Florian Stöhr

AMEBA Technologies GmbH & Co. KG

Kontakt	Lerchenweg 7, 86492 Egling +49-173-3514000 ich@florian-stoehr.de
Profil	Freiberuflicher Softwareentwickler Rust, Python (CPython und IronPython), Assembler (16/32/64 bit, DOS/Win32/Win64/Linux/BSD), C++ (Qt+QML, wxWidgets, VCL, MFC, std-C++, STL, Threading, boost), Flutter/Dart, Kotlin, Swift, C#, Reversing (native IDA, CIL IDA+ILDASM+Telerik) Auditor für QM-Systeme nach ISO 9001 AWS Certified Developer Associate
Berufserfahrung	10/2009–jetzt **Freiberufliche Tätigkeit** 06/2005–09/2009 Aipermon GmbH & Co. KG München **Abteilungsleiter Server-Entwicklung** 06/2001–05/2005 LogIn & Solutions AG Gersthofen **Entwicklungsleiter, stellv. Abteilungsleiter** 01/2001–06/2001 Skybeamer GbR Aulzhausen **Gesellschafter, Bereich Softwareentwicklung** 02/1997–12/2000 Stöhr Software Egling **Nebenberuflich EDV-Dienstleistung und Softwareentwicklung**
Ausbildung	10/2009–10/2009 TÜV Süd Akademie GmbH Augsburg **Ausbildung zum QM-Auditor ISO 9001** 09/1999–06/2001 BOS (Technik) Augsburg **Fachhochschulreife** 09/1995–02/1999 Deutsche Bahn AG München **Ausbildung zum Kommunikationselektroniker**
Sprachkenntnisse	• Muttersprache Deutsch • Hervorragende Kenntnisse in Englisch in Wort und Schrift • Grundkenntnisse in Hindi in Wort und Schrift (Devanagari)
Sonstiges	• Medizinprodukteberater (EUROCAT)
Programmiersprachen	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
Versionskontrolle	git, subversion, cvs, Mercurial, TFS
Datenbanken	PostgreSQL (primäre Datenbank), Access, BDE, MySQL, ODBC, SQLite, SQL-Server

Florian Stöhr

AMEBA Technologies GmbH & Co. KG

Hardware	ARM, Alpha, Amiga, Apple, Ascii/X-Terminals, Drucker, embedded, HP, Mikrocontroller, Motorola, PC, Silicon-Graphics, SUN, MFA-8085, RaspberryPi
Betriebssysteme	CP/M, Mac OSX, MS-DOS, Solaris, Unix allgemein, Windows (ab 2.0), Windows CE (Mobile), NetBSD/FreeBSD/OpenBSD, Linux (Alpine, Debian und Derivate, SuSE, CentOS), iOS, Android
Kommunikation	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, u.v.a. (Eigenbau)
Sonstige Technologien	docker, bash, gcc, Borland C++ Builder, Borland Delphi, Lazarus, FASM, NASM, TASM, Watcom C++, MS-Visual C++, MS-Visual Basic, MS-Visual C#, CIL, Mono, MonoTouch, MonoGame, XNA, IronPython, TurboPascal, vim, (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, Firebase, Samba, spamd, fail2ban, QtCreator, Django, CherryPy, PySide, C++/Python-Schnittstellen mit boost::python, ImageMagick, InstallShield, NSIS, Windows Services, FinBanks, NUnit, WPF, WCF, jQuery, nginx, grafana/prometheus, htmx
Branchen	Militär, Rüstungsindustrie, 3D-Rendering, Handelsnetz, eCommerce, Bildverarbeitung, Logistik, Lagersteuerung, Lagerverwaltung, Medizintechnik, Medizingeräte, Bildungswesen, Verkehr, Eisenbahn, öffentliche Verwaltung, Computerspiele, Pharmazulassung, Finanzen, Gastronomie, Dating, Baubranche, Marktforschung

Projekte

–Auszug–

07/2025–07/2025: Verbesserung an Steuergerät

Branche:
Motorroller (NIU N-GT BJ 2019)

Rolle:
Reversing, Entwicklung

Projekt:
Mein NIU N-GT stürzte alle 30 Minuten ab mit der Folge, dass das Fahrzeug unabhängig von der Verkehrssituation von 75 auf 45 abbremst. Die "Experten" des Herstellers behaupten, dass das Hauptsteuergerät "kaputt" wäre. Der eigentliche Grund ist, dass die vom Hersteller mangelhaft implementierte Firmware leider nicht in der Lage ist, mit einer deaktivieren SIM-Karte umzugehen. Die Lösung bestand darin, Kommunikation auf dem PCB abzuschnüffeln, dann das SIM-Modul abzulöten und durch einen eigenen Microcontroller zu ersetzen. Meine eigene Firmware auf diesem emuliert das SIM-Handling komplett inkl. dem Internettraffic zum chinesischen Server. Das Qualitätsfahrzeug meldet jetzt auf seinem Dashboard, dass sich der "Cloud Service" im Zustand "Connected" befindet und die Signalstärke der (deaktivierten und ausgebaut auf meinem Tisch liegenden) SIM-Karte "hervorragend" wäre. Jedenfalls ist der Absturz weg und Verkehrssicherheit ist wieder hergestellt.

