

# Profil: Software-Entwickler, System-Integrator

## 1 Persönliche Daten

---

Name:	Enrico Weigelt
Geburtsjahr:	1980
EDV-Erfahrung seit:	1992
Ausbildungsabschluß:	Abitur, Leistungskurs Informatik (bei Prof.Dr.Fothe)
Nationalität:	Deutsch
Sprachen:	Deutsch (Muttersprache), Englisch (praxistauglich)
Verfügbarkeit:	kurzfristig, nach Absprache auch Vollzeit
Regionale Flexibilität:	Deutschsprachiger Raum (evtl. englischsprachiger Raum)
Besondere Bemerkungen:	Politiker, 2004-2008 Abgeordneter ( <i>Kommunalparlament</i> )

---

Kontaktdaten:	Anschrift: Heydear Straße 23, 99338 Plaue
	Telefon: +49-174-7066481
	eMail: weigelt@metux.de
	WWW: <a href="http://www.metux.de/">http://www.metux.de/</a>

---

## 2 Spezialgebiete

- GNU/Linux-Experte (from scratch und alle gängigen Distros)
- Embedded-Linux / Small-Devices
- Plattformübergreifende Portierung
- Code-Audits, Cleanups, Build-Management
- Laufzeit-Migration
- Paketierung, Distribution, Configuration-Management
- Protokoll- und Datenanalyse, Forensik

## 3 Branchenerfahrung

- Finanzwesen (Börsenhandel)
- IT Geräteentwicklung (Embedded systems)
- Internetprovider + Telco (seit 5 Jahren selbst ISP)
- Medizin (Ärzteverbände)
- Politik und NGOs (selbst Abgeordneter)
- Meßtechnik und -Kalibrierung
- KFZ-Werkstätten
- Werbebranche
- Einzelhandel
- Textil-Manufaktur

## 4 Softwaremethoden / -Gebiete

- OO-Methoden und -Programmierung (components, signals, etc)
- Constraint based programming + Unit-Tests
- Hardwarenahe Programmierung
- Embedded Systems
- Netzwerkprogrammierung, Protokollentwicklung und -Analyse
- Datenmodellierung (logisch, workflow, physisch)
- Extreme Programming
- Verteilte Systeme

## 5 Skills: Programmiersprachen

- Pascal (12 Jahre Projekterfahrung)
- BASIC (13 Jahre Projekterfahrung)
- C/C++ (14 Jahre Projekterfahrung)
- Unix-Shellprogrammierung (12 Jahre Projekterfahrung)
- Oberon/Modula (9 Jahre Projekterfahrung)
- postgresQL PL/SQL (9 Jahre Projekterfahrung)
- Perl (9 Jahre Projekterfahrung)
- PHP (9 Jahre Projekterfahrung)
- Java (9 Jahre Erfahrung)
- HTML + CSS (seit 10 Jahren Projekterfahrung)
- LaTeX (Seit 7 Jahren Projekterfahrung)
- Javascript/ECMA-Script (gute Kenntnisse)
- Prolog (gute Kenntnisse)
- Assembler (gute Kenntnisse)
- XUL - XML Userinterface Language
- Haskell
- m4
- tcl/tk

## 6 Skills: Programmierwerkzeuge, Toolkits, Libraries

- GNU make
- Imake
- Autoconf, automake, libtool
- UnitTool, TreeBuild
- Git, CVS, Subversion, Bugzilla, Trac,
- GNU Toolchains, GDB (GNU debugger)
- Gtk, Qt, Motif, Xt, Xaw
- Yacc/Lex/Bison
- \*nix utils (awk,sed,perl,...)
- unixODBC, JDBC, libpq (PostgreSQL), mySQL, sqLite
- OpenSSL, GNU TLS
- SAX, Expat
- X11, SDL, DirectFB, Linux-FB
- 9P2000, Plan9Port (Plan9 on Unix)

