



Software and Network Solutions

SANS Ltd.; Hohenzollernstr. 20; 88605 Meßkirch-Igelswies

SANS Software and Network Solutions LIMITED

Director:

Dipl. Ing.(BA) Florian Mayer

Niederlassung Deutschland:

Hohenzollernstr. 20

88605 Meßkirch

Kontakt

Telefon: +49(0)7575-92 41 595

Fax: +49(0)7575-92 49 62

Mobil: +49(0)176-222 09 624

Mail: info@sans-ltd.com

Datum:

Zu meiner Person (Director Dipl.Ing.(BA) Florian Mayer)
geboren 1980

Schulischer Hintergrund / Berufliche Fortbildungen

1997-2000	Technisches Gymnasium Sigmaringen
2001-2004	Berufsakademie Friedrichshafen, technische Informatik
2008, 2009	Fortbildungen für Telenot Sicherheitssysteme (Einbruch- und Brandmeldesysteme)
2010	Zertifizierung zur Fachkraft nach DIN-14675
2013	Fortbildung zum Squish GUI Testing Framework
2017	Zertifizierung zum Bizerba Retail Vision Expert und Retail App Developer

Schwerpunkte

- Software Entwicklung (Planung, Architektur und Implementierung) mit Schwerpunkt Embedded:
 - C++ / Qt / QML für Desktop-Applikationen
 - C/C++ (inkl. STL & boost) für System-Programmierung
 - C für hardwarenahe Applikationen oder Micro-Controller
 - JavaScript / JScript für Application Extension / Addon Development (speziell QtScript / QML, Mozilla Addons, Enterprise Architect Scripting, Windows Script Host)
 - Skriptsprachen: BASH, CMD, PowerShell und Python
 - UML Modellierung mittels Enterprise Architect (inkl. Profiles, Templates, Scripting und Versionierung)
 - DotNET ASP Core oder Java Jetty für Web Anwendungen (zusammen mit SQLite, PostgreSQL oder MySQL sowie HTML5)
 - Node RED / MQTT / InfluxDB / Grafana für IOT Anwendungen & Gebäudesteuerungen
 - Versionsverwaltungssysteme (GIT und SVN)

Bankverbindung

BIC: GENODE61MES
IBAN: DE23 6936 2032 0002 6257 09
Volksbank Meßkirch

Registered Office:

Kandoy House
2 Fairview Strand
Dublin 3

Registered in Ireland

Company No.: 647237
Steuer-Nr: 85483 / 18903
Amtsgericht Ulm HRB 742614



Software and Network Solutions

- Build-Systeme und Build Automation (Continuous Integration) mit Docker und Jenkins oder Microsoft TFS Team Foundation Server für sämtliche Plattformen
 - Deployment Lösungen (Paket-Management, Repositories und Windows-Installer)
 - RPM, DEB, MSI (WiX) NSIS und CPack
- Linux, auf dem Server, als Desktop System und Embedded
 - Speziell: Ubuntu, Debian, Fedora / RedHat / CentOS, Android
 - Systemintegration / Paket-Management mit RPM
- Anpassung von Linux für spezielle Anforderungen, z.B. Steuerung, Machine Controller, im Docker Container oder für IOT Anwendungen
- Virtualisierung mit VirtualBox, VMWare ESX/ESXi und XEN / Citrix XenServer und Docker
- Netzwerk-Techniken:
 - WiFi (WLAN), 4G Mobile Networks, Bluetooth
 - Routing/Firewall-Applikationen mit Mikrotik Routern
 - CAN / CANOpen, I²C, Dallas One Wire Sensor-Netzwerk, RS-485
 - proprietäre Bus / Ring-Bus Systeme in der Gebäude und Sicherheitstechnik
- Elektronik-Entwicklung, hauptsächlich Adapter, Anpassungen, IOT Bausteine etc.
- Video-Technik, Framegrabber, Video-Überwachung
- Sicherheitstechnik
 - Alarm-Anlagen, Brandmelde-Technik, Zugangskontroll-Systeme
 - Planung, Montage, Dokumentation mittels CAD und Wartung

Abgewickelte Projekte (Auswahl)

