

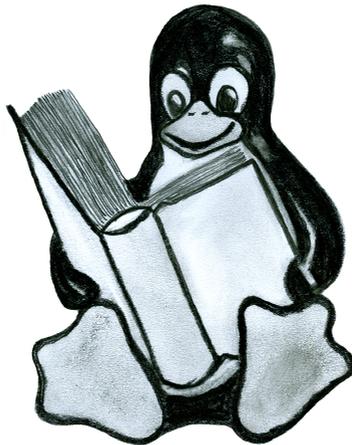
**Team der Chemnitzer Linux-Tage:  
Chemnitzer Linux-Tage 2014  
– Tagungsband –  
15. und 16. März 2014**



Team der Chemnitzer Linux-Tage

# Chemnitzer Linux-Tage 2014

15. und 16. März 2014



– Tagungsband –



TECHNISCHE UNIVERSITÄT  
CHEMNITZ

Universitätsverlag Chemnitz  
2014

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Technische Universität Chemnitz/Universitätsbibliothek  
Universitätsverlag Chemnitz

Herstellung und Auslieferung:  
Verlagshaus Monsenstein und Vannerdat OHG  
Am Hawerkamp 31  
48155 Münster  
<http://www.mv-verlag.de>

ISBN 978-3-944640-08-2

URL: <http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-133499>  
URN: <urn:nbn:de:bsz:ch1-qucosa-133499>

Satz und Layout: Jens Pönisch und Monique Kosler  
Titelgraphik: Wikimedia Commons (User Jszigetvari), Jens Pönisch (CC-BY-SA 3.0)  
Graphik Innentitel: Petra Pönisch  
URL der Chemnitzer Linux-Tage: <http://chemnitzer.linux-tage.de>

## Premiumsponsoren



## Weitere Sponsoren



## Medienpartner





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b>	<b>11</b>
<b>2</b>	<b>Inhalt der Hauptvorträge</b>	<b>13</b>
2.1	Adventures with Linux in the Caribbean (Banking etc.) . . . . .	13
2.2	Festplattenverschlüsselung mit Yubikey und LinOTP . . . . .	19
2.3	Grüner verschlüsseln – Energieverbrauch dieser Algorithmen . . . . .	25
2.4	Heutige Möglichkeiten von Prozessoren in eingebetteten Linux-Systemen	33
2.5	Informationelle Selbstbestimmung und Informationsfreiheit aus Sicht von Recht und Ökonomik . . . . .	41
2.6	Open Source Enterprise Resource Planning und der Mittelstand . . . . .	51
2.7	Markdown und eine Prise pandoc . . . . .	61
2.8	Mikrocontroller stromsparend programmieren . . . . .	81
2.9	Projektautomatisierung am Beispiel von µracoli . . . . .	91
2.10	Sicheres MultiSeat . . . . .	99
2.11	Vertrauen in Spracherkennung ...? . . . . .	107
2.12	Vertraulichkeit in (kleinen) Unternehmen – (k)eine Frage der IT? . . . .	117
2.13	X2Go – Einsatzmöglichkeiten für Privatanwender . . . . .	123
<b>3</b>	<b>Zusammenfassungen der weiteren Vorträge</b>	<b>131</b>
3.1	20 Dinge über Verschlüsselung, die Sie schon immer wissen (w   s)ollten	131
3.2	Aktuelle Entwicklungen beim Linux-Kernel . . . . .	131
3.3	AMaViS und Kaspersky vs. Malware . . . . .	131
3.4	Amazon Linux – Betriebssystem für die Cloud . . . . .	132
3.5	Anwendung, Implementierung und Sicherheit von Kryptografie zur Datei- und Sprachverschlüsselung . . . . .	132
3.6	Automatisierte Systemkonfiguration mit Capistrano und Puppet . . . .	132
3.7	B.A.T.M.A.N. Beginners: WLAN-Meshing für Einsteiger . . . . .	132
3.8	Best of Geany-FAQ – alles was ihr schon immer über den Editor wissen wolltet . . . . .	133
3.9	Ceph und Gluster im Vergleich . . . . .	133
3.10	CloudStack – Aufbau und Struktur . . . . .	133
3.11	Das Debian-LAN Projekt: Installation eines Debian-Netzwerks einfach gemacht . . . . .	133
3.12	Data Leakage Protection: Zukünftige Herausforderungen zur Sicherung von Vertraulichkeit . . . . .	134
3.13	Datenintegration im Service – mit OTRS die passenden Daten für den Service-Prozess liefern! . . . . .	134
3.14	Den Schlapphüten die Ohren verstopfen – Transportverschlüsselung für alle . . . . .	134

3.15	Der Linux-Multimediastack . . . . .	134
3.16	Deutschlands Sicherheitspolitik im Cyberspace . . . . .	135
3.17	Die «Deutsche Wolke»: Open Source Cloud für den Mittelstand . . . . .	135
3.18	Die private Cloud mit ownCloud . . . . .	135
3.19	Die Technik des elektronischen Personalausweises . . . . .	136
3.20	E-Mail Made in Germany – steckt da was dahinter? . . . . .	136
3.21	Effiziente Kommunikation und Arbeit im IT-Team . . . . .	136
3.22	Einführung in SSL mit Wireshark . . . . .	136
3.23	Entfernt einloggen – Grundlagen der SSH-Nutzung . . . . .	137
3.24	Geotagging: Fotos mit Geoinformationen verknüpfen . . . . .	137
3.25	Graylog 2 – Log-Management einfach gemacht . . . . .	137
3.26	Hier geht nix rein! Storage Performance im Virtualisierungsumfeld . . . . .	138
3.27	High Availability und Disaster Recovery: Metro Storage Cluster mit ZFS . . . . .	138
3.28	I got root – I can read your mail . . . . .	138
3.29	Icinga 2 – Secure Cluster Stack Monitoring and More . . . . .	138
3.30	Installation und Arbeiten mit einer (La)T <sub>E</sub> X-Distribution unter Linux . . . . .	139
3.31	Introduction to Software Collections . . . . .	139
3.32	Keine Angst vor den Befehlen – die Welt der Linux-Kommandozeile . . . . .	139
3.33	Kerberos – sichere Authentifizierung seit 30 Jahren . . . . .	139
3.34	KMS UXA DRM OMG WTF BBQ – Durchblick im Linux-Grafikdschungel . . . . .	140
3.35	KryptoRide – Kryptographie zum Mitmachen . . . . .	140
3.36	Kryptoschlüssel, Zertifikate und Smartcards in der Praxis . . . . .	140
3.37	Linux im Automotive-Umfeld – wie baue ich mir mein eigenes Fahrerassistenzsystem . . . . .	140
3.38	Linux-Booten leicht gemacht: der Barebox Bootloader . . . . .	141
3.39	Linux-Dienstleister stellen sich vor (Business-Forum) . . . . .	141
3.40	LPI-Zertifizierung, aber wie? . . . . .	141
3.41	Medienalphabetismus – heilbar? . . . . .	141
3.42	Methoden zur Gewinnung neuer Teammitglieder . . . . .	142
3.43	Nachrichtenschlüsselung im Alltag . . . . .	142
3.44	NeDi – Network Discovery that Really Works . . . . .	142
3.45	nftables – der neue Paketfilter im Linux-Kernel . . . . .	142
3.46	Open Source in der brasilianischen Regierung . . . . .	142
3.47	Open-Source-Lizenzen in der kommerziellen Praxis . . . . .	143
3.48	PDF-KungFoo mit Ghostscript & Co. . . . .	143
3.49	Perfekte Silbentrennung in E-Books mit präreformatrischen Texten . . . . .	143
3.50	Pond – E-Mail sicher und vertraulich . . . . .	143
3.51	PostgreSQL: Killing NoSQL . . . . .	144
3.52	PREEMPT-RT – More than just a kernel . . . . .	144
3.53	Quelle: Internet? Das können wir besser! – Mit Metadaten Ordnung ins Chaos bringen . . . . .	144
3.54	Samba 4 und OpenLDAP als Home Server mit UCS . . . . .	144
3.55	SCSI EH and the real world . . . . .	145
3.56	SELinux: Bitte nicht deaktivieren . . . . .	145
3.57	Shell lernen und günstig tanken . . . . .	145

3.58	Sichere entfernte Rechnernutzung und Dateitransfer . . . . .	145
3.59	Sichere Netze mit OpenVPN . . . . .	146
3.60	Sicheres Anwendungsmonitoring mit SNMP . . . . .	146
3.61	Statische Codeanalyse – wo ist der Fehler in meinem Programm? . . .	146
3.62	Systemmanagement mit Puppet und Foreman . . . . .	146
3.63	Thin Clients von morgen, booten via WLAN . . . . .	147
3.64	truecrypt.sh: Deniable File System with bash . . . . .	147
3.65	Tux im Passivhaus – Klimaschutz und Smarthome mit Freier Software	147
3.66	Vertrauen ist gut, Kontrolle ist besser: 7 Aspekte der Vertrauenswürdigkeit von freien Office-Suiten . . . . .	147
3.67	Vollautomatische Betriebssystemtests mit openQA . . . . .	148
3.68	Vom Aussterben bedroht: die Universalmaschine Computer . . . . .	148
3.69	Wald und Bäume – Log-Analyse für Serverparks . . . . .	148
3.70	Wanderreise mit OpenStreetMap . . . . .	149
3.71	Warum Kinder eine Open-Source-Community brauchen . . . . .	149
3.72	Was kommt nach SysVinit? . . . . .	149
3.73	WebODF – gemeinsame Dokumentenbearbeitung in der eigenen Website	149
3.74	Wenn Geeks Langeweile haben – reloaded . . . . .	150
3.75	Wie kann man Zertifikate von CAcert verwenden . . . . .	150
3.76	Wie wir einmal 500 Server mit 150 Personen in 3 Tagen migriert haben und was wir alles gelernt haben . . . . .	150
3.77	Zur eigenen Linux-Distribution in 30 Minuten . . . . .	150
3.78	Zur Geschichte der Verschlüsselung: Von der Kopfrasur zur Kopfkrobatik . . . . .	151
<b>4</b>	<b>Zusammenfassungen der weiteren Workshops</b>	<b>153</b>
4.1	darktable – die digitale Dunkelkammer . . . . .	153
4.2	Die Schale um den Kern – Einstieg in die Bash und den GNU-Werkzeugkasten . . . . .	153
4.3	Django: Schnell performante Web-Applikationen entwickeln . . . . .	153
4.4	Einführung in die 3D-Visualisierung mit Blender . . . . .	154
4.5	Einführung in Python . . . . .	154
4.6	Elektronikbasteln für Kinder . . . . .	154
4.7	Hardware-Workshop . . . . .	154
4.8	KDE/Kubuntu-Grundeinstellungen . . . . .	154
4.9	Kreatives Programmieren mit Processing . . . . .	155
4.10	Open Knowledge: Dein Wissen als interaktiver Online-Kurs . . . . .	155
4.11	openATTIC – offenes Storage Management . . . . .	155
4.12	PyMove3D – Vorbereitung zum Programmierwettbewerb . . . . .	155
4.13	Raspberry Pi zum Anfassen . . . . .	156
4.14	SSL-gesicherte Web-Seiten – was ist da wie «sicher»? . . . . .	156
<b>5</b>	<b>Personen</b>	<b>157</b>



## 1 Vorwort

Vertrauen ist ... gut, werden manche unser Motto gedanklich weiterführen und damit den Spruch zitieren, der – möglicherweise zu Unrecht – Lenin zugeschrieben wird. Kontrolle ist besser, heißt es weiter und drückt damit letztendlich fehlendes Vertrauen aus.

Wir vertrauen immer und überall und natürlich auch im Kontext der Linux-Tage. Wir vertrauen auf richtig rechnende Prozessoren – auch wenn sich der Pentium-Bug in diesem Jahr zum 20. Mal jährt. Wir vertrauen auf unsere Software oder zumindest darauf, dass genug Menschen kontrollieren, ob das Vertrauen auch gerechtfertigt ist. Menschen, denen wir wiederum ... vertrauen müssen.

In den letzten Monaten wurde unser Vertrauen mehrfach erschüttert. Der zweite Teil des Spruchs gewann damit an Bedeutung: Kontrolle ist besser. Leicht gesagt und schwer getan: Kontrolle erfordert Wissen über das zu Kontrollierende. Wer kann von sich sagen, dass er wirklich verstanden hat, wieso https «sicher» ist? Oder zumindest sein soll? Wer kann beurteilen, ob die Implementierung eines Algorithmus keine Fallen enthält?

Ohne Wissen sind weder Vertrauen noch Kontrolle möglich. Das ist für uns ein Grund, die Linux-Tage zum nunmehr 16. Mal zu veranstalten: Wissen vermitteln. Aber wir freuen uns auch über die zahlreichen Begegnungen und neuen Kontakte. Eine Atmosphäre zu schaffen, in der Neugier und Offenheit dominieren, ist unser Ziel. Dass das gelingt, verdanken wir vielen, vielen Referenten, Standbetreuern, Sponsoren, Helfern, Unterstützern und Besuchern.

Tatsächlich belegt ist von Lenin übrigens lediglich die Aussage «Vertraue, aber prüfe nach.» Das klingt schon sympathischer. Sie kann gut als Einleitung zu diesem Tagungsband mit seinen vielfältigen Beiträgen stehen.

Ralph Sontag im Februar 2014



# **Adventures with Linux in the Caribbean (Banking etc.)**

Mark Courtenay

## **1 Abschnitt**

Much has been achieved by the Linux development community to provide components which enable the integration of Linux into “hostile” environments, most notably those dominated by Microsoft Windows. The teams developing these components recognise that the key to acceptance of Linux in such environments is not the “big bang” replacement of Windows on the desktop or even in the server-room, but adding value in areas where Linux has particular strengths.

Linux's key strengths in such a heterogeneous environment include a open, standardised set of scripting languages for use in a batch or Web-based framework, the latter using the well-supported, powerful and flexible Apache web server. MySQL was used as the underlying database, providing robustness and power. Code examples and modifiable source components (for example to automate the sending of e-mail) are readily available in the Open Source tradition. This framework lends itself well to rapid development of the automation of tasks to be achieved in the business environment.

On assuming responsibility for the IT environment of a Caribbean bank, the author was faced with a proprietary banking system “RIBS” running on an IBM AS/400

mid-range system, with the desktop and server environment provided by Microsoft Windows NT4.0. An IBM RS/6000 AIX system was responsible for the processing and logging of ATM transactions and was the only Unix-derived platform. It soon became clear that there were many efficiency improvements possible if only the power of Linux could be introduced in this environment, but there was initially strong management opposition to this possibility.

Initial developments to facilitate acceptance were to use SAMBA networking to automate the extraction of statement information from a Windows based file server. Proprietary database connectivity on the RS/6000, in due course replaced by ODBC (Open DataBase Connectivity), was used to access information from the ATM transaction log to monitor the health of the ATM system. These improvements led to the acceptance of the concept of Linux coexisting with other systems at the bank, because it was inconceivable that such functionality could be implemented with the existing systems. The system was further developed to facilitate the accounting of credit card merchant transactions, saving a few hours a day effort by a key member of staff, allowing him to focus on customer service. A later development enabled the automatic secure e-mailing of credit card statements to customers in soft form, as had been implemented for normal account statements earlier.

The key "RIBS" banking database remained a challenge to integrate with the new Linux, but a straightforward nightly export of key tables proved remarkably effective for mirroring that database in Linux. In addition, an ODBC component for the AS/400 was also implemented. This led to the new Linux capability being called "NotRIBS", because: "It's not RIBS"! The mirrored database had numerous uses, a key one being to facilitate building a new customer database to conform to current banking regulations which would be implemented by the introduction of a new

banking system (Temenos T24).

To support the seamless integration of the Linux-based Apache Web server into the new framework, it was deemed a high priority to support Windows-based browser authentication (NTLM). This is a component developed within SAMBA and its implementation is critical to avoid the need for a separate user database to that provided by Windows Active Directory. Once this capability is in place, access lists can be readily built with Active Directory ids to control access to privileged functions with requiring the user to remember a separate id/password or consciously execute a new login action. This capability is also known as "Single Sign On" (SSO).

The „NotRIBS“ system was so successful it was adopted by management as a basic Intranet server platform, even though it was never particularly intended as such. The original Intranet platform had been envisaged to be Microsoft Sharepoint, but this initiative had insufficient resource and/or skills devoted to it to be successful. To satisfy this need, some work was done with Drupal to assess its suitability as a fully fledged Content Management System framework to incorporate all functionality so far developed. An advantage of Drupal is that it offers some support for an external authentication mechanism like NTLM.

A wide range of reporting capabilities were implemented, including analyses of loans performance and reports to facilitate Anti-Money Laundering (AML). Specific reports required on a monthly basis for regulatory purposes were improved upon, for example a report which required about „3 days worth of effort sifting through about 30 sheets of paper“ was replaced by an automatically generated report providing the exact information required. An archiving mechanism was

implemented, allowing retrospective reports to be generated „as at“ the end of a particular month in the past.

The Linux distribution used was CentOS, a free version of RedHat Enterprise Linux, a very sold and stable platform suitable for banking. The Red Hat commercial version was chosen by the bank as the underlying operating system for the T24 banking system, to which all jurisdictions would in due course be migrated. CentOS is also a popular distribution for the Asterisk Open Source PBX, which formed the basis of initial implementations of the „NotRIBS“ system. A number of developments specifically related to Asterisk were also achieved as a „side-project“.

Initially a basic Asterisk PBX configuration was built alongside the Nortel PBX serving the main location of the bank, with about 60 extensions. Through an analog „bridge“ adapter it was possible for calls to the IT department to be transferred to a number of different departments. This system evolved to providing the „Virtual Internet Phone Service“, a trial service offering low-cost Internet telephony to international destinations such as North America and Europe. A key feature of the system was the use of voice menus to ease the user experience for commonly called numbers, i.e. avoiding complicated international dialling.

In summary, it was demonstrated that the introduction of Linux into a „hostile“ environment is possible when the best use is made of the components which exist to provide integration. Some of these components are still a challenge to install, configure and get working and the related developments should be supported as much as possible to provide examples and documentation to overcome these

challenges. In particular the importance of these integrations needs to be wider recognised within the Linux community as a key enabler to introducing Linux into traditional business environments. There is still a lot of suspicion of Linux in traditional IT departments as being experimental in nature and this can be overcome by focussing on functionality, robustness and reliability rather than incorporating the latest features.



# Festplattenverschlüsselung mit Yubikey - verwaltet mit LinOTP

Cornelius Kölbel

cornelius.koelbel@lsexperts.de

<http://www.linotp.org>

Spätestens seit Snowdens Enthüllungen sollte es jedem klar sein, dass man sich über den Schutz seiner Daten Gedanken machen muss. Verschlüsselung ist hier das Mittel der Wahl. Doch auch ein starker Verschlüsselungsalgorithmus ist nicht stärker als ein schwaches Passwort. Deswegen betrachten wir hier die Möglichkeit, die Verschlüsselung mit einer Zweifaktor-Authentisierung zu schützen. Die mit LUKS verschlüsselte Festplatte wird erst mit der Eingabe eines starken Passwortes und dem Besitz eines Yubikeys aufgeschlossen. Die Verwaltung dieser Yubikeys erfolgt mit LinOTP.

## 1 Zweifaktor-Authentisierung

Bei der Zweifaktor-Authentisierung wird zusätzlich zum Faktor *Wissen* des Passwortes der Faktor *Besitz* eingeführt. Dieser Faktor ist i.d.R. ein Stück Hardware, das nicht kopierbar sein sollte. Mit der Zeit wurde ein solcher Besitz-Faktor auch als Software ausgeführt, doch erfolgreiche Angriffe, die das Kopieren eines Software-Besitzes durchführen<sup>1</sup>, zeigen, dass der Besitz in Form eines schwerer zu kopierenden Stück Hardware ausgeführt sein sollte.

Im strengen Sinne scheidet also bspw. ein USB-Stick, ein PC oder ein Smartphone auf dem sich ein Schlüssel befindet aus, da der Schlüssel von solchen Geräten kopiert und auf andere Geräte aufgebracht werden kann.

Der klassische Besitz-Faktor ist eine Smartcard, die einen privaten Schlüssel enthält, auf den nicht direkt zugegriffen werden kann, sondern dessen Nutzung durch das Kartenbetriebssystem geregelt ist. Der Löwenanteil des mäßigen Wachstums des Smartcard-Marktes wird den Mobilelefonen (SIM) und Finanzsektor (Kreditkarten) zugeordnet [3]. Sie erfordern Treiber, oft den Aufbau einer PKI und die Erneuerung der Laufzeit-begrenzten Zertifikate. Diese Komplexität hat viele Anwender vor dem Einsatz zurückschrecken lassen und zu einer Renaissance anderer Hardware-Token geführt, die basierend auf einem symmetrischen, geheimen Schlüssel nicht vorher-sagbare Einmalpassörter erzeugen<sup>2</sup>.

<sup>1</sup><http://arstechnica.com/security/2012/05/rsa-securid-software-token-cloning-attack/>

<sup>2</sup>Der erste Einmalpasswort-Token wurde bereits 1986 von RSA mit einem proprietären Algorithmus auf den Markt gebracht. [1]

## 2 Festplattenverschlüsselung mit LUKS

LUKS (Linux Unified Key Setup)<sup>3</sup> ist eine Erweiterung des Device Mappers dm-crypt und in allen gängigen Linuxdistributionen als Verschlüsselungslösung der Festplatte vertreten. LUKS arbeitet mit Keyslots, die vereinfacht gesagt den mit einem Passwort verschlüsselten Verschlüsselungskey (DEK<sup>4</sup>) enthalten. Nach Eingabe des Passwortes kann der DEK entschlüsselt und damit auf die gesammte verschlüsselte Festplatte zugegriffen werden.

Die Sicherheit hängt hier also an dem Passwort, mit dem der Keyslot aufgeschlossen werden kann. Leider bietet LUKS heute noch keinen Plugin-Mechanismus, so dass die Herausforderung besteht, wie man einen zweiten Besitz-Faktor mit einem Slot koppeln kann.

## 3 Der Yubikey

Der Yubikey ist ein USB-Device, das zur Authentisierung dient. Er kann selber initialisiert werden und wird im Betrieb als Tastatur erkannt. In dieser klassischen Betriebsart sind keine Treiber für den Yubikey erforderlich. Die Funktionsweise von Einmalpasswörtern wurde bereits in [2] detailliert dargestellt. Der Yubikey unterstützt verschiedene Moden. Er kann Einmalpasswörter auf Basis des HOTP Algorithmus[8] berechnen oder eines OTP-Algorithmus, den Yubico selber entworfen und offengelegt hat. In beiden Moden wird ein symmetrischer, geheimer Schlüssel verwendet, der nicht-auslesbar auf dem Yubikey gespeichert ist<sup>5</sup>. Überlicherweise wird vom Yubikey der OTP-Wert erzeugt, indem der Knopf auf dem Yubikey gedrückt wird. Der Yubikey kann aber auch in einem Challenge-Response-Modus betrieben werden, der den HMAC-SHA1-Algorithmus[7] benutzt, der die Basis für HOTP darstellt. Dabei geht nicht wie bei HOTP der interne, inkrementierte Zähler in die Berechnung ein, sondern es wird über USB eine Challenge an den Token gesendet, die in den HMAC-SHA1-Algorithmus eingeht. Das Ergebnis wird nicht wie bei HOTP auf 6 Ziffern gekürzt sondern als 20 Byte lange Response zurückgesendet. Dies erfordert keine Interaktion mit dem Benutzer und wird in dieser Form auch in dem PAM-Modul libpam-yubico<sup>6</sup> verwendet.

Dieser Challenge-Response-Modus soll nun auch für die Festplattenverschlüsselung mit LUKS verwendet werden. Dabei spielen die folgenden Aspekte die entscheidende Rolle:

- **Secret Key** Der Secret Key befindet sich nur auf dem Yubikey und wird nicht auf dem zu verschlüsselnden Computer gespeichert.

---

<sup>3</sup><http://code.google.com/p/cryptsetup/>

<sup>4</sup>data encryption key

<sup>5</sup>Gegenüber der Speicherung eines Schlüssels in Software wird der Key hier als „nicht-auslesbar“ angesehen. Eine Sicherheitsbetrachtung wie in [4],[5], [6] findet hier nicht statt.

<sup>6</sup><https://github.com/Yubico/yubico-pam>

- **Wiederholbarkeit** Im Challenge-Response-Modus können Ergebnisse reproduziert werden. Eine gleiche Challenge erzeugt immer wieder den gleichen Response.

#### 4 Einbindung in LUKS

Yubico selber gibt allgemeine Empfehlungen zur Integration des Yubikeys in Festplattenverschlüsselungen[9].

Die Idee ist nun, dass in einem LUKS Keyslot nicht mehr das Passwort, sondern die Response auf eine Challenge enthalten ist. Die Challenge, die diesen Response erzeugt, liegt mit einem Passwort verschlüsselt auf dem Computer. In der Pre-Boot-Phase wird in den Distributions-üblichen GUIs ein Passwort abgefragt. Mit diesem Passwort wird nicht wie üblich direkt ein Keyslot aufgeschlüsselt, sondern die Challenge entschlüsselt. Die Challenge wird an den Yubikey gesendet und der Keyslot wird erst mit dem Response aufgeschlüsselt.

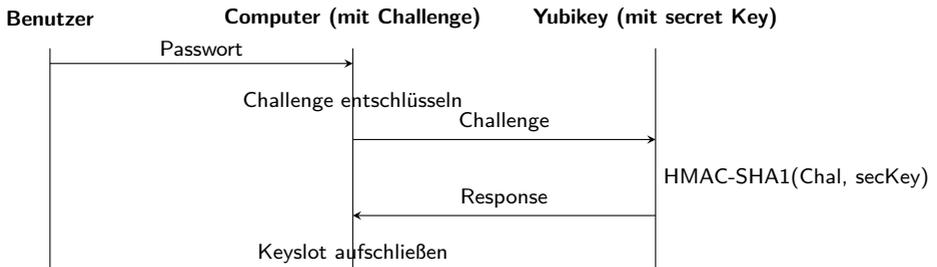


Abbildung 1: Bootvorgang mit Yubikey

Dieses Vorgehen könnte bei jedem Bootvorgang genauso wieder durchlaufen werden. Hierzu existieren bereits einige ähnliche Implementierungen [10] [11] für die Anpassungen der Initial Ramdisk, um eine solche Funktionalität beim Booten zu gewährleisten.

Um aber zu verhindern, dass mit der gleichbleibenden Challenge bzw. dem gleichbleibenden Response der Keyslot auch ohne Besitz des Yubikeys aufgeschlüsselt werden kann, sollte ein zweiter Schritt durchgeführt werden:

Es wird eine zweite zufällige Challenge  $chal_2$  erzeugt, diese wird wieder an den Yubikey gesendet und man erhält eine Response  $resp_2$  zurück.

$chal_2$  wird mit dem Passwort verschlüsselt und auf der Festplatte abgelegt und  $resp_2$  wird in den Keyslot geschrieben.

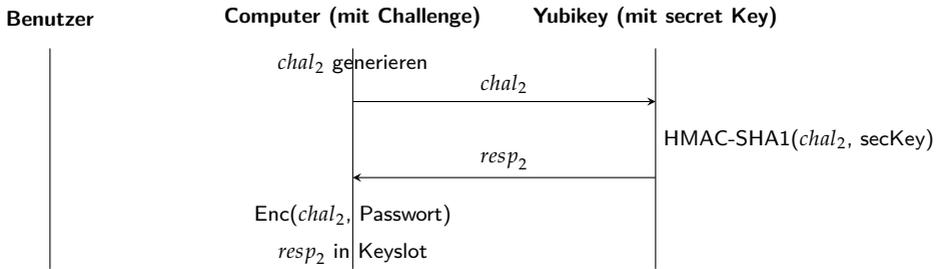


Abbildung 2: Neuschreiben der Challenge und des Keyslots

Somit wird beim nächsten Bootvorgang das gleiche Passwort aber ein anderer Wert vom Yubikey verwendet.

Der jeweils nächste Wert kann nur mit dem Besitz des Yubikeys erzeugt werden, da sich der geheime Schlüssel nur auf dem Yubikey befindet. Somit kann der Computer nur mit dem Passwort und dem Besitz des Yubikeys gestartet werden.

## 5 LinOTP

Um die umgebenden verwaltenden Arbeitsabläufe abzurunden, muss außerdem der Yubikey im Challenge-Response-Modus initialisiert werden und eine initiale Challenge und Response bereitgestellt werden. LinOTP unterstützt in der Version 2.6 bereits den Rollout von Yubikeys im Challenge-Response-Modus. Die modulare Struktur von LinOTP wurde bereits in [2] detailliert erklärt. Um ausgerollte Yubikeys einzelner Benutzer entsprechenden Computern zuzuordnen, kann LinOTP um ein Skript erweitert werden, das ein initiales Challenge-Response-Paar vom LinOTP-System erfragt und damit den Yubikey-Boot-Prozess auf einem entsprechenden Computer initialisiert. Somit kann auch ein Administrator, der selber keinen Zugriff auf den entsprechenden Yubikey hat, einem andere Benutzer das Booten mit dem Yubikey ermöglichen.

Listing 1: LinOTP script zum initialisieren der Yubikey LUKS Umgebung

```

linotp-luks-init -H https://linotpserver --admin=admin --
  serial=UBOM123456 --keyslot=3
linotp-luks-init -H https://linotpserver --admin=admin --
  user=mustermann --keyslot=4
  
```

In Listing 1 wird der Yubikey mit der Seriennummer UBOM123456 dem Keyslot 3 und der Yubikey des Benutzers *mustermann* dem Keyslot 4 zugeordnet.

Genau wie Benutzer bzw. Yubikeys einem Keyslot zugeordnet werden, müssen sie auch wieder entfernt werden können. LinOTP merkt sich hierzu die Zuordnung des Yubikeys anhand der Seriennummer zum Computernamen und zum Keyslot. Somit können zu einem späteren Zeitpunkt diese Informationen abgerufen werden (siehe Listing 2) oder auch einzelne Keyslots wieder gelöscht werden, um den Benutzern die Möglichkeit einen Computer zu booten, wieder zu entziehen (Listing 3).

Listing 2: Script zum Anzeigen der Keyslots

```
linotp-luks-show -H https://linotpserver --admin=admin
slot 3      user: musterfrau      serial: UBOM123456
slot 4      user: mustermann  serial: UBOM234567
```

Listing 3: Script zum Entfernen eines Yubikeys aus einem Keyslot

```
linotp-luks-remove -H https://linotpserver --admin=admin --
serial=UBOM123456
```

## 6 Sicherheitsbetrachtung

Ein USB Sniffer könnte beim Erzeugen der Response, wenn sie zum Verschlüsseln das erste Mal erzeugt wird, diese abfangen und für ein späteres Login weglegen und abspeichern. Im zweiten Schritt könnte der Angreifer diesen in der PreBoot-Phase zum Entschlüsseln nutzen. Das gleiche Problem hat man aber prinzipiell aufgrund des Designs von LUKS auch bspw. bei einer USB-basierten Smartcard. LUKS verwaltet Slots mit symmetrischen Passwörtern. Diese würden entweder auf der Smartcard gespeichert und direkt von der Smartcard wieder ausgelesen und dabei ebenfalls über USB wandern. Selbst in PBA-Lösungen, die den Data-Encryption-Key mit einem Public Key eines Benutzers verschlüsseln, muss der verschlüsselte DEK an den zweiten Faktor geschickt werden, um dort asymmetrisch entschlüsselt und wieder über USB zurückgeschickt zu werden. So wäre sogar der DEK kompromittiert. Insofern wird sogar im Falle des Yubikeys nicht der DEK kompromittiert, sondern eben nur ein Slot, der im Bedarfsfall gelöscht werden kann, ohne alle Daten umschlüsseln zu müssen.

## 7 Ausblick

Das hier vorgestellte Konzept erscheint gerade durch die Managebarkeit für Unternehmen und anderen größeren Benutzergruppen vielversprechend, so dass es zeitnah in LinOTP integriert werden könnte.

## Literatur

- [1] RSA: *RSA History*. <https://www.rsa.com/node.aspx?id=2760>
- [2] Kölbel: *Starke Zweifaktorauthentisierung mit LinOTP - modular, skalierbar, frei*. Tagungsband, Chemnitzer Linux-Tage 2012.
- [3] Transparency Market Research: *Smart Card Market - Global Industry Analysis, Size, Share, Growth, Trends, And Forecast 2012 - 2018* <http://www.transparencymarketresearch.com/smart-card.html>
- [4] Yubico: *YubiKey Security Evaluation*. [http://static.yubico.com/var/uploads/pdfs/Security\%20Evaluation\%202\\_0\\_1.pdf](http://static.yubico.com/var/uploads/pdfs/Security\%20Evaluation\%202_0_1.pdf). 2012
- [5] Vamana: *Formal Analysis of Yubikey*. <http://n.ethz.ch/~lvamanu/download/YubiKeyAnalysis.pdf>. 2012
- [6] Oswald, Richter, Paar: *Side-Channel Attacks on the Yubikey 2 One-Time Password Generator*. [http://link.springer.com/chapter/10.1007\%2F978-3-642-41284-4\\_11](http://link.springer.com/chapter/10.1007\%2F978-3-642-41284-4_11). 2013
- [7] Krawczyk, Bellare, Canetti: *RFC2104: HMAC-SHA-1 Algorithm*. <https://www.ietf.org/rfc/rfc2104.txt>
- [8] M'Raihi, Bellare, Hoornaert, Naccache, Ranen: *RFC4226: HOTP Algorithm*. <https://www.ietf.org/rfc/rfc4226.txt>. 2005.
- [9] Yubico: *YubiKey Integration for Full Disk Encryption*. <http://www.yubico.com/wp-content/uploads/2012/10/YubiKey-Integration-for-Full-Disk-Encryption-with-Pre-Boot-Authentication-v1.2.pdf>. 2012.
- [10] Heen: *ykfde* <https://github.com/tfheen/ykfde>. 2011.
- [11] Klien: *initramfs\_ykfde*. [https://github.com/flowolf/initramfs\\_ykfde](https://github.com/flowolf/initramfs_ykfde). 2012.

# Grüner verschlüsseln – Messung des Energieverbrauchs von Verschlüsselungsalgorithmen

Jens Lang

TU Chemnitz

jens.lang@informatik.tu-chemnitz.de

Der Artikel gibt einen Überblick über hardware- und softwarebasierte Energiesparmethoden. Er erläutert, wie sich ohne zusätzliche Messhardware der Energieverbrauch moderner CPUs ermitteln lässt. Mit der vorgestellten Technik wird der Energieverbrauch verschiedener in OpenSSL implementierter Verschlüsselungsalgorithmen analysiert. Auch auf die Energieeffizienz spezieller Maschinenbefehle zur Verschlüsselung, die moderne CPUs anbieten, wird eingegangen.