## 7 Skills: Betriebssysteme

- GNU/Linux - Embedded, from Scratch und div. Distros (Experte)
- xBSD (gute Kenntnisse)
- andere UNIXe (gute Kenntnisse)
- Plan-9
- Native Oberon (gute Kenntnisse)
- MS-DOS (sehr gute Kenntnisse)
- Novell Netware (gute Kenntnisse)
- Windows 3.1 (4 Jahre Projekterfahrung)
- Windows 9x (4 Jahre Projekterfahrung)
- Windows NT/2000/XP (gute Kenntnisse)

## 8 Skills: Datenbanken

- postgresQL (7 Jahre Projekterfahrung)
- mySQL (seit 3 Jahren Projekterfahrung)
- sqLite
- LDAP
- Berkeley-DB + nDBM + GNU DBM
- Xindice
- Synthetische Filesysteme

## 9 Skills: Softwarepakete (incl. Entwicklung)

- Apache Webserver (Entwickler des Multiplexer MPM)
- sendmail Mailserver (incl. selbstentwickelte Admin-Tools)
- qMail (Entwickler)
- X-Window, GNOME, KDE, gnuStep (GUI-Programmierung)
- GCC (GNU compiler collection) und andere GNU Tools
- Mozilla + XUL (Admin, Entwickler, Crosscompiling)
- OpenOffice (Admin, Entwickler)
- Ghostscript (gute Kenntnisse)
- (La)Tex (4 Jahre Projekterfahrung)
- CRM: vTiger (Entwickler)
- CMS: Joomla, Content\*Builder, Typo3
- VMware, QEmu, VirtualBox, DosBOX, WinE
- Plan9 from outer space

## 10 Skills: Hardware

- Architekturen/Plattformen: intel/x86, ppc, arm, Commodore, div. Embedded (zB. Router)
- Interfaces: RS282, ISA, USB, ATA/IDE, SCSI, WLAN, Ethernet, DSL, Cable, ISDN, SDH

## 11 Skills: Netzwerke

- Transport-Protokolle: TCP/IP (IPv6, IPv6, IPSEC, IPIP, route management), IPX, NetBEUI, PPP, SLIP
- Anwendungs-Protokolle: SSL, HTTP, SMTP, UUCP, Streaming
- Management-Protokolle: SNMP, RADIUS, HTCPCP
- Server-Anwendungen: Apache HTTPD, Tomcat, Sendmail, QMail, ISC DHCP, NPFS
- Netzwerk-Filesysteme: NFS, 9P2000, SMB, Coda

## 12 Skills: Web-Technologien

- HTML, CSS, XML, Templates
- XUL
- AJAX, Meshups, XMLRPC
- Java Applets
- Scanner / Crawler, Suchmaschinen
- Web-Services

## 13 Beteiligung an Opensource-Projekten (kleine Auswahl)

- Joomla CMS + Extensions: Entwicklung, QM
- Apache Webserver - multiplexer MPM: Initiator/Maintainer
- Expat XML Parser: Cleanups, Crosscompiling
- BusyBox: Cleanups, Crosscompiling
- zLib Compression Library: Entwickler, Buildsystem + Crosscompile
- bZip2 Compression Tool + Library
- Mozilla Suite: Buildsystem, Cleanups
- Crosstool (crosscompiling suite)
- OSS QM Taskforce: Initiator/Maintainer
- Comprehensive Source Database: Initiator/Maintainer
- OpenVZ (Linux virtualization): Fixes, Crosscompiling
- Linux-Vserver (Linux virtualization): Fixes, Crosscompiling
- coLinux (Linux virtualization): Fixes
- GPM Console mouse server and library
- OpenSSL: crosscompilation + cleanups
- PkgConfig: crosscompile support
- CODA distributed filesystem
- PHP engine: crosscompilation + cleanups
- Linux VLAN: crosscompilation + cleanups
- Xwindow/X11/Xorg: modular buildsystem cleanups + crosscompilation
- BIND (Nameserver): build cleanups + crosscompilation
- iproute2: build cleanups + crosscompilation
- patTemplate (PHP template engine): Entwickler
- Links web-browser: buildsystem + crosscompilation
- Midnight Commander: buildsystem + crosscompilation, virtual filesystem (MVFS)
- MII-Tool standalone package: maintainer + buildsystem
- ISC DHCPd - modular branch
- ISSTV (sstv protocol package): maintainer
- Spamassassin: standalone spamc package
- SU-wrapper: Entwickler/Maintainer
- Xmeter: Entwickler/Maintainer
- CryoSleep: Entwickler
- JBidWatcher: Entwickler
- Unitool (universal toolchain frontend): Entwickler/Maintainer

