Senior Embedded Linux Architekt
(über 20 Jahre Erfahrung)

- 25 Jahre Erfahrung Linux Architekt / Entwickler / DevOps
- 20 Jahre Erfahrung in Linux/Embedded
- Kernel-Entwicklung und OS-Architektur
- Linux-Kernel/-Treiber Entwicklung
- Offizieller Kernel-Maintainer für verschiedene Treiber
- Anwendungs-Architektur und -Entwicklung
- Build-/Deployment - DevOps
- IoT
- Security

1 Skills

1.1 Hadware-Plattformen:

Treiber-/BSP-Entwicklung für verschiende Hardware-Plattformen:

- **CPU-Architekturen:**
  intel/x86, x86_64, m68k, ppc, arm
- **Interfaces:**
  RS282, RS485, CAN, ISA, USB, ATA, SCSI, WLAN, 3G/4G baseband, GPIO, Ethernet, DSL, Cable, ISDN, SDH, SPI, SSI, MVB, ...
- **System-on-chip:**
  Freescale/NXP i.MX-Serie, TI Sitara-Serie, STM32, AMD-G412, Intel/Ubox SOFIA, uvm.
- **Boards/Vendors:**
  PC-Engines (Linux Kernel maintainer), phyCore, Karo, F+S, Congatec, TQ-Systems, custom boards
- **Peripherie:**
  Video-Codecs, GPUs, ADCs, Encoder, Meß-/Steuergeräte, Sensoren/Camera, Displays, Touch

1.2 Branchenerfahrung (Embedded)

- Automotive / Bahntechnik / Gefahrengut-Transporter
- Medizinprodukte / Medizintechnik *(zB. IEC-62304)*
- Internetprovider + Telco
- Maschinenbau / Anlagenbau / Gebäudetechnik
- Meßtechnik und -Kalibrierung
- Autohäuser und KFZ-Werkstätten

1.3 Betriebssysteme, Programmierwerkzeuge, Toolkits, Libraries

- **Programmiersprachen:**
  C/C++ Oberon/Modula/Pascal, BASIC, Shell, SQL, Perl, Python, Javascript, Java, ASM, tcl/tk, uvm.

- **Build-Tools:**
  GNU make, Imake, Autoconf/Autotools, Unitool, TreeBuild, uvm.
- **Embedded-Toolkits:**
  PTXDist, ELBE, OpenWRT, Embedian, Yocto, Debkins
- **SCM:**
  Git, CVS, P4, Subversion
- **Toolchains:**
  GCC, Clang, OSELAS, Linaro, uvm.
- **GUI-Toolkits:**
  Gtk, Qt, Motif, Xt, Xaw, twtk, uvm.
- **Graphics:**
  X11, SDL, DirectFB, Linux-FB, Wayland, MESA, KMS/DRI, ...
- **DevOps+HIL:**
  Jenkins, Debkins, Labgrid, DI/APT, uvm

2 Referenzprojekte

Im Folgenden Abschnitt wird eine Auswahl realisierter Projekte im Bereich Linux-Embedded *(unsortiert)* vorgestellt.

2.1 (IOT) Beratung/Entwicklung Embedded-Linux/Diagnose - Bahntechnik, Hochsee-Schiffahrt, Kraftwerke

- Einführung nachhaltiger Entwicklungsprozesse für ein Telemetrie-/Diagnose-Gerät in der Bahntechnik, Hochsee-Schiffahrt und Kraftwerken.
- Source control management und Release management
- Hohe Ausfallsicherheit und einfache Langzeit-Wartung
- Konsolidierung der Embedded-Linux Plattform *(uA. generische Treiber-APIs, zB. IIO)*
- robuste, vollautomatische Build-/Deployment-Prozesse *(DevOps)*
- Kernel-Portierung / Treiberentwicklung für div. Controller *(uA. Duagon IONIA)*
- Configurations- und Lizenzmanagement
- Portierung von nodejs auf embedded systems
- Beratung zur SW-Architektur und Entwicklungs-Prozessen
- Beratung zur Chipset-Auswahl / HW-Architektur für Custom Boards
- Evaluation div. HW *(zB. NI cRIO)* auf SW-technische Nutzbarkeit für GNU/Linux-Systeme

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, Nodejs, Barebox, ADC, Git, RS485, CAN, RS232, Duagon IONIA, MVB, cmake, javascript, DevOps