## 1 Einleitung

Rechenzentren in Deutschland haben im Jahr 2011 ca. 35 PJ elektrischer Energie verbraucht [10]. Das entspricht ca. 1,9 % des Stromverbrauchs dieses Jahres in Deutschland [6]. Zum Schutze unserer Umwelt sollte sich dieser Anteil nicht weiter erhöhen, auch wenn vermutlich die Gesamtrechenleistung der in Deutschland installierten Server weiter ansteigen wird. Dafür ist es notwendig, sowohl für die Hardware- als auch für die Softwareebene Energiespartechiken zu entwickeln. Jedes für Berechnungen eingesparte Watt Leistung lohnt sich doppelt: Mit sinkender Leistungsaufnahme der Rechner sinkt auch der Bedarf an Kühlleistung und für weitere Infrastruktur wie unterbrechungsfreie Stromversorgung. Insbesondere ist auch bei Mobilgeräten wie Laptops oder Mobiltelefonen der Energieverbrauch eine entscheidende Größe. Da der Strom bei diesen Geräten nicht aus der Steckdose kommt, ermöglichen energieeffiziente Implementierungen von Hard- und Software längere Laufzeiten.

In diesem Artikel werden die wichtigsten Energiesparmethoden heutiger CPUs erläutert und einige Strategien zur Implementierung energiesparender Algorithmen vorgestellt. Außerdem wird erläutert, wie man rein softwarebasiert, also ohne zusätzliche Messhardware, den Energieverbrauch von CPUs herausfinden kann. Diese Methode wird anschließend an Verschlüsselungsalgorithmen als ein Beispiel für eine Klasse häufig genutzter Algorithmen demonstriert. Es wird ermittelt, welcher der zahlreichen verfügbaren Verschlüsselungsalgorithmen am energieeffizientesten arbeitet. Dabei wird auch auf die Frage eingegangen, ob spezielle Hardwarebefehle zur Verschlüsselung eine Energieeinsparung bringen.

## 2 Hardware-Energiesparmechanismen

Die beiden wichtigsten Energiesparmechanismen heutiger sind die dynamische Spannungs- und Frequenzregulierung (engl. *dynamic voltage and frequency scaling*) über  $P$ -

*States* sowie das Abschalten nicht benötigter CPU-Komponenten über *C-States*. Auch Architekturmerkmale, die dazu führen, dass Programme schneller ausgeführt werden, wie spezielle Maschinenbefehle oder große Caches, ermöglichen eine energieeffizientere Ausführung. Architekturmerkmale hingegen, die die Komplexität der Schaltkreise stark erhöhen, wie eine Out-of-Order-Execution-Einheit führen tendenziell zu einem höheren Energieverbrauch. Die beiden erstgenannten Mechanismen werden im Folgenden näher erläutert.

## 2.1 Prozessorzustände – C-States

Durch Abschalten einiger CPU-Komponenten kann ihre Leistungsaufnahme gesenkt werden. Dafür definiert ACPI verschiedene *C-States* (Prozessorzustände) [15]. Im Zustand *C0* arbeitet der Prozessor normal mit voller Leistung. Über den einen speziellen Maschinenbefehl (z. B. *HLT* bei *x86*-CPUs) kann in den *C1*-Zustand *Halt* mit verringerter Leistungsaufnahme gewechselt werden. Der Haupttakt stoppt, bis der nächste Interrupt an die CPU gesendet wird. Die CPU wechselt dann ohne Verzögerung in den Zustand *C0* und die Anwendung wird fortgesetzt, ohne dass sie von der Unterbrechung etwas merkt. Zum Wechseln in den *C2*-Zustand *Stop-Clock* wird kein Befehl per Software, sondern ein Signal an einen speziellen CPU-Pin gesandt. Fast alle internen Takte werden nun angehalten. Im Zustand *C3* *Sleep* wird zusätzlich u. a. die Cache-Kohärenz nicht mehr sichergestellt. In den weiteren *C-States* werden nacheinander mehr und mehr Teile der CPU und des Chipsatzes stillgelegt bzw. Takte angehalten, um weitere Leistung einzusparen. Das genaue Verhalten ist jedoch nicht spezifiziert. Die Zeit für den Übergang von dem jeweiligen *C-State* in den Zustand *C0* erhöht sich allerdings mit steigender *C-State*-Nummer. Im Linux-Kernel werden die *C-States* vom *cpuidle*-Subsystem verwaltet.

## 2.2 Leistungszustände – P-States

Die Leistungsaufnahme eines CMOS-Schaltkreises lässt sich näherungsweise über die Formel

$$P = P_{\text{dyn}} + P_{\text{stat}} = CU^2f + UI_{\text{leak}}$$

berechnen [4]. Das heißt, die Gesamtleistung setzt sich zusammen aus der dynamischen und der statischen Leistung  $P_{\text{dyn}}$  und  $P_{\text{stat}}$ . Die statische Leistung wird durch die Leckströme z. B. zwischen den Leiterbahnen verursacht und ist unabhängig von der Taktfrequenz oder der Auslastung der CPU [4]. Sie ergibt sich als Produkt der Spannung  $U$  und des Leckstroms  $I_{\text{leak}}$ . Die dynamische Leistung fällt nur an, wenn Berechnungen durchgeführt werden, sich also interne Zustände ändern. In die dynamische Leistung geht die Spannung  $U$  quadratisch ein, die Taktfrequenz  $f$  und die Kapazität  $C$  der Leitungen und Transistoren jeweils linear. Es erscheint also sinnvoll, die Versorgungsspannung abzusenken, um elektrische Leistung einzusparen.

Wird die Versorgungsspannung abgesenkt, ist proportional dazu auch die Taktfrequenz zu senken [16], damit die Transistoren mehr Zeit zum Umschalten bekommen

[5]. Die Anzahl pro Zeiteinheit durchgeführter Berechnungen einer CPU fällt mit sinkender Taktfrequenz linear ab. Gleichzeitig verringert sich die dynamische Leistung in dritter Potenz. Sinnvoll eingesetzt birgt also Frequenzabsenkung ein großes Potenzial zur Verringerung der Aufnahme elektrischer Leistung. Sicherzustellen, dass das Einsparen von Leistung auch zu einer Energieeinsparung führt, ist allerdings Aufgabe des Softwareentwicklers.

Zum Absenken der Taktfrequenz spezifiziert ACPI sogenannte *P-States*. Sie werden etwa seit dem Jahr 2000 implementiert und sind beispielsweise bei Intel-CPU's unter dem Namen *SpeedStep* bzw. bei AMD-Prozessoren unter den Namen *PowerNow!* und *Cool'n'Quiet* bekannt. Jedem der P-States von P0 bis P<sub>k</sub> ist in absteigender Reihenfolge eine Taktfrequenz zugeordnet. Im P-State P0 hat die CPU die volle Rechenleistung, also aber auch die höchste Leistungsaufnahme. Die Leistungsaufnahme in den folgenden P-States nimmt immer weiter ab. Zwischen den P-States kann im laufenden Betrieb jederzeit gewechselt werden. In der Regel setzt das Betriebssystem den P-State automatisch unter Berücksichtigung der aktuellen CPU-Auslastung, im Linux-Kernel das Modul *cpufreq*.

### 3 Software-Energiesparmechanismen

Damit die in der Hardware implementierten Mechanismen zur Energieeinsparung auch tatsächlich in einem verringerten Energieverbrauch bei der Ausführung von Software resultieren, muss der Softwareentwickler sie auch nutzen. Im Folgenden werden einige Ideen angegeben, wie sich Algorithmen entwerfen bzw. implementieren lassen, um möglichst energieeffizient zu sein.

Die wohl effektivste als auch schon am längsten eingesetzte Methode zum Einsparen von Energie ist *Race-to-Idle*: Der Algorithmus wird so implementiert, dass er so schnell wie möglich ausgeführt wird. So kann die CPU schnellstmöglich die nächste Aufgabe ausführen bzw. abgeschaltet werden. Für Programmabschnitte hingegen, deren Ausführungszeit allein von der Speichergeschwindigkeit abhängt, kann es sinnvoll sein, von vornherein einen P-State mit verringerter Frequenz zu wählen. Die Daten müssen von der CPU genau so schnell verarbeitet werden wie sie vom bzw. zum Speicher fließen. Der genaue P-State ist hierbei vom Problem sowie vom Typ der CPU und des Speichers abhängig.

Bei paralleler Verarbeitung kann es vorkommen, dass für einen Datenaustausch eine CPU auf eine andere wartet. In einem solchen Fall sollte die wartende CPU in einen möglichst energiesparenden C-State wechseln, um sich beim Eintreffen von Daten aufzuwecken zu lassen. Ist eine Prognose der Wartezeit möglich, kann sogar der C-State passend gewählt werden, dass der rechtzeitige Wechsel in den Status C0 möglich ist. In manchen Fällen ist es auch sinnvoll, die meist sehr energieintensive Übertragung von Daten über ein Netzwerk zu umgehen, indem an anderer Stelle benötigte Daten einfach dort erneut berechnet werden.

Manche Berechnungen können auf Beschleunigern wie Grafikprozessoren energieeffizienter ausgeführt werden. In diesem Fall sollten sie dorthin ausgelagert werden. Existieren für eine Operation existieren, sollten diese verwendet werden, da sie die Operation effizienter ausführen als generische Befehle.

## 4 Energieverbrauch messen

Zwei wichtige Methoden zum Ermitteln des Energieverbrauchs werden in diesem Abschnitt vorgestellt. Weitere Möglichkeiten, die Leistung von CPUs ohne zusätzliche Messhardware wie [7] zu ermitteln, bestehen beispielsweise über die ACPI-Funktion zum Auslesen des Ladestandes von Laptop-Batterie oder über IPMI.

### 4.1 RAPL

Die Technologie *Running Average Power Limit* (RAPL) wurde von Intel mit der *Sandy-Bridge*-Prozessorgeneration eingeführt [13] und dient dazu, eine Obergrenze für die Leistungsaufnahme einer CPU festzulegen. Dadurch kann beispielsweise sichergestellt werden, dass die Leistungsaufnahme eines Clusters nicht über dem spezifizierten Höchstwert liegt und so die Spannungsversorgung oder die Kühlung überlastet. Für diesen Zweck muss selbstredend die momentane Leistung ermittelt werden. Dadurch ergibt sich die Möglichkeit, den Energieverbrauch der CPU softwarebasiert auszulesen. Zu beachten ist, dass bei RAPL keine „Messung“ im Wortsinne durchgeführt wird, sondern vielmehr eine indirekte Schätzung der Leistungsaufnahme anhand verschiedener Hardware-Performance-Counter [13]. Verschiedene Untersuchungen zeigen, dass die Schätzung ziemlich genau ist [9, 13].

Die geschätzten Werte für den Energieverbrauch seit einem bestimmten Zeitpunkt werden über maschinenspezifische Register bereitgestellt [11, Bd. 3B, Kap. 14-28], die einmal pro Millisekunde aktualisiert werden. Das Register `MSR_PKG_ENERGY_STATUS` enthält den Energieverbrauch des kompletten CPU-Packages, also einschließlich aller Caches und der „Uncore“-Energie. Den Energieverbrauch der *power planes* 0 und 1 lässt sich über die Register `MSR_PPO_ENERGY_STATUS` und `MSR_PP1_ENERGY_STATUS` auslesen. Die *power plane* 0 versorgt die CPU-Kerne, *power plane* 1 meist die Onchip-Grafikkarte. Über das Register `MSR_DRAM_ENERGY_STATUS` erhält man den Energieverbrauch der Speicherriegel, die der CPU zugeordnet sind. Der Umrechnungsfaktor der Registerwerte in Joule ergibt sich aus  $\frac{1}{2}$  potenziert mit den Bits 8 bis 12 des Registers `MSR_RAPL_POWER_UNIT`. Nicht alle MSRs sind auf allen CPU-Typen verfügbar.

Auslesen lassen sich maschinenspezifische Register unter Linux mithilfe des Kernel-Moduls `msr`. Es legt für jede CPU die Datei `/dev/cpu/k/msr` an, die Zugriff auf deren MSRs bietet. Das Register mit der Nummer `m` lässt sich auslesen, indem das `m`-te 64-Bit-Wort der Datei gelesen wird. Der Benutzer muss für die Datei Leserechte besitzen und die ausführbare Datei muss die Capability `CAP_SYS_RAWIO` besitzen, die der Root-Benutzer mit dem Befehl `setcap cap_sys_rawio=ep dateiname` setzen kann.

## 4.2 APM

AMD implementiert in seinen CPUs ab der *Bulldozer*-Generation das *Application Power Management* (APM) [2, Kap. 2.5.2.1.1]. Ähnlich wie RAPL handelt es sich hierbei um eine Technik zur Beschränkung der Leistungsaufnahme, bei der die tatsächliche Leistungsaufnahme anhand von Hardware-Performance-Countern geschätzt wird. Das Aktualisierungsintervall beträgt 10 ms [9]. Auch bei APM können die Werte aus einem maschinenspezifischen Register, nämlich dem MSR C001 0077, ausgelesen werden. Anders als bei RAPL handelt es sich jedoch nicht um Energie-Werte, sondern um die durchschnittliche Leistungsaufnahme des letzten Messintervalls. Außerdem sind die Register nicht direkt von einem CPU-Programm auslesbar, sondern nur über das Sideband-Interface [3], also den SMBus. Das macht Verfahren ein wenig komplizierter als bei RAPL.

## 5 Verschlüsselung

Die wohl bekannteste und am weitesten verbreitete Verschlüsselungsbibliothek ist OpenSSL [1]. Sie, genauer die `libcrypto` aus diesem Paket, wurde in diesem Artikel genutzt, um den Energieverbrauch verschiedener symmetrischer Verschlüsselungsalgorithmen zu vergleichen. Die verwendete OpenSSL-Version 1.0.1e unterstützt die folgenden Algorithmen, die untersucht wurden: AES, Blowfish, Camellia, CAST5, DES, Triple-DES, RC2 und SEED.

Alle genannten Algorithmen sind Blockchiffren. *Blockchiffre* bedeutet, dass die Nachricht in Blöcke einer festgelegten Länge zerteilt wird, von denen jeder einzeln verschlüsselt wird. Die *Betriebsart* legt fest, wie der Rundenschlüssel, also der Schlüssel für jeden der Blöcke, aus dem Ausgangsschlüssel bestimmt wird [14]. Beim Verfahren ECB wird jeder Block direkt mit dem Ausgangsschlüssel verschlüsselt. Bei den Verfahren CBC, CFB und OFB werden jeweils Geheim- oder Klartexte aufeinanderfolgender Blöcke auf eine definierte Art und Weise mit dem Ausgangsschlüssel oder dem vorangehenden Rundenschlüssel verknüpft. So wird sichergestellt, dass gleiche Klartextblöcke unterschiedliche Geheimtextblöcke ergeben. Bei den Verfahren CTR und XTS wird der Rundenschlüssel aus der Blocknummer erzeugt, um einen wahlfreien Zugriff auf die Daten zu ermöglichen.

Die Ver- und Entschlüsselung ist sowohl über Bibliotheksaufrufe der `libcrypto` als auch über das Kommandozeilenwerkzeug `openssl` möglich. Für die Messungen zu diesem Artikel wurde das Kommandozeilenwerkzeug verwendet.

### 5.1 Energieverbrauch von Verschlüsselungsalgorithmen

Für die Messung des Energieverbrauchs der verschiedenen Algorithmen wird die folgende Kommandozeile auf der `bash` ausgeführt:

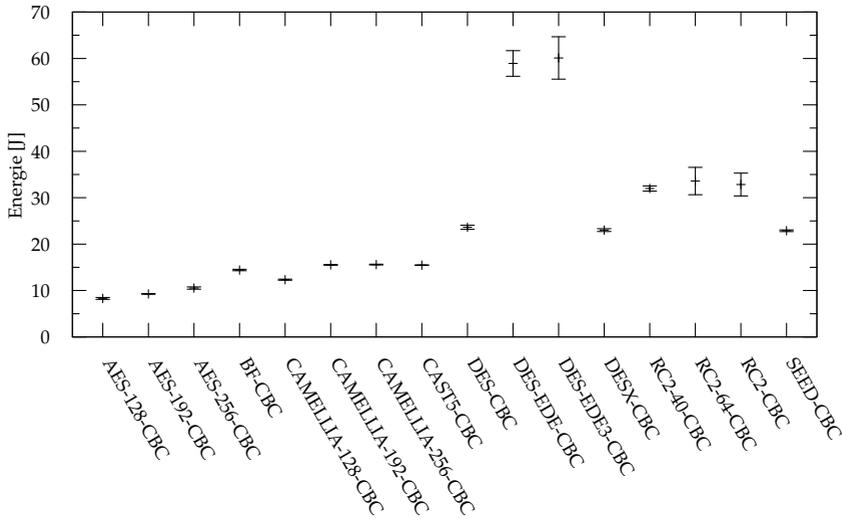


Abbildung 1: Energieverbrauch verschiedener Verschlüsselungsalgorithmen zur Verschlüsselung von 50 MiB Daten

```
head -c 50M < /dev/zero | openssl aes-256-cbc -salt -pass geheim
> /dev/null.
```

Es werden also 50 MiB Nullen aus dem Device `/dev/zero` gelesen, diese vom Programm `openssl` mit dem angegebenen Algorithmus, in diesem Fall AES mit 256 Bit Schlüssellänge und der CBC-Betriebsart, verschlüsselt und danach in das Device `/dev/null` geschrieben. Für `aes-256-cbc` werden nacheinander alle verfügbaren Verschlüsselungsalgorithmen eingesetzt. Unmittelbar vor Beginn und nach Beendigung der Ausführung werden die Uhrzeit und die RAPL-MSRs nach der oben beschriebenen Methode ausgelesen.

Die Ergebnisse der Messung sind in Abb. 1 dargestellt. Die Bezeichnung der Algorithmen entspricht der von OpenSSL verwendeten Terminologie. Es wurden alle verfügbaren Algorithmen mit der Betriebsart CBC untersucht, da CBC die sichere Betriebsart, die vermutlich am häufigsten verwendet wird, ist.

Die Algorithmen benötigen eine Energie von ca. 8 bis 60 J für die Verschlüsselung von 50 MiB Daten. Zum Vergleich: Ersetzt man in oben angegebener Kommandozeile den Aufruf von `openssl` durch `cat`, beträgt der Energieverbrauch ca. 0,7 J. Die restliche Energie ist also der Verschlüsselung zuzuschreiben. Aus der Reihe fällt der Algorithmus DES-EDE, der besonders viel Energie benötigt. Das ist nicht verwunderlich, da es sich dabei um Triple-DES handelt, also DES dreimal hintereinander ausgeführt wird. AES ist der Algorithmus, der mit der wenigsten Energie auskommt. Ein entscheidendes Kriterium bei der Wahl des Algorithmus für AES war unter anderem eine möglichst kurze Laufzeit. Daher überrascht das Ergebnis nicht.

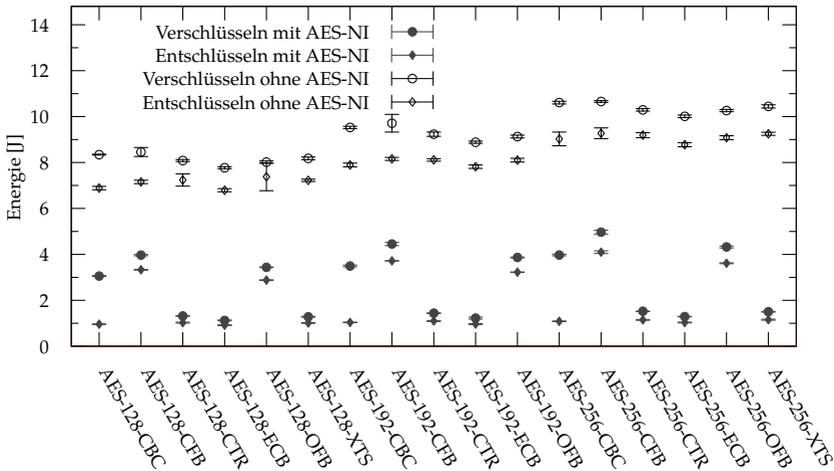


Abbildung 2: Energieverbrauch verschiedener AES-Varianten mit und ohne AES-Hardwarebefehle

## 5.2 AES Native Instructions

Seit etwa 2008 beherrschen Intel- und AMD-CPU's die Befehlssatzerweiterung *AES-NI* [8]. Sie definiert 6 neue Maschinenbefehle, die das Ver- und Entschlüsseln nach dem AES-Standard beschleunigen sollen. Die Befehle *AESENC* und *AESDEC* dienen dazu, einen Block zu ver- bzw. entschlüsseln. Analog dazu existieren die Befehle *AESENCCLAST* und *AESDECLAST* für den jeweils letzten Block einer Nachricht, für den eine spezielle Funktion nicht ausgeführt werden muss. Die Befehle *AESKEYGENASSIST* und *AESIMC* werden beim Erzeugen des Rundenschlüssels beim CBC-Verfahren genutzt [12].

OpenSSL nutzt die AES-Maschinenbefehle, wenn sie von der CPU unterstützt werden. In einem zweiten Experiment wurden daher alle für AES verfügbaren Varianten einmal mit und einmal ohne Befehlssatzerweiterung untersucht: 50 MiB Daten werden verschlüsselt in eine RAM-Disk geschrieben und von dort wieder entschlüsselt. Die Ergebnisse der Energiemessung für beide Vorgänge sind in Abb. 2 dargestellt: Mit Nutzung des AES-Befehlssatzes sinkt der Energieverbrauch auf etwa ein Drittel gegenüber der Implementierung mit generischen Befehlen. Außerdem ist bemerkenswert, dass die Verschlüsselung bei den Betriebsarten CFB, CTR und XTS, die eine parallele Verschlüsselung ermöglichen, nur ca. 1 J benötigt, während bei den anderen Verfahren ca. 4 J benötigt werden. Für das Entschlüsseln bei der Betriebsart CBC, das auch parallel möglich ist, wird ebenfalls nur eine Energie von ca. 1 J benötigt. Die aufgenommene elektrische Leistung liegt ohne AES-Befehle bei ca. 44 W fürs Verschlüsseln und bei 39 W fürs Entschlüsseln. Mit Nutzung der AES-Befehle erhöht sich die Leistung fürs Verschlüsseln auf 50 W bei den Betriebsarten CTR, ECB, XTS und sinkt auf ca. 41 W für die anderen Betriebsarten.

## 6 Zusammenfassung

Der Artikel hat eine Einführung in Energiesparmechanismen moderner CPUs gegeben. Das von Intel-CPU's bereitgestellte *RAPL* zum Messen des Energieverbrauchs von CPU's wurde näher vorgestellt und genutzt, um zu untersuchen, wie viel Energie verschiedene symmetrische Verschlüsselungsalgorithmen benötigen. Es hat sich herausgestellt, dass AES von den in OpenSSL implementierten Algorithmen der energieeffizienteste ist. Die Energieeffizienz wird sogar noch gesteigert, wenn der spezielle Maschinenbefehlssatz *AES-NI* genutzt wird.

## Literatur

- [1] *OpenSSL: The Open Source toolkit für SSL/TLS*. <http://www.openssl.org/>.
- [2] Advanced Micro Devices: *BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 00h-0Fh Processors*, Oktober 2012. Rev 3.12.
- [3] AMD: *Advanced Platform Management Link (APML) Specification*, August 2009. <http://developer.amd.com/wordpress/media/2012/09/41918.pdf>. Rev 1.02.
- [4] BELOGLAZOV, ANTON, RAJKUMAR BUYYA, YOUNG CHOON LEE und ALBERT Y. ZOMAYA: *A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems*. *Advances in Computers*, 82:47–111, 2011. doi: 10.1016/B978-0-12-385512-1.00003-7.
- [5] BENZ, BENJAMIN: *Nachbrenner: Prozessor-Turbos von AMD und Intel*. c't – Magazin für Computertechnik, (16):170–175, 2010.
- [6] *Zahlen und Fakten Energiedaten*, Bundesministerium für Wirtschaft und Technologie. <http://www.bmwi.de/BMWi/Redaktion/Binaer/energie-daten-gesamt.property=blob.bereich=bmwi2012.sprache=de.rwb=true.xls>. 2013. [2014-01-10].
- [7] GE, RONG, XIZHOU FENG, SHUAIWEN SONG, HUNG-CHING CHANG, DONG LI und KIRK W. CAMERON: *PowerPack: Energy Profiling and Analysis of High-Performance Systems and Applications*. *IEEE Trans. Parallel Distrib. Syst.*, 21(5):658–671, Mai 2010. doi: 10.1109/TPDS.2009.76.
- [8] GUERON, SHAY: *Intel Advanced Encryption Standard (AES) New Instructions Set*. White Paper 323641-001, Intel Corporation, September 2013. Revision 3.01.
- [9] HACKENBERG, DANIEL, THOMAS ILSCHKE, ROBERT SCHÖNE, DANIEL MOLKA, MAIK SCHMIDT und WOLFGANG E. NAGEL: *Power measurement techniques on standard compute nodes: A quantitative comparison*. 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 0:194–204, 2013. doi: 10.1109/ISPASS.2013.6557170.
- [10] HINTEMANN, RALPH und KLAUS FICHTER: *Energieverbrauch und Energiekosten von Servern und Rechenzentren in Deutschland: Aktuelle Trends und Einsparpotenziale bis 2015*. Kurzstudie, Borderstep Institut für Innovation und Nachhaltigkeit, Berlin, Mai 2012.
- [11] Intel Corporation: *Intel 64 and IA-32 Architectures Software Developer's Manual*, May 2012.
- [12] LAU, OLIVER: *Spezialkommando: Schnelle AES-Chiffres mit Intrinsic*. c't – Magazin für Computertechnik, (14):174–177, 2013.
- [13] ROTEM, EFRAIM, ALON NAVEH, AVINASH ANANTHAKRISHNAN, DORON RAJWAN und ELIEZER WEISSMANN: *Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge*. *IEEE Micro*, 32(2):20–27, 2012. doi: 10.1109/MM.2012.12.
- [14] SCHMEH, KLAUS: *Kryptografie: Verfahren – Protokolle – Infrastrukturen*. dpunkt.verlag, 5. Auflage, 2013. isbn: 978-3864900150.
- [15] TORRES, GABRIEL: *Everything You Need to Know About the CPU C-States Power Saving Modes*, Hardware Secrets. <http://www.hardwaresecrets.com/article/-/611>. September 2008. [2014-01-12].
- [16] ZHUO, J. und C. CHAKRABARTI: *Energy-efficient dynamic task scheduling algorithms for DVS systems*. *ACM Transaction on Embedded Computing*, 7(2):17:1–17:25, Januar 2008.

# Heutige Möglichkeiten von Prozessoren in GNU/Linux-basierten eingebetteten Systemen

Wolfram Luithardt

Hochschule für Technik und Architektur Freiburg, Boulevard de Perolles 80,  
CH-1705 Fribourg, Schweiz

## 1. Einleitung

Während zu Beginn des Computerzeitalters Hardware und Software sehr eng verknüpft waren, haben sich diese beiden Themenfelder mit der Zeit immer mehr voneinander abgekoppelt. Software wurde mit den Jahren immer weiter abstrahiert, um eine möglichst große Plattformunabhängigkeit der Programmiermodelle zu erreichen. Die Verbindung zu den meist sehr plattformabhängigen Hardwaremodulen wird dann über Device-Treiber hergestellt, die für jede Prozessorfamilie speziell angepasst werden müssen. Allerdings gibt es in modernen Prozessoren auch Module, die nur sehr schwierig über Device-Treiber eingebunden werden können, da in den gängigen Programmiermodellen solche Zusatzhardware nicht vorgesehen ist. Die Anzahl solcher sehr spezifischen Module hat sich in den letzten Jahren aufgrund der immer kleineren möglichen mikroelektronischen Strukturen stark vergrößert. Allerdings sind häufig die Spezifikationen für diese Module überhaupt nicht oder nur unvollständig verfügbar, was eine transparente Einbindung in offene Betriebssysteme wie GNU/Linux nicht gerade erleichtert. Im Gegensatz zu Mikroprozessoren, die reine Rechenmaschinen mit stark optimiertem Befehlsdurchsatz sind, bieten Mikrocontroller sehr häufig wesentlich mehr Hardwaremodule, die spezifische Funktionalitäten, wie z.B. die Bedienung von Schnittstellen übernehmen. Weiterhin gibt es Prozessoren, die mehrere symmetrische (d.h. gleichartige) oder asymmetrische (unterschiedliche) Recheneinheiten enthalten, die bei guter Einbindung in ein System, wesentliche Performancesteigerungen erzielen können. Solche Systeme werden häufig auch als SoC (Systems on Chip) bezeichnet; ein vollständiger Computer inklusive aller notwendiger Zusatzmodule sind also auf einem einzigen Chip vereinigt.

Die Frage stellt sich nun, wie diese neuen Recheneinheiten möglichst transparent in ein GNU/Linux-System eingebunden und damit von den enormen Möglichkeiten heutiger SOC profitiert werden kann. Nach einem kurzen Überblick über SoC in

Kapitel 2 berichten wir in Kapitel 3 von einem Forschungsprojekt, welches einen möglichen Ansatz für die transparente Einbindung aufzeigt. In Kapitel 4 wird über Möglichkeiten berichtet, graphische Co-Prozessoren auch für nicht graphische Anwendungen zu verwenden. Dieses Kapitel gibt allerdings nur einen groben Überblick über diese Möglichkeiten. Kapitel 5 gibt einen Ausblick auf zukünftige Entwicklungen.

## 2. Hardwaremodule von SoC

Heutige SOCs für eingebettete Systeme bestehen meist aus mehreren CPU-Cores, speziellen Recheneinheiten wie digitale Signalprozessoren (DSP), Vektorrecheneinheiten (SIMD: Single Instruction, Multiple Data), Recheneinheiten zur Verschlüsselung usw. Weiterhin verfügen solche Systeme über eine ganze Reihe von Modulen für alle gängigen Schnittstellen (USB, Ethernet, SPI, I2C, GPIO usw.), On-Board Speicher (Ram und/oder Flash), Interfaces für externen Speicher sowie Module zur internen Spannungsversorgung und Takterzeugung. Einzig Leistungskomponenten werden noch extern benötigt, da diese wegen dem großen Siliziumflächenbedarf und der Wärmeerzeugung nicht weiter integriert werden können.

Die genannten Schnittstellenmodule können als autonome, sehr hoch spezialisierte Prozessormodule angesehen werden, also meistens Zustandsmaschinen-basierte Automaten, welche ihre Arbeiten bei richtiger Konfigurierung vollkommen unabhängig vom Hauptprozessor im Hintergrund verrichten. Die Kommunikation findet über Register, gemeinsamen Speicher, DMA und Interrupts statt. Diese Module sind heute absoluter Standard und eine Vielzahl von Implementierungen sind vollkommen transparent in GNU/Linux integriert. Dies ist allerdings nicht der Fall für komplexere Recheneinheiten wie DSP, SIMD oder gar FPGA (Field Programmable Gate Array). Diese Cores sind vollständig programmierbar und ermöglichen prinzipiell eine vollständige Parallelisierung von Aufgaben. Je nach Typ der Recheneinheit sind sie natürlich jeweils für gewisse Berechnungen wesentlich besser geeignet als für andere.

Während symmetrische Multiprozessoren, also Chips, welche 2 oder mehrere gleichartige Recheneinheiten besitzen, schon seit langer Zeit sehr gut von Linux unterstützt werden (symmetrisches Multiprocessing: SMP), sieht es bei asymmetrischen SOCs noch relativ schlecht aus. Aufgrund der spezifischen Eigenart jeder Recheneinheit ist es viel schwieriger, diese generell, also unabhängig von der spezifischen Funktionalität eines Prozesses, einzusetzen, sondern es muss vielmehr dynamisch eine Entscheidung getroffen werden, ob eine gewisse Aufgabe besser auf diesem oder jenem Core ausgeführt werden sollte. Dieses müsste vom Programmierer der Anwendung durchgeführt werden, der aber häufig nicht über

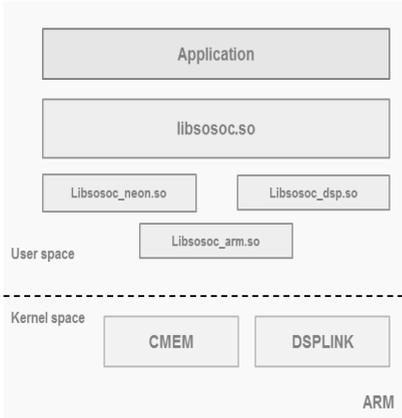
die notwendigen Detailkenntnisse verfügt, um diese Entscheidung richtig zu treffen. Weiterhin werden die Programmiermodelle für asymmetrisches Multiprocessing wesentlich komplexer, da Prozesse bzw. Threads nicht als Unterscheidungskriterium herangezogen werden können. Was liegt also näher, als dem Programmierer ein Framework in die Hand zu geben, das ihm bei dieser Entscheidung hilft?

### **3. Das Projekt SOSoC**

An der Westschweizer Fachhochschule (HES-SO) wurde in den letzten Jahren an einem Projekt gearbeitet, welches die Verwendung zusätzlich vorhandener Hardware auf SoC wesentlich vereinfachen kann. Ziel dieses Projekts mit dem Namen „System Optimization using Systems on Chip (SOSoC) [1] war es, ein Framework bereitzustellen, mit dem vorhandene Hardware transparent in ein Projekt eingebunden werden kann, ohne dass sich der Programmierer allzu sehr mit den Hardwaredetails auseinandersetzen muss. Natürlich übernimmt dieses Framework nicht das eigentliche Erstellen oder ein Übertrag des Programmcodes auf die unterschiedlichen Cores, es kann aber dabei helfen, den besten Core für eine bestimmte Aufgabe auszuwählen bzw. sogar selbst zu entscheiden, auf welchem Core die Aufgabe am schnellsten ausgeführt wird. Weiterhin hilft es mit seinem asynchronen Mode bei der Umsetzung von echt paralleler Ausführung von mehreren Routinen.

Als Entwicklungsplattform wurde der DM3730 von Texas Instruments gewählt, ein Mitglied einer Familie von SoC, die insbesondere durch das Beagleboard [2] starke Verbreitung fand. Dieser Chip beinhaltet neben einem ARM Cortex A8 Prozessor mit L1 und L2 Cache, einen NEON SIMD Coprozessor sowie ein TMS320DMC64X+ DSP-Megamodul, welches ebenfalls über einen eigenen Cache-Speicher verfügt. Natürlich verfügt das SOC noch über eine ganze Reihe weiterer Module, welche allerdings bisher noch nicht in SOSoC eingebunden sind. Während der DSP als vollkommen getrennte Recheneinheit bezeichnet werden kann, ist die Kopplung von NEON an den Cortex-8 wesentlich enger. Der NEON verfügt über einen eigenen Befehlssatz im Cortex und kann damit direkt programmiert werden, ohne über mehrere Ebenen des Speichers Daten austauschen zu müssen. Die Übertragung von Daten zwischen Cortex und DSP benötigt hingegen einen spezifischen Mechanismus, um die Datenintegrität beider Einheiten immer zu gewährleisten. Texas Instruments bietet hierfür bereits eine geeignete Lösung namens DSP-Link an, die ermöglicht, den DSP richtig zu starten und Daten in ihn zu schreiben bzw. von ihm zu lesen. DSP-Link bietet verschiedene Komponenten zum Austausch von Daten, z.B. einen shared Memory, einen Ringbuffer und ein Message-System [3].