Technologien:
Saleae Logicanalyzer, C++, ATmega4809, Skalpell, Mikroskop, Lötstation, Geduld

Plattformen:
Bare metal

02/2025–laufend: Portierung Scala nach Python

Branche:
Hosting (noris network AG)

Rolle:
Architektur, Entwicklung

Projekt:
Eine bisher in Scala implementierte REST-API nebst Erzeugungsmodul für stored procedures soll auf Python portiert werden

Technologien:
Python, FastAPI, PostgreSQL

Plattformen:
Linux/Docker

01/2025–06/2025: Gegenstellenemulator Rüstungsindustrie

Branche:
Rüstungsindustrie (Rohde & Schwarz SIT GmbH)

Rolle:
Entwicklung

Projekt:
Funkgeräteemulator zum Testen verschlüsselter SIP/SCIP-Kommunikation gemäß NATO-Vorgabe

Technologien:
Python, SIP/SCIP, proprietäre Verschlüsselung

Plattformen:
Linux/Docker

09/2024–laufend: Navigationslösung für TMS-Medizingerät

Branche:
Medizingerät (SEBERS Medical)

Rolle:
Design, Entwicklung

Projekt:
TMS-Geräte dienen zur Behandlung von Depressionen durch Beschießen des Gehirns mit sehr starken Magnetfeldern an einer bestimmten Stelle. Bisher musste die Stelle manuell ausgemessen werden. Dieses Projekt ist die Erweiterung bestehender TMS-Maschinen um Navigation: Mit Hilfe von sehr präzisen Sensoren wird die Position der Spule im Raum bestimmt. Die Software hilft dem Arzt, die Spule exakt auf die richtige Stelle am Kopf zu positionieren. Die Nagivation erfolgt als 3D-"Flug" über Navigationspunkte am Kopf mit Tiefenillusionseffekten. Die Positionsdaten werden über USB von den Sensoren eingelesen. Als Toolkit für die UI kam Slint zum Einsatz.

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

Plattformen:
ARM / Embedded Linux

11/2023–06/2024: Rust-Backend für Monitoring

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Innerhalb von FUSE wird Monitoring durch Collector-Server gemacht, die Informationen über Docker-Nodes einholen und auch aus den Applikationen innerhalb der Nodes mit Statisik-Daten beschickt werden. Die Informationen werden in einer Datenbank gespeichert und dann durch dieses Tool einem Webfrontend via server-seitigem HTML-Renderung zur Verfügung gestellt. Updates werden im SPA-Stil partiell eingebracht, die Steuerung dazu erfolgt mittels htmx im HTML-Code.

Technologien:
Rust, tokio (als async-Runtime), PostgreSQL, sailfish, html, htmx, Bootstrap

Plattformen:
MacOS, Alpine Linux/docker

09/2023–07/2024: Requirements Engineering für Militärprojekt

Branche:
Bundesrepublik Deutschland / Bundesministerium der Verteidigung

Rolle:
Requirements Engineer

Projekt:
NuKOM ist ein Nachfolger der Kommunikationslösung von Airbus, die seinerzeit das ursprüngliche Fernschreibnetz abgelöst hat. Es ist ein System für sichere EMail-ähnliche Kommunikation (etwa Übermittlung militärischer Befehle) sowohl zwischen Dienststellen der Bundeswehr als auch zu NATO-Partnern. NuKOM ist bis zur Geheimhaltungsstufe NATO SECRET zugelassen und wird auf einem dedizierten Netzwerk betrieben welches physisch vom öffentlichen Internet getrennt ist. Meine Rolle im Projekt bestand aus dem Requirements Engineering für den Software-Anteil.

Technologien:
(Geheim/NDA)

Plattformen:
(Geheim/NDA)

09/2023–11/2023: Rust WebSocket server

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Benachrichtigungen über neue Nachrichten werden an die mobilen Apps über WebSockets gemeldet, weil Push-Benachrichtigungen dazu zu träge sind. Der ursprüngliche WebSocket-Server in FUSE war in x86-64-Assembler implementiert und musste durch eine Rust-Version ersetzt werden weil neue Anforderungen für das Eingangsprotokoll und das Reporting von Statistiken zu aufwändig in die Assembler-Version einzubauen gewesen wären. The Server ist "manuell" auf TCP implementiert (rohes Socket-I/O) entsprechend der RFC6455. Er bekommt Input vom Backend über ein kompaktes Binärprotokoll und spricht das WebSocket-Protokoll mit dem mobilen Clients. Statistiken und Logs werden an die FUSE- Monitoring-Infrastruktur verschickt.

Technologien:
Rust, tokio (als async-runtime), tokio-sockets

Plattformen:
MacOS, Alpine Linux/docker

07/2023–08/2023: Maschinensteuerungsbibliothek

Branche:
Maschinenbau (Aucos AG, Aachen)

Rolle:
Entwicklung

Projekt:
Für eine Installation bei einem Kunden von Aucos muss ein Hallenkran über eine SPS-Schnittstelle eines weiteren Lieferanten angebunden werden. In diesem Projekt habe ich eine C++/Qt-Bibliothek entwickelt, die die nötige Kommunikation mit der SPS ausführt.

