

## ABOUT ME

Born on September 23, 1979, in Stolberg, near Aachen, Germany, I am a experienced Senior Full-Stack Software Developer with 20 years in the automotive, e-commerce, and donation/member management sectors.

Technical expertise in LAMP, Windows, (No-)SQL, .NET, and Node, as well as frontend and backend development.

Focused on delivering innovative and efficient solutions, continuous learning, and team-oriented project successes.

## EXPERIENCES

Senior Software Engineer 2023 - Present  
Next.e.GO Mobile SE, Aachen, Germany

Ensuring and conceptioning of backend services, apps, system architecture as well as development of software solutions, technology scouting and testing, enterprise-wide software engineering coaching.

- Cloud based software development
- Database development
- DevOps
- Frontend development (web & mobile)
- Open Source
- System administration

Senior Software Developer 2021 - 2023  
Next.e.GO Mobile SE, Aachen, Germany

Full-Stack Software Development and Cloud Administration.

- Flutter
- Go
- Kubernetes
- Next.js
- Open AI
- PostgreSQL

Senior Software Developer 2018 - 2021  
e.GO Digital GmbH, Aachen, Germany


Full-Stack Software Development and Cloud Administration.

- Docker
- Microsoft Azure
- NoSQL
- TypeScript
- Vue.js

IT Developer & Administrator 2016 - 2018  
doppeltplus GmbH, Aachen, Germany


# Marcel J. Kloubert


Senior Full-Stack Developer


 marcel@kloubert.dev  
(mailto:marcel@kloubert.dev)

 Germany


 Europe/Berlin

 blog.kloubert.dev/  
(http://blog.kloubert.dev/)

 marcel-kloubert-410013282  
(https://linkedin.com/in/marcel-kloubert-410013282)

 Marcel\_Kloubert078926  
(https://www.xing.com/profile/Marcel\_Kloubert)

 mkloubert  
(http://github.com/mkloubert)

 Download PDF  
(/CV\_\_Marcel\_J\_Kloubert\_\_Senior\_FullStack\_

## EDUCATION

FH Aachen, Germany  
2016 - 2018

GRÜN Software AG  
2006 - 2009

RWTH Aachen, Germany  
2001 - 2004

## LANGUAGES

German (Native)

English (Professional)

## INTERESTS

Black Fun Shirts

Books

Flipper Zero (<https://flipperzero.one/>)

Movies

Music

Working student

Open Source  
(<https://github.com/egomobile>)

- E-Commerce
- GitLab
- Linux
- PHP
- Shopware

Software Developer

2015

Mayersche, Aachen, Germany

- E-Commerce
- MySQL
- OXID
- PHP
- Visual Basic .NET (VB.NET)

Software Developer

2013-2014

TN CuRA GmbH, Aachen, Germany

- ASP.NET
- C#
- Cryptography
- Subversion
- Visual Basic .NET (VB.NET)
- Windows Presentation Foundation (WPF)

Junior Software Developer

2010-2013

EVOCURA GmbH, Aachen, Germany

- .NET framework
- C#
- Entity Framework
- Microsoft SQL Server
- Windows Communication Foundation (WCF)
- Windows Presentation Foundation (WPF)

## PROJECTS

**After Sales web app & backend for service partners and workshops** - *Docker, Microsoft Azure, Microsoft Dynamics CRM, Microsoft Power Automate, Next.js, Node.js, PTC ThingWorx, TypeScript*

This internal web application combines the most important workflows in the after-sales segment:

- creation of warranty claims by service partners
- documentation of workshop orders
- communication with service partners
- provision of vehicle documentation, such as technical documentation and instructions
- announcements of service campaigns
- provision of digital vehicle files
- ordering of spare parts

Both the frontend and backend are proprietary developments, in which I am involved in both cases.

Nevertheless, the external backend, which is connected to the frontend, is also connected to numerous external systems: This includes in particular CRM, production, and cloud systems from various providers.

**Desktop app for article maintenance** - .NET Framework, C#, GitLab, MVVM, Win32 API, Windows, Windows Presentation Foundation (WPF)

For a client, article data had to be entered into a merchandise management system.

Since they were not in an organized list, such as Excel/CSV files, they had to be entered manually.

For this purpose, I developed a Windows desktop application in C# that optimized the process as much as possible, so that as many articles as possible could be entered in the shortest possible time.