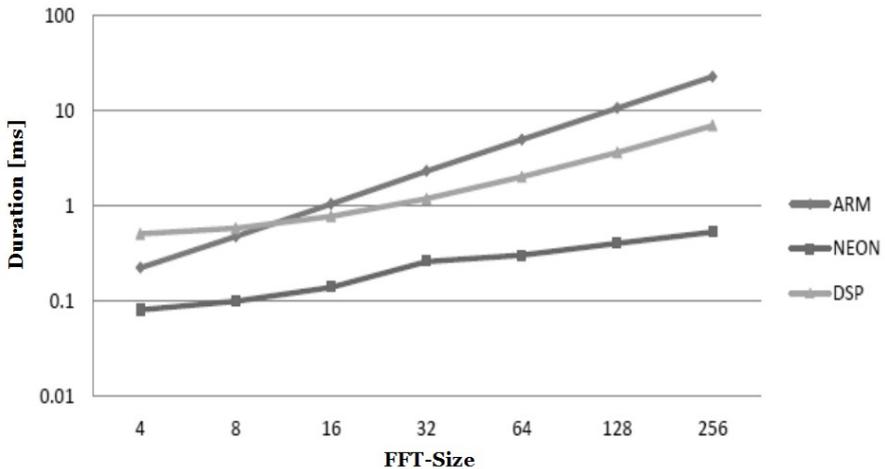
Innerhalb des SOSoC Projekts wurde eine Library erstellt, welche Funktionalitäten zur transparenten Einbindung von Hardware-Modulen zur Verfügung stellt. Diese Library besteht aus einem SOC-unabhängigen Main-Runtime Code (libsosoc.so) sowie einem Code für jeden Core, auf dem eine Berechnung ausgeführt werden kann: Libsosoc\_neon.so Libsosoc\_arm.so, und Libsosoc\_dsp.so. Diese sind natürlich



abhängig vom jeweils verwendeten Core und müssen für jede verwendete Plattform angepasst werden. Interessant wird das Framework natürlich besonders dann, wenn Standardroutinen und -algorithmen für die vorhandenen Cores bereits zur Verfügung stehen und nicht speziell programmiert werden müssen.

Libsosoc kann dann so konfiguriert werden, dass es automatisch Performancemessungen durchführt und eine bestimmte Aufgabe dann auf den Core dispatched, der diese Aufgabe am schnellsten durchführt. Beim Aufruf von Funktionen in LibSOSoC kann auch ausgewählt werden, ob Funktionen synchron oder asynchron ausgeführt werden sollen. Im synchronen Mode wartet das Framework bis die Aufgabe erledigt ist, im asynchronen Modus wird der Code auf der CPU weiter ausgeführt und das Ende der gepacheden Routine wird über ein Interrupt gemeldet. Somit können Aufgaben vollkommen parallel verteilt werden und das Framework übernimmt die Optimierung. Wichtig dabei ist, dass Libsosoc im asynchronen Modus durch die echte Parallelisierung auch dann Vorteile bringt, wenn, aus welchem Grunde auch immer, einmal nicht der beste Core ausgewählt wurde.

Für die Performancemessungen wurde ftrace verwendet, ein Kernel-Tracer, der sich dafür auszeichnet, einen relativ geringen Overhead zu erzeugen [4]. Neben vorprogrammierten Standardtracepoints kann das System auch mit eigenen Tracepoints erweitert und damit optimal an die Gegebenheiten angepasst werden. Eine Erweiterung des Sched\_switch Converters [5] erlaubt sogar, die SOSoC-Tracingdaten in ein VCD-Format umzuwandeln, um es dann z.B. mit GTK-Wave analysieren zu können.



Erste Ergebnisse sind in der obenstehenden Abbildung dargestellt. Es handelt sich hierbei um eine Fast Fourier Transformation (FFT) mit unterschiedlicher Anzahl von Signalpunkten. Während der ARM-Prozessor ein streng logarithmisches Verhalten zeigt (Achtung: die Skalierung auf der Zeitachse ist logarithmisch) sieht man eine deutlich kürzere, aber nicht vollständig logarithmische Dauer für den DSP und den Neon. Erstaunlich ist das relativ schlechte Abschneiden des DSP. Dies ist darauf zurückzuführen, dass die Datenübertragung zwischen ARM und DSP relativ viel Zeit benötigt. Schon bei wenigen Daten benötigt die Übertragung einige 100µs. Der SIMD NEON ist für alle Datengrößen der optimale Core, da er für die Multiplikations- Additionsaufgabe einer FFT optimiert ist. Bei etwas komplexeren Berechnungen, wie sie häufig in der digitalen Signalverarbeitung verwendet werden, würde sicherlich der DSP wesentlich besser abschneiden.

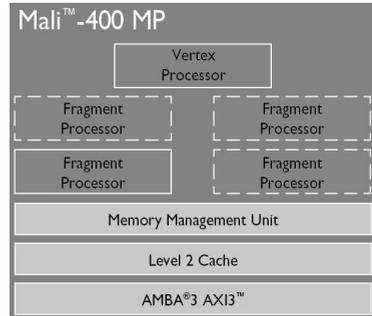
Die SOSoC Libraries stehen unter der GPL zur Verfügung [1] und wir hoffen, dass in Zukunft das Framework weiterentwickelt wird und es in vielen Anwendungen eingesetzt wird. Es sind auch bereits einige Portierungen auf andere Plattformen erfolgreich durchgeführt worden.

#### 4. Verwendung von Graphischen Co-Prozessoren

Auch graphische Prozessoren (GPU), welche teilweise in Prozessoren für embedded Plattformen vorhanden sind, bieten sich natürlich ebenfalls an, nicht graphische Aufgaben parallelisiert umzusetzen. Aufgrund der interessanten Möglichkeiten solcher Prozessoren wurden bereits einige Standards hierfür entwickelt. Auf

Desktop-Basis hat sich CUDA als Plattform etabliert um Nicht-graphische Berechnungen für wissenschaftliche und wirtschaftliche Problemstellungen hoch parallelisiert durchzuführen. Diese von NVIDIA vorgeschlagene Lösung läuft allerdings nur selten auf embedded Prozessoren. Auch OpenCL [6] ist eine Schnittstelle, um parallel laufende Rechnereinheiten, inklusive GPUs in ein System einzubinden. Die in eingebetteten Systemen eingesetzten GPU sind sicherlich etwas weniger leistungsfähig als Desktop-GPU, dafür benötigen sie wesentlich weniger Chipfläche und Energie als Verwandten für Desktops.

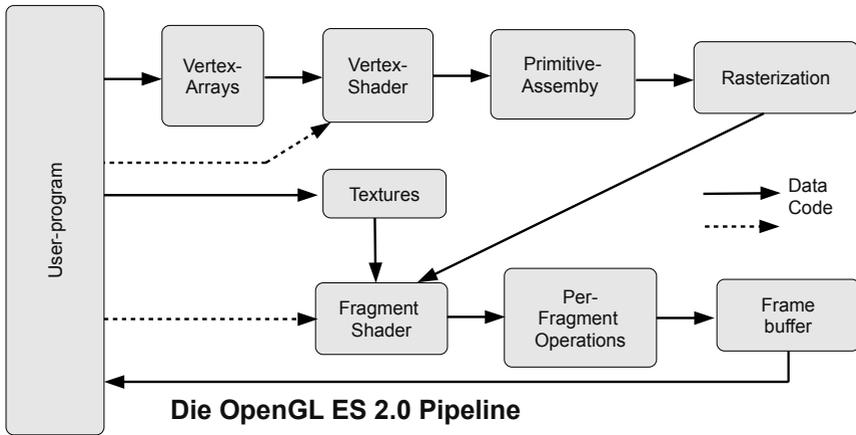
Eine GPU, welche häufig in ARM-basierten Plattformen eingesetzt wird ist der Mali-400 [7], dessen Struktur in nebenstehenden Bild dargestellt ist. Er besteht neben einer eigenen Speicherverwaltungseinheit aus einigen hochspezialisierten Pixelprozessoren, welche zahlreiche graphische Elemente parallel verarbeiten können. Die genaue interne Struktur und die API auf Bitebene sind häufig nicht bekannt und wird vom Hersteller geheim gehalten. Die Programmierung eines solchen graphischen Co-prozessors ist auch deshalb nicht ganz einfach, da sich die Programmiermodelle und die vorhandenen Instruktionen meist sehr von den gewohnten unterscheiden. Aus diesem Grund wurden auch für graphische Aufgaben Standards entwickelt, welche eine API zur Verfügung stellen, um die GPU mit bekannten Mitteln unabhängig vom Betriebssystem programmieren zu können. Ein sehr bekannter Standard ist OpenGL oder sein kleiner Bruder OpenGL ES (ES=Embedded System). Der Mali-400 „spricht“ OpenGL ES in der Version 2.0, welche ein API für eine Pipeline von typischen Funktionen, die zur Darstellung von graphischen Objekten und zum Rendern verwendet werden, anbietet. Neben diesen vorprogrammierten und gut parametrierbaren Funktionen können gewisse Elemente, die sogenannten Shader, auch mit einer eigenen Programmiersprache speziell angepasst werden. Dabei können sowohl Elemente in einem 3-dimensionalen Raum, sogenannte Vertexes, als auch Objektfragmente in einer 2-dimensionalen Projektion mit Hilfe einer C-ähnlichen Programmiersprache frei bearbeitet werden, was eine enorme Vielfalt an Möglichkeiten bietet. Die Shader-Programme werden im Userspace geschrieben, dann kompiliert, in die GPU geladen und dort ausgeführt.



Können nun diese enormen Hardwareressourcen auch verwendet werden, um nicht graphische Berechnungen durchzuführen und damit die Leistung des Prozessors zu erhöhen?

Dafür stellt sich natürlich zuerst die Frage, ob das zu bearbeitende Problem auf eine graphische Funktion gemappt werden kann. Dies ist sicherlich nicht immer der Fall,

jedoch kann mit etwas Erfahrung und Kreativität die eine oder andere Aufgabe



zumindest teilweise übertragen werden. Sehr viele optische Operationen werden über Matrixfunktionen durchgeführt und so besteht auch für andere Aufgaben die Möglichkeiten, diverse Matrixoperationen auf der GPU optimiert durchzuführen. Ebenfalls interessant sind die sogenannten internen Funktionen der GPU, mathematische Operatoren, die in Hardware umgesetzt sind und im optimalen Fall für jeden Pixel angewendet werden. Auch wenn aufgrund von Overhead- und eventuellen Genauigkeitsproblemen die eigentliche Berechnung auf dem Hauptcore prinzipiell schneller bzw. genauer durchgeführt werden kann, erhält man doch den Vorteil einer echten parallelen Durchführung, wenn gewisse Berechnungen auf die GPU ausgelagert werden können. Die Hardware ist vorhanden, warum sie also nicht nutzen?

Seit einiger Zeit wird auch an einem Reverse Engineering Treiber für die Mali GPU für Linux gearbeitet [8]. Bei Erfolg dieses Projekts hätte man dann die Möglichkeit, noch spezifischer auf die vorhandene Hardware zuzugreifen und weitere interessante Frameworks aufzubauen. Allerdings ist ein reverse-Engineering immer ein komplizierter und zeitraubender Prozess, der nicht notwendig wäre, wenn die Hersteller die internen Details ihrer Module freigeben würden. Dieses würde sicherlich auch zu einer wesentlich größeren Verbreitung ihrer Module führen.

## 5. Zusammenfassung und Ausblick

Es ist immer wieder faszinierend zu untersuchen, welche Möglichkeiten uns heutige Prozessoren prinzipiell bieten. Gerade eingebettete Prozessoren bestehen

aus zahlreichen zusätzlichen Hardware-Modulen, welche zur Zeit nur sehr reduziert eingesetzt werden. Durch die Quelloffenheit bieten GNU/Linux-Systeme natürlich gute Möglichkeiten, selbst mit diesen Modulen zu experimentieren und diese früher oder später voll transparent in ein System einzubinden. Der Weg dorthin ist allerdings nicht einfach, solange die Hersteller die Details dieser Module nicht offenlegen. Mit SOSoC ist ein erster, wenn auch kleiner Anfang gemacht, neue Modelle zu diskutieren und damit praktische Erfahrungen zu sammeln.

Allerdings geht die Entwicklung neuer Hardware sehr schnell vonstatten, sodass kaum Zeit bleibt, ein System richtig zu etablieren. Bei den graphischen Co-Prozessoren stehen neue Module am Start (z.B. die Mali T6xx-Familie), die weitere interessante Möglichkeiten versprechen. OpenGL ES 3.0 ist noch wesentlich besser für nicht graphische Berechnungen geeignet. Erste Hardware für diese Version ist seit kurzem erhältlich und so werden wir sicherlich neue, faszinierende Beispiele dazu sehen können

Ein großes Dankeschön an dieser Stelle an das gesamte SOSoC-Team der Westschweizer Fachhochschule (HES-SO) für ihre tolle Arbeit, an das RCSO ISYS für die Ermöglichung dieses Projektes und an Texas Instruments für die technische Unterstützung. Weiterhin möchte ich hier auch dem gesamten Team der Chemnitzer Linuxtage danken, jedes Jahr diese tolle Veranstaltung zu organisieren und durchzuführen.

## **Literatur**

[1] <https://sourceforge.net/projects/sosoc/>

[2] Beagleboard: <http://beagleboard.org/Products/BeagleBoard>

[3] [http://processors.wiki.ti.com/index.php/DSPLink\\_Overview](http://processors.wiki.ti.com/index.php/DSPLink_Overview)

[4] <https://www.kernel.org/doc/Documentation/trace/ftrace.txt>

[5] <http://www.spinics.net/lists/linux-rt-users/msg04268.html>

[6] <http://www.khronos.org/opencv/>

[7] <http://www.arm.com/products/multimedia/mali-graphics-hardware/mali-400-mp.php>

[8] <http://limadriver.org/>

alle Links: Stand Dezember 2013.

# Informationelle Selbstbestimmung und Informationsfreiheit aus Sicht von Recht und Ökonomik

Falk Zscheile

falk.zscheile@gmail.com

<http://www.pirschkarte.de/>

Der Vortrag versucht anhand einfacher Denkmodelle aus Rechtswissenschaft und Ökonomik darzustellen, welche Antworten beide Disziplinen auf das Verhältnis von Datenschutz und Informationsfreiheit geben.

## 1 Einleitung

Nichts ist für Sozial- und Geisteswissenschaften spannender als gesellschaftlicher Wandel. Durch ihn eröffnen sich nicht nur neue Forschungsgebiete, sondern auch die Möglichkeit, bestehende Theorien und Modelle zu überprüfen und gegebenenfalls zu überarbeiten.

Eine dieser gesellschaftlichen Veränderungen ist die Fülle an Daten, welche die Informationsgesellschaft verursacht. Damit verbunden ist die Betonung von zwei Aspekten. Zum Einen die Sorge um die eigenen Daten, die jeder als Teil der Informationsgesellschaft verursacht und hinterlässt. Zum Anderen die Möglichkeit auf unzählige Daten zugreifen zu können und mit Hilfe dieser Daten und ihrer Kombination neue Informationen zu gewinnen, mit welchen sich dann ein wirtschaftlicher oder gesellschaftlicher Mehrwert generieren lässt.

Im Folgenden soll dargestellt werden, wie Rechtswissenschaft und Ökonomik (Ökonomie als Hilfswissenschaft der Juristerei) mit Fragestellungen in der Informationsgesellschaft im Bereich der personenbezogenen Daten und der Informationsfreiheit umgehen und welche Antworten sie geben.

## 2 Informationen als digitaler Rohstoff

Die mit der Digitalisierung der Gesellschaft verbundene Fülle an speicherbaren Daten und den daraus zu gewinnenden Informationen haben zu völlig neuen Möglichkeiten bei der Generierung von wirtschaftlichen Werten geführt. Daten werden als digitaler oder virtueller Rohstoff der Zukunft angesehen. Es geht im Folgenden nicht um die Datenerhebung durch Geheimdienste unter sicherheitspolitischen Aspekten. Welche Rechte und Pflichten der Staat diesbezüglich gegenüber seinen und den Bürgern anderer Länder hat, ist ein eigener Themenbereich. Dieses Gebiet der Datenerhebung richtet sich primär nach rechtlichen und politischen Gesichtspunkten. Die

Ökonomie kann hierzu fast nichts beitragen. Anders verhält es sich, wenn man die in der Informationsgesellschaft anfallenden Daten mit und ohne Personenbezug im Verhältnis Bürger – Wirtschaft betrachtet (digitaler Rohstoff).

Bei digital vorliegenden Informationen handelt es sich um einen von der Rechtsordnung nur fragmentarisch geregelten Bereich. Informationen allgemeiner Art werden von der Rechtsordnung insbesondere geschützt, wenn sie Inhalte aus dem Recht des Geistigen Eigentums aufweisen. Dies ist beispielsweise der Fall, wenn eine geistig-schöpferische Leistung vorliegt oder die Informationen in Form einer Datenbank zusammengestellt sind. Ansonsten genießt die Information an sich keinen Schutz durch die Rechtsordnung.

Etwas anderes gilt wiederum in zwei Fällen. Zum einen, wenn es sich um Informationen mit Personenbezug handelt. Zum anderen, wenn es um Informationen aus allgemein zugänglichen Quellen geht.

Schutzgegenstand ist in beiden Fällen aber nicht die Information selbst. Das Recht auf informationelle Selbstbestimmung (Datenschutzrecht) schützt die Person, auf welche sich die Information bezieht. Das Recht auf Informationen aus allgemein zugänglichen Quellen schützt den Zugang einer Person zu diesen Quellen. Die Existenz solcher Quellen wird vorausgesetzt.

Ob eine Person geschützt wird oder die Information selbst, macht einen entscheidenden Unterschied. Dies wird im Folgenden noch deutlicher werden.

### **3 Was sagt die Rechtswissenschaft?**

Der Rechtswissenschaft geht es (unter anderem) um ein gerechtes Zusammenleben innerhalb einer Gesellschaft. Wie sich der Gesetzgeber eine gerechte Gesellschaft vorstellt, ergibt sich aus den Gesetzen. Wie sich die Juristen Gerechtigkeit vorstellen ergibt sich aus der Auslegung von Gesetzen und hierzu ergehender Gerichtsentscheidungen. Über all dem wacht das Bundesverfassungsgericht und prüft, ob diese Rechtsausgestaltung mit der Idee des Grundgesetzes, insbesondere mit den Grundrechten vereinbar ist.

Dabei gelten die Grundrechte nicht nur im Verhältnis Bürger – Staat, sondern sind als Elemente einer objektiven Wertordnung bei der Ausgestaltung der gesamten Rechtsordnung zu berücksichtigen.<sup>1</sup> Diese Wertordnung wird durch das Menschenbild des Grundgesetzes flankiert, welches von der einzelnen Person im Spannungsfeld zwischen Individuum und Gemeinschaft ausgeht.<sup>2</sup>

Diese bereits aus der Anfangszeit der Bundesrepublik stammenden Äußerungen des Bundesverfassungsgerichts beanspruchen nach wie vor Gültigkeit und können als Ausgangspunkt für die weitere Betrachtung genommen werden.

<sup>1</sup>So schon *BVerfG*, Urteil v. 15. Jan. 1958 (1 BvR 400/51) *BVerfGE* 7, 198-230 (205 f.) – Lüth-Urteil –.

<sup>2</sup>So schon *dass.*, Urteil v. 20. Juli 1954 (1 BvR 459/52) *BVerfGE* 4, 7-27 (15 f.) – Investitionshilfe –.

### 3.1 Recht auf informationelle Selbstbestimmung (Datenschutz)

In Deutschland ist der Schutz personenbezogener Daten im Recht auf informationelle Selbstbestimmung, Art. 2 Abs. 1 i. v. m. Art. 1 Abs. 1 GG, verankert.<sup>3</sup> Es ist nicht als eigenes Grundrecht geregelt, sondern wurde erst später vom Bundesverfassungsgericht entwickelt. Es ergibt sich aus dem Zusammenspiel von allgemeiner Handlungsfreiheit, Art. 2 Abs. 1 GG, und der Menschenwürde, Art. 1 Abs. 1 GG. Es ist entsprechend seiner Herkunft stark von der Rechtsprechung des Bundesverfassungsgerichts geprägt. Daneben bestehen noch datenschutzrechtliche Vorgaben aus dem Europarecht und eine Absicherung über die Europäische Menschenrechtskonvention.

Personenbezogene Daten sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbarer natürlicher Person, vgl. § 3 Abs. 1 Bundesdatenschutzgesetz (BDSG). Dabei sprechen die Juristen von personenbezogenen Daten, meinen damit aber eigentlich die in den Daten enthaltenen personenbezogenen Informationen.

Kerngedanke hinter dem Schutz ist die Überzeugung, dass sich eine Person, die nicht genau abschätzen kann, wer in ihrer Umwelt was und wieviel über sie weiß, sich in einer Gesellschaft nicht ungewollt und frei bewegen kann.<sup>4</sup>

Das geltende Datenschutzrecht geht im Anschluss an das Volkszählungsurteil des Bundesverfassungsgerichts davon aus, dass es „unter den Bedingungen der automatischen Datenverarbeitung kein ‚belangloses‘ Datum“ mehr geben kann.<sup>5</sup>

Personenbezogene Informationen werden infolgedessen als Informationen über einen Menschen angesehen, deren Erheben, Verarbeiten und Nutzen grundsätzlich eingeschränkt oder ganz verhindert werden muss. Ausgangspunkt ist die Überzeugung, dass personenbezogene Daten etwas sehr Gefährliches für den Menschen darstellen.

Der Einzelne soll „grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten [...] bestimmen.“<sup>6</sup>

Andererseits hat das Bundesverfassungsgericht gesehen, dass es im gesellschaftlichen Miteinander nicht ohne diese Daten geht.<sup>7</sup>

Wie mit personenbezogenen Daten im Einzelnen umzugehen ist, das regelt das Bundesdatenschutzgesetz. Dabei richtet es sich an diejenigen, die Daten erheben, verarbeiten und nutzen. Die Person, von der die Daten in rechtmäßiger Weise erhoben wurden, ist dann weitestgehend Zuschauer in eigener Sache. Der Gesetzgeber bestimmt im Wesentlichen, was geschehen darf und was nicht.

<sup>3</sup>Vgl. *dass.*, Urteil v. 15. Dez. 1983 (1 BvR 209/83) BVerfGE 65, 1-71 (41 ff.) – Volkszählungsurteil –.

<sup>4</sup>Ebd., 1-71 (42 f.) – Volkszählungsurteil –.

<sup>5</sup>Ebd., 1-71 (45) – Volkszählungsurteil –.

<sup>6</sup>Ebd., 1-71 (43) – Volkszählungsurteil –.

<sup>7</sup>Ebd., 1-71 (44) – Volkszählungsurteil –.

### 3.2 Informationsfreiheit

Die Informationsfreiheit (Art. 5 Abs. 1 Satz 1 GG) wird vom Bundesverfassungsgericht als ebenso wichtig für den Meinungsbildungsprozess eingeschätzt, wie die gleichfalls in Art. 5 Abs. 1 GG geregelte Meinungs- und Pressefreiheit. Sie ist gleichermaßen konstituierend für die freiheitliche Demokratie.<sup>8</sup> Voraussetzung ist, dass es sich um eine Quelle handelt, die geeignet ist, dass sich aus ihr ein nicht individuell bestimmbarer Personenkreis Informationen verschaffen kann.<sup>9</sup>

Klassischer Weise wird die Informationsfreiheit anders als oben angedeutet nicht als Ausdruck einer objektiven Wertordnung verstanden, sondern als reines Abwehrrecht gegenüber dem Staat.<sup>10</sup> Es schützt also nur, wenn der Staat den Zugang zu vorhandenen allgemeinen zugänglichen Quellen einschränkt, gibt aber keinen Anspruch auf die Schaffung neuer Informationsquellen.<sup>11</sup>

Ein Anspruch auf Schaffung frei zugänglicher Informationsquellen kann sich allerdings aus dem Europarecht oder dem einfachen Recht (Informationsfreiheitsgesetze) ergeben.

Das Internet ist eine allgemein zugängliche Quelle im oben skizzierten Zusammenhang.<sup>12</sup>

Besteht eine allgemein zugängliche Informationsquelle personenbezogene Daten, so genießt auch diese Quelle den Schutz des Art. 5 Abs. 1 Satz 1 GG. Der Staat ist also zunächst daran gehindert, allgemein zugängliche Quellen, die personenbezogene Daten enthalten, generell zu verbieten oder ihre Löschung anzuordnen (Abwehrrecht der Informationsinteressierten). Dies gerät in einen gewissen Konflikt mit dem staatlichen Schutzauftrag im Bereich der informationellen Selbstbestimmung, lässt sich aber bewältigen.<sup>13</sup>

### 3.3 Ausgleich zwischen Informationsfreiheit und informationelle Selbstbestimmung

Damit stehen auf der einen Seite die Interessen der Allgemeinheit, sich aus allgemein zugänglichen Quellen zu informieren, und auf der anderen Seite das Recht auf informationelle Selbstbestimmung. Wenn zwei Rechte mit Verfassungsrang zum Ausgleich gebracht werden sollen, so geschieht das bei Juristen durch den Versuch, für

<sup>8</sup>BVerfG, Beschluss v. 3. Okt. 1969 (1 BvR 46/65) BVerfGE 27, 71-88 (81) – Leipziger Volkszeitung –.

<sup>9</sup>Ebd., 71-88 (83) – Leipziger Volkszeitung –.

<sup>10</sup>Bethge, in: *Sachs*<sup>6</sup>, Rn. 59.

<sup>11</sup>BVerfG, Urteil v. 24. Jan. 2001 (1 BvR 2623/95) BVerfGE 103, 44-81 (59) – ntv-Entscheidung –; anders demgegenüber *Caspar*, in: *Informationsfreiheit und Informationsrecht*, 297–304 (300 ff.); *Mutius*, in: *Informationsfreiheit und Informationsrecht*, 45–53 (48 f.).

<sup>12</sup>Starck, in: *Mangoldt*<sup>6</sup>, Rn. 45.

<sup>13</sup>Vgl. dazu auch *Schoch*, in: *Informationsfreiheit und Informationsrecht*, 123–155 (126 ff.).

beide Rechte eine Interpretation zu finden, mit deren Hilfe beide zu maximaler Geltung gelangen (Herstellung praktischer Konkordanz).

Ausdruck dieser juristischen Abwägung sind § 28 Abs. 1 Satz 1 Nr. 3 und § 29 Abs. 1 Satz 1 Nr. 2 BDSG. Sie erlauben der nicht öffentlichen Stelle, für ihre Geschäfte aus allgemein zugänglichen Quellen personenbezogene Daten zu gewinnen, es sei denn, der Verwendung solcher Daten stehen schutzwürdige Interessen des Betroffenen gegenüber. Auch die Informationsfreiheitsgesetze enthalten einen Abwägungsvorbehalt zu Gunsten des Datenschutzes, vgl. z. B. 5 IFG. Das Abwägungsgebot der Verfassung wurde hier also ins einfache Recht übertragen. Die konkrete Anwendung bleibt eine Frage des Einzelfalls und ist damit für beide Seiten unter dem Gesichtspunkt der Rechtsklarheit letztlich unbefriedigend. Wessen Interesse nun überwiegt, kann im Streitfall nur ein Gericht am konkreten Fall entscheiden.

## **4 Was sagt die Ökonomik?**

Das Grundmodell der Ökonomik ist der homo oeconomicus – eine Modellierung menschlichen Verhaltens, das im Wesentlichen durch das zweckrationale Eigeninteresse geprägt wird. Rational handeln heißt nach diesem Grundmuster in erster Linie den eigenen Vorteil suchen. Ein Ziel mit möglichst wenig Aufwand erreichen oder bei gegebenen Mitteln möglichst viel vom Ziel erreichen. Moralisches Verhalten hat in diesem Modell keinen Platz.

### **4.1 Ökonomisches Interesse an Informationen**

Informationen werden dabei ausschließlich unter Nützlichkeitsgesichtspunkten betrachtet. Ob es sich dabei um personenbezogene Daten oder Geodaten handelt ist völlig unerheblich, entscheidend ist, was zur Zielerreichung benötigt wird.

Der Trend hin zu Open (Government) Data hat aus Sicht des homo oeconomicus nichts mit staatlicher Transparenz oder echter Teilhabe zu tun. Für ihn ist entscheidend, dass er nun über Informationen verfügen kann, die er sonst nicht oder zu wesentlich schlechteren Konditionen, z. B. hohe Erwerbskosten, erhalten hätte. Diese freie Verfügbarkeit von Informationen versetzt ihn in die Lage, Innovationen hervorzubringen, von denen er früher aus Kostengründen abgesehen hätte. Dieser Aspekt zeigt, eigennütziges Verhalten kann, muss aber nicht gut für die Gesellschaft sein.

Für den homo oeconomicus ist das Recht auf Informationsfreiheit nur insoweit relevant, als es ihm kostengünstige Informationsquellen gewährt.

## 4.2 Ökonomisches Interesse an personenbezogenen Daten

Wie verhält sich das Ganze nun im Hinblick auf personenbezogene Daten? Diese Informationen haben, wenn man etwas verkaufen will, besonderen Wert – dank ihrer Hilfe kann man versuchen, das menschliche Verhalten zu analysieren, um so potentielle Kunden in der Masse zu identifizieren oder sogar das Verhalten von Kunden vorherzusagen und ggf. zu beeinflussen. Hier sind die Betätigungsfelder von Marketing, Verhaltensökonomie und Spieltheorie.

Aus Sicht des homo oeconomicus ist es also vernünftig, so viele und so lange (personenbezogene) Informationen zu sammeln, wie es seinem individuellen Nutzen zuträglich ist.

Rechtliche Ge- und Verbote, wie beispielsweise der Datenschutz, stellen sich dabei als sogenannte Transaktionskosten dar. Es handelt sich hierbei um verschiedenste Kostenfaktoren, die auf dem Weg zum Ziel zu kalkulieren sind. Der homo oeconomicus überlegt sich dabei auch, ob es für ihn (finanziell) günstiger ist, die datenschutzrechtlichen Regeln einzuhalten, die Sanktionierung eines Verstoßes zu bezahlen oder in ein Land mit niedrigeren datenschutzrechtlichen Standards auszuweichen. Natürlich bezieht er auch Überlegungen mit ein, wie sein Gegenüber, über das er die personenbezogenen Daten erhebt, reagiert. Dies setzt natürlich voraus, dass das Gegenüber überhaupt weiß, dass es beobachtet wird. Wäre dies nicht der Fall, so würde man von einer Informationsasymmetrie sprechen.<sup>14</sup> Diesen Fall gibt es aber nur in einem vom klassischen ökonomischen Modell abweichenden Ansatz. Das klassische Modell des homo oeconomicus geht von vollkommen informierten Marktteilnehmern aus. Dort würde sich nur die Frage stellen, ob personenbezogene Daten für beide Seiten einen Preis haben oder nur für eine Seite. Die Frage nach einem ökonomischen Wert von personenbezogenen Daten für diejenigen, über die sie erhoben werden, ist bisher empirisch nicht eindeutig beantwortet.<sup>15</sup> Es handelt sich hierbei auch um eine besonders schwierige Fragestellung, da auch die Grenze zwischen privatem und öffentlichem Bereich ambivalent ist und nicht genau feststeht.<sup>16</sup>

Es kommt also darauf an, ob Personen bei der Erhebung personenbezogener Daten nur ein ungutes Gefühl haben, die Erhebung aber ohne weiteres akzeptieren, ob sie mit der Erhebung einverstanden sind, weil sie dafür Gegenleistung unentgeltliche Internetdienstleistungen bekommen<sup>17</sup> oder ob sie bereit sind, Geld zu bezahlen, wenn dafür keine personenbezogene Daten erhoben werden. Man bewegt sich hier im ökonomischen Bereich der Handlungs- oder Verfügungsrechte (Property Rights)<sup>18</sup>

Im ersteren Fall handelt es sich aus Sicht des homo oeconomicus lediglich um externe Effekte, die für ihn keinerlei Bedeutung haben.<sup>19</sup> In den anderen beiden Fällen geht es

<sup>14</sup>Vgl. Richter/Furubotn, Neue Institutionenökonomik, S. 100 f.

<sup>15</sup>Vgl. Hess/Schreiner, DuD 2012, 105–109 (108 f.).

<sup>16</sup>Vgl. Rössler, Der Wert des Privaten, S. 305 f.

<sup>17</sup>Vgl. auch Caspar, in: Informationsfreiheit und Informationsrecht, 297–304 (297).

<sup>18</sup>Vgl. Schäfer/Ott, Ökonomischen Analyse des Zivilrechts, S. 589 f.

<sup>19</sup>Richter/Furubotn, Neue Institutionenökonomik, S. 90.

dagegen um eine Frage des Marktpreises. Der zweite Fall wird problematisch, wenn sich aufgrund von Monopolstrukturen (Marktversagen) kein Markt bilden kann, auf dem für Geld der Verzicht auf personenbezogene Daten angeboten wird.

## 5 Personenbezogene Daten in Recht und Ökonomik

Während die Ökonomie als sozialwissenschaftliche Disziplin in erster Linie beschreibend und erklärend tätig wird (mit gewissen normativen Tendenzen), hat das Recht von vornherein einen regelnden und gestaltenden Anspruch. Dabei können sozialwissenschaftliche Erkenntnisse bei der Ausgestaltung des Rechts von großem Vorteil sein, um einer Norm auch zur faktischen Anerkennung und Durchsetzung zu verhelfen. Andererseits kann die Rechtswissenschaft durch Gesetz und Rechtsprechung versuchen, auf die soziale Wirklichkeit einzuwirken, um so unerwünschte Effekte zu beseitigen.

So kann ein Gesetz auf die Beseitigung von Informationsassymetrien hinwirken, indem es Informationspflichten normiert. Diesen Effekt hat auch § 4 Abs. 1 BDSG, indem er die Einwilligung des Betroffenen in die Datenerhebung verlangt und § 33 Abs. 1 BDSG, der eine echte Informationspflicht beinhaltet.

Der Gesetzgeber kann außerdem rechtliche Gestaltungen wählen, mit deren Hilfe externe Effekte internalisiert werden.<sup>20</sup> Im Bereich des Emissionshandels werden solche Modelle bereits umgesetzt. Hier sollen die ansonsten von der Allgemeinheit zu tragenden Kosten der Umweltverschmutzung in Form von Emissionszertifikaten von den verursachenden Unternehmen getragen werden.