Technologien:
C++, Qt

Plattformen:
Win64, MacOS

03/2022–07/2023: Rust monitoring infrastructure

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Um den Systemzustand zu überwachen und technische Statistiken zu erfassen wurde eine Monitoring-Lösung für FUSE entwickelt. Sie besteht aus einer Datenbank, einem Informations-Collector-Server der einmal pro Docker-Node installiert wird sowie aus einem Client-Crate der in andere auf dem Knoten laufende Backend-Prozesse einkompiliert wird. Der Server selbst sammelt CPU, Speicherbelegung und Uptime ein, Docker-Informationen werden extern über ein shared directory zur Verfügung gestellt. Andere Services reporten sowohl statistische Daten (etwa aktive WebSocket-Verbingunden oder API- Aufrufstatistiken und Timings) wie auch Logmeldungen über eine TCP-Schnittstelle die von der Client-Bibliothek beschickt wird. Der Collector sammelt diese Infomrationen ein, verdichtet sie und schreibt sie in regelmäßigen Abständen in die Datenbank.

Technologien:
Rust, tokio (als async-runtime), tokio-sockets, PostgreSQL, tokio-postgres

Plattformen:
MacOS, Alpine Linux/docker

12/2022-02/2023: Dating-App native Android-Implementierung

Branche:
Eine zuvor als Prototyp in ReactNative implementiere app sollte nativ auf Android komplett neu Implementiert werden

Rolle:
Design, Entwicklung

Projekt:
FUSE ist eine Datingplattform. Die zugehörige App lag als ReactNative vor, aufgrund der schlechten Wartbarkeit und der mäßigen Performance war eine Neuimplementierung in Android nötig.

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

Plattformen:
Android API-Level 24+

11/2022-01/2023: Dating-App native iOS-Implementierung

Branche:
: Eine zuvor als Prototyp in ReactNative implementiere app sollte nativ auf iOS komplett neu Implementiert werden

Rolle:
: Design, Entwicklung

Projekt:
: FUSE ist eine Datingplattform. Die zugehörige App lag als ReactNative vor, aufgrund der schlechten Wartbarkeit und der mäßigen Performance war eine Neuimplementierung in iOS nötig.

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

Plattformen:
: iOS 15+

01/2022–03/2023: Test-Infrastruktur

Branche:
: Eisenbahn/Schienenfahrzeugbau (SIEMENS Mobility, Erlangen)

Rolle:
: Entwicklung

Projekt:
: Siemens Mobility setzt ein eigens (auf behaverunner aufsetzendes) Framework ein, um die Software von Schienenfahrzeugen in der Entwicklung zu testen (Fahrzeugsteuerung). Hier mussten Fehler behoben und das Framework erweitert werden.

Technologien:
: Python, docker, bash

Plattformen:
: Linux, Windows

06/2022–10/2022: Rust background services für Datingplattform

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Im Rahmen der Infrastrukturportierung auf Rust (siehe nächstes Projekt) mussten zusätzliche Backend-Services die ausßerhalb kritischer Ausführungspfade laufen nach Rust portiert werden. Das behinhaltet die Kommunikation mit Google Firebase für Pushnachrichten an mobile Endgeräte, periodische Datenbankbereinigungen und Teile des Registrierungsprozesses für neue User.

Technologien:
Rust, tokio (als async-runtime), PostgreSQL

Plattformen:
MacOS, Alpine Linux/docker

12/2021–04/2022: Flutter-Frontend für Plattformadministration

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Für die FUSE-Plattform musste eine webbasierte Administrationsoberfläche ertellt werden, die idealerweise das Look and Feel der nativen App abbildet um neue Admins möglichst schnell produktiv einsetzen zu können. Parallel sollte die Entwicklung zügig abgeschlossen werden. Daher fiel die Wahl der Implementierung auf Flutter mit primärem Target "Web" aber auch macOS als Zielplattform im Auge. CSS/JS wurden nicht verwendet, da sich die komplexe UI leichter in Flutter implementieren ließ. In der Applikation können statische Daten zu Userregistrierungen eingesehen werden, außerdem können Chats überwacht, Bild- und Audiomaterial von Usern gelöscht und User deaktiviert werden.

Technologien:
Flutter/Dart

Plattformen:
nginx/Web

11/2021–06/2022: Rust-Backends für Datingplattform

Branche:
Dating/Social Network (FUSE UG)

Rolle:
Design, Entwicklung

Projekt:
Für ein Startup sollten die Backends von C++ nach Rust portiert werden. Die Backends haben REST-Interfaces via async HTTP (actix-web) und greifen mit Tokio-Postgres auf die Datenbank zu, wobei mehrere Lese- und Schreibpools verwendet werden. Tests der Backends erfolgen extern mit pytest-Skripten. Die Releaseversion des Backends wird in einem Docker-Build-Container gebaut und anschließend in einem weiteren Container verpackt und ausgeführt.

Technologien:
Rust, actix-web, tokio (als async-runtime), PostgreSQL, Python, pytest, bash, docker

Plattformen:
MacOS, Alpine Linux/docker

07/2021–10/2021: Fehlerbehebung Medizingerät

Branche:
Medizingeräte (Inveox GmbH, München)

Rolle:
Entwicker

Projekt:
Der Kunde hat ein Medizingerät zum Entpacken von Gewebeproben im Prototypenstadium. Die Steuerungssoftware (in Python) hatte diverse Probleme und musste für nachfolgende klinische Studien in einen funktionierenden Zustand gebracht werden.