- TreeBuild (buildsystem w/ declarative model): Entwickler/Maintainer
- Linux-Kernel: crosscompiling, 9P-fs, Coda-FS, div. Treiber
- Plan9
- Freenet (anonymous distributed storage / anonymos web)
- u.v.m

## 14 Referenzprojekte

### 14.1 Media-Portal Plattform: MediaCloud

Für den Die beiden Ärzteverbände "Deutsches Forum für Psychotherapie" (DPI) und die "Vereinigung psychotherapeutisch tätiger Kassenärzte" (VPK) haben metux IT service ihre komplette EDV überantwortet. Neben alltäglicher Wartung von EDV-Anlagen und redaktionellen Diensten zählt hierzu eine ASP-Lösung (Anwendungsdienst), mit der die Verbände ihre Mitglieder verwalten, Beiträge berechnen und sich in zahlreichen geschlossenen und offenen Mailinglisten austauschen können.

Derzeit wird zudem eine Lösung erarbeitet, mit der sie Verbände ihre Rundschreiben via Web erstellen, in Druckereiqualität automatisch setzen und anschließend sowohl als Newsletter ebenso wie als Briefpost verteilen lassen können.

*Tools+Spachen: PHP, Perl, PostgreSQL, Javascript, LaTeX, Majordomo, Java, CVS, Shellscrip, Lighttpd, LAMP*

### 14.2 Virtualisierte und Redundante Serverumgebung (XrentX)

Der Kunde betreibt mehrere Server mit verschiedensten Anwendungen (Mailserver, Webserver, Datenbanken, eigenes Web-Portal), die für den Geschäftsbetrieb kritisch sind. Aufgrund eines bereits erfolgten Angriffs ist dem Kunden die Sicherheit besonders in den Mittelpunkt gerückt.

Um die Schadmöglichkeiten im Falle eines Angriffs auf ein Minimum zu reduzieren, wird das ganze System mittels OpenVZ virtualisiert: jeder Dienst erhält eine eigene virtuelle Umgebung.

Die Software innerhalb der einzelnen VZ's stammt nicht aus einer gängigen Distribution, sondern wird mittels der Eigenentwicklung Briegel (siehe Projekte) für jede einzelne VZ speziell erzeugt. Dabei enthält jede VZ nur jeweils nur einen Service und ausschließlich den dafür benötigten Programmcode. Updates, regelmäßige Integritätsprüfung und automatische Wiederherstellung der einzelnen VZ's finden vom Host-System aus statt. Das Host-System selbst ist im Normalbetrieb nicht via Internet erreichbar.

Eine nicht zu unterschätzende Herausforderung besteht in diesem Projekt darin, sämtliche benötigten Pakete für Crosscompiling/Sysroot vorzubereiten - viele Pakete sind dafür noch nicht von Haus aus geeignet.

*Tools+Spachen: C/C++, Java, Perl, Shellscrip, Briegel, Sendmail, lighttpd, PHP, Apache, qMail, IMAP, POP3, LDAP, PostgreSQL, mySQL*

### 14.3 Embedded-Linux Entwicklungsplattform und Paket-Portierung

Der Kunde (*aufgrund NDA nicht näher genannt*) entwickelt eine ISP-Plattform zum Aufbau von öffentlichen WLAN-Netzen mittels gängigen DSL-Anschlüssen und den dort verfügbaren Endgeräten.