Für den Bereich des Datenschutzes kommt ein solches Modell nach geltender Rechtslage nicht in Betracht. Ungeachtet der derzeit ungeklärten Frage, ob personenbezogene Daten auch für den Menschen einen ökonomisch messbaren Wert haben, wäre der Gesetzgeber in der Lage, derartige Informationen mit einem solchen Wert zu versehen.<sup>21</sup>

Hierfür fehlen aber bisher die Voraussetzungen. Wie eingangs festgestellt, schützt die Rechtsordnung Informationen nur ausnahmsweise als ökonomischen Wert (z. B. Urheberrecht, Patentrecht). Es fehlt die Anerkennung jedes einzelnen personenbezogenen Datums als vermögenswertes Recht. Zwar muss der Betroffene in die Datenerhebung einwilligen, § 4 Abs. 1 BDSG, darf Auskunft über gespeicherte Daten begehren, § 34 BDSG, und kann unter bestimmten Voraussetzungen eine Berichtigung, Löschung oder Sperrung der Daten verlangen, § 34 BDSG. Schadensersatzansprüche oder das Recht, Gewinne herauszuverlangen, die mit unrechtmäßig erhobenen Daten erzielt wurden, hat er jedoch nicht.

---

<sup>20</sup>Ebd., S. 109 f.

<sup>21</sup>Ebd., S. 109 f.

Der Gesetzgeber hat sich im Bundesdatenschutzgesetz bzw. in den Informationsfreiheitsgesetzen für die Durchsetzung des Rechts auf informationelle Selbstbestimmung und Informationsfreiheit mittels Regulierung und Kontrolle der Datenverwender entschieden. Eine wichtige Rolle nehmen in diesem Kontrollsystem die Datenschutzbeauftragten ein. Dieses Kontrollmodell steht und fällt jedoch mit der Ausgestaltung der Position der Kontrolleure.

Die mit der Verfassung garantierten Grundrechte stellen den Schutz des Menschen in den Mittelpunkt. Die Verknüpfung der informationellen Selbstbestimmung mit der Menschenwürde macht dies zu einer wichtigen staatlichen Aufgabe. Sie kann auch nicht (nur) dem Einzelnen überlassen werden, weil es keine Ausgestaltung der Information über eine Person im Sinne eines ökonomischen Handlungsrechts gibt. Ob ein echtes „Recht an den eigenen Informationen“ im Sinne eines absoluten Verfügungsrechts in das gegenwärtigen Datenschutzrecht eingefügt werden könnte, muss eingehenderen Untersuchungen vorbehalten bleiben.

Im Augenblick sieht es danach aus, als würde der informationshungrige homo oeconomicus trotz aller rechtlichen Regeln im Wettlauf um (personenbezogene) Daten das Feld anführen und seinen Vorsprung gegenüber den Personen, die diese Daten liefern, weiter ausbauen.

## Literatur

Caspar, Johannes, Informationsfreiheit als Verfassungsgrundrecht – Analyse und Argumente für ein Grundrecht auf staatliche Transparenz, in: Dix, Alexander u. a. (Hrsg.), Informationsfreiheit und Informationsrecht. Jahrbuch 2011, Berlin 2012, S. 297–304, *zitiert als: Caspar, Informationsfreiheit und Informationsrecht.*

Hess, Thomas / Schreiner, Michael, Ökonomie der Privatsphäre, DuD 2012, S. 105–109.

Mangoldt, Hermann von (Hrsg.), Kommentar zum Grundgesetz, Bd. 1, 6. Aufl., München 2010, *zitiert als: Bearbeiter, in: Mangoldt<sup>6</sup>.*

Mutius, Albert von, Deutschland auf dem Weg zu einem Informationsgrundrecht?, in: Dix, Alexander u. a. (Hrsg.), Informationsfreiheit und Informationsrecht. Jahrbuch 2010, Berlin 2010, S. 45–53, *zitiert als: Mutius, Informationsfreiheit und Informationsrecht.*

Richter, Rudolf / Furubotn, Eirik G., Neue Institutionenökonomik. Eine Einführung und kritische Würdigung, 3. Aufl., Tübingen 2003, *zitiert als: Richter / Furubotn, Neue Institutionenökonomik.*

Rössler, Beate, Der Wert des Privaten, Frankfurt am Main 2001, *zitiert als: Rössler, Der Wert des Privaten.*

Sachs, Michael (Hrsg.), Grundgesetz. Kommentar, 6. Aufl., München 2011, *zitiert als: Bearbeiter, in: Sachs<sup>6</sup>.*

Schäfer, Hans-Bernd / Ott, Claus, Lehrbuch der ökonomischen Analyse des Zivilrechts, 5. Aufl., Berlin, Heidelberg 2012, zitiert als: Schäfer / Ott, Ökonomischen Analyse des Zivilrechts.

Schoch, Friedrich, Informationsfreiheit versus Datenschutz, in: Dix, Alexander u. a. (Hrsg.), Informationsfreiheit und Informationsrecht. Jahrbuch 2012, Berlin 2013, S. 123–155, zitiert als: Schoch, Informationsfreiheit und Informationsrecht.



# Open Source Enterprise Resource Planning (OSS-ERP) und der Mittelstand - Eine schier unlösbare Aufgabe ?

**Frederik Kramer und Markus Schneider**

{frederik.kramer}@ovgu.de, {markus.schneider}@initos.com

<http://www.mrcc.eu> <http://blog.initos.com>

Das Interesse an Open Source Enterprise Resource Planning (OSS-ERP) steigt kontinuierlich. Zwar fallen keine Lizenzkosten an, die Einführung ist jedoch komplex und damit in der Regel kostenintensiv. Es müssen Anforderungen aufgenommen, Geschäftsprozesse dokumentiert und implementiert, Rollenkonzepte erarbeitet, Schulungen vorbereitet, Support sichergestellt und zusätzliche Anwendungssysteme angebunden werden. In Anbetracht dieser Aufwände, treten wichtige Vorteile von OSS-ERP wie Flexibilität und Vermeidung von Anbieterabhängigkeiten (sogenannten Lock-Ins) häufig in den Hintergrund. Der vorliegende Beitrag diskutiert die in kleinen und mittleren Unternehmen (KMU) auftretenden Probleme und diskutiert Lösungsansätze zu deren Vermeidung.

## 1 Einleitung

Das Akronym „SAP“ ist jedem Fachmann der mit Informationstechnologie im betrieblichen Kontext zu tun hat ein Begriff. Dies liegt nicht nur am Gewicht der SAP AG in ihrem Markt, sondern insbesondere auch an der Enterprise Resource Planning (ERP) Lösung SAP ERP. Diese hat eine sehr hohe Marktdurchdringung von derzeit knapp 25% weltweit<sup>1</sup>.

Waren insbesondere teure Hardware und die elektronische Steuerung der Unternehmensprozesse mit diesen ERP-Systemen zu Beginn der 1970er Jahre großen und damit kapitalstarken Unternehmen vorbehalten, so gerät diese Art der Unternehmenssteuerung im Zuge der Standardisierung von Hardware und der Verbreitung des Internet immer stärker auch in den Fokus von kleinen und mittleren Unternehmen (KMU).

Die Aufgaben, Problemstellungen und Lösungsstrategien zur Einführung großer betrieblicher Anwendungssysteme sind für Großunternehmen mindestens seit der Entstehung der Forschungsdisziplin Wirtschaftsinformatik Gegenstand entsprechender Forschung. Für KMU sind deren Einführung, Erfolgsfaktoren und Risiken jedoch weit weniger intensiv erforscht. Deshalb kommt es bei der Anwendung üblicher, aus der Praxis von Großunternehmen bekannter Methodiken bei KMU immer wieder zu Problemen, die im Rahmen dieses Beitrags diskutiert werden [3].

---

<sup>1</sup>[http://www.aktiencheck.de/exklusiv/Artikel-SAP\\\_Aktie\\\_Greift\\\_Microsoft\\\_Walldorfer\\\_Software\\\_Schmiede-5102242](http://www.aktiencheck.de/exklusiv/Artikel-SAP\_Aktie\_Greift\_Microsoft\_Walldorfer\_Software\_Schmiede-5102242), zugegriffen am 11.01.2014

Dieser Beitrag verfolgt das Ziel die Unterschiede zwischen Großunternehmen und KMU hinsichtlich der Auswahl und des Einsatzes von betrieblicher Standardsoftware zu untersuchen. Insbesondere verfolgt er dabei das Ziel die Probleme von KMU mit dem aus Großunternehmen bekannten Standardvorgehen zu illustrieren. Mit dem Kick-Start Vorgehen wird ein Lösungsansatz zur Verringerung / Vermeidung dieser Problem vorgestellt von dem die Autoren glauben, dass dieser die offensichtlichsten Probleme lösen kann ohne zu große Risiken in Kauf zu nehmen. Auf eine fallstudienbasierte oder gar großzahlige empirische Validierung des Vorschlags wird jedoch zu Gunsten einer kondensierteren Darstellung und logischer Argumentation verzichtet. Auch wenn die Argumentationsgrundlagen und vor allem beobachteten Fälle in dem vorliegenden Beitrag nicht explizit dargestellt sind, so kann der Leser von einer großen Zahl begleiteter Auswahlverfahren in Großunternehmen und KMU ausgehen, die Grundlage für die Argumentation in diesem Beitrag sind.

## 2 Mittelständische Unternehmen

Es gibt eine Vielzahl von Mittelstandsdefinitionen [1, 2]. Praktisch jedes Land und jeder Kontinent hat andere Gepflogenheiten bei der Definition des Begriffes. In Deutschland sind drei Definitionen gebräuchlich. Einerseits sind dies die in Paragraph 267 des Handelsgesetzbuches (HGB) beschriebenen Größenklassen der kleinen und mittelgroßen Kapitalgesellschaft, die Definition des Instituts für Mittelstandsforschung (IfM) in Bonn und die der Europäische Kommission. Während die Definitionen des HGB und der EU als Mittelstand Unternehmen bezeichnen, die nicht mehr als 250 bzw. 249 Mitarbeiter haben, schließt die Definition des IfM auch noch solche mit nicht mehr als 499 Mitarbeitern ein. Daneben wird auch je eine Grenze für Umsatz und / oder Bilanzsumme festgelegt. Für die IfM Definition sind das zum Beispiel 50 Mio. Euro Umsatz oder weniger.

Der Mittelstandsbezug des IfM umfasst neben den quantitativen auch qualitative Aspekte. Dazu gehört zum Beispiel die Einheit von Eigentum, Leitung, Haftung und Risiko bei dem / den Entscheidungsträgern [2]. Auch wenn sich die Definitionen unterscheiden, so macht der zahlenmäßige Unterschied zwischen der Anwendung der einen oder anderen Definition nur einen kleinen Anteil an der Grundgesamtheit der Unternehmen aus. D.h. der Anteil mittelständischer Unternehmen an der deutschen Volkswirtschaft zum Beispiel liegt unabhängig von der konkreten Definition bei über 99%. Die Besonderheiten des Mittelstandes liegen also nicht alleine an quantitativen Kenngrößen und auch nicht an den wenigen qualitativen Eigenschaften, die zusätzlicher Gegenstand der IfM Definition sind. Viel mehr ist jedes Unternehmen durch seine spezielle Struktur gekennzeichnet. Unstrittig ist die Rolle des Unternehmers. Er hat in der Regel wesentlichen Einfluss auf alle strategischen Entscheidungen des Unternehmens.

### 3 Einführung betrieblicher Anwendungssysteme

Die Einführung betrieblicher Anwendungssysteme in großen Unternehmen erfolgt in der Regel mittels planmäßigem Vorgehen. Egal ob es sich dabei um die Einführung von Produktivitätswerkzeugen wie zum Beispiel Textverarbeitungs- oder Tabellenkalkulationslösungen, um Betriebssysteme wie Windows und Linux oder um große betriebliche Anwendungssysteme wie ERP- oder CRM-Systeme handelt. In großen Unternehmen gibt es üblicherweise klare Regeln und Bewertungsmaßstäbe für die Einführung neuer Software. Den Mitarbeitern ist es häufig per Arbeitsvertrag untersagt selbst Software zu installieren. Eine Sonderrolle spielt hier Open Source Software, weil sie insbesondere von den IT-Abteilungen gerne genutzt wurde und wird, um Probleme des Alltagsbetriebs zu lösen für die die Beschaffung einer proprietären Software schlicht zu lange dauert (wegen der strikten Regelung). Der dadurch entstehende Wildwuchs ist ein Problem für Großunternehmen geworden, dem jedoch nicht Gegenstand diese Beitrags sein soll.



Abbildung 1: Klassischer Auswahlprozess

In Großunternehmen stellt in der Regel der Fachbereich, also eine Abteilung, die direkt mit der primären Leistungserstellung betraut ist oder das Management oder eben der IT-Betrieb selbst eine entsprechende Anforderung bzw. wird diese Anforderung aus der IT-Strategie bzw. der Unternehmensstrategie abgeleitet. Darauf werden verschiedene Alternativen ermittelt und im Anschluss einer entsprechenden u.U. mehrstufigen Bewertung unterzogen. Diese Bewertung erfolgt zumeist quantitativ (z.B. ROI, TCO) und häufig auch qualitativ (unformal, semiformal, formal). Auf Basis der zuvor festgelegten Entscheidungsmethodik wird dann eine Einführungsentscheidung zu Gunsten einer Alternative getroffen, ggf. Anforderungen modifiziert und Alternativen reevaluiert oder die Einführung verschoben (siehe Abbildung 1). Zwar sieht das Vorgehen der Sache nach auch in KMU nicht erheblich anders aus (Anforderung, Alternativenbestimmung, Alternativenbewertung, Entscheidung), allerdings sorgen verschiedene, für KMU charakteristische Restriktionen dafür, dass das Vorgehen nicht mit der selben Güte und Stringenz umgesetzt werden kann und deshalb vermeidbare Risiken entstehen.

### 4 Charakteristische Besonderheiten des Mittelstandes

KMU sind durch einige charakteristische Besonderheiten gekennzeichnet [4], die obgleich der grundlegende Prozess der Einführung betrieblicher Anwendungssoftware

ähnlich verläuft, andere Risiken hervorrufen bzw. den Ausgang des Vorhabens beeinflussen. KMU leiden an notorischer Knappheit von Ressourcen [5]. Dies betrifft den Zugang und die Verfügbarkeit zu gut ausgebildetem Personal und insbesondere auch die Verfügbarkeit von Spezialkenntnissen und finanziellen Mitteln zur Durchführung strategischer Projekte. Während in Großunternehmen die IT üblicherweise die Rolle einer eigenen Unternehmensfunktion mit eigener Leitung, Organisation und Mittelausstattung inne hat, obliegt sie in vielen KMU entweder dem technisch versiertesten Mitarbeiter, dem innovativen und experimentierfreudigen Unternehmer oder einem externen Dienstleister. Verantwortlichkeiten im Kontext der IT sind oft unklar verteilt. Strategische Planung bzw. auch nur sinnvolle Budgetierung erfolgt eher selten [6]. Sowohl was die Kenntnis der potentiell für eine Anforderung zur Disposition stehenden Alternativen, als auch die geeignete Entscheidungsmethodik angeht, fehlen oft entsprechende Kenntnisse. Alternativen werden eher durch Empfehlung befreundeter Unternehmer oder Mitarbeiter in den Entscheidungsprozess aufgenommen, als auf Basis zuvor festgelegter Auswahlkriterien.

Oft fehlt auch die grundlegende Akzeptanz zur Anwendung einer Auswahlmethodik. Zum einen ist die Anwendung einer solche Auswahlmethodik komplexer als das Treffen reiner Bauchentscheidungen, zum anderen sorgt bereits die Anwendung einer solchen Methodik für Personal- und Sachaufwand (Recherche / Tests / Evaluierung). Während diese Art der Auswahl in Großunternehmen üblicher Beratungsstandard ist, ist die Akzeptanz formaler Planung im Mittelstand vergleichsweise gering. Im Allgemeinen sind KMU durch einen operationalen Fokus gekennzeichnet. D.h. die kurzfristige Aufgabenerfüllung sowie Planung und Umsetzung (in Tages- oder Wochenscheiben) stehen gegenüber langfristiger, abstrakter und damit strategischer Planung in tendenziell unausgewogenem Verhältnis. Die durch den operativen Fokus von KMU gewährleistete Flexibilität wird im Allgemeinen als wichtige Eigenschaft von KMU angesehen.

Leider sorgen der operative Fokus und die mangelhafte Akzeptanz formaler Planungsmethodik, sowie das Fehlen der Kenntnisse über entsprechende Planungstechniken auch für einen vergleichsweise schlechten Dokumentationsstand der betrieblichen IST-Situation. Sowohl auf Ebene der Prozesse sowie auch auf Ebene der Anwendungssysteme finden sich in KMU damit äußerst selten Dokumentationen. Die konkreten Ausprägung der oben genannten Eigenschaften eines KMU sind erheblich von der Person des Unternehmers / der Unternehmerin abhängig. So liegt es an der, in der Regel durch die Unternehmensleitung bestimmten, Art der Verteilung von Verantwortung und Entscheidungsbefugnissen, deren Bereitschaft Geld in Personal und formale Planung zu investieren bzw. auch entsprechend gut geschultes Personal / Dienstleister am Markt zu beschaffen, ob die Auswahl- und Einführung betrieblicher Anwendungssysteme effizient erfolgt oder nicht. Bei den über 99% Anteil der Unternehmen, die KMU im Euroraum ausmachen, sind praktisch alle Branchen vertreten.

Zwar lassen sich für einzelne Branchen Standardprozesse definieren und damit entsprechend Standardsoftwareprodukte entwickeln, allerdings geschieht dies eher auf Grund regulatorischer Vorgaben, wie etwa des Herkunftsnachweises bei Lebensmit-

tel, als auf Grund ökonomischer Bestrebungen des Unternehmens oder entsprechender Branchen. Auch wenn der Anteil an administrativen Kosten mit dem Grad der Standardisierung, der Unternehmensgröße und auf Grund von Lernkurveneffekte abnimmt, rechtfertigt dies in vielen KMU noch keine strategische Auswahl von betrieblichen Anwendungssystemen ausschließlich für die Supportaktivitäten (z.B. Einkauf und Buchhaltung). Erst wenn der üblicherweise weitaus größere Teil der Unternehmensbelegschaft, der direkt mit dem Kern der betrieblichen Leistungserstellung in Kontakt kommt, Ausschussraten und Durchlaufzeiten verringern, Beratungsleistung verbessern oder Kunden- und Auftragsinformationen schneller verarbeiten kann und die wettbewerbsrelevanten und zuvor standardisierten Besonderheiten des KMU in einem flexiblen betrieblichen Anwendungssystem integriert werden können, rechtfertigt dies die Kosten eines mit Großunternehmen vergleichbaren Auswahlprozesses.

## **5 Einfluss auf den Einführungsprozess großer betrieblicher Anwendungssysteme**

Die im Abschnitt charakteristische Besonderheiten von KMU (siehe Abschnitt 4) beeinflussen den Auswahlprozess betrieblicher Anwendungssysteme (siehe Abbildung 1) an verschiedenen Stellen. So bestehen zum Beispiel erhebliche Schwierigkeiten Anforderungen in geeigneter Weise, d.h. auf die Auswahl von Anwendungssystemen bezogen zu formulieren. Zu beobachten ist daneben, dass selbst zwei Mitarbeiter, die grundsätzlich ein ähnliches oder sogar identisches Aufgabenspektrum haben, Anforderungen unterschiedlich formulieren bzw. sogar unterschiedliche Anforderungen an ein Anwendungssystem haben. KMU verfügen in der Regel über keine oder wenige Kenntnisse im Bereich des Requirements Engineering, so dass die fachliche Formulierung (z.B. Vorbedingungen, Änderungszustände, Akteure, System, Nachbedingungen) oder sogar die Modellierung von Anforderungen (z.B. UML / BPMN / EPK) extern beauftragt werden müsste. Dies geschieht aus Kostengesichtspunkten leider oft nicht. Zudem ist festzustellen, dass viele IT-Dienstleister und sogar Anbieter von Standardsoftware im KMU Segment über solche Fähigkeiten nicht verfügen. Wie bereits beschrieben sorgt das Fehlen eigenen Fachpersonals dafür, dass Alternativen erst gar nicht Gegenstand des Auswahlverfahrens werden, obwohl diese grundsätzlich geeignet wären. Dies liegt unter anderem auch daran, dass es im KMU-Segment deutlich mehr sogenannter Standardsoftware insbesondere aus dem proprietären Spektrum gibt, als das beispielsweise für die Großunternehmen der Fall ist.

Ein weiterer, deutlich zu beobachtender Effekt ist, dass die Güte der branchenspezifischen Softwaredemonstrationen, Marketing (Präsentation, Flyer, Anzeigen, Websites etc.), ein möglichst niedriger Preis, und die Empfehlung von bekannten Unternehmen, sowie letztlich ein professionell erscheinendes Auftreten des Verkaufspersonals ein höheres Entscheidungsgewicht zu haben scheinen, als die technische Prüfung des Produktes, dessen Flexibilitätsoptionen, die Vergleichbarkeit von angebotenen Leistungen oder die Möglichkeiten der Erweiterung eines Produktes. Zwar ge-

langen immer häufiger eben auch Open Source Produkte und alternative Bezugsverfahren wie Cloud Computing bzw. Application Service Providing in die Auswahl von KMU, die Erwartungshaltung an das Marketing bzw. das Auftreten des Verkaufspersonals ist jedoch das selbe wie bei proprietären Lösungen. Im Detail sorgen die Unkenntnis der besonderen Eigenschaften von Open Source Software (z.B. Zugang zum Source Code, Veränderbarkeit, Lizenzen) bzw. Cloud Computing oder Application Service Providing (drohender Daten Lock-In, fehlende Service Level Agreements) daher dafür dass diese Alternativen gar nicht oder undifferenziert bewertet werden.

Ein durch entsprechendes Fachpersonal und / oder Fertigkeiten unterstützter Auswahlprozess kann diese Risiken verringern. Allerdings werden diese Spezialfähigkeiten im Alltag eines KMU vergleichsweise selten benötigt und sind zudem relativ teuer und veralten schnell. Deshalb leistet sich kaum ein KMU entsprechendes Personal, sondern verlässt sich wenn überhaupt auf externe Dienstleister. Diese Dienstleister müssten jedoch zum einen hinsichtlich der Alternativen, als auch der Bezugs- und Lizenzierungsverfahren sowie insbesondere auch der Auswahlmethodik stets auf dem letzten Stand und vor allem unabhängig sein. D.h. sie fürden nicht auf eine spezielle Lösung speziell fokussiert sein. Der insgesamt schlechte Dokumentationsstand des IST-Zustandes in KMU, die Kosten eines Auswahlverfahrens und der operative Fokus sorgen dafür, dass selbst bei Anwendung eines derartigen Verfahrens und bei vorhandenen fachlich versierten Ressourcen stets erhebliche Risiken eines erfolgreichen Projektabschlusses vorhanden bleiben. Wird auf fachliche Unterstützung und ein formales Verfahren ganz verzichtet, ist die Auswahl zwar u.U. erheblich schneller erledigt aber die Risiken sind u.U. deutlich höher.

## **6 Das Kick-Start Verfahren**

Ein radikaler Ansatz dieses Problem für KMU zu lösen, ist ein Kick-Start Verfahren. Bei diesem Verfahren wird der Auswahlprozess (siehe Abbildung 1) ebenfalls erheblich vereinfacht. Es wird keine detaillierte Anforderungsanalyse mehr durchgeführt, sondern der Umfang des Kick-Start Verfahrens verglichen. Beim Kick-Start Verfahren wird davon ausgegangen, dass bei einem KMU betriebliche Standardfunktion, wie etwa das Erstellen von Rechnungen oder die Beschaffung von Waren zu großen Teilen außerhalb einzelner Anwendungssystemdomänen bzw. auf mehrere Anwendungen verteilt und damit im Verantwortungsbereich einzelner Mitarbeiter und in mehreren Softwareprodukten verteilt liegt. Die Annahme ist, dass das unterstützen, harmonisieren und integrieren einzelner, zuvor ausgewählter Standardtätigkeiten, wie etwa der Angebots- oder Rechnungsstellung einen deutlichen Vorteil hinsichtlich der Effizienz, Transparenz und Datenqualität gegenüber dem Status Quo erzeugt.

Das heißt nicht, dass KMU z.B. bei der Rechnungsstellung zuvor ohne Informationstechnologie ausgekommen sind, aber oft erfolgt das Erstellen einer Rechnung unter Nutzung einer Kontaktdatenbank (z.B. in Exchange), einer Rechnungsvorlage bzw. historischer Rechnungen in einem Textverarbeitungswerkzeug, dem Absenden des

entsprechenden Dokumentes per Post und / oder Email und seiner Ablage auf einem Fileserver sowie der Übermittlung einer Kopie an Steuerberater. Die tatsächlich gelieferten Produkte / erbrachte Leistungen werden zudem oft entweder von Stundenzetteln oder handschriftlichen Notizen übernommen. Ein Medienbruch ist die Folge, d.h. eine Informations bzw. Informationssystemintegration von Angebot, Lieferschein und Rechnung findet oftmals nicht statt. Zwar kann mit diesem Kick-Start Vorgehen in der Regel noch keine ganze Arbeitskraft abgelöst werden, aber die Arbeit kann erheblich vereinfacht und der Mitarbeiter für andere, potentiell höherwertige Tätigkeiten entlastet werden. Ohnehin ist die Einsparung von Personal in KMU weit weniger häufig das Optimierungsziel als zum Beispiel in börsennotierten Großunternehmen.

Bei einem solchen Kick-Start Vorgehen wird das letztendliche Projektziel analog zur agilen Softwareentwicklung (z.B. mittels SCRUM) nicht mehr vorher detailliert geplant. Stattdessen wird eine Grundausrüstung von einfachen Funktionen wie oben beschrieben zum festen, vorher genau definierten Preis und im genauen zeitlichen Rahmen installiert und auf die niedrigsten Erfordernisse des jeweiligen Kunden beschränkt angepasst (z.B. Logoanpassung in Reports). Mitarbeiter werden daraufhin mit einer standardisierten Schulung auf die Arbeit mit dem Anwendungssystem vorbereitet. Diese Grundausrüstung wird möglichst schnell in den produktiven Arbeitseinsatz überführt. Parallel dazu wird ein, gängigen Industriestandards vergleichbares, Supportsystem aufgesetzt, das aus einem First-Level Anwendungssupport durch den mit der Umsetzung des Kick-Start Verfahrens beauftragten Dienstleister und einem Second-Level Applikationssupport durch den Hersteller bzw. die Community besteht.

Der einmalige Aufwand für dieses Kick-Start Verfahren, variiert ausschließlich bezüglich der installierten und initial konfigurierten Module des Anwendungssystem und der Anzahl unterschiedlicher zu schulender Rollen bzw. anzulegender Nutzer. Die Kosten für das Aufsetzen der Hardwareinfrastruktur bleibt im Rahmen der Anwendbarkeit dieses Kick-Start Vorgehens in der Regel gleich. Der regelmäßig wiederkehrende Supportaufwand ist allerdings abhängig von der Anzahl der Nutzer. Lizenzkosten entfallen im Falle von Open Source Software.

Die Weiterentwicklung und Anpassung des Anwendungssystems nach erfolgreich abgeschlossenem Kick-Start Vorgehen, erfolgt desweiteren im Rahmen der Anwendung von Grundsätzen der agilen Softwareentwicklungsmethodik. Insbesondere werden Veränderungswünsche zeitnah zwischen Dienstleister und Kunden abgestimmt und in kurzen Iterationszyklen umgesetzt. Am Ende jeder Umsetzungsphase erfolgt das Einspielen geänderter Versionsstände, so dass ein zeitnaher Feedbackzyklus entsteht, der im Übrigen mittels des bereits am Anfang instanziierten Supportverfahrens abgewickelt wird.

## 7 Kritische Betrachtung

KMU und ERP-Systeme sind kein einfaches Unterfangen. Nicht zuletzt deswegen ist der Markt stark segmentiert und die SAP AG, der Marktführer in Großunternehmen versucht seit mehr als einem Jahrzehnt einen ähnlichen Markterfolg wie in Großunternehmen auch im Mittelstand zu erlangen. Die strukturellen Besonderheiten kleiner und mittlere Betriebe sind in Anbetracht ihres zahlenmäßigen Gewichts und ihrer Verteilung auf alle denkbaren Branchen sehr mannigfaltig.

Ökonomische Beschränkungen wie die Knappheit an Personal, Kapital und Know-How, sowie die Rolle des Unternehmers, vergleichsweise geringe Akzeptanz formaler, langwieriger und damit kostenintensiver Planungsverfahren und der, in der Regel, schlechte Dokumentationsstand des Status Quo erschweren die Einführung betrieblicher Standardsoftware zusätzlich. Zudem erfordern gerade der operative Fokus von KMU und deren hohes Flexibilitätsbedürfnis besonders agile und gute Anpassungsmöglichkeiten.

Zwar ist nicht von Vorneherein ausgeschlossen, das auch proprietäre und / oder als ASP bzw. Cloud angebotene Software dies leisten kann, aber die Besonderheiten des Open Source Entwicklungsprozesses (z.B. Einsicht in den Quellcode, keine Nutzungseinschränkungen, leichte indentifizierbare Verfügbarkeit von Ressourcen und geeigneten Dienstleistern), die weltweite Verteilung der Community der Nutzer und die damit verbundene grundsätzliche Möglichkeit der Vergleichbarkeit von Kosten und Qualität der Dienstleistung sowie verhältnismäßige einfache Austauschbarkeit der Dienstleister macht OSS-ERP Lösungen wie OpenERP, OpenBravo oder Adempiere gegenüber dieser Konkurrenten zu einer sehr attraktiven Alternative.

Das Kick-Start Vorgehen reduziert das ökonomische Risiko gerade für kleine KMU erheblich ohne zu große Risiken in Kauf zu nehmen. Der Einschätzung der Autoren nach, sollten selbst zweifelnde Unternehmer dazu bewegt werden können einen entsprechenden Versuch zu wagen. Dennoch stecken diese Kick-Start Verfahren noch in ihren Anfängen. Zwar haben sie der Einschätzung der Autoren nach das Potential die Einführung Open Source basierter betrieblicher Standardsoftware erheblich zu beflügeln, allerdings wird erst die großzahlige Anwendung zeigen, an welchen Stellen sie wirklich das zielführendste Verfahren darstellen. Schon jetzt scheint klar, dass für einen großen mittelständischen Automobilzulieferer, der über 200 Mitarbeiter Personal und mehr und einen Umsatz von nahe der 50 Mio. Euro verfügt, ein Kick-Start Verfahren kaum gangbar ist. Es ist anzunehmen, dass ein solcher Betrieb zum einen bereits seit vielen Jahren über ein Altsystem verfügt, das abgelöst und dessen Daten möglicherweise migriert werden müssen. Daneben wird er Restriktionen durch seine Abnehmer unterworfen (große Automobilhersteller zwingen mitunter kleinere Zulieferer SAP Anwendungssysteme einzusetzen) oder den Hürden des Change Management erheblich stärker ausgesetzt sein. Allerdings sind solche Betriebe der Definition nach zwar u.U. noch KMU, stellen aber einen verschwindend geringen Teil der Grundgesamtheit der Selbigen dar.

## Literatur

- [1] Karoline Gundermann. Kmu-mittelstand: Theoretische abgrenzung und betrachtung der forschungsaktivitäten in den jeweiligen bereichen. 2009.
- [2] Brigitte Günterberg and Hans-Jürgen Wolter. *Unternehmensgrößenstatistik 2001/2002*. Institut für Mittelstandsforschung Bonn, 2002.
- [3] Frederik Kramer, Jorge Marx Gómez, and Key Pousttchi. *HMD-Praxis der Wirtschaftsinformatik*, volume 285, chapter Enterprise Resource Planning in KMU - eine vergleichende Studie von proprietärer und Open Source Software, pages 20–33. dpunkt.verlag, 2012.
- [4] Pascal Schaumlöffel. *Mittelstand und planung*. 2009.
- [5] Juhan Teder and Urve Venesaar. Strategic management in estonian smes. Working Papers 126, Tallinn School of Economics and Business Administration, Tallinn University of Technology, 2005.
- [6] Calvin Wang, Elizabeth A. Walker, and Janice Redmond. Explaining the lack of strategic planning in smes: The importance of owner motivation. *International Journal of Organisational Behaviour*, 12(1):1–16, 2007.



# Markdown und eine Prise *pandoc*

Kurt Pfeifle <[kurt.pfeifle@mykolab.com](mailto:kurt.pfeifle@mykolab.com)>

Dieser Beitrag gibt eine Einführung in das Textformat *Markdown*. Außerdem deutet er an, wie man mittels des Dokumenten-Konverters *pandoc* dieses einfache Textformat nach ausdrückstärkeren Formaten wie HTML,  $\LaTeX$ , PDF, ODT, EPUB und andere wandelt.

Der Beitrag stellt hierzu diverse praktische Beispiele vor.

---

## 1 Das Kleine Einmaleins von Markdown (mit etwas *pandoc*)

Markdown und *pandoc* sind zwei verschiedene Sachen:

- Markdown ist ein *Format* zum Schreiben von Texten.
- *pandoc* ist ein Kommandozeilen-Tool zum Konvertieren von Markdown-Texten in andere 'reiche' Formate.

Markdown beruht auf reinem (ASCII- oder UTF-8-)Text. Man kann dem Text damit ziemlich leicht bestimmte Auszeichnungen einzelner Elemente mitgeben. Beim Konvertieren des Markdown-Textes in andere Formate erscheinen die markierten Stellen dann als die beabsichtigten Formatierungen. Aufgrund der Einfachheit der Auszeichnungs-Elemente und ihrer fast schon intuitiv erscheinenden Arbeitsweise bleibt das Ursprungs-Dokument auch in der Quelltext-Form weiterhin gut lesbar.

Man stelle sich vor: man soll eine E-Mail schreiben, ohne HTML zu verwenden zu dürfen. *Wie würde man den Email-Text gestalten, um damit gewisse Formatierungs-Absichten auszudrücken? Um Betonungen, Hervorhebungen, Überschriften, Listen usw. darzustellen?*

Eben.

Man hat das doch schon öfters gesehen... oder nicht?!

**Github**<sup>1</sup> übrigens unterstützt Markdown ebenfalls. Viele der dort hinterlegten Software-Dokumentationen sind als README.md vorhanden. Auf **StackOverflow**<sup>2</sup> basiert die Eingabe von Frage- oder Antwort-Beiträgen gleichermaßen auf Markdown.

Wer einmal etwas vertraut mit dem Schreiben von Markdown und dem Umgang mit *pandoc* geworden ist, kommt schnell in die Versuchung, seine gesamte Produktion von größeren Dokumenten auf diese beiden Komponenten zu stützen. Denn damit kann

---

<sup>1</sup><http://github.org/>

<sup>2</sup><http://stackoverflow.com/>

man in Windes-Eile sehr viele verschiedene Ausgabe-Formate erzeugen, alle aus ein und demselben Quelltext.

Markdown-Quelltext und `pandoc` versetzen selbst Anwender, die keinerlei Ahnung von  $\LaTeX$  haben, in die Lage, mit dem Erstellen dieses anspruchsvollen Dokumenten-Formats erfolgreich zu beginnen. Selbst fortgeschrittene  $\LaTeX$ -Gurus können enorm profitieren: denn Markdown-Schreiben geht um einiges schneller von der Hand als direkt  $\LaTeX$  zu coden – und die aller-anspruchsvollsten Feinheiten, die `pandoc` und Markdown (noch) nicht abdecken, kann ein Profi im Anschluss an die `Markdown=>LaTeX`-Konvertierung immer noch händisch dem `pandoc`-generierten  $\LaTeX$ -Code beifügen.