The application was also designed to allow multiple people to work on processing the list at the same time.

**e.GO Configurator (<https://configurator.e-go-mobile.com>)** - Azure Active Directory, Azure DevOps, Docker, MongoDB, Next.js, PostgreSQL, React.js, Redis, TypeScript

Under this website, potential customers can configure an e.wave X and then place an order.

In addition to setting up the backend and frontend projects in Azure DevOps, I actively supported and accompanied both frontend and backend development.

One challenge was to meet the high number of accesses after the launch in addition to the UX.

In addition to this configurator, a sales portal for e.GO sales partners was developed at the same time, which I implemented as a Next.js application with its own UI library.

Through this portal, sales partners are able to independently receive orders and send them to e.GO with all necessary documents.

**e.GO Connect App (<https://apps.apple.com/de/app/e-go-connect/id1573365927>)** - Azure DevOps, Azure Kubernetes Service (AKS), Docker, Helm Charts, MongoDB, Node.js, PostgreSQL, TypeScript, React Native

This mobile app served in its first form to connect a customer with their vehicle based on TCU data.

The app is currently being developed platform-independently based on React Native.

My main task is to ensure the connection of the app to the backend, which is implemented as a collection of various microservices.

This includes not only the implementation itself, but also the smooth functioning of the build and release processes in Azure DevOps.

**Note:** Due to a change of TCU provider, the functionality is currently deactivated.

**e.GO internal administration tool** - Azure DevOps, Docker, Microfrontends, Next.js, TypeScript, Webpack

At e.GO Mobile we are developing an internal administration tool that is used by the following departments:

- After Sales
- Data Science
- Marketing
- Sales

This tool is used to manage projects such as the "e.GO Connect App", "e.GO Configurator", "e.GO Website", and "After Sales web app & backend for service partners and workshops".

In addition, e.GO-specific QR codes can be created and managed and a self-developed wiki is also part to share information between the departments.

The web app is protected by a rights/roles system, which is implemented by employees from different departments.

I initiated this web app as a Next.js/React application using our own e.GO specific UI framework, and I am now leading the refactoring to a microfrontend application with other e.GO developers.

In addition, I ensure the functionality of the build and release pipelines in our DevOps environment.

**e.GO sales portal** - *Azure Active Directory, DevOps, Next.js, Node.js, REST API*

This is a web application that was developed simultaneously with the e.GO Configurator (<https://configurator.e-go-mobile.com/>) in the second half of 2022 and was released on time for the launch of the e.wave X model.

I worked on both the frontend and the underlying backend. While I gave support in the backend development, I personally focused on developing the frontend part, while the phase of implementation.

A potential customer can go to one of the e.GO sales partners with a code that belongs to a configuration created through the e.GO Configurator (but doesn't have to). The partner is then able to make final changes using a simplified editor, if needed. Finally, the partner can trigger an order with all documents in the e.GO system.

**e.GO URL shortener and QR code generator** - *Barcodes, Docker, Go, MongoDB, REST API, SVG*

The URL shortener / QR code is a microservice, which is part of a large API cluster at e.GO.

In addition to redirecting to the actual address, it generates on-the-fly QR codes, for example as SVG files.

These are managed in the internal tool of e.GO, which I have already mentioned as "e.GO internal administration tool".

In addition, after creating a link, URL parameters can be added as needed and at any time, without the need to adjust the data record in the administration tool. Since these URL parameters are not part of the data record itself, the same data record can be used to distinguish between different partners and campaigns.

Examples:

- <https://link.e-go-mobile.com/ego-connect-app-store?language=german>

- <https://link.e-go-mobile.com/ego-connect-app-store/qrcode?language=english>

**e.GO Website (<https://e-go-mobile.com/>)** - *Azure DevOps, Docker, Helm Charts, Kubernetes, Microfrontends, Next.js, PostgreSQL, REST API, React.js, TypeScript, Webpack*

The website as the flagship of e.GO Mobile, I have particularly supervised in the sense that I, as one of 2 DevOps engineers, was responsible for the smooth functioning of the build and release processes.

In addition, I prepared the site in such a way that parts of it could be integrated as micro frontends.

Currently I am preparing the relaunch of the current website.