Dazu wird für die gängigen DSL-WLAN-Router (zB. FritzBox) eine eigene Linux-basierte Firmware mit einer Virtualisierungslösung benötigt. Existierende Distributionen/Werkzeuge (zB. OpenWRT) wurden für ungeeignet befunden.

Ich habe für dieses Projekt ein Buildsystem geliefert, die sämtliche Build-Schritte - vom Herunterladen der Quellen, Patching, Compilieren in einer virtuellen Umgebung (SYSROOT) bishin zur Paketierung und Image-Generierung - vollautomatisch ausführt.

Dazu gehören zahlreiche Anpassungen und Fixes an den verwendeten Paketen (zB. um ein sauberes Crosscompiling zu ermöglichen), für die von mir Patches geliefert wurden.

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

## 14.4 Maßgeschneiderte GNU/Linux-Firmware für WLAN-VoIP-Router (TU-Ilmenau)

Die TU-Ilmenau entwickelte in einem Forschungsprojekt eine Infrastruktur für mobile Telefonie- und Videodienste auf Basis von gängiger WLAN-Technik. Es werden dabei sowohl die Zellrouter, das MPLS-basierte Traffic-Management-System und Referenzfirmware für mobile Endgeräte entwickelt.

Meine Aufgabe bestand vor allem darin, eine möglichst schmale und optimierte GNU/Linux-basierte Firmware auf Grundlage weltweit entwickelter Komponenten zu implementieren und zukünftig auch ständig auf dem neuesten Stand zu halten. Gängige Distributionen sind hier aufgrund der speziellen Systemanforderungen nicht geeignet.

Dank meines selbst entwickelten **Briegel Buildsystem** war diese Aufgabe - im Vergleich zu gewohnter Handarbeit - recht unkompliziert und rasch zu meistern.

Viele der verwendeten Pakete waren noch nicht für Embedded-Systeme und Crosscompiling geeignet. Hierzu habe ich entsprechende Patches geliefert.

Wichtig war jedoch auch die Anpassung und Weiterentwicklung des Linux-Kernels. Beispielsweise waren zu diesem Zeitpunkt die MPLS+RSVP-Implementation, wie auch verschiedene WLAN-Treiber noch nicht produktiv einsetzbar. Hier habe ich dem Kunden einen produktiv benutzbaren Kernel geliefert.

*Tools+Sprachen: C/C++, Perl, Java, MPLS, WLAN, Intel Geode, Crosstool, Briegel, GNU make, GNU autotools, diffutils, CVS, Shellscript*

## 14.5 Refactoring vom QMail Mailserver-Paket (XrentX)

Der Kunde möchte das QMail-Paket aufgesetzt bekommen und ggf. selbst administrieren. Zwar ist das QMail-Paket für seine Sicherheit berühmt, aber bisher noch extrem schwer zu installieren/warten.

Erschwerend kommt hier hinzu, daß ein wichtiges Produktionssystem von Sendmail auf QMail umgestellt werden soll. Eine minimale Downtime und keinerlei Datenverlust sind hier unabdinglich.

Im Sinne einer längerfristig leichteren Administration und Pflege haben wir uns zunächst zu einem Refactoring des gesamten Build- und Installationsprozeß entschlossen. Dabei ist vor allem wichtig, daß dieser ohne manuelle Eingriffe sauber in einem Zug abläuft.

Das von mir bereitgestellte Patch sorgt für einen sauberen Build- und Installations-Ablauf und macht sogar Crosscompiling möglich.

*Tools+Sprachen: C/C++, GNU make, GCC, qmail, Subversion, LaTeX*

## 14.6 Portierung von PostgreSQL auf Collax Linux Appliance

Der Kunde (*anonym*) ist Collax-Partner und betreut mittlerweile eine Vielzahl von Kunden mit Collax-Servern (*einer Linux-basierten Business-Server Appliance mit einfach bedienbarem Web-Interface*).

Trotz der Vielzahl der Funktionen mangelt es dem Collax noch an einem Relationalem Datenbank-System. PostgreSQL bietet sich hier aufgrund des hohem Funktionsumfangs und guter Performance an.