2.2 Security-Beratung - Android - Anti-Tampering

- Beratung zu Android-Security im Zahlungsverkehr/Banking
- Anti-Tamperating / Integritätsprüfung innerhalb Android-Apps

Tools+Spachen: C/C++, GNU make, Linux, Android, Java

2.3 (DevOps) Ansible Modulentwicklung

- Entwicklung kundenspezifischer Ansible-Module
- Provisionierung von edge computing Systemen

Tools+Spachen: Git, Ansible, Python, Debian packaging, DevOps, Edge Computing, Embedded

2.4 Kernel-Treiber / BSP für PCEngines APUv2/APUv3

- Treiber-Entwicklung für AMD G-series (GX-412TC) GPIO
- Treiber-Entwicklung für PC-Engines APUv2 / APUv3 (LEDs + Buttons)
- Integration in die Mainline
- offizieller Kernel-Maintainer
- Build-Umgebung + Debian-Paketiereung
- Debian-basiertes Board Support Package
- Testautomatisierung

Tools+Spachen: C/C++, GNU make, Linux-Kernel, Debian, Devuan, Jenkins, Git, Perl, Python, Shellscript, GNU/Linux

2.5 (IOT) Notrufsystem für Aufzugssteuerung

- Entwicklung von EN-81 konformer Notruf-Funktionalität in einer existierenden Aufzugs-Telemetrie
- Anbindung von Callcenter-Backends
- automatische Provisionierung und ERP-Anbindung
- Treiber-Entwicklung / Kernel-Anpassung *(HMI, Codecs, GSM-TA, ...)*
- Weiterentwicklung des Board Support Package
- Konsolidierung der SW-Architektur zur Unterstützung verschiedener Produktvarianten und landesspezifischer normativer Vorgaben
- Vereinfachung und Optimierung der gewachsenen Codebase, Ablösung von Legacy-Code durch Linux-Bordmittel
- Schulung zur Entwicklung unter GNU/Linux / Linux-Embedded
- Automatisierung und Optimierung von Build- und Deployment- Prozessen und Source-Control-Management

- Beratung bzgl. Software-Architektur und Entwicklungsstrategie
- Schulung der Kollegen bzgl. Entwicklung unter GNU/Linux

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, Azure, Git, Perforce, Redmine, LaTeX, RS485, CAN, RS232, cmake, javascript

2.6 (IOT) Beratung/Entwicklung Embedded-Linux Aufzugstechnik

- Beratung bei einer Linux-Embedded-basierten Aufzugssteuerung
- Automatisierung und Optimierung von Build- und Deployment- Prozessen und Source-Control-Management *(DevOps)*
- Beratung bzgl. Software-Architektur und Entwicklungsstrategie
- Schulung der Kollegen bzgl. Entwicklung unter GNU/Linux
- Ablösung von Legacy-Code durch Linux-Bordmittel
- Code-Reviews und Cleanup

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, Git, Perforce, LaTeX, RS485, CAN, RS232, cmake, javascript, DevOps

2.7 (IOT) Beratung/Entwicklung Embedded-Linux Aufzugstechnik

- Weiterentwicklung eines Telemetrie-/Diagnose-Gerät für Aufzugsanlagen *(predictive maintenance)*
- BSP- und Kernel-Anpassungen
- Ablösung von Legacy-Code durch Linux-Bordmittel und Optimierung
- Refactoring der HW-Ansteuerung / Treiberschicht *(HMI, Modem/GSM-TA, etc)*
- Refactoring der Techniker-Funktionen
- Refactoring der Cloud-Kommunikation
- Implementierung von Selbstdiagnose *(zB. Verbindungstest, Aufzugswärter, etc)*
- Anbindung verschiedener Aufzugssteuerungen
- Harmonisierung auf internationale Gegebenheiten *(div. Gehäuse/LED-Panel, Bedienfunktionen, etc.)*
- Architektur- und Code-Review

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, Azure, Git, Perforce, Redmine, LaTeX, RS485, CAN, RS232, cmake, javascript

2.8 Code-Generator für CAN-Protokolle (automotive)

- Toolkit für die formale Beschreibung von CAN-Bus'en, -Protokollen und deren Teilnehmern
- automatisierte Konsistenzprüfung *(zB. ID-Konflikte, Erreichbarkeitsgraphen, Payload-Sizes, etc)*
- Code-Genratoren für verschiedene Sprachen und MCU-Plattformen
- autogenerierte Code für Router/Firewalls, Codierung/Decodierung, Message handlers, etc-
- Build-Engieering / Integration in DevOps-Umgebung