## 2 Etwas Markdown-Geschichte

Angefangen hat es 2004 mit einer Initiative von [John Gruber](#)<sup>3</sup>, der dabei die Unterstützung von [Aaron Swartz](#)<sup>4</sup> hatte. Diese wollten eine möglichst einfache Methode schaffen, um aus Text mit simplen Hervorhebungen vollwertiges HTML zu erzeugen. Die beiden legten eine Reihe von Regeln fest, wie diese Hervorhebungen im Ursprungs-Text beschaffen sein sollten. Zugleich erschuf John ein (ebenfalls `markdown` genanntes) Tool, um solchen Text nach HTML zu konvertieren.

Andere Initiative und Projekte erweiterten John's und Aaron's anfängliche Regeln mit der Zeit, um noch mehr Formatierungs-Optionen zu erhalten. Auf diese Weise entstanden bis heute mehrere verschiedene Markdown-Dialekte (die leider z.T. nicht immer miteinander harmonieren). Ebenso kamen neue Konvertierungs-Pfade hinzu: nach HTML, nach  $\LaTeX$  oder nach ODT.

Dabei entstanden im Lauf der Jahre mehrere neue Markdown-Konverter. Der mächtigste dieser Konverter heutzutage ist `pandoc`<sup>5</sup>, geschrieben von [John MacFarlane](#)<sup>6</sup> von der Berkeley University of California.

## 3 `pandoc`

`pandoc` kann nicht nur das ursprüngliche, einfache Markdown verarbeiten:

1. Erstens kommt `pandoc` mit den meisten modernen Markdown-Dialekten ebenfalls sehr gut zurecht.
2. Als *Eingabe* können ebenso HTML,  $\LaTeX$ , DocBook, RST (reStructuredText) sowie praktisch alle aktuellen Markdown-Varianten dienen.
3. Auch bei den *Ausgabe*-Formaten erweist es sich als Tausendsassa: nicht nur HTML, DocBook,  $\LaTeX$ , RST und alle möglichen Markdown-Dialekte sowie PDF sind

---

<sup>3</sup><http://daringfireball.net/projects/markdown/>

<sup>4</sup><http://www.aaronsw.com/weblog/001189>

<sup>5</sup><http://johnmacfarlane.net/pandoc/demos.html>

<sup>6</sup><http://johnmacfarlane.net/>

möglich, sondern auch ConTeXt, MediaWiki, ODT (Open/LibreOffice Text), EPUB, EPUB3, JSON, XML, ManPage, DOCX (MS Word), FictionBook2, AsciiDoc und Rich Text Format.

4. Schließlich kann man beim HTML-Output auf einfache Weise auch so anspruchsvolle Dinge wie *RevealJS*<sup>7</sup>- und *ImpressJS*<sup>8</sup>-Präsentationen erzeugen, oder für's PDF-Zielformat *Beamer*<sup>9</sup>-PDF-Folien. (Der [Vortrag Nr. 176](#)<sup>10</sup> stellt die Erzeugung speziell diese Ausgabeformate in den Mittelpunkt.)

Die Entwicklung von *pandoc* schreitet nach wie vor zügig voran. Während der letzten beiden Jahre kamen praktisch monatlich neue Features hinzu. Seit der Version 1.0 im September 2008 gab es ca. 40 verschiedene Releases, also ca. alle 7 Wochen eine. Der aktuelle Stand ist Version 1.12.3.3 (9. Februar 2014).

## 4 Markdown: die Grundlagen

Also, nochmal zur Frage: *Wie würde man Text gestalten, um folgende Formatierungs-Elemente darzustellen?*

- Listen wie diese hier
- *kursive* Worte
- **fette** Worte
- ***fett-kursive*** Worte
- Satz-Teile, welche code-Elemente enthalten
- Absätze (werden durch Leerzeilen voneinander getrennt)

Die Antwort ist ganz einfach. Nämlich so:

```
* Listen wie diese hier
* *kursive* Worte
* **fette** Worte
* ***fett-kursive*** Worte
* Satz-Teile, welche `code`-Elemente enthalten
* Absätze werden durch Leerzeilen voneinander getrennt
```

Es gibt für manche der angestrebten Formatierungen mehrere Möglichkeiten, das entsprechende Markdown zu schreiben. Man kann z.B. als Listen-Markierer statt dem Sternchen (‘\*’) auch Plus- oder Minus-Zeichen (‘+’ oder ‘-’) verwenden. Zudem lassen sich einige *Syntax-Erweiterungen* nutzen, die *pandoc* unterstützt, in der Ur-Form von Markdown

<sup>7</sup><http://lab.hakim.se/reveal-js/#/>

<sup>8</sup><http://bartaz.github.io/impress.js/#/>

<sup>9</sup>[http://de.wikipedia.org/wiki/Beamer\\_\(LaTeX\)](http://de.wikipedia.org/wiki/Beamer_(LaTeX))

<sup>10</sup><http://chemnitzer.linux-tage.de/2014/de/vortraege/detail/176>

jedoch nicht vorhanden sind. Beispielsweise könnte man direkten HTML- oder  $\LaTeX$ -Code in den Quelltext einfügen, sofern man mit diesen Sprachen vertraut ist und die `pandoc`-Erweiterungen `+raw_html` und `+raw_tex` nutzt..

Wofür ist Markdown also gut? Na, für mehreres:

- *Erstens* hat man ein Dokument in Text-Form, welches ziemlich übersichtlich aussieht und sich sehr gut lesen lässt.
- *Zweitens* lässt sich dieses Dokument (da in Text-Form) ausgezeichnet in Versions-Kontroll-Systemen verwenden.
- *Drittens* kann man aus Markdown-Dokumenten mittels verschiedener Tools auf sehr einfache Weise andere Formate erzeugen:

- i. HTML
- ii. PDF
- iii. RTF
- iv.  $\LaTeX$
- v.  $\LaTeX$ Beamer Präsentationen
- vi. ODT (OpenOffice/LibreOffice)
- vii. DOCX (MS Word)
- viii. Text
- ix. AsciiDoc
- x. reStructuredText
- xi. DocBook XML
- xii. groff man pages
- xiii. GNU texinfo
- xiv. Text
- xv. MediaWiki markup
- xvi. ConTeXt
- xvii. EPUB, nämlich:
  1. EPUB v2
  2. EPUB v3

Es fehlt noch die Auflistung von Elementen, die man benötigt, um die Dokumentenstruktur ein bißchen zu gliedern. Damit wären die Basics dann bereits fertig:

- Überschriften 1. Ordnung
- Überschriften 2. Ordnung
- [...]

- Überschriften 6. Ordnung

Zu Beginn dieses Dokuments befindet sich eine *Überschrift 1. Ordnung*. Folgende Markdown-Textzeile hat zu dieser Überschrift geführt:

```
# Das Kleine Einmaleins von Markdown (mit etwas `pandoc`)
```

## 4.1 Überschrift 2. Ordnung

Die vorhergehende *Überschrift 2. Ordnung* kam so zustande:

```
## Überschrift 2. Ordnung
```

### 4.1.1 Überschrift 3. Ordnung

Die obige *Überschrift 3. Ordnung* hatte diesen Quell-Text:

```
### Überschrift 3. Ordnung
```

Wie Überschriften 4., 5. und 6. Ordnung gehen, sollte jetzt leicht zu erraten sein.

**Das war jetzt eigentlich schon das Wichtigste. Mehr braucht man nicht zu wissen, um mit Markdown-Schreiben anzufangen!** Wie man das mittels pandoc konvertiert, braucht man an dieser Stelle noch gar nicht zu wissen. Das kann man sich aneignen, sobald man seinen ersten Markdown-Text geschrieben hat.

## 5 Vertiefende Details

Die Liste im vorhergehenden Abschnitt war übrigens verschachtelt, und sie benutzte in der 2. und 3 Ebene jeweils ein anderes Aufzählungszeichen. Das ging so:

```
* Erstens hat man ein Dokument in Text-Form, welches ...
* Zweitens lässt dieses Dokument (da in Text-Form) au...
* Drittens kann man aus Markdown-Dokumenten mittels v...
  i. HTML
  i. PDF
  i. RTF
  i. \LaTeX
  i. \LaTeX Beamer Präsentationen
  i. ODT (OpenOffice/LibreOffice)
```

```
i. DOCX (MS Word)
i. Text
i. AsciiDoc
i. reStructuredText
i. DocBook XML
i. groff man pages
i. GNU texinfo
i. Text
i. MediaWiki markup
i. ConTeXt
i. EPUB, nämlich:
    1. EPUB v2
    1. EPUB v3
```

Beim Verschachteln von Listen sollte man einfach die nächste Ebene um 4 zusätzliche Leerzeichen einrücken. Die Auswahl der Aufzählungs-Zeichen bietet selbstverständlich einige Optionen mehr als die oben gezeigten: a), A., (i), um nur drei zu nennen.

Wie man unschwer erkennen kann, braucht man nummerierte Listen nicht händisch zu pflegen, um die richtige Reihenfolge der Nummerierungs-Zeichen zu erhalten – das macht pandoc bereits ganz automatisch.

## 5.1 Weitere Elemente

Bisher fehlen noch ein paar weitere Elemente, um einen Text aufzupeppen:

1. [Hyperlinks](#)<sup>11</sup>
2. Bilder und Grafiken
3. Einrückungen/Zitate
4. Definitions-Listen
5. Nummerierte Listen (wie diese hier)
6. Code-Blöcke
7. Tabellen
8. Horizontale Linien

### 5.1.1 Hyperlinks

Die schnellste Möglichkeit, einen Hyperlink einzufügen, besteht darin, folgendes Markdown zu schreiben: [Hyperlinks] (<http://en.wikipedia.org/wiki/Hyperlink>). So habe ich das in der vorhergehenden Liste auch gemacht. Das darf sogar mitten im Satz vorkommen:

---

<sup>11</sup><http://en.wikipedia.org/wiki/Hyperlink>

- In eckigen Klammern, [...], steht der im Zieldokument anzuzeigende Link-Text.
- Direkt dahinter, ohne Leerraum, folgt in runden Klammern, (...), das Sprungziel des Hyperlinks.

---

### 5.1.2 Horizontale Linien

Horizontale Linien wie diejenige vor der obigen Überschrift entstehen, indem man mindestens 4 "Dash"-Zeichen hintereinander eingibt, auf einer eigenen Linie:

```
----
```

### 5.1.3 Einrückungen/Zitate

Ein Zitat bzw. eine Einrückung ist dadurch markiert, dass sich vor dem Absatz ein > (plus 1 Leerzeichen) befindet:

**Dies ist ein Zitat. Es besteht aus 2 Absätzen.** Lorem ipsum dolor sit amet, consetetur sadipscing elitr.

Stet clita kasd *gubergren*, no sea takimata sanctus est Lore ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy.

Es ist aus folgendem Markdown entstanden:

```
> **Dies ist ein Zitat. Es besteht aus 2 Absätzen.** Lorem ipsum dolor
> sit amet, consetetur sadipscing elitr.
>
> Stet clita kasd *gubergren*, no sea takimata sanctus est Lore ipsum
> dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing
> elitr, sed diam nonumy.
```

### 5.1.4 Bilder und Grafiken

Die schnellste Möglichkeit, ein Bild einzufügen, besteht darin, folgendes Markdown zu schreiben: ![Bild-Unterschrift: Ghostscript-Logo](../my-resources/logo.png). Das darf allerdings **NICHT** mitten im Satz vorkommen, sondern es muss auf einer separaten Zeile stehen, mit jeweils einer Leerzeile davor und dahinter! Statt PNG kann man natürlich auch JPEG verwenden.

Eine dergestalt eingebundene Abbildung rendert pandoc als eigenen Absatz und verleiht ihr eine automatisch nummerierte Bild-Unterzeile<sup>12</sup>.

**WICHTIG:** Der Markdown-Code für ein Bild fängt immer mit einem Ausrufezeichen an. Also so:

```
![Bild-Unterschrift: Ghostscript-Logo](../my-resources/logo.png)
```

Das erzeugt dann folgendes Bild:



Abbildung 1: Bild-Unterschrift: Ghostscript-Logo

Dabei erscheint das Bild unskaliert in seinen Original-Abmessungen bei einer Auflösung von 72 PPI, horizontal mittig auf der Seite. Sollte das Bild zu groß sein, wird es entsprechend verkleinert dargestellt, damit es auf die Seite passt. Bei diesem Runterskalieren werden die Bild-Daten nicht verändert – entsprechend steigt die PPI-Zahl im Bildbereich der Seite an.

Leider ist es derzeit (noch) nicht möglich, den Bildbereich der Seiten vom Markdown-Code aus zu skalieren, oder Bilder rechts oder links am Seitenrand zu platzieren, um sie von Text umfließen zu lassen.

---

<sup>12</sup>Leider funktioniert dies bei den Ausgabeformaten ODT, RTF und OpenDocument (noch) nicht – hier wird der Bild-Untertitel ausgelassen.

### 5.1.5 Definitions-Listen

Definitionslisten kann man mittels folgender Markdown-Syntax erzeugen. Dabei sind innerhalb der Listen-Elemente weitere Markdown-Auszeichnungen zulässig:

```
Terminus 1.

:   Hier kommt die Definition 1*, welche den Terminus 1* erklärt.
    Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam.

Shellfunktion mit `Ghostscript`.

:   Definition 2.* Lorem ipsum dolor sit amet, consetetur sadipscing
    elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore.

    Nach dem zweiten Absatz fügen wir sogar noch einen Code-Block mit
    Syntax-Highlighting ein.

    ``` {.bash}
    # Etwas eingefügter Code, Teil der Definition 2. Es handelt sich
    # um ein Beispiel, welches eine Bash-Funktion darstellt, die es
    # ermöglicht, Ghostscript als Taschenrechner für die Kommando-
    # Zeile zu benutzen:
    function gscalculator()
    {
        IFS=" ";
        gs -dNODISPLAY -q -c "$* == quit" | head -n 1
    }
    # Anwendungsbeispiel:
    #   gscalculator 13 2 div 5 5 mul add
    # berechnet: (13/2)+(5*5)
    ```

    Vierter Absatz der Definition 2. Lorem ipsum dolor sit amet,
    consetetur.

Terminus 3

:   Definition 3*. Innerhalb von Definitionslisten kann man
    natürlich auch weitere Auszeichnungen einbauen.
```

Da kommt dann folgendes raus:

**Terminus 1.** Hier kommt die *Definition 1*, welche den *Terminus 1* erklärt. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam.

**Shellfunktion mit Ghostscript.** *Definition 2.* Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore.

Nach dem zweiten Absatz fügen wir sogar noch einen Code-Block mit Syntax-Highlighting ein.

```
# Etwas eingefügter Code, Teil der Definition 2. Es handelt sich
# um ein Beispiel, welches eine Bash-Funktion darstellt, die es
# ermöglicht, Ghostscript als Taschenrechner für die Kommando-
# Zeile zu benutzen:
function gscal()
{
    IFS=" ";
    gs -dNODISPLAY -q -c "$* == quit" | head -n 1
}
# Anwendungsbeispiel:
# gscal 13 2 div 5 5 mul add
# berechnet: (13/2)+(5*5)
```

Vierter Absatz der Definition 2. Lorem ipsum dolor sit amet, consetetur.

**Terminus 3** *Definition 3.* Innerhalb von Definitionen kann man *natürlich* auch weitere Auszeichnungen **einbauen**.

### 5.1.6 Code-Blöcke

Falls man einen Code-Block benötigt, der so aussehen soll (ohne Syntax-Highlighting):

```
%!PS
%
% PostScript-Code

/Helvetica findfont 48 scalefont setfont
72 600 moveto
1 0 0 setrgbcolor
(Hello, World!) show
showpage
```

rückt man den entsprechenden Abschnitt im Markdown-Quelltext einfach um jeweils (mindestens) 4 Leerzeichen ein:

```
    %!PS
    %
    % PostScript-Code

    /Helvetica findfont 48 scalefont setfont
    72 600 moveto
    1 0 0 setrgbcolor
    (Hello, World!) show
    showpage
```

Alternativ zur Einrückung um 4 Leerzeichen kann man auch sogenannte *fenced code blocks* (“*eingezäunte Code-Blocks*”) verwenden. Diese erfordern keine Einrückung, sondern das “Einzaunen” des Code-Bereichs: darunter versteht man das Umschließen des Code-Blocks durch zwei Zeilen mit je (mindestens) 3 `backticks`. Vorteil von *fenced code blocks*: hier kann man ein Syntax-Highlighting dazuschalten. Dieses Dazuschalten bewirkt man, indem man am Ende der ersten ‘Zaun’-Zeile in geschweiften Klammern eine der unterstützten Sprachen, eingeleitet durch einen Punkt, hinzufügt. Hier das Beispiel für das Anfordern von Syntax-Highlighting bei PostScript-Code:

```
``` {.postscript}
%!PS
%
% PostScript-Code

/Helvetica findfont 48 scalefont setfont
 72 600 moveto
 1 0 0 setrgbcolor
 (Hello, World!) show
showpage
```
```

Das Ergebnis sieht dann z.B. so aus (Details wie Farben, Schrift-Größe und -Fonts sind selbstverständlich abhängig vom jeweiligen CSS-File (bei HTML-Ausgabe), Referenz-Dokument (bei ODT-Ausgabe) oder  $\LaTeX$ -Vorlage (bei  $\LaTeX$ - oder PDF-Ausgabe), welches man verwendet):

```
%!PS
%
% PostScript-Code

/Helvetica findfont 48 scalefont setfont
 72 600 moveto
 1 0 0 setrgbcolor
 (Hello, World!) show
showpage
```

An den meisten Stellen dieses Dokuments kommen solche ‘fenced code blocks’ zum Einsatz, bereits ab dem 1. Kapitel. Dabei war die Syntax-Hervorhebung meist ausgeschaltet, indem der erste ‘Zaun’ wie folgt definiert ist:

```
``` {.noweb}
```

Wenn man wissen möchte, welche Sprachen `pandoc` per Highlighting unterstützt, kann man dies mit folgendem Kommando abfragen:

```
$> pandoc --version
```

Es gibt z.Zt. allerdings keinen Code, der z.B. `{.no-syntax-highlight}` oder ähnlich intuitiv lautet. Die Methode mit `{.noweb}` funktioniert als Übergangs-Lösung jedoch ebenso gut. ...

Um für die Code-Blöcke trotzdem den farbigen (=dunklen) Hintergrund erhalten, kam beim Konvertieren der Kommandozeilen-Parameter `--highlight-style=espresso` zum Einsatz.

## 6 Tabellen

Das Erzeugen von Tabellen geht ganz einfach.

### 6.1 Einfachste Möglichkeit: *Simple Tables*

Aus folgendem Markdown...

```

  Right      Left      Center      Default
  -----
  1245      1245      1245      1245
012345    012345    012345    012345
   123      123      123      123
    1        1        1        1
Table: Beispiel für eine einfache Tabellen-Syntax.

```

...entsteht die Darstellung dieser Tabelle:

Right	Left	Center	Default
1245	1245	1245	1245
012345	012345	012345	012345
123	123	123	123
1	1	1	1

Tabelle 1: Beispiel für eine einfache Tabellen-Syntax.

Anmerkungen:

- Die Tabellen-Unterschrift kommt nur dann zustande, wenn im Markdown das (englische!) Schlüsselwort `"Table:"` (mit Doppelpunkt) vorhanden ist und mit einer

Leerzeile Abstand vor oder hinter dem Tabellen-Code am Zeilenbeginn steht. (Abgekürzt kann man diesen Effekt ebenfalls erhalten, indem man nur einen einzelnen Doppelpunkt schreibt...)

- Die Spalten-Grenzen gibt's nur dort, wo die Strich-Linie an den entsprechenden Stellen unterbrochen ist.
- Eine zweite gestrichelte Linie ist hier überflüssig.
- Im obigen Fall ist die linke Spalte rechtsbündig (Ausrichtung an Dezimal-Kommas geht momentan noch nicht). Die zweite Spalte ist linksbündig, die nächste rechts davon ist zentriert, und die letzte ist wieder rechtsbündig (weil dies die Voreinstellung ist).
- Man darf zum Ausrichten der Spalten keine Tabulatoren benutzen!
- Die Kopf- und die Tabellenzeilen müssen im Quelltext ohne Umbruch in einer einzigen Zeile stehen.
- In der Ausgabe als  $\LaTeX$  oder PDF erscheint die Tabelle in ihrer Gesamtheit immer zentriert.
- Die jeweiligen Spaltenbreiten lassen sich nicht beeinflussen – sie werden automatisch auf die erforderlichen Maße eingestellt.
- Die Ausrichtung der Spalten steuert die jeweilige Position des Kopf-Textes zur darunterliegenden Strich-Linie. Dabei gelten folgende Regeln:
  1. Falls die Strich-Linie auf der rechten Seite bündig mit dem Kopf-Text ist, jedoch auf der linken Seite übersteht, dann ist die Spalte rechtsbündig.
  2. Falls die Strich-Linie auf der linken Seite bündig mit dem Kopf-Text ist, jedoch auf der rechten Seite übersteht, dann ist die Spalte linksbündig.
  3. Falls die Strich-Linie relativ zur Länge des Kopf-Textes auf beiden Seiten übersteht, dann ist die Spalte zentriert.
  4. Falls die Strich-Linie auf beiden Seiten bündig zum Kopf-Text ist, kommt die standard-mäßig voreingestellte Spalten-Ausrichtung zu Tragen (meistens wird dies Linksbündigkeit sein).

Wie leicht zu sehen ist, brauchen die Texte der Tabellen-Zellen nicht selbst exakt ausgerichtet zu sein. Selbstverständlich liest sich der Quelltext etwas schöner, wenn er ebenso formatiert ist, wie man für das End-Ergebnis beabsichtigt.

Alle Formen der folgenden *Simple Table*-Quelltexte werden zur selben Tabellen-Darstellung wie oben in *Tabelle 1* führen.

Right	Left	Center	Default
----- 1245	----- 1245	----- 1245	----- 1245
012345	012345	012345	012345
1	1	1	1

Right	Left	Center	Default
-----	-----	-----	-----
1245	1245	1245	1245
012345	012345	012345	012345
1	1	1	1

Right	Left	Center	Default
-----	-----	-----	-----
1245	1245	1245	1245
012345	012345	012345	012345
1	1 1		1

Right	Left	Center	Default
-----	-----	-----	-----
1245	1245	1245	1245
012345	012345	012345	012345
1	1	1	1

Right	Left	Center	Default
-----	-----	-----	-----
1245	1245	1245	1245
012345	012345	012345	012345
1	1	1	1

Alle o.a. Markdown-Beispiele führen zum selben Ausgabe-Ergebnis (nur 1x gezeigt, oben). Welche dieser Formen einem innerhalb seines Quelltextes besser gefallen, bleibt jedem selbst überlassen.

pandoc unterstützt mehrere Erweiterungen, um Tabellen per Markdown zu definieren: `grid_tables`, `pipe_tables` und `multiline_tables`. Diese werden in dem Artikel [“pandoc Power Features”](#)<sup>13</sup> ausführlicher besprochen.

## 7 Mathe-Formeln à la LaTeX

Man kann auch LaTeX-Formeln mit Markdown einsetzen. *(Aber die folgenden Beispiele habe ich zugegebenermaßen er-google-t – ich könnte sie nicht selbst schreiben, wie es wohl richtige LaTeX-Gurus könnten; LaTeX ist momentan noch ein Buch mit 7 Siegeln für mich. . .)*

Das funktioniert allerdings nur mit der neuesten pandoc-Version. Man schreibt einfach den originalen LaTeX-Code für eine Formel und schließt diese innerhalb von 2  $\$$ -Zeichen ein:

<sup>13</sup><http://chemnitzer.linux-tage.de/2014/vortraege/proceedings/>

```

 $\forall x \in X, \quad \exists y \leq \epsilon$ 
 $\cos(2\theta) = \cos^2 \theta - \sin^2 \theta$ 
 $\lim_{x \rightarrow \infty} \exp(-x) = 0$ 
 $\frac{n!}{k!(n-k)!} = \binom{n}{k}$ 
 $\frac{\frac{1}{x} + \frac{1}{y}}{y-z}$ 
 $\int_0^\infty \mathrm{e}^{-x} \mathrm{d}x$ 

```

Wenn es funktioniert, sieht das Ergebnis für die Formeln so aus:

$$\forall x \in X, \quad \exists y \leq \epsilon$$

$$\cos(2\theta) = \cos^2 \theta - \sin^2 \theta$$

$$\lim_{x \rightarrow \infty} \exp(-x) = 0$$

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

$$\frac{\frac{1}{x} + \frac{1}{y}}{y-z}$$

$$\int_0^\infty e^{-x} dx$$

Im HTML-, ODT- und DOCX-Output kommen diese Formeln auf Anhieb allerdings nur teilweise korrekt rüber – im  $\LaTeX$ -Output erscheinen sie tadellos (und ebenfalls im PDF, falls man es via  $\LaTeX$  erzeugt). Damit *alle* Formeln korrekt funktionieren, muß man an mehreren Stellen zusätzlich Hand anlegen – ein Thema, welches über den Rahmen dieses Beitrags hinausgeht.

## 8 Diverses

### 8.1 Superscript, Subscript

Nun zu einer kleinen Spezialität, die oft ganz nützlich ist: Superscript und Subscript. Hier zunächst der verwendete Markdown-Code:

```
* H20 ist (meist) eine klare Flüssigkeit. 210 ergibt 1024.
```

Mal sehen, ob das beim Konvertieren einwandfrei klappt:

- H<sub>2</sub>O ist (meist) eine klare Flüssigkeit. 2<sup>10</sup> ergibt 1024.

## 8.2 Durchstreichen von Textstellen

Man kann bestimmte Worte auch durchstreichen durchstreichen. Das geht im Markdown so...

```
auch durchstreichen durchstreichen.
```

...also durch Einfassung der durchzustreichenden Stelle in jeweils doppelte "Tilden". – Alles klar?

## 8.3 Fußnoten

Als nächstes: Fußnoten. Und zwar zuerst eine Form, die im Markdown-Quelltext als *inline*<sup>14</sup>-Fußnote angelegt ist).

Diese Fußnote kommt vom folgenden Markdown:

```
Als nächstes: Fußnoten. Und zwar zuerst eine Form, die im
Markdown-Quelltext als inline[Fußnoten sind sehr leicht zu schreiben.
Man legt diese an beliebigen Stellen des Textes einfach an, indem man
das Zeichen ^^ tippt, und direkt anschließend daran in eckigen Klammern
([...]) den Text, den die Fußnote enthalten soll.]-Fußnote angelegt
ist).
```

## 9 Konvertierung dieses Dokuments

Wie bereits gesagt: zum Konvertieren von Markdown-Text in andere Formate verwendet man am besten `pandoc`. (`Pandoc` kann nicht nur Markdown einlesen, sondern auch `AsciiDoc`, `Textile`, `reStructuredText`, `HTML`, `Textile`, `DocBook`, `MediaWiki markup` und `LaTeX`!).

Daraus erzeugt `pandoc` dann die Formate, die bereits weiter oben aufgelistet sind.

Für manche Ausgabe-Formate benötigt `pandoc` evtl. noch ein paar anderweitig vorbereitete Templates, damit das Ergebnis dann auch so aussieht, wie gewünscht. So lässt sich z.B. das `ODT/OpenDocumentText`-Format ganz unterschiedlich stylen, je nach dem, was man in dem entsprechenden Template hinterlegt hat.

### 9.1 Konvertieren nach HTML

HTML-Erzeugung aus Markdown ist sehr einfach:

---

<sup>14</sup>Fußnoten sind sehr leicht zu schreiben. Man legt diese an beliebigen Stellen des Textes einfach an, indem man das Zeichen `^^` tippt, und direkt anschließend daran in eckigen Klammern (`[...]`) den Text, den die Fußnote enthalten soll.

```
pandoc --to=html --from=markdown
      -o output.html clt14-markdown-und-ne-prise-pandoc.mmd
```

Allerdings erhält man jetzt ein HTML, welches nach wie vor externe Referenzen beinhaltet (z.B. Bilder, Javascripts, CSS-Stylesheets). Falls man ein *transportables* HTML-File erzeugen möchte, welches alle Referenzen eingebettet in sich trägt, muss man die zusätzliche Parameter `--standalone` `--self-contained` verwenden. Des weiteren bietet sich noch an, per `--css=my-stylesheet.css` ein eigenes CSS einzubinden, welches das Erscheinungsbild des HTML-Dokuments nach eigenem Gusto prägt:

```
pandoc \
  --to=html \
  --standalone \
  --smart \
  --highlight-style=espresso \
  --normalize \
  --filter=pandoc-citeproc \
  --bibliography=./refs.bib \
  --from=markdown+mmd_title_block\
+definition_lists+pipe_tables+multiline_tables\
+grid_tables+table_captions+implicit_figures \
  --toc \
  --css=./resources/kp.css \
  --self-contained \
  --include-after-body=./resources/footer.html \
  --output=./build-full/clt14-markdown-und-ne-prise-pandoc.html \
  clt14-markdown-und-ne-prise-pandoc.mmd
```

Das erzeugte HTML enthält jetzt alle erforderlichen Komponenten und kann (auch bei fehlender Netzverbindung) auf jedem Computer und jedem Betriebssystem betrachtet werden, sofern dort ein Browser vorhanden ist, der modern genug ist.

## 9.2 Konvertieren nach ODT (Libre/OpenOffice Text-Dokument)

Folgendes Kommando kann zum Erzeugen eines ODTs aus dem Markdown-Quellcode dieses Dokuments dienen:

```
pandoc \
  --to=odt \
  --standalone \
  --smart \
  --reference-odt=./resources/reference--kp.odt \
  --output=./build/markdown-minidocu.odt \
  --from=markdown+mmd_title_block\
```

```
+definition_lists+pipe_tables+multiline_tables\  
+grid_tables+table_captions+implicit_figures \\  
    clt14-markdown-und-ne-prise-pandoc.mmd
```

Die lange Zeile mit den vielen +-Zeichen (...+*mmd\_title\_block*+*definition\_lists*+...) teilt pandoc mit, dass das vorliegende Markdown an manchen Stellen entsprechende Syntax-Erweiterungen verwendet. Nur dann erfolgt die Konvertierung in der beabsichtigten Weise.

Der Parameter `--reference-odt=reference---kp.odt` legt fest, dass das resultierende OpenDocument dieselben Schriftarten, Absatz- und Überschriften-Stile (und auch dieselben Seiten-Formate, Kopf- und Fußzeilen) verwenden soll wie das genannte Referenz-Dokument.

### 9.3 Konvertieren nach PDF

Zur Konversion nach PDF gibt es natürlich auch die Möglichkeit, pandoc zuerst ein ODT erzeugen zu lassen (s.o.), dieses in LibreOffice/OpenOffice zu öffnen und dann nach PDF zu exportieren.

Für eine direkte "Markdown-nach-PDF"-Konversion benötigt pandoc eine lokal installierte L<sup>A</sup>T<sub>E</sub>X-Engine: *pdflatex*, *lualatex* oder *xelatex*.

Das aller-aller-einfachste Kommando zum Erzeugen von PDF aus dem vorliegenden Markdown (die Quelldatei trägt den Namen *clt14-markdown-und-ne-prise-pandoc.mmd*) sieht so aus:

```
pandoc -f markdown -o my-output.pdf clt14-markdown-und-ne-prise-pandoc.mmd
```

Allerdings gefällt mir das Ergebnis noch nicht besonders, denn es enthält z.B. noch kein Syntax-Highlighting meiner Beispiel-Codeblocks, für deren Erstellung ich mir so große Mühe gegeben habe... Zudem soll das vorbereitete Literatur-Verzeichnis ins fertige Dokument automatisch eingebaut werden. Das verlangt nach ein paar ergänzenden Kommandozeilen-Parametern.

Tatsächlich verwendete ich folgendes Kommando zum Erzeugen des PDFs, welches meine Einreichung als Beitrag für den vorliegenden [Tagungsband des CLT 2014](#)<sup>15</sup> darstellt:

```
pandoc \\  
  --standalone --smart \\  
  --toc \\  
  --highlight-style=espresso \\  
  --normalize \\  
  --template=./clt2014-template.tex \\  
  --latex-engine=pdflatex \\  
  \
```

<sup>15</sup><http://chemnitzer.linux-tage.de/2014/vortraege/proceedings>