Das PostgreSQL-Paket für Collax habe ich mit meinem selbst entwickelten **Briegel-Builder** und einem Crosscompiler erzeugt und Patches für das saubere Crosscompiling geliefert.

*Tools+Sprachen: C/C++, Perl, Javascript, Crosstool, Briegel, PostgreSQL, GNU make, GNU autotools, Subversion*

## 14.7 Automatisierte Prüfung von Meßgeräten. (IVB Meßtechnik)

Der Kunde ist Sachverständiger für Kalibrierung von Prüfung von Druckmeßtechnik. Es wurde eine Softwarelösung für die Erfassung der Meßdaten, Erstellung von Prüfberichten sowie der Verwaltung von Kundendaten, Kundengeräten und Prüfmitteln entwickelt.

In enger Zusammenarbeit wird das Produkt nun auch für andere Meßgerätearten (z.B. Längenmeßtechnik) ausgebaut und in Kürze als Branchenlösung auf dem Markt angeboten.

*Tools+Sprachen: PHP, PostgreSQL, Apache, LaTeX, Shellscript, LAMP*

## 14.8 IT-Outsourcing für Ärzteverbände

Die beiden Ärzteverbände "Deutsches Forum für Psychotherapie" (DPI) und die "Vereinigung psychotherapeutisch tätiger Kassenärzte" (VPK) haben metux IT service ihre komplette EDV überantwortet.

Neben alltäglicher Wartung von EDV-Anlagen und redaktionellen Diensten zählt hierzu eine ASP-Lösung (Anwendungsdienst), mit der die Verbände ihre Mitglieder verwalten, Beiträge berechnen und sich in zahlreichen geschlossenen und offenen Mailinglisten austauschen können.

Derzeit wird zudem eine Lösung erarbeitet, mit der sie Verbände ihre Rundschreiben via Web erstellen, in Druckereiqualität automatisch setzen und anschließend sowohl als Newsletter ebenso wie als Briefpost verteilen lassen können.

*Tools+Spachen: PHP, Perl, PostgreSQL, Javascript, LaTeX, Majordomo, Java, CVS, Shellscript, LAMP*

## 14.9 Webcrawler / Scanner und Mailgate für Web-Foren (Alphacron)

Bei seiner täglichen Arbeit benötigt der Kunde stets aktuelle Informationen aus vielerlei Web-Foren, Newsportalen, udgl. Als Kommunikationsmedium bevorzugt der Kunde E-Mail.

So lautet hier die Aufgabenstellung, Scanner-Anwendungen zu programmieren, die verschiedene Portale und Foren regelmäßig abfragen, Information extrahieren und die **neuen** Nachrichten per Email weiterleiten. Für die Foren gibt es zudem umgekehrt auch einen Email-Roboter, der empfangene Mails wieder ins Forum schreibt - so erhält man den Komfort einer Mailingliste bzw. Newsgroup.

*Tools+Spachen: C/C++, PHP, PostgreSQL, Apache HTTPd, CVS, GNU make, GCC, Shellscript, LAMP*

## 14.10 Entwicklung von Monitoring-Utills für sendmail (IGEPA ITS)

Es wurde eine kundenspezifische Software entwickelt, die Logdaten des Sendmail-Mailserver auswertete, Berichte erstellte und den aktuellen Mailverkehr im Web sowie im Terminal darstellt.

*Tools+Spachen: Perl, PHP, ncurses, Sendmail, Apache, PostgreSQL, LaTeX, Shellscript, LAMP*

## 14.11 Netzwerk-Optimierung und Dokumentation (Thuecob AG)

Die Projektaufgabe bestand darin, existierende Netzwerkinfrastruktur und Adreßzuteilungen zu erfassen, zu optimieren und anschließend für die Servicetechniker zu dokumentieren. Im Anschluß wurden auch Wartungspläne erstellt.