**In-house Visual Studio Code extension** - *Azure Open AI, Azure DevOps, Node.js, TypeScript, VSCode*

To make the approximately 400 projects more easily accessible in the Visual Studio Code editor, I have written an extension that every developer can install.

Some features of this extension:

- auto-setup of the project
- detection of the underlying Azure DevOps project, depending on the currently open git repository
- opening of a list of own work items like bugs, user stories, or tasks

- ChatGPT functions:
  - creation and execution of shell commands from natural language for a file or directory
  - creation of git branch from natural language
  - explanation and documentation of code
  - translation of text in editor or (i18n) JSON files.
- opening, browsing, and saving content of and from zipped files
- scriptable notebooks similar to <https://github.com/egomobile/vscode-powertools/wiki/Notebooks> (<https://github.com/egomobile/vscode-powertools/wiki/Notebooks>)

**Mail merge function via SOAP as Word add-in** - *Microsoft Word, PHP, SOAP, C#, Windows Forms, XML, Visual Studio Tool for Office (VSTO)*

I had developed a Word add-in for a connection to the web-based membership management system developed by GRÜN, which was able to retrieve member data through a SOAP interface.

Through a field selection dialog, these could be integrated into the current Word documents and also stored as XML in the document itself.

Anyone who had the corresponding add-in could later access all the data again.

**Modular WCF backend** - *.NET Framework, C#, Entity Framework, Managed Extensibility Framework (MEF), MSSQL, WCF*

I implemented a multi-modular backend at EVOCURA.

This was not a traditional web application like ASP.NET, but rather a Windows Communication Foundation (WCF) server with various transport bindings.

The modules were distributed across separate projects as various DLLs and assemblies using MEF (Managed Extensibility Framework).

The database connection was established using the Entity Framework with the help of POCOs.

The system could be administrated via a web interface from a browser.

**Refactoring merchandise management system** - *CodeIgniter, Doctrine, MySQL, PHP, Shopware, MVC*

I have initiated the process of restructuring an existing and complex merchandise management system in such a way that it could be gradually converted to current PHP modules and techniques.

At the end it was not necessary to rewrite the entire system at once, but it could be done behind the scenes as needed.

**Ride pooling backend and driver app** - *MongoDB, Node.js, PostgreSQL, React Native, TypeScript*

I was responsible for the development of a ride pooling app, including a corresponding backend.

While the ride pooling algorithm and the customer app were developed by students, I developed a corresponding driver app.

**Telemetry Service** - *Azure Kubernetes Service (AKS), Docker, Go, MongoDB, REST API*

This microservice is used to capture any application data for later evaluation during the testing and production phase, for example by the developers or the data science department.

The implementation of the underlying REST API was done in Golang, which speeds up access by the respective applications and leads to a low consumption of resources within a Kubernetes cluster.

The underlying applications can send data of any kind and are distinguishable from each other by IDs.

## SKILLS

**.NET (C# / VB.NET)**

**Active Directory / LDAP (Azure, on-premise)**

**AI (OpenAI)**

**API (REST, SOAP)**

**C / C++**

**Caching (Memcached, Redis)**

**Cloud (Azure)**

**Container (Docker, Kubernetes)**

**DevOps (Azure, GitHub)**

**E-Commerce (Magento, OXID, Shopify, Shopware)**

**Data structure (INI, JSON, YAML, XML)**

**Go**

**HTML5 / CSS / LESS / SASS**

**IDE (Eclipse, JetBrains, NetBeans, Microsoft)**

**Java (OpenJDK, Spring Boot)**

**JavaScript / TypeScript (Browser, Node.js, V8)**

**Mobile development (Android, Cordova, Flutter, iOS)**

**MVC (ASP.NET, CodeIgniter, Express.js, self developed)**

**Network (HTTP, TCP/IP, WCF)**

**NoSQL (Mongo DB)**

**Operating System (DOS, Linux, MacOS, Windows)**

**ORM (Doctrine, Entity Framework, self developed)**

**PHP**

**React (Next.js, React Native)**

**SQL (MSSQL, MySQL, PostgreSQL, SQLite)**

**Terminal (bash, CMD.EXE, PowerShell, sh, zsh)**

**UI (Bootstrap, MUI for React, self developed, Vuetify, WinForms, WPF)**

**Version Control (Git, Subversion, Team Foundation Server)**

**Vue**