```

--from=markdown+blank_before_header+mmd_title_block+definition_lists\
+pipe_tables+multiline_tables+grid_tables+table_captions\
+implicit_figures+tex_math_dollars+superscript+subscript\
+inline_notes+fenced_code_blocks+line_blocks+startnum\
+header_attributes+fancy_lists+example_lists+simple_tables\
+pandoc_title_block+yaml_metadata_block+raw_html\
+markdown_in_html_blocks+raw_tex+footnotes+inline_notes+citations\
+lists_without_preceding_blankline+link_attributes \
-V geometry:paperwidth=4.13193in \
-V geometry:paperheight=21.0cm \
-V geometry:vmargin=14.5mm \
-V geometry:tmargin=39pt \
-V geometry:bmargin=55pt \
-V fontsize:10pt \
--filter=pandoc-citeproc \
--csl=mhra.csl \
--biblio=mybiblio.bib \
--output=./build-full/clt14-markdown-und-ne-prise-pandoc.pdf \
clt14-markdown-und-ne-prise-pandoc.mmd

```

Wie man sieht, kann man die Geometrie-Maße sowohl in Inch (in), als auch in Zentimeter (cm), Millimeter (mm) und Punkten (pt) angeben.

## 10 Weitere, detaillierte Dokumentation

Eigentlich ist Markdown nicht nur eine 'Syntax für Text'. Zu Anfang seiner Geschichte war es zusätzlich ein Tool, welches aus ebendieser Syntax HTML erzeugen konnte. Aber diese vergleichsweise kleine Aufgabe hat inzwischen pandoc übernommen – zusätzlich mit einigen weiteren, viel größeren... Heutzutage ist Markdown allerdings in erster Linie als Text-Auszeichnungsformat bekannt.

Der vorliegende Artikel deckt selbstverständlich das Thema nicht erschöpfend ab. Manche pandoc-Begriffe hat er nur erwähnt, jedoch nicht erklärt: *Was ist denn z.B. ein 'pandoc\_title\_block'?*

Folgende Adressen können als Ausgangspunkt für eigene Recherchen in der Welt der weiteren Markdown- und pandoc-ereien dienen:

- Download des ursprünglichen markdown-Tools:  
<http://daringfireball.net/projects/markdown/>
- Dokumentation der Markdown-Syntax:  
<http://daringfireball.net/projects/markdown/syntax>
- Download (und Dokumentation) von pandoc:  
<http://johnmacfarlane.net/pandoc/>

- Pandoc-verständliche Syntax-Erweiterungen für Markdown:  
<http://johnmacfarlane.net/pandoc/demo/example9/pandocs-markdown.html>
- Vim-Plugin für pandoc:  
<https://github.com/vim-pandoc/vim-pandoc>

## 11 EBook-Publishing bei Leanpub.com

- Ein Internet-Verlag, der eine modifizierte markdown+pandoc-Toolchain verwendet und exzellente Konditionen sowohl für Autoren wie auch für Leser/Käufer bietet. Die dort erhältlichen Formate sind PDF, MOBI und EPUB: [Leanpub.com](https://leanpub.com)<sup>16</sup>
- *“PDF-KungFoo with Ghostscript & Co.”* (englisch) – mein EBook bei Leanpub als ‘work in progress’ mit Minimal-Preis \$US 0,00: <https://leanpub.com/pdfkungfoo>
- *“PDF-KungFoo Workshop bei der Ubucon 2013”* (deutsch) – ein kollaborativ erstelltes EBook mit Minimal-Preis \$US 0,00: <https://leanpub.com/pdfkungfoo-ws1-deu>

---

<sup>16</sup><https://leanpub.com/authors/>

# Mikrocontroller stromsparend programmieren

**Uwe Berger**  
bergeruw@gmx.net

In vielen Datenblättern moderner Mikrocontroller<sup>1</sup> (MCU) ist der Begriff "Ultra-Low Power" zu finden. Was ist damit gemeint? Kann man programmier-technisch Einfluss auf den Stromverbrauch einer Mikrocontrollerschaltung nehmen? In der Folge sollen diese und ähnliche Fragen aus der Sicht eines Softwareentwicklers diskutiert werden.

## 1 Motivation

Nicht erst seitdem der Begriffe „Green IT“<sup>2</sup> aktuell wurde, kümmern sich Hard- und Softwareentwickler um die Minimierung des Energieverbrauchs ihrer Produkte. Viele Einsatzszenarien erfordern den effizienten Umgang mit der Verfügung stehenden Energie. Geräte wie eine elektronische Armbanduhr oder Fernbedienung sollen nicht jeden Tag an ein Ladegerät angeschlossen werden. Ein Mobiltelefon soll in der Hosen- oder Handtasche Platz finden. Unbemannte Messstationen, Signalbojen etc. müssen monatelang oder, im Extremfall, über Jahre permanent und wartungsfrei funktionieren. Nicht immer steht eine Steckdose zur Stromversorgung in unmittelbarer Reichweite zur Verfügung.

Mit modernen Batterien und Akkumulatoren (Akkus), vielfach in Verbindung mit sogenannten „Energy Harvesting“<sup>3</sup>-Energiequellen (z.B. Photovoltaik), stehen Technologien zur Verfügung, die die Konzeption und den Aufbau von platz- und gewichtssparenden Geräten ermöglichen. Dabei reicht es nicht aus, „Ultra-Low Power“-Hardware nur einfach einzusetzen. Vielmehr muss auch die Firmware die zur Verfügung gestellten Möglichkeiten nutzen.

Der Autor betreibt eine Außenstation<sup>4</sup> mit diversen Sensoren zur Wetterbeobachtung, welche drahtlos an eine Hauptstation<sup>5</sup> angebunden ist. Die Energieversorgung sollte dabei unabhängig von stationären Stromquellen gelöst werden. Um dies leisten zu können, setzte er sich mit den Begriffen „Ultra-low Power“ und „energieeffi-

---

1 CPU plus weitere Peripherie (Speicher, ADC, DMA, Interrupt-Kontroller etc.) in einem IC

2 [http://de.wikipedia.org/wiki/Green\\_IT](http://de.wikipedia.org/wiki/Green_IT)

3 „drahtlose“ Energiegewinnung ([http://de.wikipedia.org/wiki/Energy\\_Harvesting](http://de.wikipedia.org/wiki/Energy_Harvesting))

4 <http://bralug.de/wiki/RFM12-Funkbrücke>

5 [http://bralug.de/wiki/Wetterdaten\\_mit\\_Linux\\_aufzeichnen\\_und\\_verarbeiten](http://bralug.de/wiki/Wetterdaten_mit_Linux_aufzeichnen_und_verarbeiten)

ziente Programmierung“ intensiv auseinander<sup>6</sup>.

## 2 Ultra-Low Power Hardware

Grundsätzlich werden die elektrischen Eigenschaften von Halbleiterschaltungen, und damit auch deren Stromverbrauch, durch die eingesetzten Technologien und den damit gegebenen physikalischen Gesetzmäßigkeiten bestimmt. Die Auswahl der geeigneten elektronischen Bauteile für ein konkretes Produkt ist eine verantwortungsvolle Aufgabe. Diese wird meist dem Hardwareentwickler überlassen, der diese Zusammenhänge<sup>7</sup> kennen sollte.

Einige dieser Eigenschaften sind aber auch für den Softwareentwickler interessant. Viele moderne MCUs bieten die Möglichkeit die interne CPU und weitere interne Peripherie-Komponenten (z.B. ADC, DMA, WDT)<sup>8</sup> unabhängig voneinander, teilweise oder vollständig abzuschalten. Dazu sind softwareseitig meist nur entsprechende Steuerbits in den dafür vorgesehenen MCU-Registern (siehe Datenblätter) zu manipulieren. Durch vorher konfigurierte interne und externe Ereignisse kann dieser Ruhezustand jederzeit wieder verlassen werden. Ähnliches gilt auch für viele externe Komponenten und Baugruppen diverser Hersteller<sup>9</sup>.

Dabei wird der Stromverbrauch teilweise drastisch gesenkt. Im Fall einer ATmega8L-MCU reduziert sich der Bedarf beispielsweise von 3mA im Aktiv-Modus auf unter 1µA bei der Abschaltung der CPU und aller weiteren internen Komponenten. Weitere Stromeinsparungen können durch die Absenkung der CPU-Taktfrequenz (z.B. ATmega8L mit 3V Versorgungsspannung: 8MHz, 6mA → 4MHz, 3mA) und die Minimierung der Zugriffe auf die verschiedenen Speicherbereiche (RAM, FLASH, EEPROM etc.) außerhalb der CPU erzielt werden.

## 3 Energieeffiziente Software entwickeln

Der überwiegende Teil, der durch eine MCU abzuarbeitenden Aufgaben lässt sich mit dem EVA-Prinzip<sup>10</sup> beschreiben. Es tritt ein Ereignis (Signal, Timer etc.) ein, auf welches der Prozessor entsprechend reagieren muss (Daten ermitteln, Berechnun-

---

6 [http://bralug.de/wiki/Mikrocontroller\\_stromsparend\\_programmieren](http://bralug.de/wiki/Mikrocontroller_stromsparend_programmieren)

7 z.B.: „embedded projects Journal“ (<http://journal.embedded-projects.net/>); Ausgabe 18; S.13 ff.; Riedenaier, A., „Ultra Low Power Design mit Atmel Microcontrollern“

8 Analog/Digital-Konverter, Direct Memory Access Controller, Watchdog

9 Beispiel Funkmodul RFM12: <http://www.hoperf.com/upload/rf/RFM12.pdf>

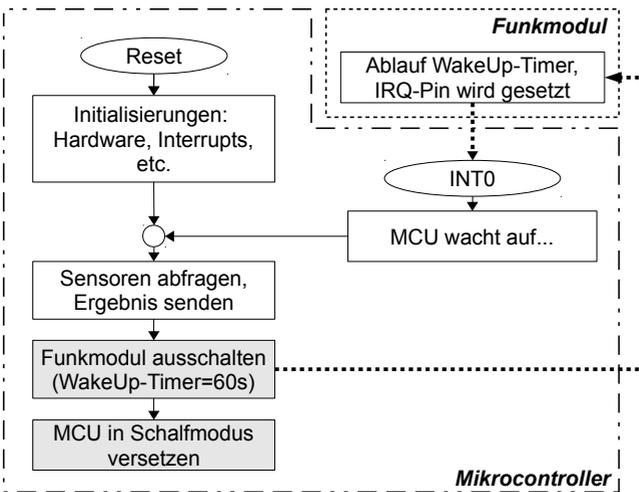
10 Eingabe/Ereignis → Verarbeitung → Ausgabe (<http://de.wikipedia.org/wiki/EVA-Prinzip>)

gen etc.). Die Verarbeitung endet mit einer geeigneten Ausgabe des Ergebnisses. Eine ausreichend schnelle MCU vorausgesetzt, wird man feststellen, dass zwischen diesen „EVA-Zyklen“ Wartezeiten entstehen. Der Prozessor und weitere nicht benötigte Peripherie kann abgeschaltet und damit in einen der zur Verfügung stehenden stromsparenden Zustände versetzt werden. Dies bedeutet, kurze Verarbeitungszeiten innerhalb der MCU verlängert deren Wartezeiten und reduziert damit den Strombedarf des Gesamtsystems. Die zeitliche Optimierung der Programmstruktur und -algorithmen ist Aufgabe des Firmwareentwicklers.

Für ihre Entwicklungsumgebung „Code Composer Studio“<sup>11</sup> bietet die Firma Texas Instruments<sup>12</sup> das Analysetool „ULPAdvisor“<sup>13</sup> an. Dieses Werkzeug ist auf die Analyse von C-Quelltexten für die haus eigene MCU-Familie MSP430 zugeschnitten. Die zugrunde liegenden Kriterien sind aber ohne weiteres verallgemeinerbar und damit auch auf vergleichbare Hardwareprodukte anderer Hersteller anwendbar. Die folgenden Empfehlungen beziehen sich deshalb auf die implementierten Analyse kriterien des „ULPAdvisor“. Die Reihenfolge spiegelt ihre Wichtigkeit aus Sicht des Autors wider.

**Nutze die Stromspar-Modi der MCU, wann immer es geht**

In einem der „Low Power“-Betriebsmodi verbraucht eine MCU am wenigsten Strom. Deshalb sollte einer dieser Zustände immer Ausgangs- und Endpunkt einer jeden Verarbeitung sein. Die Realisierung dieser wichtigen Empfehlung erfordert bereits in der Konzeptionsphase eine gewissenhafte Planung der Softwarestruktur:



Beispiel: Struktur Firmware Funkbrücke

11 <http://www.ti.com/tool/ccstudio>  
 12 <http://www.ti.com/>  
 13 [http://processors.wiki.ti.com/index.php/ULP\\_Advisor](http://processors.wiki.ti.com/index.php/ULP_Advisor)

Nach den notwendigen Initialisierungen und dem ersten Durchlauf der Verarbeitung (Messwerte ermitteln und senden), werden alle Komponenten der Funkbrücke (MCU und Funkmodul RFM12) in einen stromsparenden Zustand versetzt (im Bild grau dargestellt). Zuvor wurde das RFM12-Modul so konfiguriert, dass es nach ca. 60s selbstständig aufwacht und ein Signal ausgibt. Mit dem Auftreten dieses Signals am INT0-Eingang der MCU, wird hier der gleichnamige Interrupt ausgeführt. Die MCU wird in den Aktiv-Modus geschaltet und die Verarbeitungsroutinen wieder durchlaufen. Danach kehrt sie in den Stromsparmodus zurück. Im Fall der Funkbrücke beträgt das Verhältnis zwischen aktiv/inaktiv ca. 0,5s/60s (Stromverbrauch dabei 30mA/1µA!; siehe auch weiter unten).

Damit unterscheidet sich die Steuerung der Software von den üblichen Konzepten. Statt ständig aktiv Zustände abzufragen, *wartet* man auf deren Auftreten und aktiviert erst dann die notwendigen Komponenten.

Hinweis: Als Schlafmodus ist jeweils derjenige zu wählen, der ein Wecken durch den oder die zu erwartenden Interrupt-Ereignisse zulässt<sup>14</sup>. Welche das sind, ist in den entsprechenden Datenblättern der jeweiligen MCU-Hersteller zu finden.

### ***Verwende Interrupts statt „Flag-Polling“***

Viele der internen Peripherie-Baugruppen einer MCU spiegeln ihren momentanen Zustand mit Hilfe von Status-Bits (Flags) wider. Stößt man beispielsweise auf einem ATmega<sup>15</sup> eine A/D-Wandlung an, wird das Ende der Messung mit einem Low-Wert des ADSC-Bit des ADCSRA-Register gemeldet. Das Löschen dieses Flags könnte man mit einer while-Schleife im Programm abfragen. Dabei ist aber die CPU ständig aktiv und verbraucht viel Strom.

Besser ist es, vor Anstoß der A/D-Messung den ADC-Interrupt zu aktivieren. Jetzt kann man die MCU in den „ADC Noise Reduction“-Schlafmodus<sup>16</sup> versetzen (Stromverbrauch ca. 0,3mA) oder währenddessen andere Aufgaben erledigen lassen, um die Verarbeitungszeit insgesamt zu verkürzen. Ist die A/D-Wandlung beendet, wird automatisch die ADC-Interrupt-Routine gestartet. Es können nun die ermittelten ADC-Werte verarbeitet werden. Ähnliches ist auch mit vielen anderen Komponenten diverser MCU möglich. Für Einzelheiten wird auf die jeweiligen Datenblätter verwiesen.

### ***Verwende Timer statt Pausenschleifen***

Viele Softwareentwickler realisieren zeitlich definierte Programmpausen mit while-, until- oder for-Schleifen. Dabei muss die CPU der MCU aber aktiv sein und ständig die entsprechenden Schleifenbefehle ausführen.

Zielführender ist es, einen der internen MCU-Timer so zu konfigurieren, dass nach der gewünschten Pausendauer ein Interrupt ausgeführt wird. Während der Pause

---

14 Beispiel AVR-MCUs: [http://www.mikrocontroller.net/articles/AVR-Tutorial:\\_Power\\_Management](http://www.mikrocontroller.net/articles/AVR-Tutorial:_Power_Management)

15 <http://www.atmel.com/devices/atmega8.aspx>

16 [http://www.mikrocontroller.net/articles/Sleep\\_Mode](http://www.mikrocontroller.net/articles/Sleep_Mode)

schaltet man in einen der möglichen Schlafmodi.

Hinweis: Bei sehr kurzen Pausen muss man abwägen, ob sich das Umschalten lohnt, da sowohl hierfür als auch das Wecken ebenfalls Rechenzeit benötigt wird.

### ***Vermeide Funktionsaufrufe innerhalb von Interrupt-Routinen***

Aus den bisherigen Punkten ist erkennbar, dass Interrupt-Routinen eine zentrale Rolle bei der Steuerung stromeffizienter Firmware spielen. Durch sie sollen vor allem schnelle Reaktionszeiten auf interne und externe Zustände realisiert werden. Dazu müssen Interrupt-Routinen möglichst kurz sein und dürfen sich nicht überlappen. Im Idealfall setzt man dort nur Flags, auf deren Zustände später im Hauptprogramm entsprechend reagiert wird.

Hinweis: Diese Flags sollten als volatile Variablen<sup>17</sup> deklariert sein.

### ***Vermeide rechenintensive Operationen***

Auf MCUs ohne FPU<sup>18</sup> sollten Floating-Point-Berechnungen vermieden werden, da hierfür sehr langer Maschinencode vom Compiler generiert wird. Dessen Abarbeitung verkürzt die Verweildauer in einem der stromsparenden Modi. Gleiches gilt auch für Ganzzahlmultiplikationen ohne Hardware-Multiplizierer, Modulo-/Divisions-Operationen und viele weitere mathematische Funktionen. Jede Verwendung im Programmcode sollte deshalb kritisch überdacht und auf ein Minimum beschränkt werden. Für einige MCU-Plattformen existieren speziell angepasste Mathematik-Bibliotheken<sup>19</sup>. Weiterhin sind beispielsweise Festkomma-Arithmetik<sup>20</sup> und Lookup-Tabellen<sup>21</sup> ebenfalls geeignete Mittel zur Code-Reduzierung.

Aus diesen Gründen sollte man außerdem die Verwendung von „Universal“-Funktionen (z.B. printf(), sprintf() und String-Funktionen) aus den C-Standard-Bibliotheken vermeiden. Besser ist hier eine Neu-Implementierung mit genau dem Funktionsumfang, der in dem konkreten Anwendungsfall tatsächlich benötigt wird.

### ***Verwende, wenn vorhanden, DMA***

Steht in der MCU ein DMA-Controller zur Verfügung, ist dieser zum Transfer großer Datenmengen zwischen zwei Speicherbereichen (z.B. anstatt der C-Funktion memcpy()) oder interner Peripherie und Speicher (z.B. Senden/Empfangen über die serieller Schnittstelle)) zu verwenden. Während des DMA-Betriebs kann die CPU abgeschaltet und damit insgesamt Strom gespart werden.

### ***Benutze, wenn möglich, lokale statt globale Variablen***

Viele Compiler versuchen bei der Übersetzung lokale Funktionsvariablen freien Registern der CPU zuzuordnen. Der daraus resultierende Maschinencode ist kürzer

17 [http://www.mikrocontroller.net/articles/Interrupt#Volatile\\_Variablen](http://www.mikrocontroller.net/articles/Interrupt#Volatile_Variablen)

18 Floating Point Unit

19 Beispiel MSP430: <http://www.ti.com/tool/mspmathlib>

20 [http://www.mikrocontroller.net/articles/Festkommaarithmetik#ITOA\\_selbst\\_gemacht](http://www.mikrocontroller.net/articles/Festkommaarithmetik#ITOA_selbst_gemacht)

21 <http://de.wikipedia.org/wiki/Lookup-Tabelle>

und damit schneller. Es entfällt das Umkopieren vom Speicher in die CPU-Register.

### *Verwende „call by reference“ bei großen Variablen*

Wenn Funktionsparameter über den Zeiger auf ihre (globale) Speicheradresse referenziert werden („call by reference“), generiert der Compiler keinen Maschinencode zum Umkopieren der entsprechenden Speicherinhalte in die CPU-Register bzw. den Stack. Es reduziert sich die aktive Zeit der CPU.

Diese Empfehlung steht im Widerspruch zur vorhergehenden Regel, weil hierfür global deklarierte Variablen Voraussetzung sind. Hier muss der Softwareentwickler entscheiden bzw. experimentieren, welche der beiden Empfehlungen zum gewünschten Ergebnis führt. Bei kleineren Variablengrößen ist meist „pass by value“<sup>22</sup> vorzuziehen.

### *Verwende „const“ und „static“ bei Variablen-Deklarationen*

Wenn lokale Variablen in einer Funktion als "static" deklariert sind, werden sie nur einmal erzeugt und stehen während der gesamten Lebensdauer der Anwendung zur Verfügung. Dies reduziert den Maschinencode um den Teil, welcher sonst bei jedem Funktionsaufruf zur Reservierung und Initialisierung des entsprechenden Speicherbereiches ausgeführt werden muss.

Für Variablen, die mit dem Typ-Qualifizierer „const“ deklariert sind, wird kein Maschinencode zum Umkopieren vom Programmspeicher in den dynamischen Speicherbereich erzeugt. Der Inhalt dieser Variablen wird direkt aus dem Programmspeicherbereich gelesen, ist aber damit auch nicht veränderbar.

In beiden Fällen wird, durch die Reduzierung des auszuführenden Maschinencodes, die Dauer des stromverbrauchenden Aktiv-Modus der CPU verringert.

### *Verwende keine vorzeichenbehafteten Variablen, wenn nicht erforderlich*

Jede Rechnung mit vorzeichenbehafteten Variablen erzeugt zusätzlichen Maschinencode zur Überprüfung der Wertegrenzen, der die Aktivität der CPU verlängert. Aus diesem Grund sollte jede Variablendeklarationen nach ihrem voraussichtlichen Wertebereich kritisch beurteilt werden. Feld-Indizes können beispielsweise in C nur positive Werte annehmen.

Hinweis: Ähnliches gilt auch für Variablen, deren Größe die Registerbreite der CPU überschreiten.

### *Verwende Bitmasken statt Bitfelder*

Programmquelltext mit Zugriffen auf deklarierte Bitfelder<sup>23</sup> ist zwar lesbarer, aber der daraus resultierende Maschinencode meist ineffizient. Besonders der Zugriff auf einzelne Bits eines Registers oder Variablen über Bitmasken<sup>24</sup> kann vom Compi-

22 „pass by value“: Übergabe des tatsächlichen Wertes des Funktionsparameters

23 <http://www.c-howto.de/tutorial-strukturierte-datentypen-bitfelder.html>

24 Referenzierung/Manipulation einzelner Bits über geeignete logische Operationen (in C: &, |, ^, >>, <<)

ler meist zu einem Maschinenbefehl zusammengefasst werden. Ergebnis ist ein insgesamt kürzerer Maschinencode.

### *Zähle in bedingten Schleifen rückwärts statt vorwärts*

Diese Empfehlung mag auf den ersten Blick etwas ungewöhnlich erscheinen. Aber für jedes Inkrementieren einer Zählervariable in einer for-Schleife muss der Compiler in der Mehrzahl der Fälle einen zusätzlichen Maschinenbefehl einfügen. Es muss bei jedem Durchlauf die, vom Softwareentwickler festgelegte, obere Grenze der Schleife abgefragt werden, die eventuell eine Weiterverzweigung bedingen könnte. Viele Maschinencode-Befehlssätze kennen aber einen Befehl „Verzweige, wenn Ergebnis gleich 0“<sup>25</sup>. Besonders bei Schleifen mit vielen Durchläufen macht sich diese Reduzierung der Programmgröße hinsichtlich des Stromverbrauchs positiv bemerkbar.

Viele der hier aufgeführten Empfehlungen wurden auch bei der Implementierung der Firmware des Sendemoduls der Wetterstation des Autors berücksichtigt und umgesetzt. Dabei können diese aber in Gänze nicht pauschalisiert werden. Meist ist, mittels Experimentierens bei der Strukturierung des Programmablaufes, ein Kompromiss zwischen einzelnen Regeln zu suchen.

## **4 Permanente Stromversorgung?**

Nachdem nun die wichtigsten verallgemeinerbaren Empfehlungen zur Implementierung stromsparender Firmware besprochen wurden, stellt sich die abschließende Frage nach der einzusetzenden Energiequelle. Wichtigstes Kriterium ist dabei die Dauer des geplanten wartungsfreien Betriebs des Gerätes. Vernachlässigt man dabei die Lebensdauer der eingesetzten elektronischen Bauelemente<sup>26</sup>, wird das Wartungsintervall bei autarken Baugruppen durch die Kapazität der Spannungsversorgung bestimmt. Wünschenswert wäre eine permanente Stromversorgung über die gesamte Lebensdauer des Gerätes. Dem gegenüber stehen aber die Grenzen, die durch das Gerätedesign (z.B. Gewicht, Größe und Aussehen) bestimmt werden.

Heißt „permanent“ aber wirklich immer „für immer und ewig“? Die tatsächliche Nutzungsdauer elektronischer Geräte ist auf Grund verschiedener Faktoren begrenzt. Spätestens nach ca. 10 Jahren wird man in der Regel über den Ersatz nachdenken, da Abnutzungserscheinungen sowie technologische Weiterentwicklung zu weit fortgeschritten sind. Preiswerte Elektronik aus der Massenproduktion wird man in der Regel nicht länger wie 2-3 Jahre nutzen. In diesem Zusammenhang sollte auch der Begriff „geplante Obsoleszenz“<sup>27</sup> erwähnt werden. Eine ausreichende

25 Beispiel MSP430: Assemblerbefehl JNZ

26 [http://de.wikipedia.org/wiki/Lebensdauer\\_\(Technik\)](http://de.wikipedia.org/wiki/Lebensdauer_(Technik))

27 geplanter Verschleiß ([http://de.wikipedia.org/wiki/Geplante\\_Obsoleszenz](http://de.wikipedia.org/wiki/Geplante_Obsoleszenz))

Stromversorgung über mehrere Wochen, Monate oder Jahre aus ein und der selben Batterie kann dabei, je nach Anwendungsfall, ausreichend und zumutbar sein.

Der maßgebliche Kennwert zur Angabe des „Energiegehaltes“ einer Batterie ist ihre Kapazität<sup>28</sup>. Diese wird in Amperestunden angegeben. Hier einige ausgewählte Kennzahlen häufig eingesetzter Primärbatterie-Typen<sup>29</sup>:

	AA (2 Stück)	CR2032	CR123A
Nennspannung	2x1,5V=3V	3V	3V
Kapazität	2500mAh	220mAh	1500mAh
Kapazität bis 2,55V	1250mAh	220mAh	1500mAh
Selbstentladung (21°C)	3%	1%	1%

Für die Lebensdauer einer Batterie im Funkmodul der Wetterstation<sup>30</sup> des Autors ergibt sich damit z.B. folgendes:

- relevante Hardwareeigenschaften der größten Stromverbraucher:
  - MCU ATmega8L: min. Spannung ca. 2,7V (bei folgender Rechnung 2,55V angenommen); Stromverbrauch bei 4MHz, 3V ca. 3mA (Sleep-Mode <math><1\mu\text{A}</math>)
  - RFM12-Funkmodul<sup>31</sup>: min. Spannung 2,2V; Stromverbrauch Sendebetrieb (433MHz-Band) ca. 21mA (Sleep-Mode 0,3μA)
  - daraus ergibt sich ein Stromverbrauch der gesamten Baugruppe im Sendebetrieb von insgesamt (stark aufgerundet) ca. 30mA; der Stromverbrauch im Sleep-Mode wird in der folgenden Berechnung vernachlässigt
- Zeitverhalten der implementierten Firmware: jede Minute werden MCU und Funkmodul aus dem Sleep-Mode geholt, die angeschlossenen Sensoren abgefragt und das Ergebnis gesendet; Verarbeitungsdauer ca. 0,5s:  $1\text{h}=60\text{min} \rightarrow 60 \cdot 0,5\text{s}=30\text{s} \rightarrow$  entspricht 0,83% einer Stunde
- für die Spannungsversorgung kommen zwei AA-Zellen mit einer Kapazität von 1250mAh (bis 2,55V) zum Einsatz:  $1250\text{mAh}/30\text{mA}=41,6\text{h}$  Lebensdauer bei permanenter Entnahme von 30mA
- da nur 0,83% jeder Stunde Strom entnommen wird, ergibt dies für die Batteriebensdauer:  $41,6\text{h}/0,0083=5012\text{h} \rightarrow 209\text{d} \rightarrow 6,97$  Monate

28 [http://de.wikipedia.org/wiki/Kapazit%C3%A4t\\_%28galvanische\\_Zelle%29](http://de.wikipedia.org/wiki/Kapazit%C3%A4t_%28galvanische_Zelle%29)

29 <http://www.sensormag.com/sensors-expo/symposium2deeplyembeddedssystemsthatlastforever>

30 <http://bralug.de/wiki/RFM12-Funkbrücke>

31 <http://www.hoperf.com/upload/rf/RFM12.pdf>

Fast 7 Monate sind für diesen Anwendungsfall ein komfortabler Wert für die Nutzungsdauer einer einzelnen Batterieladung. Selbst mit einer kleinen CR2032-Knopfzelle (siehe Tabelle „Batterie-Kennzahlen“) könnte man das beispielhaft besprochene Funkmodul fast 37 Tage mit Strom versorgen. Und noch etwas anderes ist aus obiger Rechnung deutlich erkennbar. Jede weitere Verringerung der Verarbeitungszeit der MCU oder des Stromverbrauches der Schaltung wirkt sich positiv auf die Lebensdauer der Batterie aus. Würde man z.B. nur noch 0,25s für das Ermitteln und Versenden der Messdaten benötigen, könnte man die beiden AA-Zellen über ein Jahr verwenden.

Verwendet man statt der Primärbatterien Akkus und lädt diese während des Betriebes z.B. mit Solarenergie („Energie Harvesting“) permanent auf, kann die Nutzungsdauer bis zum Lebensende der eingesetzten Komponenten verlängert werden.

## 5 Zusammenfassung

Mit dem Begriff „Ultra-Low Power“ ist nicht nur die energiesparende Hardware moderner Mikrocontroller gemeint. Vielmehr sind die dabei eingesetzten stromsparenden Technologien nur die Grundlage für die Implementierung energieeffizienter Firmware. Berücksichtigt der Softwareentwickler einige verallgemeinerbare Regeln bei der grundsätzlichen Strukturierung des Programmablaufs und der Optimierung bzw. Vermeidung rechenintensiver Operationen, kann der Stromverbrauch einer Schaltung teilweise drastisch gesenkt werden. „Ultra-Low Power“ bedeutet für den Firmwareentwickler CPU und alle weiteren Hardwarekomponenten nur solange zu nutzen, wie wirklich notwendig. Dazwischen ist der energiesparendste Betriebszustand des Mikrocontrollers einzuschalten. Damit sind im Idealfall sogar ununterbrochene Batteriezyklen bis zum physischen Ende der eingesetzten elektronischen Komponenten erreichbar.



# Projektautomatisierung am Beispiel von *µracoli*

Axel Wachtler (axel@uracoli.de)

Ralf Findeisen (rfindeis@gmail.com)

Programmieren macht Spaß, testen ist langweilig, debuggen kostet viel Zeit. Die monotonen Aufgaben zur Qualitätssicherung eines Softwareprojektes müssen aber dennoch erledigt werden. Wir programmieren uns verschiedene Helfer, die schnell zeigen, wie sich die letzten “genialen” Codeänderungen auf das gesamte Projekt auswirken. Der Vortrag ist ein Streifzug durch das *µracoli*-Projekt und stellt Lösungen zur Testautomatisierung für Embedded Software mit Bash, Python, Scons und Jenkins vor.

## 1 Einleitung

Um bei komplexen Software-Projekten den Überblick nicht zu verlieren und die Funktionalität aller Komponenten zu gewährleisten, ist es sinnvoll, stets wiederkehrende Abläufe zu automatisieren. Wenn sich das System quasi auf Knopfdruck selbst baut, verpackt und testet, bleibt der Kopf frei für kreative Arbeiten. Auch gilt dann die Entschuldigung, dass man keine Zeit für den zeitraubenden Test hatte, nicht mehr.

Das *µracoli*-Projekt stellt ein Opensource-Software-Paket für Atmel IEEE-802.15.4-Transceiver zur Verfügung, mit dem einfach und schnell Anwendungen für drahtlose Sensornetze geschrieben werden können. Das Projekt stellt Libraries, Beispielprogramme und Applikationen für mehr als achtzig verschiedene Boards, die mit verschiedenen Mikrocontrollern, Transceivern und Sensoren bestückt sind, bereit.

Als Build-Plattformen werden Linux und Windows unterstützt. Neben der aktuellen AVR-Toolchain und Doxygen werden Scons und Python verwendet. Es hat sich gezeigt, dass sich durch den Einsatz von Python der Aufwand für eine multiplattformfähige Buildumgebung stark reduziert, da nur selten OS-spezifischen Funktionen implementiert werden müssen. So zum Beispiel kann man mit Scons auf jeder Plattform den “/” in Pfadangaben verwenden und verzichtet damit auf Abenteuer beim Quoten von Backslashes. Ausserdem bringt Python sehr viele nützliche Module mit, wodurch auf die Verwendung zusätzlicher externer Tools weitestgehend verzichtet werden kann.

## 2 Das Build System

Bei der Vielzahl von Boards und Anwendungen stellte sich sehr bald heraus, dass es essenziell für ein konsistentes Buildsystem ist, die Daten von den Buildregeln zu trennen.

Als Fileformat für die Metainformationen wird das INI-File Format [3] genutzt, da es direkt von Python unterstützt wird. INI-Files haben eine einfache Syntax und können mit einem Text-Editor bearbeitet werden. Die flache Datenhierarchie, die aus Abschnitten ([section]) und Key-Wert-Paaren (value = 42) besteht, ist zur Abbildung der Metainformationen ausreichend. Ein Beispiel aus dem File `board.cfg` zeigt die Metadaten des Boards `raspbee`:

```
# File board.cfg
[raspbee]
comment    = Dresden Elektronik Raspberry Pi Module
image      = raspbee.jpg
include    = boards/board_derfa.h
cpu        = atmega256rfr2
f_cpu      = 8000000UL
sensors    = mcu_vtg mcu_t
provides   = trx hif led tmr
...
```

In weiteren Dateien wird die Konfiguration von Anwendungsprogrammen und Paketen beschrieben.

```
# File application.cfg
[sniffer]
requires:  trx hif led
excludes:  stkm8
sources:   Sniffer/sniffer.c
           Sniffer/sniffer_ctrl.c
           Sniffer/sniffer_scan.c
```

```
# File packages.cfg
[psniffer]
name = uracoli-sniffer
boards = psk230 psk230b stb230 stb230b ...
depends = build/sniffer
files = install/bin/sniffer_psk*.hex
       install/bin/sniffer_stb*.hex
...
relocate = install/bin:firmware
          Tools:script
...
```

Diese Daten werden an mehreren Stellen im Buildprozess benutzt, u.a. von `scons` zur Erzeugung von Target-Rules und von verschiedenen Codegeneratoren zur Erzeugung von Header-, Make- und Doxygen-Files. Der Bezug zwischen den einzelnen Konfigurationsdateien wird über Schlüsselwerte hergestellt, z.B. wird aus dem

`provides`-Wert in `board.cfg` und dem `requires`-Wert in `application.cfg` ermittelt, ob eine Applikation für ein Board gebaut werden kann oder nicht.

### 3 Der Release Prozess

Die *uracoli*-Releases erfolgen in unregelmäßigen Abständen, meist dann, wenn neue Features, Boards, MCUs oder Transceiver implementiert wurden. Bei einem Release werden fünf verschiedene Pakete für unterschiedliche Anwendungszwecke bereitgestellt. Das Entwicklerpaket `uracoli-devel-0.4.1.zip` enthält einen Snapshot des Repositories, `uracoli-src-0.4.1.zip` ist das Anwender-Paket, das nur `python`, `make` zum Bauen benötigt. Vorkompilierte Binaries und Scripte zum Aufbau eines Sniffers mit Wireshark [4] sind im Paket `uracoli-sniffer-0.4.1.zip` enthalten. Das Paket `uracoli-arduino-0.4.1.zip` erweitert die Arduino-IDE um die Radiofaro und ZigBit Hardware. Die aktuelle Projekt-Dokumentation befindet sich im Paket `uracoli-doc-0.4.1.zip`.

Der Inhalt der Pakete ist, wie oben beschrieben, im File `packages.cfg` definiert. Für jede Major-Release wird im Repository ein Branch angelegt, um die aktuelle Entwicklung vom Release zu entkoppeln und um eventuelle Bugfixes unbeeinträchtigt von der Entwicklung auf der Mainline veröffentlichen zu können.

```
# hg branches
rel_0.4.x           2513:4d2d9f0e0b70
rel_0.3.x           2122:24ad3a3a7e17
...
```

Für das eigentliche Release-Paket wird auf dem Release-Branch ein Tag gesetzt.

```
# hg tags
0.4.1               2512:8106324461a1
0.4.0               2404:896d32bade39
...
```

Das Tag wird vom Release-Script `makerelease.sh` verwendet, um ausgehend von einem Snapshot des Repositories alle Pakete zu erzeugen.