- **2024** Prüfstand für die Produktion der IoT Waage
 - Entwicklung von Hardware und Software zum Testen der IoT Waage während der Produktion
 - Controller: Embedded Linux Armbian / Debian auf RockPI Hardware
 - Multimeter und Netzteil im Prüfstand werden vom Controller gesteuert
 - Entwicklung der Prüfapplikation mit NodeRED
 - CI Build als GitHub Action auf Basis vom Armbian Build System
- **2023-24** Firmware Entwicklung für IoT Waage
 - Entwicklung in C++ auf ESP32 mittels FreeRTOS und Arduino SDK
 - Technologien: WiFi / NB-IoT, Google Protocol Buffers, Web Services und Web GUI
 - Entwicklungsumgebung PlatformIO mit Visual Studio Code und GitHub
 - Deployment via GitHub Actions CI Build in Docker Container und Programmer Software von Espressif
- **2023** Anbindung eines Kameramoduls in einem Embedded Linux System
 - Entwicklung eines Linux Kernel Moduls zur Konfiguration des Video4Linux und Media Subsystems auf die Eigenschaften der Kamera
 - Entwicklung eines Tools zur Initialisierung der Kamera über I2C mittels eines Hersteller-spezifischen Scripts
 - Debuggen des MIPI Datenstroms aus der Kamera
 - Eingesetzte Techniken: Linux Kernel Programmierung in C, .Net/C# für das Init Tool, Github
- **2023** Entwicklung einer QML Oberfläche für ein Haushaltsgerät
 - Entwicklung eines Import Moduls für Design Token Definitionen aus Figma (in Python)
 - Implementierung von kundenspezifischen QML Controls in enger Abstimmung mit den Designern des Kunden
 - Bereitstellung von Reviews über Qt Design Studio für HMI/Design.
 - Eingesetzte Tools: Visual Studio Code, QtCreator, Qt Design Studio, Figma, Gitlab, Confluence, Jira, Miro



Software and Network Solutions

- **2022** Entwicklung einer Intranet Virens Scannerlösung für eine IT Security Abteilung
 - Bereitstellung eines Linux Docker Containers zur Untersuchung der Daten vom ungesicherten (roten) Netz zum gesicherten (grünen) Netz, u.A. mit MS Defender und Thor.
 - Entwicklung von Auspackmechanismen für geschachtelte Archive (Zip, MSI usw. ineinander gepackt) inkl. Quarantänemechanismen
 - Entwicklung einer Web Applikation (ASPCore) in Dot Net Core als User Interface
 - Integration in die darunterliegenden Linux Linux Distributionen (Debian / Ubuntu) sowie Docker Umgebungen (Artifactory als Registry, Docker in Docker)
 - Entwicklung eines Ubuntu Linux Live Systems zum Scannen von virtuellen Maschinen.
 - Versionsverwaltung mit GIT, Build- und Projektmanagementumgebung Team Foundation Server TFS, Dokumentation mit Enterprise Architect.
- **2022** Entwicklung eines Frameworks für kundenspezifische Anpassungen mittels Qt QML
 - Bereitstellen von speziellen QML Controls für das Kassen / Waagen Umfeld
 - Spezifizierung der APIs mittels UML in Enterprise Architect
 - Refactoring im bestehenden Code
 - Anbindung des vorhandenen Codes über Fassaden
 - Anbindung von Qt Design Studio als zukünftige Entwicklungsumgebung für kundenspezifische Anpassungen, auch für Nicht-Programmierer
- **2017-2022** Entwicklung einer Steuerung für eine Maschine zum Verpacken von Lebensmitteln
 - Begleitung des kompletten Entwicklungszyklus vom frühen Prototypenstadium bis zur Produktion des Seriengeräts
 - Entwurf einer Domain Specific Language auf Basis von Google Protocol Buffers und FreeMarker zur Steuerung des Ablaufs inkl. Graphischer Modellierung mittels Enterprise Architect.
 - Einbettung in das Waagensystem des Kunden mittels proprietärer App Technologie in Java
 - Entwicklung von Java Webservices zur Abstrahierung der Hardware
 - Implementierung einer Ablaufsteuerung in C++ / Boost / STL zur Ausführung auf einem Embedded Linux ARM Board (BeagleBone)
 - Anbindung von Schrittmotoren über CAN Bus mittels CANOpen Protokoll und weiterer Peripherie über I2C Bus.
 - Anbindung einer RealSense Kamera und Implementierung von Algorithmen zur Vermessung der Verpackungsgutes in C++ / OpenCV
 - Versionsverwaltung mit GIT, Buildumgebung TFS, Toolchains als Docker Container, Dokumentation mit Enterprise Architect.
- **2017** Entwicklung eines Prototypen zur Steuerung eines Allrad-Antriebes bei Kleinserien-Fahrzeugen (Transporter)
 - Steuerung mittels Atmel AVR 8-Bit Microcontroller in C/C++
 - Prototypenplattform: Arduino, selbst-gelöteter Lochraster-Aufbau für erste Tests mit dem modifizierten Fahrzeug
 - Entwicklungsumgebung: Atom Editor mit Platform IO
 - Eingesetzte Techniken:
 - Atmel 8-Bit AVR Micro-Controller, C / C++, Arduino Framework (Hardware und Software), Atom / Platform IO, Prototypischer Elektronik-Aufbau
- **2016 - 2017** Weiterentwicklung einer Zahn-Medizinischen CAD Applikation
 - Planung der anstehenden Refactorings mit Enterprise Architect
 - Umsetzung mit C++11 / STL unter Windows (Visual Studio 2015)