Tools+Spachen: C/C++, GNU make, Linux-Kernel, CAN-Bus, Jenkins, Git, Perl, Python, Shellscript, GNU/Linux, FreeRTOS, QNX

2.9 CAN/RS485 datalogger (Maschinenbau / IOT)

- Konzeption / Entwicklung eines CAN- und RS485 Datalogger *(mobile und im Feld schwer zugängliche Geräte)*
- Für mobile und im Feld schwer zugängliche Geräte mit oft nur sporadischer Datenverbindung
- Minimales Linux-System *(PtxDist)* mit kundenspezifischem BSP *(incl. Kernel-/Treiber-Entwicklung)*
- Logging über verschiedene Schnittstellen *(zB. CAN, RS485, RS232, MODBUS, IP)* und kundenspezifischen Protokollen
- ggf. Sonderlösungen zur Datenanbindung *(zB. AX25 / packet radio)*
- Logging auf interne microSD und Upload via UUCP und RSYNC.
- Vollautomatisches Build, Deployment, Field-Updates
- Konfigurations-Management
- Factory-Tools

Tools+Spachen: C/C++, Linux-Kernel, Barebox, PtxDist, Git, CAN, RS232, RS485, UUCP, DevOps

2.10 Vollautomatischer HIL-test via CI-RT und Labgrid

- Integration von CI-RT, Labgrid, Jenkins, Docker
- Vollautomatische Provisionierung der CI-Umgebung via Docker
- Tool-Paketierung für ELBE

Tools+Spachen: C/C++, GNU make, Linux-Kernel, ELBE, Debian, Jenkins, CI-RT, Git, Perl, Python, Labgrid, Shellscript, GNU/Linux

2.11 Container-Lösnug für Embedded/IOT/Edge

- Container-Lösung für Embedded/IOT/Edge-Devices
- Crosscompiling-Umgebung für Google Go
- Portierung von conainerd *(Docker)*
- Integration in CI/CD/HIL

Tools+Spachen: C/C++, GNU make, Linux-Kernel, Debian, Jenkins, Docker, Go, Git, Python, Labgrid, Shellscript, GNU/Linux

2.12 Crosscompile-Umgebung für Rust (Embedded)2019

- Crosscompiling-Umgebung für Rust
- Anbindung an PtxDist-Buildumgebung
- Integration in CI/CD/HIL

Tools+Spachen: C/C++, GNU make, Rust, Debian, Jenkins, Docker, Go, Git, Python, Labgrid, Shellscript, GNU/Linux

2.13 Kernel-Treiber / BSP und SW-Architektur für Outdoor display panels

- Treiber-Entwicklung / BSP für iMX.6-basierte Display-Boards
- Build-Umgebung + Debian-Paketiereung
- Debian-basiertes Board Support Package
- Testautomatisierung
- Software-Architektur
- Surf-Browser basierte display server software *(Content vom Datacenter bespeist)*

Tools+Spachen: C/C++, GNU make, Linux-Kernel, PtxDist, Git, Perl, Python, Qt, Webkit, Surf, Shellscript, GNU/Linux

2.14 (DevOps) Docker-Buildpackage: Build-Automatisierung für Debian-/Ubuntu-Pakete

- Tool zum einfachen und vollautomatischen Build von deb-Paketen und Paket-Repositories
- strikt isolierte Build-Umgebung via Docker
- deutlich leichter aufzusetzen als bisherige Tools *(zB. pbuilder oder buildd)*
- dient uA. der Pflege kundenspezifischer oder auch eigener LTS-Repositories
- einfache Integration in CI *(zB. Jenkins)*

Tools+Spachen: Debian, APT, dpkg, Docker, docker-buildpackage, CI, Jenkins, DevOps

2.15 Embedded-Linux Entwicklungsplattform und Paket-Portierung

- Entwicklung kundenspezifischer Firmware für DSL-Router
- ermöglicht die Bereitstellung mehrere virtueller Anschlüsse und öffentlicher WLAN-Zugänge über bereits im Feld vorhandene DSL-Router
- Kernel-/Treiber-Anpassung, Build-/Deployment-Prozesse
- MPLS-Integration
- Anpassung zahlreicher Opensource-Pakete für Embedded/Crosscompiling

