

## 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 Developer 2024 - 2025  
Aktion Mensch e.V., Bonn, Germany

Contributing to agile product development by focusing on front-end tasks, integrating features into apps, and ensuring alignment of concept, design, and technology through collaborative sprints.

- AI development (Azure OpenAI)
- API development (Node.js)
- Cloud based software development (Azure)
- DevOps (Azure, Turorepo)
- Frontend development (Next.js / React.js)
- Tooling (Go, TypeScript)

Senior Full-Stack Engineer 2024  
Ordio GmbH, Cologne, Germany

Contributing to agile product development by focusing on front-end tasks, integrating features into apps, and ensuring alignment of concept, design, and technology through collaborative sprints and creative problem-solving.

Additionally, working with AWS services, including managing resources in S3 buckets, accessing AWS Secrets Manager, and setting up resources in GitLab pipelines using Terraform.


- API development (PHP, Symfony)
- Cloud based software development (AWS)
- Database development (MongoDB, MySQL)
- DevOps (GitLab, Terraform)
- Frontend development (Next.js / React.js, React Native)
- Tooling (Node.js)

Senior Software Engineer 2023 - 2024  
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.


## Marcel J. Kloubert


Senior Full-Stack Developer


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

 Germany


 Europe/Berlin

 kloubert.blog/ (<http://kloubert.blog/>)

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

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

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

- 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

doppelplus GmbH, Aachen, Germany

Working student

- 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

**Refactoring and modernizing React component library with multi-format support** - 2024-06 to 2024-09 - *Jest, Node.js, React, Rollup, Vite, TypeScript*

In this project, I was responsible for refactoring a React component library, evolving it to version 2.0. The goal was to enhance performance and maintainability while embracing modern development practices.

This is a list of key tasks I did:

- transitioned the build system to Vite, improving development speed
- incorporated Vitest for faster feedback and higher code quality in testing
- reorganized file/folder structure to enforce scalability and navigation standards
- supported multiple output formats (CommonJS, ESM, UMD) for broader integration
- integrated latest Storybook for visual testing, documentation, and collaboration

**Development of unified modules for cross-project standardization** - 2024-06 to 2024-09 - *ESLint, Node.js, Prettier, Rollup, TypeScript*

I developed a **SDK**, a platform-agnostic TypeScript module aimed at consolidating reusable backend types and functions. This module was designed to be easily integrated across various projects, allowing teams to leverage consistent, well-defined structures and utilities without duplicating code. It significantly improved cross-project collaboration and reduced development time by promoting the reuse of key components.

To enhance code quality and enforce uniform standards, I implemented a **centralized ESLint configuration** that acted as the foundation for all projects within Ordio. This configuration ensured that all teams adhered to the same coding standards and best practices, leading to cleaner, more maintainable codebases. The adoption of a single ESLint setup streamlined the onboarding process for new projects and reduced configuration overhead for developers.

I also developed a standardized **tsconfig.json** file, designed to unify TypeScript configurations across multiple projects. This initiative simplified the project setup process, allowing developers to focus on writing code rather than configuring environments. By establishing a consistent TypeScript configuration, I ensured uniformity in project structure and improved the overall maintainability and scalability of the codebases.

**Custom Visual Studio Code extension for enhanced development workflow optimization** - 2024-06 to 2024-09 - *GitLab, Node.js, React, REST APIs, TypeScript, Visual Studio Code*

In this project, I developed a custom Visual Studio Code extension to enhance the development workflow, which:

- enables temporary local linking of NPM modules without deployment, improving flexibility and speeding up iterations
- optimizes GitLab's Merge Request workflow by integrating custom tools for seamless transitions from local development to code review
- integrates AI-driven suggestions (e.g., ChatGPT, Ollama) to assist with Git commit messages and improve code documentation

- automates common tasks, such as running Tailwind Watch scripts, reducing manual steps and errors
- adds custom status bar buttons for quick access to frequently used commands
- builds dynamic WebViews using React to embed interactive tools directly into the Visual Studio Code environment

**Command Line Interface tool to optimize developer and build workflows** - 2024-06 to 2024-09 - *GitLab, Node.js, Open AI/ChatGPT, REST APIs, TypeScript*

I developed a command-line tool at Ordio to enhance the developer experience and enforce best practices with the following key features:

- strict directory structure checks
- validation of exports for React components, TypeScript types, and functions
- enforcing strict dependency versioning
- automation of exports of new types and functions using file watchers to streamline workflows
- integration with GitLab CI/CD to ensure builds failed if rules were violated, guaranteeing high-quality code
- providing security audits for NPM, PHP, and Python packages
- optimization of Git commands (checkout, push, and merge) to improve workflow speed and efficiency

**Package Manager for Go (<https://github.com/mkloubert/go-package-manager>)** - 2024-05 - *Agile Project Management, AI, DevOps, Go, Open Source, Platform-Independent Software Development*

This is a robust multi-platform command-line tool crafted to enhance developer workflows, especially for Go projects. It tackles key challenges in package management, project automation, and dependency handling, making it a vital tool for developers seeking efficiency and simplicity.