```
# hg up rel_0.4.x
# bash Tools/makerelease.sh 0.4.1
```

Um sicherzustellen, dass keine lokalen Änderungen in die Release-Pakete kommen, wird der gesamte Build-Prozess in einem leeren Verzeichnis ausgeführt.

## 4 Interaktive Tests

Vor dem Upload der Pakete auf die Webseite muß deren Inhalt noch auf Vollständigkeit und Funktion geprüft werden. Eine manuelle Überprüfung ist sehr zeitaufwändig und hängt von der Gewissenhaftigkeit und Konzentrationsfähigkeit des Release-Verantwortlichen ab. Die Pakete müssen entpackt, kompiliert und auf realer Hardware getestet werden. Meist wird dieser Vorgang mehrfach ausgeführt, da nicht alles beim ersten Mal komplett funktioniert. Es spricht also vieles dafür, auch diesen Arbeitsschritt zu automatisieren.

Im Verzeichnis **Regression** des *uracoli*-Projektes befinden sich alle Scripte, die für die Tests von Release-Paketen erforderlich sind. Jedes der Scripte kann interaktiv gestartet werden und hat eine Hilfsfunktion.

```
# python Regression/test_src_pkg.py -h
This script unpacks and completely builds the source package
```

Usage:

```
python test_src_pkg.py [options] uracoli-src-<version>.zip
```

Options:

```
-h          show this help and exit
-b BUILD_DIR
            set build directory (default: ./build)
```

Nach dem Entpacken und Compilieren werden die erzeugten Image-Files auf existierende Hardware geflasht und verifiziert. Die Wireless-UART Applikation wird zum Beispiel mit dem Kommando `python test_wuart.py -b bin_dir setup.cfg` getestet. Im File `setup.cfg`, das ebenfalls ein INI-File ist, ist die aktuelle Hardwarekonfiguration beschrieben. Es enthält Informationen zum verwendeten Programmierer und zur seriellen Schnittstelle, über die das Board angebunden ist. Mit der Option `-b bin_dir` wird der Pfad zu den Image-Files angegeben.

## 5 Test Automatisierung

**Continuous Integration** Im Laufe der Zeit wachsen Umfang und Anzahl der Testscripte, und die Testabdeckung steigt. Mit wachsender Testanzahl wird aber auch das interaktive Testen mühsam. Um den Testvorgang vollständig zu automatisieren, kommt der Continuous-Integration-Server Jenkins [2] zum Einsatz.

Ein eigener Jenkins-Server kann sehr einfach in Betrieb genommen werden.

```
# wget http://mirrors.jenkins-ci.org/war/latest/jenkins.war
# java -jar jenkins.war
Running from: /home/axel/Work/uracoli-aw/jenkins.war
webroot: $user.home/.jenkins
...
INFO: Started SelectChannelConnector@0.0.0.0:8080
...
```

Danach ist der Server über <http://localhost:8080> erreichbar. Zunächst müssen noch notwendige Plugins installiert werden, z.B. das Mercurial-Plugin für den Zugriff auf das *µracoli*-Repository.

Nun kann man neue Jobs anlegen oder aber vorgefertigte Job-Konfigurationen verwenden. Ein Job wird vollständig durch die Datei `config.xml` beschrieben, die sich im Verzeichnis `$JENKINS_HOME/jobs/<jobname>/` befindet. Startet man Jenkins als normaler Benutzer, dann werden die Konfigurationsdaten im Verzeichnis `$HOME/.jenkins` gespeichert.

**Build- und Test-Server** Soll Jenkins für mehrere Nutzer eingesetzt werden, bietet sich der Betrieb einer zentralen Server-Instanz an. Das ermöglicht es, eine gemeinsame Maschine mit ausreichend Rechenleistung für Build- und Test-Aufgaben zu verwenden. So wird auch sichergestellt, dass alle Benutzer die gleiche Build- und Testumgebung verwenden. Bei größeren Projekten kann der Buildprozess durch Jenkins-Slave-Nodes oder Tools wie `distcc` auf mehrere Hosts verteilt werden. Die Slave-Nodes werden weiter unten noch im Zusammenhang mit dem verteilten Hardware-Test vorgestellt. Das oft keineswegs triviale Setup einer solchen Umgebung legt ebenfalls einen zentralen Jenkins-Server nahe.

Als zentraler Server wird Jenkins unter einem eigenen Nutzer installiert. Alle Konfigurations- und Arbeitsdaten werden im Homeverzeichnis des Jenkins-Nutzers gespeichert. Die Jenkinsinstallation, welche mit einigen Linuxdistributionen geliefert wird, verwendet das Verzeichnis `/var/jenkins`. Dieses Verzeichnis sollte regelmäßig gesichert und unter Versionskontrolle gestellt werden. Das kann entweder mit Werkzeugen wie `git` direkt geschehen oder über das Plugin `JobConfigHistory`.

Zum Start des Servers gibt es mehrere Optionen:

1. Ein manueller Start auf einer zentralen Serverinstanz, wie oben beschrieben, ist für Tests eines einzelnen Benutzers ausreichend, für die Arbeit im Team aber ungeeignet (Zugriffsrechte, Verfügbarkeit, Sicherheit).
2. Eingebettet in eine Servicestruktur wie Systemd oder Upstart, so wie es beispielsweise in Ubuntu realisiert ist.
3. Mit Hilfe von `mod_proxy` und `mod_proxy_ajp` im Apache. Die Proxies reichen Zugriffe an Jenkins weiter. Damit erreicht man, dass kein Port mehr mit angegeben und sich gemerkt werden muss. Die URL `http://ciserver:8081` wird damit zu `http://ciserver/jenkins`.

4. Durch Nutzung eines Applicationsservers, wie etwa Tomcat. Ein solches Setup bietet vielfältige Setupoptionen. Seine Vorteile spielt es erst aus, wenn Jenkins zusammen mit anderen Services in einer Enterprisearchitektur betrieben wird.

Für den Einsatz auf einem Server, der vorrangig für Continuous Integration verwendet wird, sehen wir ein Setup mit der Option 2, eventuell ergänzt um Option 3 als am günstigsten an.

Mit einem solchen Serversetup erhält man auch die Möglichkeit, in Jenkins Authentisierung und Autorisierung zu verwenden. Jenkins kann Nutzer entweder selbst verwalten oder einen Dienst wie LDAP ansprechen.

**Wireless Test Setup** Im folgenden wird unser Setup zum Compilieren und Testen der *µracoli*-Software vorgestellt.

Der Buildvorgang läuft vollständig auf dem Jenkins-Server ab. Für den Hardware-Test werden Jenkins-Slaves auf Raspberry Pis gestartet. Diese programmieren das Testgerät (device under test - DUT) und lassen eine Testsuite laufen. Die Kommunikation zwischen Raspberry Pi und dem DUT wird mit Hilfe eines Erweiterungsboards zur Drahtloskommunikation auf dem Raspberry Pi sichergestellt. Hier wird das RaspBee-Board der Firma Dresden Elektronik verwendet.



Bild 1: Raspberry Pi mit aufgestecktem RaspBee (dunkel eingerahmt)

Die Ergebnisse des Tests werden an den Server zurückgemeldet.

Zum Start der Slaves gibt es grundsätzlich mehrere Möglichkeiten. Wir haben uns für einen Start via SSH entschieden. Hierzu wird zwischen dem CI-Server und dem Raspberry Pi ein SSH-Setup mit Public-Key-Authentisierung verwendet.

Jeder Jenkins-Slave hält auf dem entfernten System Arbeitsdaten, sichert diese jedoch auf den Server zurück, so dass alle Daten zum Betrieb des Servers auch auf

diesem liegen und das Aufsetzen eines neuen Slaves kein Backup eines anderen Slaves benötigt, sondern nur Daten vom Server. Slaves können benannt werden. Das dient einerseits der Organisation bei größeren Setups, erlaubt aber andererseits auch das direkte Ansprechen eines bestimmten Slaves aus dem Testscript. Mit Hilfe dieses Features können Slaves auch für bestimmte Tasks freigehalten werden. Jenkins kann so konfiguriert werden, dass es den Slave bei Bedarf einschaltet.

**Testablauf** Im Folgenden wird angenommen, dass wie im Bild gezeigt, Slaves für die Boards konfiguriert worden sind. Diese Slaves müssen für jeden Raspberry Pi die Konfiguration der angesprochenen Boards kennen. Das kann der Slave entweder selbst herausfinden, indem die Boards über das Drahtlosinterface erkannt werden, wenn die Firmware das unterstützt oder dass in der Slave-Konfiguration die Zuordnung von Board zu Anschluß (bei USB/JTAG) bzw. die IDs im Netzwerk gespeichert sind.

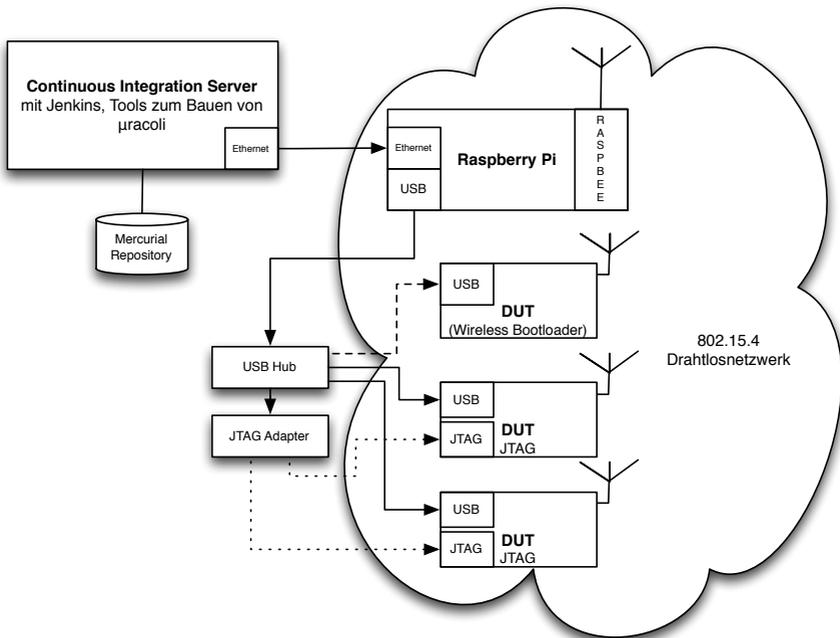


Bild 2: Testsetup

Ein Testlauf erfolgt mit den folgenden Schritten:

1. Bau der Firmware wie oben beschrieben. Jenkins holt die letzte Releaseversion aus Mercurial und baut die nötigen Images.
2. Die Jenkins-Slaves werden aktiviert. Danach sind alle Raspberry Pis gestartet, die DUTs sind eingeschaltet, die Slave-Dienste sind konfiguriert und warten auf Anweisungen.
3. Die neue Firmware muss auf die DUTs geflasht werden. Dazu gibt es zwei Möglichkeiten:
  - a. *Wireless Bootloader*. Das DUT startet in der Bootloader-Section. Jetzt kann das DUT über das Drahtlosinterface vom RaspBee aus programmiert werden. Auf dem DUT läuft in diesem speziellen Modus das Programm `wibo`, welches Softwareblöcke empfängt, die CRC prüft und die Application-Section des DUT programmiert.
  - b. *JTAG*. Über das IEEE-1149.1-JTAG-Interface [1] kann das DUT neu programmiert werden. Dieser Weg erlaubt auch den Test des Bootloaders.
4. Die Testsuite wird vom Jenkins-Server auf den Raspberry Pis durch die Slaves gestartet und die Resultate werden im Server gesammelt und dargestellt.
5. Die Testumgebung kann vom Server aus abgeschaltet werden.

## 6 Ausblick

Im Artikel wurde gezeigt, das mit einfachen Mitteln viele zeitaufwändige Arbeitsschritte des Release-Prozesses automatisiert werden können.

Da das *uracoli*-Projekt ständig weiter wächst, soll in der Zukunft an der Verteilung der Test-Aufgaben gearbeitet werden. Zum Beispiel können die Compile-Aufgaben mittels `distcc` auf mehrere Jenkins-Slaves verteilt werden. Mit der stetig wachsenden Anzahl an Boards ist auch die Verteilung der Hardware-Setups an verschiedene Standorte interessant.

Die Webseite zum Vortrag befindet sich unter <http://uracoli.nongnu.org/ct2014>.

## 7 Literatur

- [1] *IEEE1149 IEEE Standard Test Access Port and Boundary-Scan Architecture*. IEEE, 1990.
- [2] Jenkins. *Jenkins*. 2014. URL: <http://www.jenkins-ci.org>.
- [3] Wikipedia. *INI file*. 2014. URL: <http://de.wikipedia.org/wiki/802.15.4>.
- [4] Wireshark. *Wireshark*. 2014. URL: <http://wireshark.org>.

# Sicheres MultiSeat

**Axel Schöner**

axel.schoener@fh-kl.de  
<http://fh-kl.de/~axel.schoener>

Der Einsatz von MultiSeat zur Minimierung der Anschaffungs- und Betriebskosten schafft Sicherheitsprobleme. Um einen reibungslosen und sicheren Betrieb zu garantieren gilt es diese zu identifizieren und zu beseitigen. Als geeignetes Mittel hierzu dienen verschiedene Virtualisierungstechniken.

## 1 Einleitung

Einzelplatz-Computersysteme befinden sich weitestgehend im Idle-Zustand. Jedes Computersystem besitzt ein Netzteil mit lastabhängigem Wirkungsgrad, welcher nur unter bestimmten Voraussetzungen optimal ist. Zusätzlich können im Idle-Zustand nicht alle Komponenten des Computersystems deaktiviert oder energiesparend geschaltet werden. Klassisches MultiSeat ermöglicht die Effizienz zu steigern, verringert jedoch die lokale Sicherheit der Benutzer.

## 2 Vor-/Nachteile

### 2.1 Vorteile

Wird ein Computersystem von mehreren Personen gleichzeitig verwendet...

- ... sinkt die Anzahl der außerhalb des Wirkungsgrades betriebenen Computersystemen.
- ... erhöht sich die Zeitspanne des optimalen Wirkungsgrades für das verwendete Computersystem.
- ... verringern sich die Anschaffungskosten pro Arbeitsplatz.
- ... verringern sich die Wartungskosten pro Arbeitsplatz.
- ... können Serverdienste (z. B. Dateiserver) lokal für Anwender betrieben werden und belasten somit nicht das externe Netzwerk.

## 2.2 Nachteile

In Bezug auf die Sicherheit entstehen dabei folgende Probleme:

- Netzwerk-Datenverkehr ist bereits innerhalb des Computersystems nicht mehr privat.
- Wechseldatenträger können von allen Benutzern eingesehen/manipuliert werden.
- Angriffe gegen lokale Prozesse sind möglich.
- Umgehung von Authentifizierungsmechanismen sind durch lokale Angreifer möglich.
- Lokale Exploits können jedem Benutzer Schaden zufügen.
- Ressourcenengpässe können zu Verklemmungen führen und somit alle beteiligten beeinträchtigen.

## 3 Virtualisierungssoftware

### 3.1 Linux Containers(LXC)

Bei LXC [5] handelt es sich um die Userspace-basierte Komponente zum Betrieb von Linux Containern, welche mittels Kernel-Komponenten den 'virtuellen' Betrieb von Linux-Betriebssystemen ermöglicht. Linux Containers ermöglicht basierend auf Kernel-Namespaces [6] die Isolierung vorhandener Ressourcen. Durch die Verwendung von CGroups [7] können bestimmte Ressourcen Namespace-basierend freigegeben und limitiert werden (Prozessor-, Arbeitsspeicher, IO- und Netzwerkdurchsatz). Dies ermöglicht Container gegeneinander abzuschotten, als auch Verklemmungen aufgrund von Überlast einzelner Container zu vermeiden. Im speziellen ermöglicht dies auch den Zugriff auf virtuelle Terminals oder Geräte z. B. zur Ein- und Ausgabe.

### 3.2 QEMU

Bei QEMU [8] handelt es sich um einen Emulator, welcher einzelne Komponenten als auch komplette Computersysteme emulieren kann. Dies ermöglicht den Betrieb jedes Betriebssystems, welches auf einer durch QEMU unterstützten Architektur läuft. Durch die Verwendung von *KVM-Kernelmodul*, *Virtio* und gegebenenfalls *VGA Pass-through* wird eine nahezu native Performanz erreicht.

## 4 Konzepte

### 4.1 Container-basierter Desktop

Zum Betrieb eines Linux-Desktop Betriebssystem innerhalb eines Containers sind folgende Vorkehrungen zu treffen:

- Zugriff auf Eingabegeräte über Device-Nodes mittels CGroups erlauben
- Zugriff auf Grafikkarte über Device-Nodes mittels CGroups erlauben
- Laden der Grafikkarten-Treiber im Host-Betriebssystem verhindern
- X-Server innerhalb des Containers auf die speziellen Ein- und Ausgabegeräte konfigurieren

#### 4.1.1 Sicherheitsvorkehrungen

- Wahl des Netzwerktyps nach Anforderung(veth, vlan, macvlan, pyhs)
- Privater Zugriff auf Eingabegeräte
- Privater Zugriff auf eigene USB-Geräte über spezielle udev-Regeln

### 4.2 QEMU-basierter Desktop

Zum Betrieb eines Desktop Betriebssystem innerhalb einer emulierten Umgebung sind folgende Vorkehrungen zu treffen:

- X-Server Konfiguration auf die speziellen Ein- und Ausgabegeräte konfigurieren
- Eigene Session für Displaymanager erzeugen, welche die virtuelle Maschine startet

#### 4.2.1 Sicherheitsvorkehrungen

- Wahl des Netzwerktyps nach Anforderung(veth, vlan, macvlan, pyhs)
- Privater Zugriff auf Eingabegeräte, über X-Server im Host zu konfigurieren
- Privater Zugriff auf eigene USB-Geräte über zugewiesenen USB-Port

### 4.3 Bereitstellen von Betriebssystemen

Je nachdem welcher Virtualisierungsansatz (LXC/ QEMU) verfolgt wird, kann das Betriebssystem entweder direkt auf Filesystem-Ebene oder in Form einer Image-Datei zur Verfügung gestellt werden. In beiden Fällen kann das Betriebssystem von einem vorher bereitgelegten Verzeichnis abgeleitet werden. Dazu empfiehlt sich ein Snapshot-fähiges Filesystem wie BTRFS als Grundlage zu verwenden.

## 5 Minimale Konfiguration

### 5.1 Voraussetzungen

Benötigt wird ein Computer mit mehreren Ein- und Ausgabegeräten welche über USB-Hubs angeschlossen werden. Als Betriebssystem kommt ein aktuelles Linux zum Einsatz (z. B. Arch-Linux).

### 5.2 Installation

Nach Installation von LXC (*pacman -S lxc*) wird mittels *lxc-create* ein Vorlage-Container für die Seats erstellt. In diesem wird die grafische Umgebung ( z. B. KDE ) sowie die gewünschten Anwendungen eingerichtet.

### 5.3 Konfiguration der Seats

Das Zuweisen der Hardware zu den jeweiligen Seats geschieht in mehreren Schritten:

- Für jeden Seat ist eine eigene Xorg-Konfiguration anzulegen, welche die jeweiligen Ein- und Ausgabegeräte zuweist.
- Innerhalb des Containers sind die Device-Nodes für die jeweiligen Ein- und Ausgabegeräte zu erzeugen.
- Der Zugriff auf die Device-Nodes ist über CGroups zu erlauben.

Um diese Aktionen zu vereinfachen kann der Udev-Regelsatz: *50-usbseat.rules* verwendet werden. Dieser erkennt angeschlossene Input-Geräte, welche über einen USB-Hub angeschlossen sind und erzeugt für diese unter */dev/usbseat* eine entsprechende Gruppierung. Im einfachsten Fall, wenn auch die Ausgabegeräte am gleichen Hub angeschlossen sind, befinden sich dort Links auf alle benötigten Geräte. Dies hat den Vorteil, dass die Xorg-Konfiguration sehr einfach und dynamisch erstellt werden kann. Das Ableiten der Container, die Konfiguration, sowie das Starten der einzelnen Seats kann dann über ein Skript (z. B. */usr/lib/udev/usbseat.sh*) erfolgen.

## 5.4 Seats und QEMU

Um dem Benutzer die Möglichkeit einzuräumen, ein über QEMU emuliertes System zu starten, ist für den Displaymanager eine entsprechende Session anzulegen. Dies geschieht z. B. bei *LightDM* [9] über das Erstellen einer Datei unter */usr/share/xsessions* siehe Listing 1, sowie eines Skript unter */usr/bin/* z. B. nach Listing 2, welches die emulierte Maschine erzeugt.

Listing 1: win7.session

```
[Desktop Entry]
Name=QEMU Windows 7
Name[de]=QEMU Windows 7
Exec=qemu-session
TryExec=qemu-session-win7
Icon=
Type=Application
```

Listing 2: win7-session

