# Florian Stöhr

AMEBA Technologies GmbH & Co. KG

| | |
|---|---|
| Contact | Lerchenweg 7, 86492 Egling, Germany<br>+49-173-3514000<br>ich@florian-stoehr.de |
| Profile | Freelance Software Developer/Architect Rust, Python (CPython and IronPython), Assembler (16/32/64 bit, DOS/Win32/Win64/Linux/BSD), C++ (Qt, wxWidgets, VCL, MFC, std-C++, STL, Threading, boost), Kotlin, Swift, Reversing (native IDA, CIL IDA+ILDASM+Telerik)<br>Auditor for QM systems conforming to ISO 9001<br>AWS Certified Developer Associate |
| Professional experience | 10/2009–now<br>**Freelance work/own company**<br>06/2005–09/2009 Aipermon GmbH & Co. KG München<br>**Server development**<br>06/2001–05/2005 LogIn & Solutions AG Gersthofen<br>**Client and server development**<br>01/2001–06/2001 Skybeamer GbR Aulzhausen<br>**Co-founder (development)**<br>02/1997–12/2000 Stöhr Software Egling<br>**IT-Services and software development (part-time)** |
| Qualification | 09/2021 Amazon Certification<br>**AWS Certified Developer Associate**<br>10/2009–10/2009 TÜV Süd Akademie GmbH Augsburg<br>**QM auditor (ISO 9001)**<br>09/1999–06/2001 BOS (Technik) Augsburg<br>**Fachhochschulreife**<br>09/1995–02/1999 Deutsche Bahn AG München<br>**Apprenticeship (Communication/Electronics)** |
| Language skills | • Native language: German<br>• English: Business fluent<br>• Basic knowledge in Hindi (including Devanagari) |
| Additional | • Medizinprodukteberater (EUROCAT) |
| Programming languages | Rust, Assembler, C/C++, C#, Delphi/Pascal, Python, Flutter/Dart, Kotlin, Swift, bash, lex/yacc, Natural, JavaScript |
| UI-Toolkits | Slint, Flutter, Qt/QML, SwiftUI, Jetpack, VCL, wxWidgets, MFC, Win32 |
| Debugger | TurboDebugger, gdb, DDD, WinDBG, IDA Pro |
| Version control systems | git, subversion, cvs, Mercurial, TFS |

# Florian Stöhr

AMEBA Technologies GmbH & Co. KG


| | |
|---|---|
| Databases | PostgreSQL (primary database), Access, BDE, MySQL, ODBC, SQLite, SQL-Server. |
| Operating systems | Linux (Debian and derivates, Alpine), macOS, Windows, BSD, iOS, Android, DOS |
| Communication | UART, I2C, SPI, USB, CORBA, Sockets (TCP/IP, UDP/IP, UDP-Multicast), SSL/TLS, HTTP, SOAP, RS232, parallel, SMTP, Google Protocol Buffers, JSON via REST, WebSockets, ... (own protocols) |
| Additional technologies | docker, gcc, Borland C++ Builder, Borland Delphi, Lazarus, Watcom C++, MS-Visual C++, MS-Visual Basic, MS-Visual C#, Mono, MonoTouch, MonoGame, XNA, IronPython, TurboPascal, vi, (n)curses, TurboVision, cURL, libUSB, OpenGL, GNU pthreads, Multi-Threading, STL, DirectX, Orbit ORB, MIDAS, make, GNU Autotools (autoconf, automake, libtool, ltdl), bjam, gmake, qmake, boost, Apache, Postfix, Samba, spamd, QtCreator, Django, CherryPy, PySide, C++/Python-interfaces with boost::python, ImageMagick, InstallShield, NSIS, Windows Services, FinBanks, NUnit, WPF, jQuery, nginx, grafana/prometheus, htmx |
| Branches | Military, Arms industry, Medical devices, 3D-Rendering, Trading network, eCommerce, Image processing, Logistics, Warehouse management, Railways, Dating, public administration, Computer games, Pharma, Finance, Market research, Tax accounting |

# Projects

–This is an excerpt. Not all projects are included. This list contains only the projects done in my freelancing career since 2010. I can happily explain the projects of the years before (focus on C++, both server-side and desktop applications) on request.–

## 07/2025–07/2025: Improvements to electronic control unit

Branch:
Motorscooter (2019 NIU N-GT)