Key Features:

- **Streamlined Dependency Management:** gpm simplifies managing project dependencies, allowing developers to easily add, update, and remove packages, ensuring all libraries are current and compatible
- **Automated Builds and Installations:** gpm automates building and installing executables, maintaining consistency across environments and simplifying deployment, demonstrating proficiency in CI/CD practices
- **Project Initialization and Cleanup:** gpm initializes new projects from predefined templates and includes tools for cleaning up projects, helping maintain an organized and efficient codebase
- **Cross-Platform Compatibility:** gpm works seamlessly across Linux, macOS, and Windows, offering versatility and adaptability in diverse development environments, showcasing broad technical expertise
- **Advanced Scripting and AI Integration:** gpm supports custom scripts for automating tasks and integrates with AI services like OpenAI and Ollama for enhanced productivity, such as natural language branch name suggestions, highlighting innovative use of AI

The development of gpm demonstrates deep expertise in the Go ecosystem and a commitment to improving developer productivity with well-designed tools, effectively addressing package management and project automation challenges.

**In-house Visual Studio Code extension** - 2023-01 to 2024-06 - *Azure Open AI, Azure DevOps, Node.js, TypeScript, VSCode*

To enhance accessibility for the approximately 400 projects in Visual Studio Code, I developed an extension that every developer can install.

Key features of this extension include:

- **Auto-setup of projects**
- **Detection of the underlying Azure DevOps project based on the currently open git repository**
- **Access to personal work items such as bugs, user stories, and tasks**
- **ChatGPT functionalities:**
  - Creation and execution of shell commands from natural language for a file or directory
  - Creation of git branches from natural language
  - Explanation and documentation of code
  - Translation of text in the editor or (i18n) JSON files
- **Opening, browsing, and saving content from zipped files**

- **Scriptable notebooks, similar to those found in <https://github.com/egomobile/vscode-powertools/wiki/Notebooks> (<https://github.com/egomobile/vscode-powertools/wiki/Notebooks>)**

**After Sales web app & backend for service partners and workshops** - 2022-12 to 2024-06 - *Docker, Microsoft Azure, Microsoft Dynamics CRM, Microsoft Power Automate, Next.js, Node.js, PTC ThingWorx, TypeScript*

This internal web application consolidates the most important workflows in the after-sales segment, including:

- Creation of warranty claims by service partners
- Documentation of workshop orders
- Communication with service partners
- Provision of vehicle documentation, such as technical manuals and instructions
- Announcements of service campaigns
- Provision of digital vehicle files
- Ordering of spare parts

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

Additionally, the backend is connected to numerous external systems, including CRM, production, and cloud systems from various providers.

**e.GO URL shortener and QR code generator** - 2022-11 to 2024-06 - *Barcodes, Docker, Go, MongoDB, REST API, SVG*

The URL shortener and QR code generator is a microservice within a large API cluster at e.GO.

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

These are managed through e.GO's internal administration tool, which I previously mentioned.

Additionally, URL parameters can be added to links as needed without adjusting the data record in the administration tool. Since these parameters are not part of the data record, the same 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?language=german>)
- <https://link.e-go-mobile.com/ego-connect-app-store/qrcode?language=english> (<https://link.e-go-mobile.com/ego-connect-app-store/qrcode?language=english>)

**e.GO Configurator (<https://configurator.e-go-mobile.com>)** - 2022-08 to 2024-06 - *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 Sales Portal** - 2022-07 to 2024-06 - *Azure Active Directory, DevOps, Next.js, Node.js, REST API*

This web application was developed concurrently with the e.GO Configurator (<https://configurator.e-go-mobile.com/>) in the second half of 2022 and launched on time for the release of the e.wave X model.

I contributed to both the frontend and backend development, with my primary focus on the frontend.

A potential customer can visit an e.GO sales partner with a configuration code created through the e.GO Configurator (though it's not mandatory). The partner can make final adjustments using a simplified editor if needed and then trigger an order with all necessary documents in the e.GO system.