Tools+Spachen: C/C++, GNU make, Linux-Kernel, MPLS, Crosstool, Briegel, Diffutils, CVS, Perl, Shellscript, GNU/Linux, metux Briegel

2.16 (IOT) Kernel / BSP für Duagon IONIA (Automotive, Bahntechnik)

- Generisches Board Support Package für Duagon IONIA *(Bahntechnik)*
- Basis-System und reproduzierbares Build via PtxDist
- aktueller Mainline-Kernel und Barebox
- Kernel-Treiber für Backplane *(serial/tty)* und verschiedene Module *(IIO)*
- Beratung/Schulung zu IIO

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, ADC, Git, CAN, RS232, Duagon IONIA, MVB, cmake

2.17 (IOT) Lizenzmanagement für Embedded Devices (Automotive, Bahntechnik, Schiffsbau)

- Konzeption/Infrastruktur für Lizenzmanagement bei Telemetrie-/Diagnosegeräten
- Verwaltung der Anlagen und Freischaltung gebuchter Features im Embedded-Device
- Verwaltung der Lizenznutzung zugekaufter SW-Komponenten
- Autogenerierung kundenspezifischer Geräteconfigurationen
- Cryptographische Absicherung und Secure Boot
- Reporting-Werkzeuge, ERP-Anbindung, etc

Tools+Spachen: PtxDist, OpenSSL, GnuPG, Packaging, Jenkins, imx6, trusted boot, PostgreSQL, ERP, DevOps

2.18 Java/JNI-Anbindugn der Linux Kernel Crypto API

- Einbindung der Linux Kernel Crypto API in Java Virtual Machine zwecks besserer Performance und Sicherheit
- Build-Umgebung / BSP *(yocto)*

Tools+Spachen: C/C++, Java, JNI, GNU make, Linux-Kernel, Yocto, Git, Shellscript, GNU/Linux

2.19 Linux Bluetooth feature backports (automotive)

- Backports von Bluetooth-Features neuer Kernel auf alte Vendor-Kernel
- Portierung auf SOFIA-SoC (Intel/uBlocks)
- Kernel-Treiber

Tools+Spachen: C/C++, GNU make, SOFIA SoC, Linux-Kernel, Debian, Devuan, Jenkins, Git, Perl, Python, Shellscript, GNU/Linux

2.20 HMI mit Dualshock4-Controller (Maschinenbau)

- Anbindung eines Bluetooth DS4-Controller an Linux-basiertes HMI
- Anpassung des BSP und QT
- Build-Engineering / Packaging / CI

Tools+Spachen: C/C++, GNU make, Linux-Kernel, Debian, Devuan, Jenkins, Qt, Git, Perl, Python, Shellscript, GNU/Linux

2.21 (IOT) HW-Anforderungen und Embedded-Linux BSP (Bahntechnik)

- Entwicklung eines Board Support Package für ein Bahntechnik-Gerät *(predictive maintenance)*
- HW-Systemanforderungen und Beratung zur Chip-Auswahl
- Spezifikation der Schnittstelle zwischen CPU und ADC-Baugruppe/FPGA
- Anpassung von Kernel und Bootloader, Treiber-Entwicklung
- automatisierte Build-Umgebung via PtxDist und Jenkins
- Factory-Tools

Tools+Spachen: C/C++, Linux-Kernel, Barebox, PtxDist, ADC, Git, CAN, RS232, Bitbucket, Jenkins, DevOps

2.22 OPAL-Implementierung auf Linux-Embedded-Device (automotive)

- Implementierung von OPAL (SSD-Verschlüsselung) in Embedded-Devices
- Referenz-Testing für verschiedene SSD-Hersteller/-Modelle
- Kernel-Anpassungen / Treiber-Portierung
- Bugfixing in OPAL toolchain
- Testautomatisierung

Tools+Spachen: C/C++, GNU make, OPAL, sed-util, Linux-Kernel, Debian, Devuan, Jenkins, Git, Perl, Python, Shellscript, GNU/Linux

2.23 Java/JNI-Anbindugn von OpenSSL

- Einbindung von OpenSSL in Java Virtual Machine zwecks besserer Performance und Sicherheit
- Build-Umgebung + Debian-Paketiereung