Role:
Reversing, Development

Project:
My NIU N-GT's firmware crashed every 30 minutes, slowing down the vehicle from 75kph to 45kph independently of the traffic situation. The manufacturer's "experts" claimed that the electronic control unit of the vehicle was "broken". The real reason for the crash is that the vehicle's badly implemented firmware is not able to handle the case of a deactivated SIM card. The solution was to sniff some communication on the PCB, then desolder the SIM module and replace it by my own microcontroller, running my own firmware. This emulates the SIM module and the internet traffic to the manufacturer's server in China completely. The quality vehicle now reports on its dashboard that "Cloud Services" are "Connected" and the signal quality of the (deactivated) SIM (which is lying on my desk) is "excellent". The crash however is gone and the vehicle is safe to be used in traffic again.

Technologies:
Saleae logic analyzer, C++, ATmega4809, Scalpel, Micoscope, Soldering station, Patience

Platforms:
Bare metal

## 02/2025–ongoing: Porting from Scala to Python

Branch:
Hosting (noris network AG)

Role:
Architect, Developer

Project:
Porting of a REST-API as well as a stored procedures generator from Scala to Python

Technologies:
Python, FastAPI, PostgreSQL

Platforms:
Linux/Docker

### 01/2025–06/2025: Peer emulator arms industry

Branch:
Arms industry (Rohde & Schwarz SIT GmbH

Role:
Development

Project:
Wireless radio SIP/SCIP emulator for testing hardware device's protocol implementation according to NATO requirements

Technologies:
Python

Platforms:
Linux/Docker

### 09/2024–ongoing: Navigation for TMS machine

Branch:
Medical device (SEBERS Medical)

Role:
Design, Development

Project:
TMS devices are used for treatment of depression by firing strong magnetic pulses into the brain at a specific spot. Till now, finding the spot involved lots of error-prone manual measurements. This project is an extension of TMS machines to provide computerized navigation: By the help of very precise sensors, the position of the coil in space can be detected. The software helps doctors to position the coil exactly at the right spot. Navigation is done like a 3D "flight" over navigation points on the skull, with depth illusion effects. Position data is read via USB from the sensors. The UI toolkit in the project was Slint.

Technologies:
Rust, libusb, Slint, Crate "three-d", OpenGL

Platforms:
ARM / Embedded Linux

### 11/2023–06/2024: Rust backend for monitoring

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
Within FUSE, monitoring is done by collector servers that gather information from docker nodes and backend services running inside those nodes. The information is then stored in a PostgreSQL database and provided to a HTML frontend via server-side rendering supporting partial page updates with the htmx technology. This backend service "cooks" the raw monitoring information of the database and presents HTML to the frontend.

Technologies:
Rust, tokio (as an async runtime), PostgreSQL, sailfish, html, htmx, Bootstrap

Platforms:
MacOS, Alpine Linux/Docker

### 09/2023–07/2024: Requirements Engineering Military

Branch:
Federal Republic of Germany / Department of Defence

Role:
Requirements Engineer

Project:
NuKOM is a successor to the original teleprinter network replacement project done by Airbus. It is a system for exchanging messages and commands between military installations, both within Germany as well as with NATO partners. NuKOM is rated up to the NATO SECRET level for secure communication and runs on a network that is physically separated from the internet. My role in the project is requirements engineering considering the software part of the implementation.

Technologies:
(Shielded by NDA)

Platforms:
(Shielded by NDA)

### 09/2023–11/2023: Rust websocket server

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
Notifications for new messages are sent to running mobile apps via WebSocket messages as standard push notifications are too slow for this use case. The original WebSocket server in FUSE was written in x86-64 assembler and needed to be replaced by an implementation in Rust because additional requirements for operation monitoring came up which would have been too much work to add to the assembler version. The server is done in a manual way (raw socket I/O) according to RFC6455. It receives input from the FUSE backend via a compact binary TCP protocol and talks to mobile apps on the output side via WebSockets. Statistics and errors are sent to the FUSE monitoring infrastructure.

Technologies:
Rust, tokio (as an async runtime), tokio-sockets

Platforms:
MacOS, Alpine Linux/Docker

### 07/2023–08/2023: Machine control library for manufacturing

Branch:
Manufacturing (Aucos AG, Aachen)

Role:
Developer

Project:
For a manufacturing installation at a customer, Aucos needs to interface with a crane controlling system built by another manufacturer. The library will handle this interface.