*Tools+Spachen: LaTeX, PostgreSQL, PHP, X.75, X.21, Radius, SNMP, Nagios, LAMP, Apache-HTTPd*

## 14.12 Unix-Sicherheitstools (TBGS GmbH)

Entwickelt wurden einige Sicherheitstools, u.A. dem Produkt "su-wrapper" welches fein granuliert Zugriffssteuerung von Befehlsausführungen einzelner Benutzer unter anderen Benutzer (z.B. root) gestattet.

Dieses Tool wird beispielsweise eingesetzt, um unprivilegierten Benutzern gezielt Aktionen zu gestatten die betriebssystembedingt nur dem Administrator (root) erlaubt sind (z.B. öffnen und schließen von Netzwerkverbindungen, etc).

*Tools+Spachen: C/C++, GNU make, GCC, CVS, LaTeX, Shellscript*

## 14.13 Datenbank-Konzeption und RADIUS für DSL-Provider (CATS GmbH)

Der Auftraggeber benötigte eine Kundendatenbank für seinen DSL-Vertrieb. Sämtliche Aktionen wurden auch per Web zugänglich gemacht, sodaß der Kunde seine Bestellung, Tarifwechsel, Kontrolle und Pflege der Stammdaten bequem per Web erledigen kann.

Aus der Datenbank heraus wird auch der Radius-Server betrieben.

*Tools+Spachen: C/C++, PHP, Perl, PostgreSQL, plsql, CVS, Radius, X.75, DSL, IP tunneling, Shellscript, LaTeX, LAMP*

#### 14.14 fXignal - Echtzeit-Börsenhandelsplattform im Devisenmarkt (FOREX)

Der Kunde betreibt eine Plattform für den automatisierten Echtzeithandel am Devisenmarkt (FOREX) mit variablen Hebeln.

Hierzu bündelt die fXignal-Plattform Handelssignale aus verschiedensten Quellen - eigene Tradestations, externe Analysten oder manuelle Eingabe - und gruppiert diese nach verschiedenen Handelsstrategien und Währungs-paaren in separate Kanäle. Der Nutzer kann diese Kanäle mit diversen Parametern (Limits, Lot, etc) auf seine Broker-Konten leiten und so die fXignal-Plattform vollautomatisch handeln lassen.

Die Plattform war zunächst historisch aus dem eigenen Handelsgeschäft des Kunden gewachsen und wurde später für weitere Nutzer geöffnet. Mit dem weiteren Wachstum hatte sich diese gewachsene Struktur aus zahlreichen als Windows- und GNU/Linux-Systemem mit verschiedenen inkompatiblen Anwendungen als wenig zukunftsträchtig herausgestellt und mußte im laufenden Betrieb - unterbrechungsfrei - neu strukturiert werden. Nach Zulieferung einiger Software-Komponenten und verschiedener Operating-Tasks übernahm ich mit Ende 2004 hier die technische Leitung von der Weiterentwicklung und Neustrukturierung der Plattform bishin zum Operating des Rechenzentrums. Wichtige Meilensteine waren hier:

- Restrukturierung und Optimierung der Datenbank-Infrastrukturen im laufenden Betrieb, um den Echtzeit- und Stabilitäts-Anforderungen gerecht zu werden
- Ablösung der MS-SQL-Server und weiterer damit verbundener Windows-Systeme durch PostgreSQL auf einer eigens entwickelten, optimierten GNU/Linux-Appliance (\*1)
- Ablösung des Windows-/IIS-Webservers durch Apache HTTPd auf einer eigens entwickelten, optimierten GNU/Linux-Appliance (\*1)
- Ablösung der Windows-/Exchange-Mailserver durch Sendmail und diverser Web-Frontends auf LAMP-Basis, auf einer eigens entwickelten, optimierten GNU/Linux-Appliance (\*1)
- Ablösung verschiedenster Delphi-basierter Dienste, teils durch Stored-Procedures, teils durch Java- und C/C++-basierte Daemons, um den Echtzeit-Anforderungen gerecht zu werden
- Implementation eines echtzeitfähigen Reportings
- Ablösung des C#-basierten (Windows-)Clients durch eine neu entwickelte Nutzerplattform auf PHP-Basis
- Implementation eines mobilen Web-Frontends
- Ablösung der Windows-basierten Orderstations (Broker-Ansteuerung) durch durch einen GNU/Linux-basierten Order-Robot
- Anbindung weiterer Signalquellen und Broker (Refco, FXCM, etc)
- Virtualisierung der verbleibenden Windows-Systeme (Tradestations) für einfacheres Management und Kostenreduktion
- Schrittweise Implementation eines Hot-Standby für kritische Services
- Restrukturierung der RZ-Verkabelung, Redundante Anbindung und Core-Router (CISCO+GNU/Linux+Lucent-Mischunggebung, AS/BGP via Zebra)
- Restrukturierung der VPN-Infrastruktur