Technologien:
Python

Plattformen:
Windows

12/2018–06/2021: Backends Qualitäts-Infrastruktur

Branche:
Market research (GfK SE, Nürnberg)

Rolle:
Architect, Tech lead, Developer

Projekt:
In diesem Projekt mussten insgesamt drei Backends, jeweil bestehend aus einer Reihe von Microservices entwickelt werden. Alle Applikationen entnehmen marktforschungsrelevante Daten aus der Input-Pipeline der GfK und bereiten sie mit Datascience-Modellen auf (etwa detektieren von Betrug durch Auditoren oder Erkennen/Glätten von Daten-Ausreißern). Die Backends liefern die gesamte technische Infrastruktur, die nötig ist, um die Datenmodelle ablaufen zu lassen, also etwa Tasksteuerung, Benutzerkorrekturen, I/O von Datenströmen.

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

Platformen:
Linux, Web

01/2020–12/2020: Maschinensteuerung

Branche:
Maschinensteuerung (RWT GmbH, München)

Rolle:
Architect, Tech lead, Developer

Projekt:
Eine ursprünglich über eine native Windows-Anwendung und einen in C++ implementierten Telegrammserver sollte modernisiert werden als Webanwendung inklusive eines lokal laufenden Windows-Clients für Dinge, die über einen Browser hinausgehen (etwa lokale Dateien öffnen). Ein neues Backend wurde in Python/Django+GraphQL implementiert, Frontend in nuxt.js (Frontend durch meine Mitarbeiter). Außerdem wurde ein lokaler Client in C++/Qt erstellt, der alle lokalen Systemzugriffe ausführt. Die Identifikation von Browserclient zu Qt-Client erfolgt im Backend über Mapping der IP. Kommnikation via Netzwerkzugriff/GraphQL, auch vom Qt-Client zum Backend.

Technologien:
Python, Django, Qt, GraphQL

Platformen:
Windows 64

10/2018–laufend: Steuerdatenverarbeitung/Visualisierung

Branche:
Steuerdaten (U-Know! BI OHG, Regensburg)

Rolle:
Architect, Tech lead, Developer

Projekt:
Die Plattform extrahiert rohe Steuerdaten der marktführenden Software für Steuerberater (DATEV) und führt diverse Analysen und Aggregierungen auf den Daten aus. Die Ergebnisse werden dann an ein Frontend und eine App via JSON/REST weitergegeben und dargestellt. Das Projekt enthält eine Reihe von Microservices, die die Themen Import, Upload backend, API backend behandeln sowie eine native Windows-Software die den Export der DATEV-Daten auf dem DATEV-Server ausführt. Die Plattform ist für hohe Skalierbarkeit ausgelegt und kann vollständig isoliert bei größeren Kunden unter deren eigenem Hosting ausgerollt werden.

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

Platformen:
Linux, Windows

12/2016–12/2017: Bauvergleich24

Branche:
Vermittlungsportal für Bauherren/Baudienstleister

Rolle:
Co-Founder, Entwicklung, Planung

Projekt:
Bauvergleich24 ist ein webbasiertes Portal zur Vermittlung zwischen Bauherren und am Bau benötigten Handwerkern sowie sonstigen Dienstleistern. Bauvergleich ist eine SPA-Anwendung, die vollständig in HTML5/CSS/JS auf Frontendseite sowie in C++ im Backend realisiert ist.

Technologien:
HTML5, CSS, JavaScript, C++, PostgreSQL, Mercurial

Plattformen:
Linux64 (Backend)

11/2016–12/2019: CeeMii-App + Ecosystem

Branche:
Gastronimie/Lebensmittelhandel (CeeMii GmbH, Wien)

Rolle:
Co-Founder, Entwicklung, Planung

Projekt:
CeeMii besteht aus einer Single-Source-App (Qt/QML) für Android/iOS sowie aus einem C++-Backend. Es handelt sich um eine location-based-App bei der Nutzer abhängig von ihrem aktuellen geographischen Standort verbilligte Angebote von Lebensmittelhändlern und Restaurants in ihrer unmittelbaren pyhischen Umgebung ansehen und direkt in der App via Kreditkarte kaufen können. Die Ware wird gegen Vorzeigen eines 3D-Barcodes der App vom Händler herausgegeben. Ich habe die vollständige Entwicklung, sowohl der App als auch des Backends in diesem Startup gemacht und halte Anteile an der Gesellschaft.

Technologien:
C++, Qt/QML, Python, PostgreSQL, Mercurial, 3D-Barcodes, Positionssensor, Karten

Plattformen:
Android, iOS, Linux64 (Backend)

09/2015–05/2017: Embedded-Medizingerät

Branche:
Medizinbranche (Pulsion GmbH, München)

Rolle:
Entwicklung, Planung, Requirements Engineering

Projekt:
Ein Blutdruckmeßgerät für den Einsatz in der Intensivmedizin sollte in einen Arbeitsteil und einen Monitorteil getrennt werden. Der Kunde wollte den Arbeitsteil als eigenes Modul vermarkten, so daß Mitbewerber dieses in ihre eigenen Monitore integrien können. Die ursprüngliche Implementierung lag in Delphi, Assembler und C++ vor, das neue Modul sollte vollständig in C/C++ auf einem ARM Cortex M4-Controller implementiert werden. Zusätzlich wurde ein C++-SDK auf PC-Seite und ein Python-Wrapper dafür benötigt. Das Testsystem wurde komplett in Python implementiert (unittest-Modul), außerdem wurde ein Webfrontend (Twitter-Bootstrap+JQuery) mit Python/CherryPy as Backend bereitgestellt

Technologien:
Delpi, Assembler, GCC/G++-ARM, Makefiles, Subversion, Python, CherryPy