Technologies:
C++, Qt

Platforms:
Win64, MacOS

### 03/2023–07/2023: Rust monitoring infrastructure

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
For monitoring both system load/health and technical statistics, a monitoring infrastructure for FUSE was created. It consists of a database, an information collector server that is installed once per docket node and a client library crate that gets compiled into all services running on the node. The monitoring tool itself gathers CPU, memory state and uptime. Docker services and statistics are gathered directly on the host and fed to the collector service via a shared directory mount. Other services report both statistics data (e.g. active websocket connections and API call statistics) as well as log messages via a TCP interface, fed by the client library crate. Results are gathered in the collector service, condensed and regularly written to the database.

Technologies:
Rust, tokio (as an async runtime), tokio-sockets, PostgreSQL, tokio-postgres

Platforms:
MacOS, Alpine Linux/Docker

### 12/2022-02/2023: Native Android version for dating app

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
A dating app previously implemented in RectNative needed to be fully rewritten as a native Android app.

Technologies:
Kotlin, Jetpack Compose, Voice record, Facebook login, Google login, Google analytics, Firebase messaging, REST-APIs, WebSockets, Camera, Gallery, Location services

Platforms:
Android API-Level 24+

### 11/2022-01/2023: Native iOS version for dating app

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
A dating app previously implemented in RectNative needed to be fully rewritten as a native iOS app.

Technologies:
SwiftUI, Voice record, Facebook login, Google login, Google analytics, Firebase messaging, REST-APIs, WebSockets, Camera, Gallery, Location services

Platforms:
iOS 15+

### 01/2022-03/2023: Test infrastructure

Branch:
Railways (SIEMENS Mobility, Erlangen)

Role:
Development

Project:
Siemens Mobility uses a custom (based on behaverunner) framework to test the on-board software of trains (actual vehicle control, not entertainment systems). The job was to fix bugs in the framework and add new features.

Technologies:
Python

Platforms:
Linux, Docker

### 06/2022–10/2022: Rust background services for dating platform

Branch:
Dating/Social Network (FUSE UG)

Role:
Design, Development

Project:
As part an infrasture porting to rust (see backend project below), additional backend services that run outside of critical execution paths had to be implemented. This includes communication with Google Firebase for push notifications to mobile apps, database periodic cleanups and parts of the user registration process.

Technologies:
Rust, tokio (as an async runtime), PostgreSQL

Platforms:
MacOS, Alpine Linux/Docker

### 12/2021–04/2022: Flutter-Frontend for platform administration

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Development

Projekt:
FUSE required a web-base admin interface that at least partially mimics the UI interface of the native mobile apps in oder to make new operators productiv as quickly as possible. Due to the complexity of the UI, Flutter was chosen in favor of a classic CSS/HTML approach, using the Flutter/Web target. The application allows to view user's chats, location, image and audio material as well as deactivating or deleting users. Furthermore, some statistic data can be viewed.

Technologien:
Flutter/Dart

Plattformen:
nginx/Web

### 11/2021–06/2022: Rust-backend for dating platform

Branch:
: Dating/Social Network (FUSE UG)

Role:
: Design, Development

Project:
: Old C++-based backends of the platform should be ported to a new, unified Rust-based architecture. The backend provides REST APIs (using actix-web) and access several PostgreSQL databases via a connection pool done with Tokio-Postgres, with multiple pools for read-only and separate read-write connections. Testing is done with external pytest scrips against the API and a test database. The release version of the backends is built within a special docker container and then packaged into an exec-only release container.

Technologies:
: Rust, actix-web, tokio (as an async runtime), PostgreSQL, Python, pytest

Platforms:
: MacOS, Alpine Linux/Docker

### 07/2021–10/2021: Fixing medical device

Branch:
: Medical devices (Inveox GmbH, Munich, Germany)

Role:
: Developer

Project:
: A medical device prototype used for histopathology (analysis of tissue samples) which is controlled by a complex state machine implemented in Python suffers from numerous software bugs. Within this engagement, the machine should be debugged and made ready for clinical trials.