Tools+Spachen: C/C++, GNU make, Debian, Devuan, Git, Python, Shellscript, GNU/Linux

2.24 HIL-Test/Werksprüfung: Gebäudetechnik / Embedded-Linux

- Test-System zur Werksprüfung von Premium-Haustechnik
- Netzwerk-Schnittstellen *(Last- und Langzeittests, Stabilität, ...)*
- Display- und Touch: Kalibierung und Qualitätskontrolle
- Thermische Prüfung
- Prüfung von IOs.
- Test von Audio/Video-Streaming, Codec, GPU

Tools+Spachen: C/C++, Linux-Kernel, PtxDist, Git, RS485, CAN, RS232, cmake, Qt, GStreamer, Labview

2.25 Embedded: Spriometrie- und EKG-Meßcomputer für klinische Studien

- Weiterentwicklung und Moderinisierung medizinischen Meßgeräten für klinische Studien
- Spirometrie- und EKG-Messungen nach Studien-spezifischen Workflows
- Integration einer zugekauften EKG-Analyse-Kompente *(Corscience HES)*
- Portierung auf neue Hardware *(neues BSP)*
- Treiber-Entwicklung für den integrierten Drucker *(angepaßter HP OJ-470)*
- Source control management und vollautomatische Buildprozesse *(DevOps)*
- Einführung eines Configurations-Systems zur Ablösung bisheriger studienspezifischer Code-Anpassungen
- Entwicklung eines Standardgeräts zur regulären Nutzung in der Praxis *(außerhalb von Studien)*

Tools+Spachen: C/C++, Perl, GNU/Linux, Kernel-Entwicklung, Git, GCC, Crosscompiler, ARM/iMX.21, GNU make

2.26 Beratung/Entwicklung Embedded-Linux Medizintechnik

- Beratung und Entwicklung für ein Medizinprodukt *(Reinigung und Sterilisation)*
- IEC-62304 / Klasse C
- ptxdist-basiertes BSP *(Board Support Package)* nebst Treiberentwicklung/-Anpassung *(uA. auch Portierung von Qt5 auf KMS/DRI, GPU-Treiber, etc)*

- Build- und Deployment-Prozesse mit Hinblick auf Fertigung und Wartung im Feld *(DevOps)*
- Automatisierung von Dokumentation und Code-Generierung
- Sicherheitskonzepte *(zB. Manipulationssicherheit und Knowhow-Schutz)*
- Konzepte für Wartung und Langzeitpflege der Software
- cryptographisch gesicherte Wartungsschnittstelle
- Evaluation neuer Prozessor-/SOC-Generationen *(uA. IMX6)*
- Review und Abnahme der Zulieferungen externer Dienstleister
- Hardware: Custom-Boards mit imx.5, imx.6, stm32

Tools+Spachen: LaTeX, C/C++, Linux-Kernel, ptxdist, Git, shellscript, python, cmake, GNU make, OpenGL/Mesa, Qt5, imx53, imx6, CAN

2.27 (IOT/Automotive) Beratung zu Entwicklungsmethoden im Embedded-Linux-Bereich

- Kunde ist Hersteller von Meß- und Erfassungstechnik für Tanklastfahrzeuge *(zB. Treibstoff, Milch, etc)*
- Konzept zur Modernisierung der Basis-Software und Entwicklungsprozesse, Build-/Deployment, DevOps, Langzeitpflege und Portierung auf neue Hardware
- Konzeption für hochverfügbare Datenbank-Infrastruktur

Tools+Spachen: Git, SVN, C/C++, Linux-Kernel, GNU make, USB, DevOps

2.28 (DevOps) Vollautomatisches Deployment virtueller Entwicklungsumgebungen

- Bereitstellung von kundenspezifischen Embedded-Entwicklungsumgebungen als VM- und Container-Images
- Vollautomatisches Build und Provisionierung *(via Ansbile und Vagrant)*
- automatisches Rollout auf den Endgeräten
- CI-Integration *(Jenkins)* und automatische Provisierung der CI selbst
- Paketierung diverser Zusatz-SW für Debian/Ubuntu *(VTK8, Eclipse, Sigasi Studio, Vivado, ...)*
- Automatisierung des Deployment kommerzieller Software *(Vivado, Sigasi, Sysgo PikeOS/Codeo, Modelsim, Questa)*