Plattformen:
ARM32, Win64

09/2014–09/2015: C# Logistik/Lagersteuerung/Kampagnenplanung

Branche:
Modebranche (Best Secret GmbH, München)

Rolle:
Entwickung

Projekt:
Mitarbeit an der Lagersteuerungssoftware für einen großen Modeversender. Die Software steuert den kompletten Lagerablauf, also Auftragsaufbereitung, Kommissionierung, Steuerung der Packstationen, Druck der Labels + Interface zu DHL/Zoll usw. Außerdem Mitarbeit am Programm für die Kampagnenplanung und Artikelbestellung der Verwaltung. Außerdem diverse Python-basierte Hilfsprogramme zur Datenbereinigung und ein (mit Tkinter) implementiertes Frontend zur Batchkompilierung/Distribution der .NET-Services

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

Plattformen:
Win64

06/2014–07/2014: Magletics-Frontend für Gehirnstimulator

Branche:
Medizintechnik (Mag&More GmbH, München)

Rolle:
Entwicklung

Projekt:
Für eine Sonderanwendung der Ansteuersoftware für einen magnetischen Gehirnstimulator mußte ein neues Frontend auf WPF-Basis erstellt werden. Die Steuerung des Stimulators erfolgt über eine einen direkt angesteuerten FTDI-Chip via USB-Schnittstelle.

Technologien:
C#, WPF, .NET, FTDI

Plattformen:
Win64

10/2013–06/2014: Webapplikation für Pharmazulassung

Branche:
Pharma (EXTEDO GmbH, München)

Rolle:
Entwicklung (Backend)

Projekt:
Aufbau einer C#-basierten Webapplikation zur gesetzeskonformen Einreichung und Verwaltung der Zulassungsdokumente, Stofflisten und Nebenwirkungstabellen für Medikamente. Die Software unterstützt die Medikamentenzulassung über den gesamten Lebenszyklus des Produkts. Bedingt durch den enormen Zeitdruck seitens des Gesetzgebers wurde zusätzlich – bis das JavaScript-basierte Frontend fertig ist – als vorübergehende Lösung noch eine Import/Export-Schnittstelle via Excel-Dateien realisiert. Desweiteren wurde ein PDF-Generator implementiert. Ein Teil des Codes wurde über einen in Python geschriebenen Generator erzeugt.

Technologien:
C#, ASP.NET, EntityFramework, SQL-Server, TFS, Python, OpenXML/ClosedXML, "PDF-FileWriter" library

Plattformen:
Win64

10/2012–06/2013: Integration des Python-Interpreters in 3D-Visualisierungssoftware

Branche:
3D-Rendering (Realtime Technology (RTT) AG, München)

Rolle:
Architektur, Entwicklung

Projekt:
Einbau des Python-Interpreters in eine 3D-Visualisierungssoftware für die Automobil- und Luftfahrtbranche. Der Interpreter-Host wird als PlugIn geladen und kann intern beliebig viele Subinterpreter abspalten. Die Ausführung der Python-Skripte erfolgt üblicherweise (wobei die Skripte dafür zuständig sind) in Hintergrundthreads. Zusätzlich stellt das PlugIn ein internes Python-Modul (in C) zur Verfügung, über das Python-Callables auf dem Qt-Hauptthread der Anwendung ausgeführt werden können, da ggf. einige API- Aufrufe zwingend den Qt-Hauptthread voraussetzen.

Technologien:
Python, Python C API, Qt, C++

Plattformen:
Win64

10/2012–10/2012: Updateservice für iOS-Banking-Lösung

Branche:
Finanzen

Rolle:
Architektur, Entwicklung

Projekt:
Für eine neuartige iOS-Banking-Lösung, die durch Vorausladen von Umsätzen einen Geschwindigkeitsvorteil gegenüber anderen Lösungen erzielen soll, musste eine Serverkomponente erstellt werden. Diese ist als Windows-Service ausgeführt. Aufgabe war es, regelmässig Bankumsätze via FinBanks-Bibliothek von den offiziellen Gegenstellen der Banken zu laden und in einer Datenbank abzulegen. Ferner sind automatische Joberzeugung und -verteilung sowie Lastausgleich in der Komponente integriert.

Technologien:
C#, FinBanks, PostgreSQL, NUniT

Plattformen:
.NET 4.0

08/2012–09/2012: Spiel “Pancake” für iOS

Branche:
Computerspiele (selbständig)

Rolle:
Alle

Projekt:
Nachbau des LCD-Hand-Konsolenspiels "Pancake", ursprünglich von VTECH 1981 auf den Markt gebracht, vorerst für iOS (iPhone/iPad). Komplett mit MonoTouch (C# unter iOS) und dem MonoGame-Framework (Nachbau von Microsoft XNA) implementiert. Für den Verkauf im AppStore.

Technologien:
C#, MonoTouch, MonoGame

Plattformen:
OS 5.1 (iPhone, iPad)

03/2012–06/2012: EEG-Viewer für iOS

Branche:
Medizin (sh computing UG)

Rolle:
Design, Entwicklung

Projekt:
Als Demonstrator für den Messeeinsatz wird eine Software benötigt, die Gehirnstromkurven aus dem branchenüblichen EDF-Format performant und mit Touch- Unterstützung auf einem iPad anzeigen kann. Implementierung erfolgt über das MonoTouch-Framework (C#-Entwicklung auf iOS).