Software and Network Solutions

- Einführung von Enterprise Architect beim Kunden
- Aufsetzen von Build-Jobs zum automatisierten Aktualisieren des EA-Modells mittels Python/COM auf TeamCity Build Server
- Erstellung der Dokumentation mit Enterprise Architect und Polarion
- Eingesetzte Techniken:
 - C++11, STL, Python, COM, TeamCity, Polarion, Bugzilla, Plastic SCM, Windows 7/10
- **2014 - 2015** Weiterentwicklung eines generischen Systems zur Integration von multi-modaler Eingabesysteme in bestehende Applikationen
 - Anbindung von neuartigen Eingabegeräten wie Eyetrackern (SMI, Tobii), 3D-Cameras (Intel RealSense), Sprach-Erkennung (Nuance/RealSense), Gesten-Erkennung (RealSense)
 - Anbindung von Bluetooth Low Energy (BLE) Geräten über Windows APIs (Bluetooth, Setup und Raw Input HID APIs)
 - Weiterentwicklung einer hochmodularen Software-Architektur mit Plugin-Schnittstellen, IPC Infrastruktur und Bibliotheken für die Applikationsentwicklung (C / C++ (VC 2010 / 2013 Compiler inkl. Qt), Java, JavaScript für Firefox Addon)
 - Einführung der Modell-Basierten UML 2 Entwicklung mittels Enterprise Architect
 - Anpassung von EA an betriebliche Abläufe mittels UML Profilen, Scripting und Templates
 - Steuerung von bestehenden Anwendungen durch oben genannte Geräte über Emulations-Verfahren (z.B. Tastatur & Maus-Emulation, Win32 Messages, MSUIA, Addon)
 - Entwicklung unter Windows 7, 8 und 10 mit VisualStudio 2010/2013/2015 und QtCreator
 - Entwicklung zum Produktiv-System: Einarbeitung der Tester, Untersuchen und Priorisieren von Bugs sowie deren Behebung
 - Eingesetzte Techniken:
 - C/C++, boost, STL, Qt / QML, Javascript, Google Protocol Buffers, JavaScript
 - Windows API (Win32 API), Windows Hooks, Windows Messages, Windows Named Pipes, MSUIA (MS UI Automation Framework, COM basiert), OpenCV
 - NSIS / Qt Installer Framework
 - Build Automation mit Jenkins / Apache ant / cmd / bash
 - Sonstige Tools:
 - cppcheck und cpplint, doxygen, Atlassian Confluence und Jira



