistics (backend and frontend)

	<ul style="list-style-type: none"> • Superadministrator / application management features • Filtering costs • Scanning permissions • Warning system for missing proof • Workshop Angular for students (part of the final work) • Deployment application <p>Tools: Angular4, HTML/CSS, PrimeNG, Spring Boot, Bootstrap, Hibernate, MySql, Jaspersoft, SendGrid, Alexa, TravisCI, Git, NGINX</p>
--	---

Education and Extra Curricular

2015-2018	Bachelor Computer Science, Erasmus hogeschool Brussel
2013-2015	Informaticabeheer, Don Bosco Groot-Bijgaarden
2009-2013	ASO Economie Wiskunde

Extra

2019-Present	Lector Java programming (night school) at Erasmus hogeschool Brussel
2017-2018	Tutor student-independent through "Het BijlesBureau"

Knowledge and Experience Summary

Languages & Frameworks

Expert

Java, Spring, Git, Maven, CI/CD (Bamboo), Kubernetes, AWS, Kafka

Experienced

Linux, Software security, Akka (actor model), Event Driven Systems, Serverless, Terraform, Angular, SQL, Amazon Alexa, RabbitMQ, Integration, ELK stack, Atlassian stack

Knowledge

Oracle, MSQl, Windows Server, usability design, Android, Swift, Xamarin, Raspberry PI, PHP, .NET core, JEE, Javascript