- Lizenz-Prüfung *(zB. Einhaltung div. FOSS-Lizenzen)*

Tools+Spachen: Debian, APT, Ansible, Vagrant, Jenkins, ESXI, Docker, Vivado, Modelsim, Questa, PikeOS, Codeo, Eclipse, Sigasi

2.29 IOT-Lösung für Windkraftanlagen

- IOT-Lösung zur Umweltüberwachung an Windkraftanlagen
- Linux-Plattform / BSP (Debian-basiert)
- Portierung von Raspberry PI auf PC-Engines APU
- Treiber-Entwicklung für PC-Engines Mainboard + Peripherie
- Web-Interface und Cloud-Anbindung
- Remote-Update
- Build-Umgebung + Debian-Paketiereung + Deployment
- Testautomatisierung

Tools+Spachen: C/C++, GNU make, Linux-Kernel, Debian, Devuan, Jenkins, Git, Perl, Python, Shellscript, GNU/Linux

2.30 (IOT) Maßgeschneiderte GNU/Linux-Firmware für WLAN-VoIP-Router

- in einem Forschungsprojekt zu mobiler Infrastruktur für Telefonie- und Videodienste auf Basis gängiger WLAN-Technik
- angepaßte/optimierte Firmware für gängige WLAN-Router und Integration in Provider-Backbones *(zB. via MPLS)*
- Board Support Packages auf Basis eines selbst entwickelten Embedded Build-System *metux Briegel Builder*
- Build- und Deploymentprozesse für hochautomatisierte Pflege und Wartung *(DevOps)*
- Integration von MPLS in die Linux-basierte Firmware
- Container-Lösung zur einfachen Integration von Drittsoftware auf die Router *(zB. Nutzung der Leerlaufkapazitäten für Cloud-Computing)*
- Anpassung zahlreicher Opensource-Pakete für Embedded / Crosscompiling
- Weiterentwicklung von Linux-Treibern *(WLAN, MPLS/RSVP, etc)*

Tools+Spachen: C/C++, Perl, Java, MPLS, WLAN, Intel Geode, Crosstool, Briegel, GNU make, GNU autotools, diffutils, CVS, Subversion, Shellscript, GNU/Linux, metux Briegel

3 R+D: Eigene Projekte und Produkte

3.1 Debkins: vollautomatische CI und Deployment für Debian-basierte IOT-Geräte

- Vollautomatische CI- und Deployment-Umgebung für Debian-basierte Geräte
- Optimiert auf IoT-Devices
- automatisiert den gesamten Prozess vom Quellcode bis zum Zielgerät
- automatische Provisionierung via ”Configuration-As-Code”

3.2 Modellierung und Code-Generator für CAN-Umgebungen

- Modellierung von CAN-basierten Systemen *(Messages, Payloads, Bus-Topologien, Kommunikationsbeziehungen, etc)* via YAML
- Konsistenz- und Konfliktprüfungen
- leicht anpaßbare Code-Generatoren für die einzelnen Bus-Teilnehmer *(MCUs, Switches, Prüfgeräte, etc)* mit verschieden Zielsprachen und -Plattformen
- autogenerierte Firewall- und Routing-Regeln *(zB. netfilter/BPF)*
- für DevOps und agile Entwicklung gerüstet: leicht integrierbares und portierbares Baukastensystem statt monolithische SW-Suite

3.3 Plan9 subsystem für Linux

- Integration von Plan9 Basisfunktionen im Linux-Kernel
- Textuelle Usernames statt nur numerische IDs
- Deaktivierung von SUID innerhalb User-/Mount Namespace
- Mouting und Namespace-Forking für unprivilegierte User
- Authentifizierung/User-Wechsel via /dev/capuse
- Portierung des Factotum Authentifizierungs-Agent
- sukzessive Implementation diverser Plan9-Services

Tools+Spachen: Linux-Kernel, Plan9, C, LXC, coreutils, gcc, make, bash, rc, factotum

3.4 Linux-Kernel: Text-basierte Kernel-Schnittstellen via Dateisystem

- Textuelle Kernel-Schnittstellen und Devices als virtuelle Filesysteme statt klassische IOCTLs
- Vereinfacht Scripting und Debugging *(zB. keine nativen Bindings bei Scriptsprachen mehr nötig)*
- Keine Abhängigkeiten von Binärformaten *(zB. struct alignments, byteorder, etc)* mehr
- Ermöglicht direkte Verwendung von Geräten via Netzwerk-Filesysteme *(zB. NFS, 9P, etc)*