*Tools+Spachen: C/C++, PostgreSQL, MS-SQL, IIS, Apache, Zebra, PHP, Java, CISCO, Tradestation, LAMP, Sendmail*

## 15 Eigene Projekte

### 15.1 Briegel Builder - vollautomatisiertes Buildsystem für Firmware und Distros

Als Embedded-Entwickler ist man oft mit dem Problem konfrontiert, fein kofektionierte Pakete und Firmware-Images für die unterschiedlichsten Zielsysteme zu erzeugen und dabei Seiteneffekte (zB. Verwicklungen mit dem Entwicklungssystem) auszuschließen.

Viele Entwickler benutzen dazu jeweils eigene Scripte für jedes Zielsystem. Das bringt aber schnell einen erheblichen Pflegeaufwand mit sich, besonders wenn sich Anforderungen/Spezifikationen öfters ändern.

Mit dem **Briegel Builder** wird dem Entwickler dieser Aufwand praktisch vollständig aus der Hand. Der Entwickler mußjedelich jedes Zielsystem einmalig konfigurieren (zB. Crosscompiling Toolchain erzeugen) und für jedes gewünschte Paket Version und Features wählen. Der restliche Prozeßwird von Briegel vollautomatisch übernommen, einschließlich Download, Patching und Abhängigkeiten bishin zur Paketierung.

Nicht alle Pakete sind im Original für feine Konfektionierung und sauberes Crosscompiling geeignet. Hierfür werden bereits fertige Patches mitgeliefert.

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

### 15.2 OpenSource Software QM Taskforce

Das OSS-QM Projekt hat sich zum Ziel gestellt, unabhängig von einzelnen Distributionen, Patches für zahlreiche Opensource-Pakete bereitzustellen. Diese umfassen einerseits rasche Bugfixes, andererseits Erweiterungen zur besseren Anpassung an individuelle Anforderungen (zB. in Embedded-Umgebungen).

Aktuell pflegen die meisten Distros ihre eigenen Patches. Allzu oft sind diese jedoch nicht allgemeintauglich und könnn/werden nicht direkt in das Originalprojekt einfließen.

Im Zuge meiner vielfältigen Entwicklungsarbeit im Embedded-Umfeld mußten immer wieder Pakete angepaßt bzw. repariert werden. Meist sind es kleine oder eher formelle Dinge, zB. sauberes Crosscompiling aus einer Sysroot-Umgebung, Installation in ein Prefix, saubere Imports, PkgConfig-Unterstützung, etc.

Um längerfristige Doppelarbeiten zu vermeiden habe ich die Anpassungen derart allgemein ausgelegt, daß sie für die meisten Distros direkt taugen und unmittelbar in die Originalprojekte einfließen können.

Die entstandene Patch-Repository ist so organisiert, daß für jede (unterstützte) Version der jeweiligen Pakete ein Patch automatisiert heruntergeladen und eingespielt werden kann. Man könnte sich also das OSS-QM-Projekt als eine Art Filter vorstellen, der theoretisch für jedes (unterstützte) Paket in allen wichtigen Versionen eine unmittelbar reparierte Version bereitstellt, die sich aus dem Originalpaket plus dem zugehörigen OSS-QM-Patch zusammensetzt.