- **2014** Entwicklung eines Systems zur Wiedergabe von Digital Singage Content in UHD / 4K Qualität
 - Zielplattform: Set-Top Boxen (STB, Amlogic ARMv9 Architektur) unter Android 4.4 sowie Android Tablets als Controller
 - Vollständiger Unattended Betrieb inkl. Update-Funktion für Firmware und Content
 - Eingesetzte Techniken:
 - Qt 5.3, C++, Java, JNI, UPnP, RS-232, HDMI CEC, Android 4, Linux
- **2014-2013** Entwicklung einer Firmware für eine Bedienkonsole eines Kommunikationssystems mit eingeschränkten Interaktionsmöglichkeiten
 - Nur wenige Knöpfe und ein Drehrad sowie ein grafisches, jedoch sehr kleines Display stehen zur Bewältigung von komplexen Bedienabläufen zur Verfügung
 - Eingesetzte Techniken
 - C++, Qt 4.8, CORBA, UML 2, Python, Embedded Linux
 - Eingesetzte Tools:
 - VisualStudio 2008, Enterprise Architect, ClearCase, ClearQuest, MS Office, PCLint, Squish
- **2013-2012** Entwicklung eines generischen Systems zur Integration von multi-modaler Eingabesysteme in bestehende Applikationen
 - Anbindung von neuartigen Eingabegeräten wie Eyetrackern (SMI, Tobii), 3D-Cameras (z.B. Microsoft Kinect), Sprach-Erkennung (MS Speech, Nuance Vocon), Gesten-Erkennung (Microchip)
 - Entwurf einer hochmodularen Software-Architektur mit Plugin-Schnittstellen, IPC Infrastruktur und Bibliotheken für die Applikationsentwicklung (C / C++ (inkl. Qt), Java)
 - Steuerung von bestehenden Anwendungen durch oben genannte Geräte über Emulations-Verfahren (z.B. Tastatur & Maus-Emulation)
 - Entwicklung unter Windows 7 & 8 mit VisualStudio 2010/2012 und QtCreator
 - Eingesetzte Techniken:
 - C/C++, boost, STL, Qt, Google Protocol Buffers, Java mit JNA
 - Windows API (Win32 API), Windows Hooks, Windows Messages, Windows Named Pipes, MSUIA (MS UI Automation Framework, COM basiert)
 - NSIS Installer Framework
- **2010-2012** Unterstützung bei der Entwicklung eines Tablet PC Systems auf Meego 1.0 Basis
 - System-Integration
 - Paket-Management mit RPM
 - Bauen von RPM Paketen, auch für komplexe Applikationen wie VirtualBox oder KDE Applikationen
 - Hardware-nahe Anpassungen
 - Sensoren (G-Sensor, Ambient Light Sensor...)
 - Bluetooth, WiFi (WLAN) und 3G & GPS (Huawei Modem)
 - Entwicklung eines Userspace-Treibers für einen HDMI Converter
 - Entwicklung von Live-Systemen auf USB Sticks für Recovery und Service-Zwecke (mic-image-creator, kickstart)
 - Erstellung einer VM für Entwicklungszwecke auf Basis von VirtualBox inkl. Integration der VirtualBox Guest Additions, automatische Generierung mittels mic-image-creator



Software and Network Solutions

- Integration von Adobe AIR
 - Entwicklung von Applikationen mit C++, Qt, DBUS, Linux Native Libraries
 - Jeniks Build Automation / Continuous Integration mittels Apache ant, qmake & BASH Skripten
 - Aufbau und Administration eines OpenSuSE Build Service (OBS) inkl. Download Funktion auf das Endgerät
-
- **2010** Entwicklung eines Prüfstand-Systems für Server Komponenten für die Firma Pumax Remarketing
 - Alle Prüfmaschinen werden per PXE über das Netzwerk gebootet.
 - Prüfapplikationen auf Basis von Ubuntu-Linux:
 - Entwicklung von Live-Images mittels Ubuntu Casper/SquashFS
 - Test-Routinen für Hardware Komponenten als BASH Skripte
 - Integration von Windows PE
-
- **2010** Weiterentwicklung des Messdaten-Erfassungssystems für die Firma Kern Forschung:
 - Erweiterung um die Möglichkeit, beliebig viele Stromversorgungen einbinden zu können
 - die Netzteile werden über Kurven (vergleichbar mit einem Funktionsgenerator) gesteuert. Es können verschiedene Kurven-Typen durchlaufen werden
 - eine grafische Simulation unterstützt den Benutzer
-
- **2009** Montage von Sicherheits-Elektronik für die Firma Schütz Alarm-Anlagen, Altenstadt, hauptsächlich Esser IQ8, Rückbau von Altanlagen inkl. Dokumentation durch CAD System.
 - **2009** Demontage und Verwertung von Großrechner-Systemen für NEC Deutschland.
 - **2008** Entwicklung eines Steuerungssystems für eine Elektro-Heizung
 - jeder Raum kann individuell mit einer Kurve oder manuell gesteuert werden
 - Temperatur-Fühler und I/O auf Basis von Dallas One Wire Technik
 - Linux-basierter Steuerungsrechner (Embedded System von PC Engines)
 - Java-basierte Steuerung mit integriertem Webserver und Web-Interface als Benutzer-Schnittstelle
-
- **2008-2009** Montage von Sicherheits-Elektronik für die Firma RBR Braun, Reutlingen
 - Montage und Inbetriebnahme von Zugangskontroll-Systemen (Honeywell), Brandmelde-Anlagen (JOB detectomat/Telenot comfire) und Alarm-Anlagen (Telenot complex 400)
 - Teilnahme an Schulungen der Firma Telenot zum Thema Einbruchmeldetechnik und Brandmelde-Technik
-
- **2008** Entwicklung eines Meßdaten-Erfassungssystems für die Firma Kern Forschung:
 - Realisierung mit LabVIEW
 - Einbindung diversen Meßgeräten + Lambda-Netzteil als „Stimulus“
 - Protokollierung in ASCII-Datei
-
- **2008** Entwicklung und Evaluierung von Geräte-Treibern für das Institut für Gravitations-Forschung (Göde-Institut):
 - Realisierung mit LabVIEW
 - u.A. Voltmeter, Thermometer verschiedener Hersteller, vom Institut selbst entwickelte Geräte, Gero-Hochtemperatur-Ofen, Eurotherm-Regler, Leybold Vakuum-Sensoren etc.
-
- **2007-2008** Entwicklung eines Internet-Cafe Systems:
 - mit hohem Augenmerk auf Manipulations-Sicherheit und Stabilität
 - plattform-übergreifend (Linux & Windows XP)
 - Controller-Software in Java
 - Massive Eingriffe ins Betriebs-System, um die geforderte Manipulations-Sicherheit zu