Technologies:
: Python, docker, Redis

Platforms:
: Windows

### 12/2018–06/2021: Backends for quality improvement infrastructure

Branch:
Market research (GfK SE, Nuremberg, Germany)

Role:
Architect, Tech lead, Developer

Project:
The engagement consisted of in total three applications used in varous steps of the GfK data processing pipeline. All applications take various parts of raw input sales/turnover/price data, apply datascience models over the data and write results back into pipeline as well as visualizing them for supervisor users. Users can alter the data and reinject into pipeline. All subprojects need a pretty complex coordination infrastructure in order to make DS model runs efficient and scalable.

Technologies:
Python, docker, CherryPy, PostgreSQL, RabbitMQ, REST-JSON, Microservices, grafana/prometheus

Platforms:
Linux, Web

### 01/2020–05/2021: Machine control

Branch:
CNC machine control (RWT GmbH, Munich, Germany)

Role:
Developer

Project:
A Win32 desktop-based machine control program and a telegram program written in C++ needed to be modernized for a web-based approach with an additional local program running for non-browser bound local tasks, though controlled via the web. Created a new backend in Python/Django+GraphQL and a frontend in nuxt.js. The locally running windows program was implemented in Qt, communicating via GraphQL/JSON.

Technologies:
Python, Django, Qt, GraphQL

Platforms:
Windows 64

### 10/2018–09/2021: Tax processing/visualization

Branch:
Tax accounting (U-Know! BI OHG, Regensburg, Germany)

Role:
Architect, Tech lead, Developer

Project:
This application group extracts raw tax data from the market leader ("DATEV")'s database and runs various analysis and aggregation steps over it before providing the data to both a vue.js frontend and a react.native mobile app. The project contains numerous components, notably a scanner, importer, upload backend, api backend and a native Windows executable for extraction and upload processing. The system is designed in a highly optimized and scalable way and can be run by a customer also in self-hosted version.

Technologies:
Python, docker, CherryPy, PostgreSQL, REST-JSON, C++, Microservices

Platforms:
Linux, Windows

### 05/2017–06/2021: Dating platform

Branch:
Dating (FUSE Dating UG, Berlin, Germany)

Role:
Architect, Tech lead, Developer

Project:
FUSE is a social networking/dating platform running on mobile devices. It includes profile matching, chats, imaging and audio, trying to give an authentic view of the match that is going further than other platform currently do. FUSE is currently running in beta testing. I did the architecture and the backend implementation. The frontend is done in ReactNative, primarily by another developer though I did some bugfixing and parts of the flow logic in the frontend as well.

Technologies:
Python, docker, C++, nginx, PostgreSQL, ReactNative

Platforms:
Linux, Android, iOS

### 09/2015–07/2017: Embedded blood pressure/heartrate monitor

Branch:
Medical devices for ICU (PULSION medical systems (Getinge group), Munich, Germany)

Role:
Developer

Project:
The company's mother is one of the international market leaders for ventilation systems and ICU patient supervision blood pressure/heart rate monitors. The stand-alone systems run on Windows-based PCs in a small form factor with the application written in Delphi and Assembler. In this project, we ported the application/hardware to an embedded ARM32 (ARM Cortex M4) based device running FreeRTOS to be used as a (hardware) plug-in module inside a larger monitoring system of a foreign company. Additionally, a C++ SDK to talk to the device and a Python wrapper for it were developed.

Technologies:
Delphi, x86-Assembler, C++, ARM32, embedded, Python, SDK design

Platforms:
ARM32, Windows

### 09/2014–09/2015: Logistics/Warehouse control/Campaign planning

Branch:
Fashion industry (Best Secret GmbH, Munich, Germany)

Role:
Developer

Project:
In this project, we developed a logistics system that manages all goods flow processes both from the booking side as well as from the actual machine operation side (this had a highly automated warehouse where fetching the items of an oder is actually done by hardware robots). Other jobs included interfacing the label printers, weighters and also the marketing campaign planning application. Also included lots of Python-based helper tools for example for batch service compilation.

Technologies:
C#.NET, ASP.NET, WCF, WPF, Mercurial, PDFFileWriter, NHibernate, Autofac, VB.NET, Python

Platforms:
Windows64

### 06/2014–07/2014: Magletics frontend for brain stimulator

Branch:
Medical devices (Mag&More GmbH, Munich, Germany)

Role:
Developer

Project:
For a special application of their magnetic brain stimulator device, the customer needed a new C# frontend based on WPF. This version directly communicates with the stimulator via an FTDI-chip over USB.