Technologien:
C#, MonoTouch

Plattformen:
iOS 5.1 (iPad)

12/2011–08/2012: Parser für NATURAL-Sourcecode und IronPython-Tool

Branche:
Zulieferer öffentliche Verwaltung (it&more GmbH, München)

Rolle:
Entwicklung in C#, Design und Implementierung IronPython-Werkzeug, Beratung

Projekt:
Im Auftrag der öffentlichen Hand musste aufgrund einer neuen gesetzlichen Vorgabe die NATURAL-Applikation des Meldewesens auf Unicode-Fähigkeit umgestellt werden. Diese Applikation besteht aus ca. 3 Millionen Zeilen NATURAL-Quellcode, historisch über mehr als 20 Jahre gewachsen. Im Projekt musste eine C#-Lösung erstellt werden, die automatisiert den NATURAL-Code auf Unicodefähigkeit umstellt. Hierbei müssen zum Teil Konstrukte im Code automatisch durch völlig andere Konstrukte ersetzt werden, so dass simples Ersetzen nicht funktioniert - der Code muss tatsächlich geparsed und zum Teil auch "verstanden" werden (Wanderung von Daten zwischen Modulen, Werteweitergabe zwischen Variablen, Generieren von neuen Steuerelementen in Dialogen usw.).
Zusätzlich wurde ein leistungsfähiges Debug-Werkzeug auf Basis von IronPython (Python für .NET = skriptbasierter Zugriff auf das komplette Datenmodell der Applikation) erstellt.

Technologien:
C#, .NET, Python, IronPython

Plattformen:
Windows 7 64, Windows Server 2008 R2

08/2011–11/2011: Steuerungssoftware für Gehirnstimulator

Branche:
Medizin (Mag&More GmbH, München)

Rolle:
Entwicklung (Neuentwicklung), Beratung

Projekt:
Der Kunde konstruiert und vertreibt Medizingeräte, die durch Magnetfelder Impulse direkt in Gehirnregionen einleiten (nicht-invasiv). Hierzu musste eine Software neu erstellt werden, die Patientendaten verwaltet sowie Ansteuerkurven erstellt, verarbeitet und in ein Format umsetzt, welches der Controller der Geräteschnittstelle versteht. Die Verbindung zur C#-Software wird über einen FTDI-USB-Chip hergestellt, welcher im Direktverfahren angesteuert wird. Zusätzlich muss die Software eine Echtzeitvisualisierung des "Programmablaufs" in der Kurve anzeigen (der Controller meldet -gepollt- Spulentemperatur und Scriptfortschritt zurück).

Technologien:
C#, .NET, FTDI

Plattformen:
Windows XP/Vista/7

05/2010–10/2011 (Teilauslastung): Systemadministration Linux + VCS

Branche:
eCommerce (euroblaze Wapsol GmbH, Stuttgart)

Rolle:
Administration, Integration, teilweise Entwicklung

Projekt:
euroblaze erstellt nach Kunden/Designervorgaben Onlineshops auf Basis von OXID. Die Entwicklung selbst findet in Indien statt, aktuell werden über 100 Projekte von euroblaze betreut. Ich bin verantwortlich für die Systemadministration, Installation auf produktiven Maschinen und die Verwaltung des Versionskontrollsystems (subversion, bootstrap, Berechtigungen, Branchzuweisung, ...).

Technologien:
subversion, Linux, OXID, Apache, PHP, ZendGuard, ionCube, MySQL, Python

Plattformen:
Linux, FreeBSD

07/2011–07/2011: Virtuelle Maschine für OXID-Tests

Branche:
eCommerce (euroblaze Wapsol GmbH, Stuttgart)

Rolle:
sämtliche Rollen

Projekt:
Virtuelle Maschine zum Testen von Installationen der OXID eCommerce-Lösung (PHP-basiert). Die VM automatisiert die die Administration des Linux-Hosts sowie die Installation von OXID-Versionen. Sie bootet automatisch in eine Python/cdialog- Applikation, die Installationen (+Systemaccounts +Datenbank +Konfiguration +Apache- Alias), Datenbanktausch, Filemanagement, Systemupdate und Update der Installationstemplates bzw. der VM-Anwendung selbst menügesteuert auch unerfahrenen Anwendern zur Verfügung stellt. Die VM erfordert keinerlei manuelle Eingriffe mehr und konfiguriert im Fall einer IP-Änderung der VM Anpassungen der OXID- Konfigurationen automatisch durch.

Technologien:
Linux, OXID, Apache, PHP, ZendGuard, ionCube, MySQL, Python, cdialog, Shell

Plattformen:
Linux (Debian6-i386)

10/2010–07/2011: Webbasierte Kontaktplattform/Backend/PySide (QT4)-Frontend

Branche:
Vermittlung von Handelspartnern (sh computing UG)

Rolle:
sämtliche Rollen

Projekt:
Webplattform, die Vertriebsleute und Hersteller weltweit zusammenführt. 100% Python-basiertes Projekt. Frontend via Django, Backend CherryPy-Applikationsserver, einzelne Python-Minicrons und ein PySide (=Qt4 via Python)-Frontend zum Freischalten von Annoncen.