**e.GO Website (<https://e-go-mobile.com/>)** - 2021-07 to 2024-06 - *Azure DevOps, Docker, Helm Charts, Kubernetes, Microfrontends, Next.js, PostgreSQL, REST API, React.js, TypeScript, Webpack*

As one of two DevOps engineers, I played a key role in ensuring the smooth functioning of the build and release processes for e.GO Mobile's flagship website.

Additionally, I structured the site to allow parts of it to be integrated as micro frontends.

I am currently preparing for the relaunch of the website.

**e.GO internal administration tool** - 2021-07 to 2024-06 - *Azure DevOps, Docker, Microfrontends, Next.js, TypeScript, Webpack*

At e.GO Mobile, we are developing an internal administration tool utilized by the following departments:

- After Sales
- Data Science
- Marketing
- Sales

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

Additionally, it allows for the creation and management of e.GO-specific QR codes and includes a self-developed wiki for sharing information between departments.

The web app is protected by a rights/roles system, 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. I am now leading the refactoring effort to transition it into a microfrontend application with other e.GO developers.

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

**e.GO Connect App (<https://apps.apple.com/de/app/e-go-connect/id1573365927>)** - 2021-07 to 2024-06 - *Azure DevOps, Azure Kubernetes Service (AKS), Docker, Helm Charts, MongoDB, Node.js, PostgreSQL, TypeScript, React Native*

This mobile app initially served to connect customers with their vehicles using TCU data.

The app is now being developed as a platform-independent application based on React Native.

My main task is to ensure the app's connection to the backend, which is implemented as a collection of various microservices. This includes not only the implementation itself but also ensuring 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.

**Telemetry Service** - 2021-07 to 2024-06 - *Azure Kubernetes Service (AKS), Docker, Go, MongoDB, REST API*

This microservice captures application data for subsequent evaluation during both testing and production phases. This data can be analyzed by developers or the data science department.

The REST API is implemented in Golang, enhancing application access speed and minimizing resource consumption within a Kubernetes cluster.

Applications can send various types of data, which are distinguishable by unique IDs.

**Ride pooling backend and driver app** - 2018-07 to 2021-05 - *MongoDB, Node.js, PostgreSQL, React Native, TypeScript*

I was responsible for developing a ride-pooling app and its backend.

While students developed the ride-pooling algorithm and the customer app, I focused on developing the corresponding driver app and API services.

**Desktop app for article maintenance** - 2017-07 to 2018-07 - *.NET Framework, C#, GitLab, MVVM, Win32 API, Windows, Windows Presentation Foundation (WPF)*

For a client, article data needed to be entered into a merchandise management system manually, as it was not available in an organized list like Excel or CSV files.

To optimize this process, I developed a Windows desktop application in C# that streamlined the data entry, allowing for the rapid entry of as many articles as possible.

The application was also designed to enable multiple users to work on processing the list simultaneously.

**Refactoring merchandise management system** - 2016-07 to 2018-07 - *CodeIgniter, Doctrine, MySQL, PHP, Shopware, MVC*

I initiated the process of restructuring a complex merchandise management system to facilitate its gradual conversion to modern PHP modules and techniques.

This approach allowed the system to be updated incrementally behind the scenes, eliminating the need for a complete rewrite at once.

**Modular WCF backend** - 2010-01 to 2013-07 - *.NET Framework, C#, Entity Framework, Managed Extensibility Framework (MEF), MSSQL, WCF*

At EVOCURA, I implemented a multi-modular backend.

Unlike traditional web applications like ASP.NET, this was a Windows Communication Foundation (WCF) server with various transport bindings.

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

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

The system could be administered via a web interface accessible from a browser.

**Mail merge function via SOAP as Word add-in** - 2008-02 to 2009-06 - *Microsoft Word, PHP, SOAP, C#, Windows Forms, XML, Visual Studio Tool for Office (VSTO)*

I developed a Word add-in to connect to the web-based membership management system by GRÜN. This add-in retrieved member data through a SOAP interface.

Using a field selection dialog, the add-in allowed users to integrate member data into current Word documents and store it as XML within the document itself.

Anyone with the add-in could later access all the data again.



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

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

**AI (Ollama, OpenAI)**

**API (GraphQL, REST, SOAP)**

**C / C++**

**Caching (Memcached, Redis)**

**Cloud (AWS, Azure, Google)**

**Container (Docker, Kubernetes)**

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

**DevOps / GitOps (Azure, GitHub, GitLab, Gitea, Terraform)**

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

**Go**

**HTML5 / CSS / LESS / SASS**

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

**Java (OpenJDK, Spring Boot)**

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

**Messaging (NATS)**

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

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

**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**