Tools+Spachen: C/C++, GNU make, Linux-Kernel, GCC, 9P, Shellscript

3.5 GNU Droid – Android-Umgebung auf GNU/Linux

- Android-Subsystem für GNU/Linux-Systeme
- Ermöglicht den Betrieb von Android-Apps innerhalb klassischer GNU/Linux Desktops
- Verwendung von Android-Komponenten als GUI-Toolkit
- Portierung des extreme powersaving *(wakelock, etc)* auf mobile Desktops
- Dalvik als Alternative zur klassischen JVM
- Integration klassischer GNU/Linux-Komponenten in die Android-Welt

Tools+Spachen: Android, Linux-Kernel, Git, C/C++, Java, Dalvik, cmake, GNU make, auto-tools, ptxdist, Shellscript

3.6 Briegel Builder - vollautomatisiertes Buildsystem für Firmware und Distros

- Buildsystem für sehr speziell angepaßte GNU/Linux-Systeme / -Images
- ähnlich wie zB. Ptxdist, jedoch außerhalb der Embedded-Welt, zB. Container, einsetzbar
- Vollautomatisiertes, reproduzierbares Build

Tools+Spachen: Java, PHP, GNU Make, Ant, PostgreSQL , Unitool, GNU Autotools, Kaffe, GCJ, Crosstool

3.7 Comprehensive Source Database

- Normalisierte Datenbank mit vielen Opensource-Quellpaketen und entsprechenden Versionen
- Crawler zur automatischen Erfassung neuer Versionen / Releases
- Automatisierte Benachrichtigungen und Anbindung an Build-/QA-Plattformen

Tools+Spachen: C/C++, PHP, PostgreSQL , plsql, Perl, CVS, Git, SVN, Mercurial, Shellscript, DevOps

3.8 OpenSource Software QM Taskforce

- Dachprojekt zur Pflege / QM von zahlreichen Opensource-Paketen
- Normalisierte Repository-Layouts and Versions-Schemata
- Automatisiertes Sammeln von Patches/Bugfixes aus den verschiedensten Distributionen
- Langzeitpflege *(LTS)* von ausgewählten Versionen
- Sicherstellung automatisierter Buildprozesse *(insbesondere zB. für Embedded: Crosscompiling)*

Tools+Spachen: DevOps, Git, C, Java, Perl, GNU diffutils, Subversion, CVS, Git, PHP, GNU make, GCC, Crosstool, Shellscript

3.9 Linux-Kernel: Toolkit zur Meta-Configuration

- Highlevel-Configuration des Kernel auf Basis von Aufgaben und Hardware-Profilen, statt individuelle Optionen
- Autogenerierung der Kernel-Config für verschiedenste Versionen aus der Meta-Configuration heraus
- Ableitung der Treiber-Configuration aus dem Devicetree

Tools+Spachen: C/C++, GNU make, Linux-Kernel, GCC, Shellscript, Git, DevOps

3.10 Linux-Kernel: Debian-Paketierung

- Verbesserung der Debian-Paketierung des Linux-Kernels
- Patch-Management direkt via Git (anstelle bisheriger textueller Patches)
- Integration in die Debian build pipeline (dck-buildpackage, etc)

- Einführung von high-level Kernel-Configuration statt manueller Pflege
- Mittelfristig Ablösung des existierenden Pakete im Debian / Devuan
- Backports für Debian, Devuan, Unbuntu *(auch alte sehr Releases)*

Tools+Spachen: C/C++, GNU make, Linux-Kernel, GCC, Shellscript, Git, DevOps, Debian, Devuan, dck-buildpackage

3.11 Linux-Kernel: Modbus protocol stack

- Generische MODBUS Socket API *(analog zum CANSocket)*
- Treiber-Framework für MODBUS
- Integration in andere Kernel-Subsysteme, um via MODBUS angeschlossene Geräte über Standard-APIs *(zB. LED, hwmon, IIO, etc)* anzusprechen
- Security via Linux-Firewall *(netfilter/BPF)*

Tools+Spachen: Linux-Kernel, Modbus, Socket-API, Kernel-Treiber, netfilter, Linux-Embedded