Das OSS-QM-Projekt wird von meinem Distro/Firmware-Toolkit Briegel unmittelbar benutzt, sodaß dort keinerlei eigene Patches mehr vorkommen.

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

### 15.3 Comprehensive Source Database

In der CSDB werden OpenSource-Pakete mit ihren Versionen und den dazugehörigen Download-URLs (der Quellpakete) erfaßt. Die Datenbank arbeitet mit streng normalisierten Versionsnummern und läßt sich automatisiert abfragen.

Somit ist eine Integration der CSDB in automatische Build-Systeme, die Quellpakete herunterladen müssen, einfach und unkompliziert möglich.

Der Datenbestand wird durch automatische Crawler, die uA. Websites und FTP-Server zahlreicher Opensource-Projekte absuchen, regelmäßig aktuell gehalten.

*Tools+Spachen: C, PHP, PostgreSQL, plsql, Perl, CVS, Shellscript*

### 15.4 OpenVZ Virtualisierung für Collax Business Server

Collax Business Server ist eine renommierte Linux-Appliance für Office-Server in der die KMU. Die Server-Appliance ist vollständig per Web bedienbar und kann für typische Office-Anwendungen einen Windows-Server

ersetzen.

Aktuell mangelt es jedoch noch an der Unterstützung für virtuelle Server-Umgebungen. Diese wird mit dem Projekt auf Basis von OpenVZ nachgeliefert.

Die Collax-Appliance wird damit in die Lage versetzt, praktisch beliebig viele virtuelle Linux-Server zu beheimaten.

*Tools+Spachen: C, GNU make, Linux Kernel, CVS, Subversion, Javascript, Shellscript, Perl, Crosstool*

## 15.5 Linux-Kernel: Text-basierte Kernel-Schnittstellen via Dateisystem

Die Interfaces zwischen Linux-Kernel und Userland sind derzeit meist binär und damit stark von der CPU-Architektur abhängig. So benutzt zB. **netfilter** ein auf netlink-Socket-basiertes Protokoll, das sowohl von Kernel als auch Userland codiert/decodiert werden muß.

Entwicklungen auf anderen Betriebssystemen, insbesondere Plan9, haben deutlich gezeigt, daß sich textbasierte Dateisysteme hervorragend als universelle Schnittstelle zwischen verschiedenen Prozessen und Kernel eignen. Der Datentransport wird erheblich vereinfacht, der Code viel schlanker und die Schnittstellen transparenter. Unter Plan9 ist praktisch jede Resource, bishin zu Netzwerk-Schnittstellen und CPUs netzwerktransparent via 9P-Dateisystem erreichbar.

Ziel des Projektes ist es, wichtige Kernel-Schnittstellen (zB. netfilter, acpi, ipmi,...) via synthetische Dateisysteme neu zu implementieren und damit die alten Binärschnittstellen abzulösen. Probleme mit Binär-Inkompatibilitäten gehören dann der Vergangenheit an, und die Schnittstellen sind über ein passendes Netzwerk-Dateisystem direkt Netzwerk-transparent.

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

## 15.6 AuctionWatch - Auktionsuchmaschine

Mit der Auktionsuchmaschine **auctionwatch.metux.de** erhält der Schnäppchenjäger ein einfach zu bedienendes Werkzeug zum regelmäßigen Stöbern nach interessanten Auktionen.

Beliebig viele Suchabfragen können definiert werden, die von der Suchmaschine regelmäßig abgerufen werden. Der Nutzer bekommt die jeweils neu gefundenen Artikel eingeliefert und kann sie ggf. sortieren lassen (zB. zu teure Artikel anhand eines Maximalpreises aussortieren). Einmal gelöschte Artikel erscheinen auch nicht nach erneuter Suche wieder.

Zum derzeitigen Stand kann nur auf eBay gesucht werden - Anbindungen an andere Auktionshäuser sind derzeit in Arbeit.

*Tools+Spachen: C/C++, PHP, Java, Javascript, PostgreSQL, Apache, Subversion*