Software and Network Solutions

- gewährleisten
 - Fernwartung über Nomachine NX + Web-Interface (MySQL, PHP)
 - Fehler-Erkennung mit Meldung in zentraler Datenbank
- **2007** Einrichtung eines Computer-Netzwerkes für eine Spielothek:
 - IP basierte Video-Überwachung
 - Linux-Server
 - Bedienung für das Tresen-Personal über Thin-Client
 - erster Feldtest für Linux-basierte Internet-Terminals
- **2006** Verwertung des Hitachi SR-8000 Großrechners vom Leibnitz-Rechenzentrum München:
 - Demontage vor Ort in bewegliche Einheiten
 - Abtransport in eigenes Lager (insgesamt ca. 40 Tonnen Material)
 - Zerlegung nach Wertstoff-Sorten
 - Dokumentation und Nachweis der Demontage und sachgerechten Verwertung
- **2006** Realisierung eines Video-Aufzeichnungs-Systems für einen Kanal-Sanierungsbetrieb (KKS Rohrsanierung)
 - Einbinden von Framegrabber-Karten unter Windows XP
 - Einbinden eines Axis Video-Servers für Echtzeit MPEG4-Kompression
 - Steuerung eines digitalen Bild-Einblendgerätes über RS-232, GUI und Treiber in TCL geschrieben
 - Hardware-Beschaffung: Industrie-Workstation mit Dual-Head, Video-Elektronik & Framegrabber
- **2005** Entwicklung eines intelligenten Drucksensors für die Firma HF-Automation:
 - Atmel AVR basiert
 - Bedienung über LC-Display, vier Softkeys und Menüs
 - Entwicklung von Hard- und Software
- **2004** Diplom-Arbeit: Entwicklung einer graphischen Schnittstelle für das Tarif-Modell eines Telekommunikations-Großrechners bei der Firma Nortel Networks:
 - als Java/Swing-Applikation realisiert
 - Kommunikation mit dem Großrechner über Telnet
 - Zugriff auf dessen Kommando-Zeile (CLI)
 - Daten-Manipulation in dessen proprietären Datenbank über Absetzen von CLI-Befehlen
 - Schnittstelle zwischen XML und der Nortel Datenbank
 - letztendlich eine GUI, die auf diesen Strukturen aufbaut
- **1999-2000** Druckdaten-Aufbereitung für die Firma Stumpp Versand-Dienstleistungen:
 - Parsen von Spool-Dateien (von IBM AS/400)
 - umsortieren und umformatieren für Kyocera Laserdrucker
 - Programmiersprache TCL
 - Einrichtung eines speziellen Linux-Servers zur Verwaltung für 6 Kyocera Großdrucker