Technologien:
Linux, Python, Django, CherryPy, PySide, QT4, PostgreSQL, git

Plattformen:
Linux (Ubuntu Server), Multi-Plattform (QT4-Frontend)

09/2010–06/2011: Schnittstelle zu 3D-Render-Suite und Prozeßautomatisierung

Branche:
3D-Rendering (Realtime Technology (RTT) AG, München)

Rolle:
Entwicklung

Projekt:
Für eine führende 3D-Render-Suite (QT4) im Automobil/Luftfahrt/Bekleidungsumfeld sollte eine C++-Schnittstelle mit Python-Anbindung entwickelt werden. Das System arbeitet selbstkonfigurierend im Cluster (selbständiges Wählen eines Masters) und beherrscht Lastverteilung. Die komplette Schnittstelle kann auch – via boost::python – von Python aus angesteuert werden. In der zweiten Phase musste eine komplett scriptbasierte, automatisierte Renderlösung zum Generieren von Katalogbildern für den Bekleidungskonzern Adidas auf Basis von 3D-Daten erstellt werden. Auf C++-Seite zusätzliche Bildverarbeitung (u.a. combine, color profile, clipping path, PNG chunk reordering) sowie Erstellen und Warten des Installationsprogramms (NSIS)

Technologien:
C++; boost; boost::python; STL; TCP/IP; Multicast; Verschlüsselung; Eigene Kommunikationsprotokolle; Python; CherryPy; Curses; ImageMagick; WinDBG; git; subversion

Plattformen:
Win64, Linux

09/2009–07/2010: Generischer Sprachtrainer

Branche:
Privatanwender

Rolle:
sämtliche Rollen

Projekt:
Die Software ermöglicht Vokabular/Übersetzungen/Lückentexte für beliebige Sprachen zu trainieren. Sie unterstützt auch exotische Sprachen wie z.B. Hindi oder Sprachen, die von rechts nach links geschrieben werden.

Technologien:
C++; Qt4; STL; XML; Verschlüsselung; git

Plattformen:
Win32, Mac OS X, Linux, BSD

06/2005–09/2009: Systeme für medizinische Datenverarbeitung

Branche:

Medizintechnik (Konsortialprojekt; Gefördert durch Bundeswirtschaftsministerium)

Rolle:

Systemdesign; Projektmanagement; Planung; Implementierung; Teamleitung; Serveradministration; Mitarbeit an der QM-Dokumentation für ISO 9001 und ISO 13485. Wechselnd 2-10 Mitarbeiter im Team.

Projekt:

Das System besteht aus mehreren Komponenten. Zum einen ein ausfallsicherer und hochverfügbarer Server (C++ unter FreeBSD), der die von den Endgeräten roh zur Verfügung gestellten Messdaten verarbeitet und an eine elektronische Patientenakte zur weiteren Aufbereitung weiterleitet. Zum anderen eine auf Windows-Mobile-PDAs installierte Lösung, die via Bluetooth Messdaten von (hardwareseitig manipulierten = mit Bluetooth-Modulen nachgerüsteten) Messgeräten in Empfang nimmt und Rückmeldung an den Endbenutzer liefert. Ausserdem diverse Wartungs- und Hilfsprogramme.

Technologien:

C; C++; C#; STL; POSIX; Threading; FreeBSD; Windows Mobile; MFC; wxWidgets; AES (Verschlüsselung); Bluetooth; TCP/IP; Eigene Übertragunsprotokolle; embedded; PostgreSQL; SQLite; FTDI-Devices (Direktansteuerung); git; subversion; cvs

Plattformen:

FreeBSD; Linux; Win32; Windows Mobile (5, 6)

07/2003–06/2005: Softwarebasierte Personalressourcenplanung

Branche:

Logistik - Andreas Schmid Logistik AG; Siemens SBS

Rolle:

Systemdesign; Projektmanagement; Implementierung; Teamleitung; Serveradministration;

Projekt:

Es handelt sich um eine Anwendung in Client/Server-Anordnung, die dazu dient, Arbeitsabläufe elektronisch abzubilden und zu verfolgen. Hierbei können Abläufe verzweigt und Subabläufe wiederverwendet werden. Die Dauer einzelner Vorgänge wird durch Mitarbeiterinformationen oder sensorische Meldungen erfasst. Auch können so Abläufe beendet bzw. in Gang gesetzt werden. Das Programm verwaltet ferner die verwendeten Ressourcen, die entstehenden Statistiken können zur Ressourceoptimierung oder für Vorhersagen verwendet werden. Benötigte Ressourcenaufwände lassen sich auch für zukünftige Projekte vorhersagen.

Technologien:

C++; STL; VCL; PostgreSQL; ODBC; TCP/IP; Eigene Übertragungsprotokolle; cvs

Plattformen:

Win32; FreeBSD

05/2001–06/2005: Lagersteuerungssystem Gefahrgutlager + Rechnerverteilung

Branche:
Logistik, IT - Andreas Schmid Logistik AG; Siemens SBS

Rolle:
Systemdesign; Projektmanagement; Implementierung; Teamleitung; Serveradministration;

Projekt:
Das Projekt musste aus Kostengründen so aufgesetzt werden, dass es sowohl ein Gefahrgutlager als auch die Verteilung von Rechnern bei Siemens-SBS (also ebenfalls Lager, kein Gefahrgut, aber diverse Zusatzfunktionen) abbilden konnte, dementsprechend generisch und erweiterbar konzipiert. Das System besteht aus einer Serverkomponente, die die eigentliche Lagerverwaltung bzw. Warenflußsteuerung/Belegsteuerung abwickelt sowie aus diversen Clientmodulen, etwa für die Arbeitsplätze im Leitstand, die Anbindung der Scanner (Funkscanner, Echtzeit) oder sonstigen Pick-Endgeräte (Pick-by-voice, Pick-by-light). Das System beherrscht Gefahrgutlagerung, Chargen, Barcodescanner, Mehrbenutzerfähigkeit, Verteilte Standorte, Drucksysteme gesetzter Formulare, One-Touch-Logistik (=automatische Prozesskette), diverse Automatiken, Mandantenfähigkeit und beliebige Import/Exportschnittstellen.

