









# Martin Beckmann

## Softwareentwickler

Fullstack mit Fokus auf  
Backendentwicklung

-  0152 01087786 
-  martin.beckmann@protonmail.com 
-  github.com/MartinquaXD 
-  /in/beckmannmartin 

## Systems Development

Rust  
C++  
Qt

## Datenbanken

MongoDB  
Elasticsearch

## Web Development

TypeScript  
React + Redux  
Node.js  
HTML5 / Sass

## Softskills

Ewig lernend  
Vielseitig interessiert  
Engagiert  
Kreativ  
Lösungsorientiert  
Offen & immer gut gelaunt

## Sonstiges

Englischkenntnisse  
Git  
Jira / Kanban

# Umgesetzte Projekte

## Eigenmotivierte Evaluierung von Rust als Alternative für eine Backendkomponente

Um den Performanceoverhead von Node.js zu messen wurde ein existierender Server nochmals in Rust mit Tokio implementiert.

- Bis zu 50% reduzierte Latenz
- Entfernung der V8-Engine und damit verbunden reduzierte RAM-Auslastung, Binärgröße, Startzeit und genutzte Prozessorzeit
- Schaffen einer Basis für Support von Multithreading

genutzte Technologien: Rust, Tokio, Websockets

## Weiterentwicklung eines nativen Caching Systems fuer Android- und iOS-Apps

Diese Komponente abstrahiert die Netzwerkkommunikation der Apps mit dem Backend sowie das Caching empfangener Daten.

- Behebung bekannter Konnektivitätsprobleme
- Implementierung frühzeitiger Auslieferung teilweise gecachter Anfragen
- Schreiben von Integrations- und Unittests

genutzte Technologien: C++ 14, Websockets

## Konzeptionierung und Implementierung diverser Entwicklungstools

Um den Entwicklungsablauf zu beschleunigen und übliche Fehler zu minimieren, wurden teilweise neue Tools geplant und implementiert, sowie bestehende Tools weiterentwickelt.

- Umsetzung eines Schema-First-Ansatzes für die Backendentwicklung mit Node.js
- Codegenerierung für Typescript, Kotlin und Swift
- Entwicklung von Tooling für Backend- und Frontendtests mit Jest
- Entwicklung eines Plugins für JetBrains IDEs

genutzte Technologien: Typescript, Node.js, Jest, Java

## Planung und Entwicklung eines Floating-License-Servers

Um die unrechtmäßige Nutzung vom Squish GUI Testtool zu minimieren, wurden alt bekannte Lizenzschlüssel durch einen Lizenzserver ersetzt.

- Refactoring einiger Bereiche einer großen Legacy Codebase
- Entwicklung neuen Codes kompatibel mit einigen alten und exotischen Compilern und Betriebssystemen
- Steigerung des Umsatzes da Lizenzen nicht mehr kopiert werden koennen

genutzte Technologien: C++ 11, C++98, Qt 5

## Full-Stack-Entwicklung diverser Content-Management-Systeme und Webanwendungen

Um den individuellen Bedürfnissen jedes Kunden gerecht zu werden, wurde jedes Projekt von Grund auf neu entwickelt.

genutzte Technologien: Typescript, React, Node.js, MongoDB, Elastisearch

# Erfahrung

03.2020 - heute	Qt Systems Developer	Froglogic / The Qt Company
10.2016 - 03.2020	Studium Technischer Informatik	HAW Hamburg
06.2016 - 03.2020	Werkstudent Webentwicklung/Tooling	Appbase Hamburg