Technologies:
C#, WPF, .NET, FTDI

Platforms:
Windows64

### 10/2013–06/2014: Webapplication for pharma accreditation

Branch:
Pharmaceutical drugs (EXTEDO GmbH, Munich, Germany)

Role:
Development (backend)

Project:
Development of a C#-based web application for legal registration of documentation, ingredients lists and management of adverse effects of pharmaceutical drugs. The software support product management over the entire lifecycle. Due to time pressure on legal side, this temporarily contained an Excel interface as well while waiting for completion of the frontend. Also, a PDF exporter was included. The code of the application was partially generated by a C# generator implemented in Python.

Technologies:
C#, ASP.NET, EntitiyFramework, SQL-Server, TFS, Python, OpenXML/ClosedXML, PDF-FileWriter library

Platforms:
Windows64

## 10/2012–06/2013: Integration of the Python interpreter into 3D visualizer software

Branch:
3D-Rendering (Realtime Technologies (RTT) AG, Munich, Germany)

Role:
Architecture, Development

Project:
Integration of the Python interpreter into a 3D visualization software for automotive and aero industry. The interpreter hostmodule is loaded as a PlugIn into the Qt/C++ application and then split into subinterpreters. Execution of the actual script runs in background threads. In addition, the PlugIn module provides a C-based internal module that allows callables of the scrips to be executed on the Qt main thread (e.g. for UI output).

Technologies:
Python, Python C API, Qt, C++

Platforms:
Windows64

## 12/2011–08/2012: Parser for NATURAL sourcecode; IronPython tool

Branch:
E-government (it&more GmbH, Munich, Germany)

Role:
Architect, Developer

Project:
Due to changed legal requirements, the people registration software of the city of Munich needed to be changed to be unicode-compatible. The application is huge and written in non-unicode NATURAL with 3 million lines of code. As this would be too hard (especially given that the number of NATURAL programmers is pretty limited), we developed a NATURAL parser in C# that is capable of actually understanding the NATURAL code (like a compiler frontend) and emit different sourcecode with unicode support. It is not a trivial statement replace, as some commands need to be replaced by sequences of other commands and various data structures need to be resized/changed programmatically as well as automated UI module changes and so on. In addition, for cases that need to be scripted, we implemented an IronPython interface to the transpiler that allows to script it. The IronPython tool was also used to debug as it could inspect the internal tree state of the transpiler at runtime.

Technologies:
C#, .NET, Python, IronPython

Platforms:
Windows 7 64, Windows Server 2008 R2

Florian Stöhr
AMEBA Technologies GmbH & Co. KG


# Publications

–Excerpt–

- **ISBN 978-3936546-48-4 (C&L-Verlag, 2007)**
  Book "GNU Autotools - UNIX-Buildsysteme mit Autoconf, Automake und Libtool"
- **Toolbox - Entwicklermagazin, 02/2008**
  Article ".NET-PlugIns mit C#"
- **Toolbox - Entwicklermagazin, 03/2007**
  Article "Make my file - so funktioniert GNU Make"
- **iX - Magazin für professionelle Informationstechnik, 09/2005**
  Article "Verschlüsselte Block-Devices mit NetBSDs CGD"
- **freeX - Magazin für freie UNIX-Systeme, 06/2005**
  Article "VND gut komprimiert"
- **freeX - Magazin für freie UNIX-Systeme, 04/2005**
  Article "(Net)BSD auf SSH-Rootservern"
- **Postfix**
  Artikel "Postfix@NetBSD - Setting up a secure real-life mailserver with SMTP AUTH and anti-spam capabilities"
- **NetBSD-Manual ("The NetBSD Guide")**
  Section "The cryptographic device driver (CGD) - Creating an encrypted CD/DVD"
- **OpenBSD zyd(4) device driver**
  Prototype of a device driver for ZyDAS ZD-1211-based USB-WLAN-Adapters.
- **neb-cd512**
  Program for adjustment of kernel-in-core-labels of a CD/DVD to another sector size
- **neb-wipe**
  Disk eraser, implementing the Gutmann-multipass algorithm, for NetBSD.
- **neb-hdtoolbox**
  Tool for cross-installing NetBSD/amiga. Can read/write the RDSK/PART records for the amiga (comparable to a partition table on the IA32 platform)
- **cloop**
  Compressed file system for Live-CD images (compressor control program)