Technologien:
C++; Python; VCL; STL; Threading; PostgreSQL; ODBC; VT100; Delphi; TCP/IP; XML; Eigene Übertragungsprotokolle; cvs

Plattformen:
FreeBSD; Linux; Win32

01/1999–05/2001: Steuerung für internetbasiertes Shopsystem

Branche:
Shopsysteme - Skybeamer GbR

Rolle:
Systemdesign; Projektmanagement; Planung; Implementierung; 3 Mitarbeiter.

Projekt:
Die Software war das Steuerungsmodul für ein internetbasiertes Shopsystem und wurde auch als (kleine) Warenwirtschaftslösung mit minimaler Lagerverwaltung eingesetzt.

Technologien:
C++; STL; VCL; PostgreSQL; ODBC; TCP/IP;

Plattformen:
Win32

09/1997–02/1998: Echtzeit-Überwachung im Zugverkehr (S-Bahn München)

Branche:

(Eisenbahn-)Verkehr - Deutsche Bahn AG)

Rolle:

Eigenverantwortliche Planung und Implementierung (Prototyp).

Projekt:

Der sogenannte "Streckenspiegel", der den Fahrdienstleitern für ganz Süddeutschland in Echtzeit Einblick in den Zugverkehr gibt, hat informativen Charakter; Die Steuerung des Zugverkehrs (S-Bahn München) selbst erfolgt losgelöst davon über Prozeßrechner, die allerdings mit identischen Eingangsdaten arbeiten. Die alte Version des Steckenspiegels war mit speziellen Maschinen mit 8085-Prozessoren implementiert. Ich habe einen Prototyp der Software für den Einsatz unter MS-Windows (damals Win16 und Win32) implementiert während meines Aufenthalts im damaligen Rechenzentrum der DB in München im Rahmen meiner Ausbildung. Die Software umfasste sowohl den eigentlichen Steckenspiegel (Anzeigeprogramm) als auch einen Gleisplaneditor und einen speziellen Fonteditor für die Symbole. Mittlerweile (2012) ist der Streckenspiegel in Form der Smartphone-App "S-Bahn-Navi" auch für die Öffentlichkeit zugänglich.

Technologien:

C++; MFC; Watcom

Plattformen:

Win16; Win32

Veröffentlichungen

–Auszug–

- **ISBN 978-3936546-48-4 (C&L-Verlag, 2007)**
  Buch "GNU Autotools - UNIX-Buildsysteme mit Autoconf, Automake und Libtool"
- **Toolbox - Entwicklermagazin, 02/2008**
  Titelthema ".NET-PlugIns mit C#"
- **Toolbox - Entwicklermagazin, 03/2007**
  Beitrag "Make my file - so funktioniert GNU Make"
- **iX - Magazin für professionelle Informationstechnik, 09/2005**
  Beitrag "Verschlüsselte Block-Devices mit NetBSDs CGD"
- **freeX - Magazin für freie UNIX-Systeme, 06/2005**
  Beitrag "VND gut komprimiert"
- **freeX - Magazin für freie UNIX-Systeme, 04/2005**
  Beitrag "(Net)BSD auf SSH-Rootservern"
- **Postfix**
  Artikel "Postfix@NetBSD - Setting up a secure real-life mailserver with SMTP AUTH and anti-spam capabilities"
- **NetBSD-Handbuch ("The NetBSD Guide")**
  Abschnitt "The cryptographic device driver (CGD) - Creating an encrypted CD/DVD"
- **OpenBSD zyd(4) Gerätetreiber**
  Erste Version des Gerätetreibers für ZyDAS ZD-1211-basierte USB-WLAN-Adapter.
- **neb-cd512**
  Programm zur Anpassung der Kernel-in-core-Labels einer CD/DVD an eine andere Sektorengrösse (für Krypto-CDs/DVDs)
- **neb-wipe**
  Programm zur sicheren und endgültigen Vernichtung von Datenträgerinhalten. Basierend auf Forschungsergebnissen zur Wiederherstellung von magnetischen Datenträgern unter Laborbedingungen. Implementierung des von Peter Gutmann vorgeschlagenen 35-Pass-Musterverfahrens. Kann gesamte Datenträger oder einzelne Partitionen sicher zerstören. Entwickelt für NetBSD.
- **neb-hdtoolbox**
  Programm um NetBSD für die amiga-Plattform crossinstallieren zu können. Kann das spezielle RDSK/PART-Format des Amiga (vergleichbar mit der Partitionstabelle auf der Intel-Plattform) selbständig erzeugen und verändern. NetBSD/amiga verfügte vorher nicht über entsprechende Software und war auf Fremdprogramme angewiesen. Kann auch eingetragene Partitionen ändern.
- **cloop**
  Komprimiertes Dateisystem für Live-CD-Images (Kompressor/Steuerung)