```
#!/bin/bash
/usr/bin/qemu-system-x86_64 -m 1280M -machine type=pc,accel=
    kvm -net nic,model=virtio -net user -drive file=/home/
    schoenera/win7.img,if=virtio -vga qxl
```

Dies gestattet dem Benutzer die grafische Auswahl auch von komplett emulierten Systemen.

## 6 Mehrnutzen durch Virtualisierung

Das Aufbauen von VPN-Verbindungen in einer durch Linux-Container oder QEMU bereitgestellten Desktop-Umgebung ist für andere Umgebungen unsichtbar. Die verschiedenen Möglichkeiten der Netzwerkanbindung, welche für Linux-Container als auch QEMU angeboten werden, gestatten eine flexible Konfiguration je nach Anwendungsfall. Durch den Einsatz temporärer Images, die beim Starten einer virtuellen Maschine von einer zuvor erzeugten Quelle abgeleitet werden, kann jeder Zeit eine saubere Betriebsumgebung bereitgestellt werden. Zusätzlich wird bei diesem Ansatz Speicherplatz eingespart. Das Computersystem kann zusätzlich notwendige Serverdienste lokal bereitstellen, sodass zusätzliche Hardware eingespart werden kann. Durch die Verlagerung notwendiger Serverdienste direkt in das verwendete Computersystem können Angriffe auf lokale Szenarien beschränkt werden.

## 7 Hardware

Geeignete externe Hardware für Multi-Seat-Systeme:

- Plugable UD-160-M [1]
- Plugable DC-125 [2]
- Display-Link USB-VGA Adapter der Serie: DL-1x5 [3]
- Raspberry Pi [4]

## 8 Fazit

Durch den gezielten Einsatz von Virtualisierungstechniken lassen sich mehrere sichere Arbeitsplätze über ein einzelnes Computersystem abdecken. Gegenüber dem klassischen MultiSeat bietet dieser Ansatz zusätzlich die Möglichkeit dem Benutzer unterschiedliche Betriebssysteme anzubieten, als auch notwendige Serverdienste isoliert parallel zu betreiben. Finanzielle Vorteile durch geringere Anschaffungs- und Betriebskosten können in redundante und performantere Hardware investiert werden.

## Literatur

- [1] *Elektronisches Papier*. ud-160-m  
<http://plugable.com/products/ud-160-m>
- [2] *Elektronisches Papier*. dc-125  
<http://plugable.com/products/dc-125>
- [3] *Elektronisches Papier*. DL-1x5  
[http://www.displaylink.com/technology/displaylink\\_hardware.php](http://www.displaylink.com/technology/displaylink_hardware.php)
- [4] *Elektronisches Papier*. Raspberry Pi  
<http://www.raspberrypi.org/>
- [5] *Elektronisches Papier*. Linux Containers  
<http://linuxcontainers.org/>
- [6] *Elektronisches Papier*. Kernel-Namespaces  
<http://lwn.net/Articles/531114/>
- [7] *Elektronisches Papier*. CGroups  
<https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt>

- [8] *Elektronisches Papier*. QEMU  
<http://qemu.org/>
- [9] *Elektronisches Papier*. LightDM  
<https://launchpad.net/lightdm>



# Vertrauen in Spracherkennung...?

**Christina Lohr**

christina.lohr@informatik.tu-chemnitz.de

**Robert Herms**

robert.herms@informatik.tu-chemnitz.de

**TU Chemnitz, Fakultät für Informatik, Professur Medieninformatik**

Computergestützte Analyse gesprochener Sprache funktioniert optimal, wenn Akustik- und Sprachmodelle für einen gesonderten Einsatzbereich justiert sind. Um den Zeitaufwand zur Aufbereitung von Sprachdaten für die Modellierung zu reduzieren, existieren bereits unterstützende Open-Source-Werkzeuge.

Dieser Beitrag stellt einen Überblick von Zusammenhängen der maschinellen Sprachverarbeitung vor und zeigt anschließend den State of the Art einiger Werkzeuge sowie deren Architekturen, mit denen Wissensquellen in Laut-, Wort- und Satzebene für Spracherkennungssysteme aufbereitet werden.

## 1 Einleitung

Wer Anwendungen mit Spracheingabe genutzt hat, wird wahrscheinlich auf Probleme gestoßen sein und wurde nicht verstanden. Die Gründe können unterschiedlich sein und führen oft dazu, dass das Vertrauen in entsprechende Anwendungen verloren geht. Wird jedoch bedacht, dass die Zusammenhänge und Berechnungen der Spracherkennung komplex und Anwendungen funktionsfähig sind, wenn diese für den entsprechenden Einsatz angepasst sind, so ist verständlich, warum Fehler entstehen. Für unterschiedliche Einsatzbereiche existieren einige kommerzielle Lösungen, die für einige Anwendungsbereiche (z.B. Medizin) kostenintensiv sind oder zu stark an den Nutzer angepasst werden müssen. Vielen Anwendern ist nicht bewusst, dass einige Open-Source- oder frei verfügbare Werkzeuge existieren, mit denen Modelle für Spracherkennung selbst entwickelt werden können. In diesem Beitrag werden einige Hintergründe sowie freie und Open-Source-Werkzeuge für Spracherkennung vorgestellt, damit Spracheingabeanwendungen wieder *Vertrauen* entgegen kommt.

## 2 Theoretische Hintergründe zur Erkennung von Sprache

Zunächst werden einige Begriffe sowie Zusammenhänge der Spracherkennung und Verarbeitung gesprochener Sprache vorab betrachtet, da grundlegende Faktoren aus verschiedenen Disziplinen zusammenspielen. - Die Artikulation beim Sprechen, das Hörempfinden sowie ein unterschiedlicher Bildungsgrad oder eine entsprechende Herkunft mit einem Dialekt können in einer Konversation zu *Missverständnissen* führen. Einige komplexe mathematische Modelle sind für die maschinelle Erkennung von Sprache ebenfalls von großer Bedeutung.

## 2.1 Verarbeitung und Erkennung von Sprache

*Sprachverarbeitung* betrachtet alle Verarbeitungsmöglichkeiten von Sprache im Allgemeinen. Dies umfasst neben der Spracherkennung die Aufnahme, Ausgabe, Synthese von Sprache sowie Signalverarbeitung (Codierung und Filterung) als auch die Erkennung und Transformation von einzelnen Sprechern und Dialogsysteme. Auf Algorithmen zur Sprachsignalverarbeitung, die hier benötigt werden, wird an dieser Stelle nicht weiter eingegangen. Bevor Systeme Sprache erkennen, muss Sprache *gelehrt* werden. Dieser Prozess wird als *Training* bezeichnet. Nach Möglichkeit müssen dafür Aufnahmen in Form von vielen einzelnen Sätzen vorhanden sein und ein System sammelt *Wissen* über Sprache. Damit kann in einem anschließendem Prozess Sprache erkannt werden (*Decoding*). [1]

## 2.2 Die drei Komponenten der Spracherkennung

Die Elemente eines State-of-the-Art-Spracherkennungssystems sind drei unabhängige Komponenten: das *Sprachmodell*, das *Akustikmodell* und das *phonetische Wörterbuch* (*Dictionary*). Zur Spracherkennung agieren diese Modelle zusammen.

## 2.3 Sprachmodell

Das *Sprachmodell* beinhaltet einen Textkorpus und stellt eine Sammlung ohne Wissen akustischer Merkmale von *A-priori-Kenntnissen* über Sprache dar und beobachtet Vorkommnisse von Wortreihenfolgen. Die Auftrittswahrscheinlichkeit wird nach zwei Strategien ermittelt: *statistisch* und *wissensbasiert*. Wissensbasierte Modelle fundieren auf Grammatiken, die auf standardisierten Beschreibungen sowie auf linguistischen Wissen definiert sind. Der statistische Ansatz sieht Sprache als *zufällig* an und betrachtet Messungen durch Auszählen einzelner Wörter aus einem großem Korpus mit einem großen Vokabular. Sequenzen von Wörtern der Größe  $N$  (*N-Gramme*) werden in ihrer Häufigkeit ermittelt und somit Vorhersagewahrscheinlichkeiten geschätzt. Es zeigte sich, dass es für aussagekräftige Vorhersagen reicht, Wortgruppen mit drei zusammenhängenden Wörtern (*Trigrammen*) zu betrachten. Mithilfe von zusätzlich eingefügten Schlüsselwörtern wie *Start* und *Ende* (bzw.  $\langle s \rangle$  und  $\langle /s \rangle$ ) kann in der  $N$ -Gramm-Statistik definiert werden, ob ein Wort am Anfang oder Ende eines Satzes steht und es ist möglich, einzelne Wörter, die häufig am Satzanfang vorkommen, mit syntaktischen Strukturen in Verbindung zu bringen. [1] [2]

## 2.4 Phonetisches Wörterbuch

Die menschliche Sprache hat sich auf zwei Wegen entwickelt: geschrieben und gesprochen. Das phonetische Wörterbuch stellt eine *Brücke* zwischen den beiden Formen dar. Ein *Phonem* repräsentiert das kleinste Element einer gesprochenen Sprache

mit einem Unterschied in der Bedeutung. Die Aussprache einzelner Laute ist durch das *Internationale Phonetische Alphabet (IPA)* definiert und wird von der *International Phonetic Association*<sup>1</sup> verwaltet. IPA verwendet Symbole des griechischen und lateinischen Alphabets in veränderter Form. In der maschinellen Sprachverarbeitung gab es damit anfangs viele Probleme und es bildeten sich verschiedene Standards zur Darstellung der Lautschrift. In den USA entstand Anfang der 1970-er Jahre das *Arpabet*<sup>2</sup>, das mit ASCII-Zeichen die englischen Laute repräsentiert. Für die Darstellung einiger europäischer Sprachen, zum Beispiel die Umlaute der deutschen Sprache, ist das Arpabet nicht geeignet, da diese nicht definiert sind. Im europäischen Sprachraum entstand Ende der 1980-er Jahre *SAMPA (Speech Assessment Methods Phonetic Alphabet)*<sup>3</sup>, das einen Teil der Zeichen des ursprünglichen IPA in ASCII-Codierung darstellt. Mittlerweile ermöglicht UTF-8-Codierung auch, die ursprünglichen IPA-Symbole zur Programmierung zu verwenden. Dies führt jedoch schnell zu Codierungsfehlern, da nicht alle Schnittstellen alle IPA-Zeichen eindeutig unterstützen. Im *phonetischen Wörterbuch* werden alle Wörter, die ein System erkennen soll, in der geschriebenen und gesprochen Form, der phonetischen Beschreibung, definiert. Wörter können auch in mehreren Möglichkeiten der Aussprache definiert sein.

## 2.5 Akustikmodell

Alle Informationen von hörbaren Ereignissen werden im Akustikmodell vereint. Dazu zählen Audiosignale mit entsprechend berechneten extrahierten Merkmalen. Von einer gegebenen Merkmalssequenz wird eine Wortfolge mit der höchsten *A-posteriori-Wahrscheinlichkeit* aus einer Folge von Wörtern geschätzt. Anschließend wird entschieden, welche Wörter erkannt werden. *Hidden-Markov-Modelle (HMM)* beschreiben die Variabilität entsprechender Merkmalssequenzen. Der Grundgedanke ist ein Automatenmodell mit unüberwachten und nach außen nicht sichtbaren Zuständen. Ein Zustand wechselt nur in Abhängigkeit von Übergangswahrscheinlichkeiten in den nächsten Zustand. Mithilfe von beobachteten Ausgabesequenzen ist es möglich im Rahmen der Wahrscheinlichkeitstheorie Aussagen der verborgenen Zustände zu treffen.[1] [2]

## 3 Einige Werkzeuge zur Verarbeitung von Sprache

In den folgenden Abschnitten werden einige Open-Source- oder frei verfügbare Werkzeuge für die Sammlung und Aufbereitung von Trainingsdaten sowie für die Erzeugung von Sprach- und Akustikmodellen und zur Spracherkennung vorgestellt.

---

<sup>1</sup><http://www.langsci.ucl.ac.uk/ipa>

<sup>2</sup><http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

<sup>3</sup><http://www.phon.ucl.ac.uk/home/sampa/index.html>

### 3.1 Aufnahmewerkzeuge

Zu Beginn müssen für das Training von Sprach- und Akustikmodellen digitale Daten gesammelt werden. Neben ausreichend Textmaterial sind Daten in akustischer Form notwendig. Die einfachste Möglichkeit ist, mit einem beliebigen Audioaufnahmeprogramm (z. B. *Audacity*<sup>4</sup>) Sprache aufzuzeichnen. Bei größer werdenden Datenmengen und verschiedenen Sprechern verliert diese Vorgehensweise jedoch den Überblick. Vom *Bayrischen Archiv für Sprache (BAS) (LMU München)* werden verschiedene Werkzeuge zur Aufzeichnung und Bearbeitung von Sprachsignalen bereitgestellt.<sup>5</sup> Der vom BAS bereitgestellte *Speech Recorder*<sup>6</sup> ist ein sehr übersichtliches Werkzeug für Sprachaufnahmen. Der in JAVA geschriebene Recorder zeigt während der Aufnahme den einzusprechenden Inhalt und gewährleistet eine strukturelle Verwaltung von Aufnahmen und deren Inhalten. Es ist möglich viele Sätze einzusprechen und getrennt voneinander bzw. satzweise abzuspeichern. Für eine Weiterverarbeitung, wie mit *CMU Sphinx*, ist dies wichtig. Des Weiteren kann das Programm in verteilten Systemen genutzt werden und bietet die Grundlage für einen Ansatz mit *Crowdsourcing*, indem Sprachdaten im Internet verteilt gesammelt werden. Mit *Amazon Mechanical Turk*<sup>7</sup> ist es ebenfalls möglich, Trainingsdaten verteilt zu sammeln.

### 3.2 Werkzeuge für Transkription und Korpusverwaltung

Gesammelte Trainingsdaten müssen entsprechend transkribiert werden. Der Zusammenhang zwischen Sprache und Schrift muss, vor allem bei steigender Korpusgröße, entsprechend verwaltet werden, damit Daten unkompliziert austauschbar sind. Die zwei folgenden Programme unterstützen diesen Prozess.

#### 3.2.1 FOLKER

Der *FOLKER-Editor*<sup>8</sup> ist ein Werkzeug zur effizienten Unterstützung von Transkriptionen und wurde zur Erstellung von Minimaltranskripten in Anlehnung an *GAT-2 (Gesprächsanalytisches Transkriptionssystem, 2009)* entwickelt. *FOLKER* ist nicht wie andere Transkriptionswerkzeuge für viele Nutzungsszenarien ausgelegt, sondern für das Projektzenario „*Forschungs- und Lehrkorpus gesprochenes Deutsch*“ (*FOLK*). Der Editor erlaubt Importe einer Audiodatei und enthält einen Audioplayer sowie die Darstellung des Sprachsignals. Es können Zeitmarken gesetzt sowie Segmente markiert werden, um dann die entsprechende Transkription mit unterstützender Syntaxkontrolle durchzuführen. Zudem können Sprecher angelegt und jeweiligen Segmenten zugeordnet werden. Die resultierenden Transkripte liegen in einem spezifischen

---

<sup>4</sup><http://audacity.sourceforge.net>

<sup>5</sup>[http://www.phonetik.uni-muenchen.de/forschung/bay\\_arch\\_sprsig/](http://www.phonetik.uni-muenchen.de/forschung/bay_arch_sprsig/)

<sup>6</sup><http://www.bas.uni-muenchen.de/forschung/Bas/software/speechrecorder/>

<sup>7</sup><http://www.mturk.com>

<sup>8</sup><http://agd.ids-mannheim.de/folker.shtml>

XML-Format vor und können für den Austausch in andere XML-basierte Transkriptionsformate, wie das *EXMARaLDA*-Format, exportiert werden.[4]

### 3.2.2 EXMARaLDA

Zur umfangreichen und computergestützten Transkription, Annotation, Verwaltung sowie Auswertung gesprochener Sprache eignet sich die Sammlung der Werkzeuge und Datenformate von *EXMARaLDA* (*Extensible Markup Language for Discourse Annotation*). Die Transkription erfolgt, basierend auf verbreitete Systeme wie *HIAT* (*halb-interpretative Arbeits-Transkription*) oder *GAT*, durch den Partitur-Editor<sup>9</sup>. Die Verschriftlichung kann korrespondierend zur Audiospur segmentiert und annotiert werden. Definierte Sprecher können bestimmten Äußerungen zugeordnet werden. Eine Interoperabilität mit anderen Programmen wird durch mehrere Import- und Exportformate unterstützt und es ist möglich, Verschriftlichungen nach *FOLKER* zu exportieren. Das Format des Partitur-Editors liegt als XML vor, sodass auch im Vorfeld oder Nachgang der Bearbeitung die Verschriftlichung maschinell automatisiert verarbeitet werden kann.

Erstellte Transkriptionen sowie Metadaten werden vom *Corpus Manager (Coma)*<sup>10</sup> verwaltet. Dieses Werkzeug erzeugt Schema-definierte XML-Dateien, in der verschriftlichte Beiträge inklusive Audiospur, Metadaten referenziert und deren Sprecher zugeordnet werden. Die Anwendung ist nach einer Liste von Beiträgen (Kommunikationen), Metadaten und Sprechern strukturiert. Für das Durchsuchen in Korpora nach segmentierten Transkriptionen sowie deren Analyse entstand das *EXMARaLDA Analyse- und Konkordanztool (EXAKT)*<sup>11</sup>. Die Suche erfolgt durch reguläre Ausdrücke, XPath-Ausdrücke sowie XSL-Stylesheets. Das Ergebnis sind die gefundenen Wörter der Transkripte, deren Zugehörigkeit, Kommentare und Annotationen. [5] [6] [7]

## 3.3 Werkzeuge für phonetische Beschreibungen

Jedes Wort, das erkannt werden soll, muss im phonetischen Wörterbuch hinterlegt sein. Bei steigender Korpusgröße nehmen auch die Einträge im Dictionary zu und es ist sehr aufwendig für jedes Wort die korrekte phonetische Beschreibung anzugeben. Zwei Möglichkeiten, um die phonetische Beschreibung ermitteln zu können sind *espeak* und *Wiktionay API*.

### 3.3.1 eSpeak

Das Sprachsyntheseprogramm *eSpeak*<sup>12</sup> gibt Sprache vollständig synthetisch und nicht auf Basis von Sprachdaten aus. Über den Zugriff der Kommandozeile kann *eSpeak*

<sup>9</sup><http://www.exmaralda.org/partitureditor.html>

<sup>10</sup><http://www.exmaralda.org/coma.html>

<sup>11</sup><http://www.exmaralda.org/exakt.html>

<sup>12</sup><http://espeak.sourceforge.net>

die phonetische Beschreibung des eingegebenen Textes angegeben. Dieser ist jedoch maschinell erstellt und sehr einfach gehalten. Nicht alltägliche sowie seltene oder Fremdwörter können mitunter falsch dargestellt sein.

### 3.3.2 Wiktionary API

Einige Artikel der Wikipedia zeigen in der Begriffserklärung auch die phonetische Beschreibung an. Über die *MediaWiki API* kann diese Beschreibung aus dem *Wiktionary*<sup>13</sup>, einem „freien Wörterbuch“, via Webservice abgerufen werden und es ist möglich über beliebige Wörter in phonetischer Beschreibung darstellen zu lassen. Über die Eingabe des folgenden Links in die Zeile des Browsers können alle Information zu einem *Begriff*, auch die phonetische Beschreibung angezeigt werden:

<http://de.wiktionary.org/w/api.php?action=query&titles=Begriff&prop=revisions&rvprop=content&format=json><sup>14</sup>

## 3.4 Systeme für Training und Decoding von Spracherkennung

In den folgenden Abschnitten werden Bibliotheken vorgestellt, mit denen Sprach- und Akustikmodelle trainiert und zur Spracherkennung genutzt werden können.

### 3.4.1 CMU Sphinx

An der Carnegie Mellon University (CMU) entstand die Programmbibliothek *CMU Sphinx*, die seit 2000 als Open Source zur Verfügung steht. Die APIs auf der Basis von HMMs gliedern sich in verschiedene Komponenten auf, wovon jede für eine bestimmte Aufgabe entwickelt wurde.<sup>15 16</sup>

**Für Sprachmodelle - CMUlmk** Das *CMU-Cambridge Statistical Language Modeling Toolkit* besteht aus einigen Funktionen, um schrittweise aus einer \*.txt-Datei über verschiedene Häufigkeitsbetrachtungen ein Sprachmodell zu erstellen. Sind vor der Abarbeitung im Ausgangstext verschiedene Sätze markiert, so werden Grammatikstrukturen im Sprachmodell berücksichtigt. Die Tags der Strukturen werden in einer *Context-Cue*-Datei (\*.ccs) festgelegt. In den ersten Schritten wird eine Häufigkeitstabelle sowie eine Vokabelliste des vorliegenden Textes ausgegeben. Mit diesen Ausgangsdaten wird anschließend eine N-Gramm-Statistik erstellt. Das *N* kann vom Nutzer beliebig groß festgelegt werden. Es steigt dann nicht nur der Speicherplatz des Sprachmodells, die anschließende Bibliotheken von CMU Sphinx können zum Teil nicht mit  $N > 3$ . Zum Schluss wird aus der N-Gramm-Statistik, der Vokabelliste und den Context Cues das Language Model (\*.lm) ausgegeben.<sup>17</sup>

<sup>13</sup><http://de.wiktionary.org>

<sup>14</sup><http://www.mediawiki.org/wiki/API>

<sup>15</sup><http://cmusphinx.sourceforge.net>

<sup>16</sup><http://www.speech.cs.cmu.edu>

<sup>17</sup>[http://www.speech.cs.cmu.edu/SLM\\_info.html](http://www.speech.cs.cmu.edu/SLM_info.html)

**Für das Training - SphinxBase und SphinxTrain** Diese beiden Bibliotheken stellen Module für den Aufbau eines Sprachmodells bereit. SphinxBase beinhaltet APIs, die von SphinxTrain und den Recognizern *PocketSphinx*, *Sphinx3* und *Sphinx4* genutzt werden. SphinxTrain besteht aus Perl-Skripten, mit denen ein Akustikmodell trainiert wird. Sprachaufnahmen (\*.wav) werden in Merkmalsstrukturen (\*.mfc) gewandelt und dann wird das Training gestartet. Währenddessen werden HMMs erstellt und durch den *Baum-Welch-Algorithmus* trainiert.

**Zur Erkennung - Die Recognizer** Die *Spracherkenner evaluieren* ein Modell nach dem Training und geben die *Wortfehlerrate (WER - Word Error Rate)* in Prozent an. Der Wert ergibt sich aus der Anzahl aller eingefügten Wörter zusammen mit allen Auslassungen und Einfügungen geteilt durch die zu erkennenden Wörter. Von CMU Sphinx werden drei verschiedene *Recognizer* bereitgestellt: *Sphinx3*, *Sphinx4* und *PocketSphinx*. Das in C geschriebene Sphinx3 wurde darauf ausgerichtet, *viel* erkennen zu können, wobei ein schnelles Laufzeitverhalten vernachlässigt wurde. Mit der Entwicklung von Sphinx4 in Java wurden diese Probleme behoben und es ist möglich, Modelle während der Laufzeit zu erweitern. PocketSphinx entstand für den mobile Einsatz.

### 3.4.2 HTK

Das *Hidden Markov Toolkit (HTK)* <sup>18</sup> ist ebenfalls eine Sammlung von Werkzeugen um HMMs und HMM-basierte Sprachverarbeitungswerkzeuge zu entwickeln. Es ist eines der am weitest verbreiteten freien Entwicklungsumgebungen zur Erstellung von HMMs. Das Toolkit bietet für den Bereich der automatischen Spracherkennung Werkzeuge zur Datenaufbereitung, Training, Erkennung und Analyse. Das HTK ist ein typisches State-of-the-art-Spracherkennungssystem. Da die Eingabedaten ähnlich wie bei CMU Sphinx sind und da es ebenfalls auf der Basis von HMMs arbeitet, stellt dieses ebenfalls eine Alternative dar. Einige Benchmark-Tests zeigten ein paar Vorteile bzgl. der Laufzeit und auch der Erkennungsrate im Gegensatz zu Sphinx3, allerdings konnte Sphinx4 dies wieder wett machen. [9]

## 3.5 Merkmalsextraktion mit openSMILE

Ein modulares und flexibles Werkzeug zur Merkmalsextraktion für Signalverarbeitung und maschinelles Lernen ist *open Speech and Music Interpretation by Large Space Extraction (openSMILE)*<sup>19</sup>. Die in C++ geschriebenen APIs können auf verschiedenen Plattformen wie Linux, Windows und MacOS ausgeführt werden. Für die Zusammenarbeit mit anderen Werkzeugen, unterstützt openSMILE verschiedene Datenformate der Bereiche Data Mining und maschinelles Lernen wie PCM WAVE, CSV, ARFF oder HTK. Im Toolkit gibt es auch mitgelieferte Feature Sets (Zusammenfassung der

---

<sup>18</sup><http://htk.eng.cam.ac.uk>

<sup>19</sup><http://opensmile.sourceforge.net>

zu extrahierenden Merkmale) für bestimmte Anwendungsfälle wie Emotionserkennung oder auch zur Extraktion von Merkmalen für die Sprachverarbeitung und -erkennung. Das Toolkit enthält das mächtige Kommandozeilenprogramm *SMILExtract*, das alle openSMILE-Komponenten enthält. [10]

### 3.6 Projekte zur Sammlung von Sprachdaten - VoxForge

Das Projekt *VoxForge*<sup>20</sup> sammelt Sprachdaten verschiedener Nationalitäten und stellt diese als Open Source zur Verfügung. Der Wortschatz des deutschen Sprachmodells ist mit ca. 3000 Einträgen im Dictionary und 4000 eingesprochenen Sätzen sehr eingeschränkt. Der Wortschatz beinhaltet sehr viele Wörter aus der Mathematik, Informationstechnologie und Rechtswissenschaft. Das Wörterbuch ist jedoch teilweise maschinell erstellt und enthält viele Fehler.

## 4 Workflow

Die folgende Abbildung zeigt, wie Werkzeuge der Sprachverarbeitung zusammenarbeiten und diese an der Professur Medieninformatik (TU Chemnitz, Fakultät für Informatik) genutzt werden. Für die Korpusverwaltung werden durch einen *Importer* Audiodaten verschiedener Quellen gesammelt und durch FOLKER, Coma und eigene Webbasierte Anwendungen (im Zusammenspiel mit der Wiktionary API) verarbeitet. Aus diesen Korpora werden Modelle erzeugt bzw. trainiert (CMU Sphinx) oder Analysen für Spracherkennung erstellt.

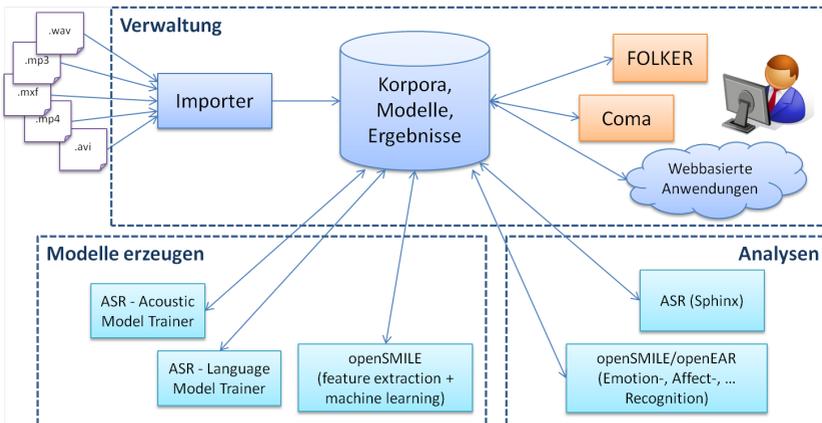


Abbildung 1: Aktueller Workflow an der Professur Medieninformatik

<sup>20</sup><http://www.voxforge.org>

## Literatur

- [1] *Sprachverarbeitung: Grundlagen und Methoden Der Sprachsynthese und Spracherkennung*. Beat Pfister and Tobias Kaufmann. Springer. 2008.
- [2] *Grundlagen der statistischen Sprachverarbeitung*. Prof. Dr. Andreas Wendemuth. Oldenbourg Verlag. 2004.
- [3] *Sprachmodelladaptation von CMU Sphinx für den Einsatz in der Medizin*. Christina Lohr. Studentensymposium Informatik Chemnitz. 2012.  
<http://www.tu-chemnitz.de/informatik/service/if-berichte/pdf/CSR-12-01.pdf>
- [4] *FOLKER Transkriptionseditor für das „Forschungs- und Lehrkorpus gesprochenes Deutsch“ (FOLK)*. Transkriptionshandbuch. Thomas Schmidt. Wilfried Schütte. 2.9.2011.  
<http://agd.ids-mannheim.de/download/FOLKER-Transkriptionshandbuch.pdf>
- [5] EXMARaLDA Partitur - Editor. Handbuch Version 1.5.1 Thomas Schmidt. 20.10.2011.  
[http://www1.uni-hamburg.de/exmaralda/files/PartiturEditor\\_Handbuch.pdf](http://www1.uni-hamburg.de/exmaralda/files/PartiturEditor_Handbuch.pdf)
- [6] EXMARaLDA EXAKT. Manual. Version. Thomas Schmidt. 1.0. 26.11.2010.  
[http://www1.uni-hamburg.de/exmaralda/files/EXAKT\\_Manual.pdf](http://www1.uni-hamburg.de/exmaralda/files/EXAKT_Manual.pdf)
- [7] EXMARaLDA Corpus-Manager Coma 1.7. Kai Wörner. 25.2.2011.  
<http://www1.uni-hamburg.de/exmaralda/files/comadoku.pdf>
- [8] *Statistical Language Modeling Using the CMU-Cambridge Toolkit From Proceedings ESCA Eurospeech*. P.R. Clarkson and R. Rosenfeld. 1997.
- [9] The HTK Book (for HTK Version 3.4). Steve Young et al. December 2006.  
[http://htk.eng.cam.ac.uk/prot-docs/htk\\_book.shtml](http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml)
- [10] openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. Florian Eyben, Martin Wöllmer, Björn Schuller. Proc. ACM Multimedia (MM), ACM, Florence, Italy, ISBN 978-1-60558-933-6, pp. 1459-1462, 25.-29.10.2010



# Vertraulichkeit in (kleinen) Unternehmen (k)eine Frage der IT?

**Georg Schütz – Dipl.Wirtschaftsingenieur (FH)**

KaMUX GmbH & Co. KG – <http://kamux.de> und KMUX-Projekt – <http://kmux.de>  
[georg.schuetz@kamux.de](mailto:georg.schuetz@kamux.de)

## 1 Vorwort

In Deutschland stellen Kleine und Mittlere Unternehmen (KMU) 99,3% aller Betriebe. Diese Quote ist über die Jahre nahezu konstant und in der gesamten EU fast gleich. Die KMU-Definition der EU geht dabei bis 250 Mitarbeiter. Nimmt man die Unternehmen mit 1-50 Mitarbeitern, dann sind das rund 96,2% aller Betriebe in Deutschland (Stand: Ende 2010, Quelle: Statistisches Bundesamt). In diesem Vortrag beziehe ich mich hauptsächlich auf diese Kleinstunternehmen (1-10 MA) und kleinen Unternehmen (11-50 MA).

Zum Begriff „Vertraulichkeit“: ich benutze die Definition, dass Vertraulichkeit die Eigenschaft von Daten, Worten oder allgemein gesprochen von Sachverhalten ist, nur für einen bestimmten Empfängerkreis vorgesehen zu sein. Vertraulichkeit wird in Deutschland durch Rechtsnormen geschützt, die zum Teil Grundrechtsscharakter haben. Siehe dazu auch: <http://de.wikipedia.org/wiki/Vertraulichkeit>

## 2 Vertraulichkeit im Unternehmen

Im Unternehmen behandle ich zunächst kurz die Vertraulichkeit im Innenverhältnis von Arbeitgeber und Arbeitnehmer. Ebenso wichtig ist die Vertraulichkeit im Außenverhältnis, also z.B. gegenüber Geschäftspartnern oder Behörden.

### 2.1 Vertraulichkeit in der Beziehung Arbeitgeber - Arbeitnehmer

Die Wahrung der Vertraulichkeit im Unternehmen gehört zu den Grundrechten

und Grundpflichten sowohl des Unternehmers als auch seiner Mitarbeiter. Zur sogenannten Treuepflicht des Arbeitsvertrags gehört die Verschwiegenheitsverpflichtung. Diese betrifft beide Vertragspartner. Beide verpflichten sich, Mitteilungen oder Informationen geheim zu halten, die geeignet sind, den Ruf oder die „Kreditwürdigkeit“ des jeweils anderen zu schädigen. Für den Arbeitnehmer gehören dazu unbedingt alle Interna über Kunden, Lieferanten, Preise, Konditionen und sonstige Geschäftsgeheimnisse. Für den Arbeitgeber gehören zu seiner Fürsorgepflicht daher auch Aussagen über die Leistungsfähigkeit und -willigkeit seiner Arbeitnehmer.

## 2.2 Vertraulichkeit in der Beziehung zu Geschäftspartnern

Der Grundsatz von „Treu und Glauben“, also das redliche und anständige Verhalten zwischen Vertragsparteien ist schon seit römischer Zeit eine anerkannte, wenn auch schwammige, Norm. Auch der Begriff des „Ehrbaren Kaufmanns“ also des wirtschaftlich tätigen Menschen, der sich in der Verantwortung für sein Handeln, sein Unternehmen und seine Umgebung sieht, drückt eine spezielle Form der Fürsorge oder „Treuepflicht“ aus. Die Vertraulichkeit von Informationen in einer Geschäftsbeziehung ist damit essentieller Bestandteil derselben. Der Unternehmer hat alles dafür zu tun, dass diese Informationen sicher sind und sicher bleiben.

## 2.3 Vertraulichkeit als Teil des Qualitätsmanagements

In QM-Systemen geht es hauptsächlich um die Beziehung des Unternehmens zu seinen Kunden. Man kann die darin proklamierte „Verantwortung der Leitung“ auch auf den Schutz von Kundendaten ausdehnen. Insbesondere, wenn man sich im Rahmen von Risiko-Managementsystemen für die Einrichtung von sogenannten Compliance-Regeln entscheidet, ist es sinnvoll, diese auch im QM-System einzubetten. Sicherheit, Verfügbarkeit und Vertraulichkeit von Daten sind ein Teil der ordentlichen Leistungserbringung, die im QM-System dokumentiert wird.

# 3 Wahrung der Vertraulichkeit

Die rechtlichen Pflichten sind eine Seite der Medaille, die tatsächlichen Möglichkeiten und Verfahrensweisen sind die andere. In der betrieblichen Praxis steht die Sicherung der Vertraulichkeit oft hinter der Notwendigkeit schneller Betriebsabläufe zurück oder kurz gesagt: **Komfort geht vor Vertraulichkeit**. Auch wenn das Be-

wusstsein dafür vorhanden ist, dass Datenschutz und Datensicherheit hohe Güter sind und für den Fortbestand des Betriebs von essentieller Bedeutung, wird doch häufig aus Bequemlichkeit darauf verzichtet, Daten und Informationen nur einem begrenzten Personenkreis zugänglich zu machen. Ohne Anspruch auf Vollständigkeit seien hier einige Punkte genannt, die sich häufig beobachten lassen:

- Gemeinsame Passwörter/Benutzerkonten – alle für einen, eines für alle...
- Keine differenzierten Rechte bei der Dateiablage
- Zugänglichkeit von Räumen, Aktenordnern etc. für alle Anwesenden
- Möglichkeit Daten auf CD, USB-Stick oder andere Medien zu kopieren
- Mitnahme betrieblicher mobiler Geräte, die nicht verschlüsselt sind
- Zugriff aus Heimbüros oder von Mobilgeräten ins Firmennetz ohne Verschlüsselung oder weitere Authentifizierung
- Zugriff Dritter auf interne IT-Geräte. Stichwort: Techniker des DSL- oder Telefonie-Providers, der den Router oder die TK-Anlage warten darf

In den folgenden Abschnitten möchte ich diskutieren, mit welchen einfachen und sicheren Methoden, Maßnahmen oder Verfahren man ein Mindestmaß an Vertraulichkeit herstellen kann.

### **3.1 Organisatorische Maßnahmen**

Bevor über Maßnahmen zum Schutz der Vertraulichkeit von Daten und Informationen nachgedacht wird, gilt es, die im Betrieb zu berücksichtigenden Stufen der Vertraulichkeit festzulegen. Dabei sind sowohl die Mitarbeiter zu klassifizieren als auch die Daten. Ein Beispiel aus der Praxis: LKW-Werkstatt mit 15 MA, davon:

- 1 Betriebsleiter
- 1 Buchhaltung/Bürokräft
- 1 Bürokräft
- 1 Ersatzteilbeschaffungs- und Lagerleiter
- 2 Mitarbeiter Kunden- bzw. Auftragsannahme, Werkstattmeister
- 9 Mitarbeiter Werkstatt

Grobe Unterteilung der Daten:

- Personal- und Buchhaltungsdaten
- Finanzdaten, Kontostände, Bankzugang, Zahlungsströme

- Kundenstammdaten
- Lieferantenstammdaten
- Fahrzeugstammdaten
- Auftragsdaten
- Warenbewegungen, Einkauf, Lieferungen

In dieser Konstellation lassen sich folgende Mitarbeiter- und Datengruppen ableiten:

- *Gruppe Betriebsleitung*
  - Betriebsleiter
  - Buchhaltungskraft
  - *Daten:* Personal- und Buchhaltungsdaten
- *Gruppe Betriebsbüro*
  - Betriebsleiter
  - Buchhaltungskraft
  - Bürokraft
  - *Daten:* Finanzdaten, Kunden- und Lieferantenstammdaten
- *Gruppe Lager- und Warenbezug*
  - Betriebsleiter
  - Buchhaltungskraft
  - Lagerleiter
  - Werkstattmeister
  - *Daten:* Kunden- und Lieferantenstammdaten, Warenbewegungen, Auftragsdaten, Fahrzeugstammdaten
- *Gruppe Werkstatt*
  - Betriebsleiter
  - Werkstattmeister
  - Werkstattmitarbeiter
  - *Daten:* Fahrzeugstammdaten, Auftragsdaten

Nach dieser groben Analyse sind die vorhandenen Papier-Datenbestände so umzugliedern, wie es das Gruppenkonzept verlangt. Erst dann lassen sich organisatori-

sche Maßnahmen sinnvoll umsetzen:

- Räumliche Trennung der Ablagen (Ordner, Mappen, Schränke)
- Zugriffs- und Zutrittsschutz durch Schlüsselgruppen
- Farbliche Kennzeichnung der einzelnen Ablagen, um Fehlalagen möglichst zu verhindern oder zu erkennen. Beispielsweise Personalsachen mit rotem Rücken, Finanzdaten mit blauem Rücken. Heißer Tipp aus dem richtigen Leben: denken Sie an Farb-Fehlsichtigkeiten wie rot-grün-Blindheit.

### **3.2 Informationstechnische Unterstützung**

Wenn das organisatorische Konzept steht und konsistent ist, dann ist die Unterstützung in der IT einfach. Schwierig wird es immer dann, wenn die IT Lösungen für mangelhafte Organisation bieten soll. Erfahrungsgemäß wird das Unterfangen chaotisch bis unmöglich, nachträglich technisch zu retten, was davor nicht sauber organisatorisch geklärt wurde. Ist die Organisation klar, dann kann die IT:

- Benutzergruppen bilden
- Mitarbeiter den Gruppen zuordnen
- Gruppenspezifische Programme und Daten an Orten ablegen, auf die nur die benannten Gruppen Zugriff haben
- Gruppenbezogene Rechte innerhalb von Anwendungen vergeben
- Zugriffe auf Hardware-Ressourcen erlauben oder verhindern
- Technische Trennungen vornehmen, z.B. ein Faxgerät für die Buchhaltung und eines für die Werkstatt

Weitere technische Möglichkeiten hängen davon ab, wie die IT im Betrieb funktioniert, hier nur einige Beispiele dafür:

- Einsatz von Thinclients und Terminalservern, keine lokalen Daten
- Rechner ohne optische Laufwerke
- Sperren von USB-Ports und Speicherkartenlesern
- Mehrfache gleichzeitige Anmeldung eines Benutzers sperren
- Zeitsteuerung für die Anmeldung an bzw. die Nutzung von Geräten
- Externe Zugriffe über VPN, offene Ports vermindern
- Verschlüsselung von Datenträgern

### 3.3 Freie und Open Source Software

Vertraulichkeit in der IT hängt auch damit zusammen, dass die verwendeten Geräte, deren Betriebssysteme und die darauf aufsetzenden Programme vertrauenswürdig sind. Diverse Quellen berichten davon, dass Hardwarehersteller absichtlich Zugänge in ihren Produkten herstellen und offen lassen. Wozu und von wem diese benutzt werden, lässt sich nicht zweifelsfrei klären. Fest steht damit jedenfalls, dass die betroffenen Geräte nicht sicher als vertrauenswürdig eingestuft werden können. Im Grunde beginnt das Problem bereits beim Start eines Systems durch das BIOS (Basic Input Output System), das ein eigenes kleines Betriebssystem ist. Schon darin könnte Code enthalten sein, der das gesamte System kompromittiert. Besonders pikant dabei: das BIOS steuert die Zugriffe auf der untersten Ebene, so dass Programme zur Sicherung von Datenintegrität und Vertraulichkeit hier keine Wirkung haben, weil sie auf höheren Ebenen ansetzen.

Schwierig wird die Beurteilung der Vertrauenswürdigkeit, wenn die Betriebssysteme, Treiber und Programme geschlossene Software sind. Dann lässt sich nur noch durch die Analyse des Systemverhaltens annähernd sagen, was gerade passiert und mit wem die Technik gefragt oder ungefragt kommuniziert.

Freie und quell-offene Software stellt dazu einen Gegenpol dar. Es bedarf zu deren Beurteilung die nötige Expertise, aber grundsätzlich lässt sich bei freier Software vor deren Einsatz feststellen, ob sie vertrauenswürdig ist. Es ist extrem schwierig, freie und quell-offene Software zu kompromittieren und Schadcode darin unterzubringen, unmöglich ist es nicht. Sehr spannende Entwicklungen gibt es bei der Entwicklung hardwarenaher Programme, z.B. *coreboot* und *OpenBIOS*. Will man das technische Vertraulichkeitsmodell konsequent umsetzen, dann muss man beim BIOS beginnen, beim eigentlichen Betriebssystem fortsetzen und bei den Anwendungsprogrammen enden.

## 4 Fazit

Vertraulichkeit im (kleinen) Unternehmen ist zunächst eine Frage der Organisation. IT alleine kann keine Vertraulichkeit herstellen, sie kann deren Herstellung nur unterstützen. Der Einsatz freier und open source Software (F/OSS) kann Betrieben helfen, Vertraulichkeit zu wahren. Aber auch dazu ist die technische Expertise notwendig, um deren Einsatz gezielt vornehmen zu können. Ein Großteil der kleinen Unternehmen ist nicht in der Lage, die technische Vertraulichkeit ihrer IT-Systeme zu beurteilen. Umso bedeutender ist die organisatorisch sichere Umsetzung von Maßnahmen zum Datenschutz, zur Datensicherheit und zur Datenintegrität. Hier gilt: Organisation geht vor Technik.

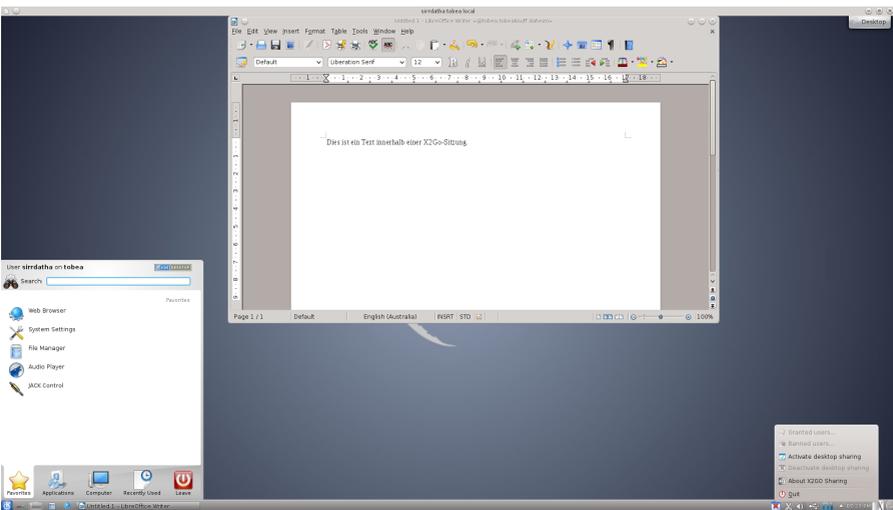
# X2Go - Einsatzmöglichkeiten für Privatanwender

Tobias Luther

tobias@x2go-community.org

x2go-community.org

X2go ist, wenn der heimische Desktop nur einen Netzanschluss entfernt ist. Per Browserplugin oder Clientapplikation auf Linux, Windows oder MacOS. Mit Ordnerfreigaben. Lokalem Druck. Sogar lokale Soundausgabe ist möglich. Wer auf den kompletten Desktop verzichten kann, startet einzelne Programme und genießt so Linux-Programme auch unter Windows.



## 1 Die beliebteste Anwendungsform: Remote Desktop

Die unter Privatanwendern beliebteste Anwendungsform von X2Go ist der Remote Desktop. Ein Rechner (X2Go-Server) bietet z.B. zu Hause aus per SSH einen Desktop an, auf den dann per Laptop über das Internet zugegriffen werden kann. Von unterwegs, von Freunden, dem Urlaubsort oder dem Hotelzimmer. So besteht immer Zugriff auf die heimische Umgebung. Auflösung und Bildschirmgröße passen sich automatisch dem Client an. Der Remote Desktop steht bereits nach Installation des Basispaketes `x2go-server` zur Verfügung. Durch Tunnelung der verschiedenen Dienste über SSH gestaltet sich auch die Dienstfreigabe über die heimische Firewall unkompliziert, lediglich der SSH-Port muß von aussen erreichbar sein (in der Ausgangskonfiguration lauscht SSH auf Port 22).

## 1.1 Für Profis: Terminalserver Thinclient-Umgebungen

Bei dieser Anwendungsform ist X2Go als PXE-Bootumgebung eingerichtet. Damit versorgt der zentrale X2Go-Terminalserver festplattenlose Clients per LAN-Boot mit einem vollständigen Betriebssystem, an dem die User zwar lokal arbeiten, das aber komplett auf dem Server ausgeführt wird. Bei entsprechender Planung kann dies den Stromverbrauch und Administrationsaufwand gegenüber einzeln zu wartenden Workstations deutlich verringern, ist aber wohl eher für öffentliche Einrichtungen und Unternehmen interessant. Wer sich hierfür interessiert findet über die unten genannten Internetlinks weiterführende Informationen zu diesem Thema. Projekte wie Linux4Afrika und Skolelinux statten nach diesem Prinzip Klassenräume aus.

## 2 X2Go - Verfügbarkeit

Die Entwicklungsplattform für X2Go ist Debian Wheezy<sup>1</sup> bzw. Debian Jessie. Das Entwicklerteam stellt für verschiedene Distributionen eigene Repositories an, auch wenn nicht jede Funktionalität sofort überall bereitgestellt werden kann. Die offiziell unterstützten Distributionen sind Debian Wheezy, Debian Jessie, Ubuntu, Fedora 20, EPEL-5 und EPEL-6. Das empfohlene und vollständigste Repository ist zur Zeit das für Debian Wheezy<sup>2</sup>.

Viele andere Distributionen haben X2Go ebenfalls im Angebot, mal mehr, mal weniger vollständig. Hier entscheiden die jeweiligen Paketierer, welche Pakete und Features sie übernehmen.

Der X2Go-Client steht für Linux, Windows und MacOS zur Verfügung.

## 3 Der X2Go-Client

Der X2Go-Client ist eine Standalone-Anwendung, die sich um die Darstellung des Remote Desktops auf dem lokalen System kümmert<sup>3</sup>. Als Minimalangaben zur Sitzungseinrichtung werden benötigt:

1. ein frei wählbarer Sitzungsname
2. Name oder IP des Hosts (entweder direkt vom X2Go-Server oder z.B. die öffentliche IP-Adresse des Routers)
3. Login-Name (und Passwort, wird beim Login-Vorgang abgefragt)

---

<sup>1</sup>das für den kurzen Installationwalkthrough in den Fussnoten als Beispielplattform dient

<sup>2</sup>echo "deb http://packages.x2go.org/debian wheezy main" » /etc/apt/sources.list

Die Installation der Repository-Schlüssel ist empfehlenswert, die Aktivierung des Pools für Quellpakete optional

```
echo "deb-src http://packages.x2go.org/debian wheezy main" » /etc/apt/sources.list
```

```
apt-key adv --recv-keys --keyserver keys.gnupg.net E1F958385BFE2B6E
```

```
aptitude update
```

```
aptitude install x2goserver
```

<sup>3</sup>aptitude install x2go-client

Per Standardeinstellung wird ein KDE-Desktop gestartet, mit dem Dropdown-Menü in der Rubrik *Sitzungsart* ist dies aber schnell auf andere installierte Desktopumgebungen abgeändert. Vorkonfiguriert sind GNOME, LXDE, XFCE, MATE und UNITY. Allerdings unterstützt X2Go keine 3D-Effekte, somit muß die jeweilige Desktopumgebung auch mit 2D funktionieren. Ist die gewünschte Umgebung nicht dabei, kann auch der Startbefehl für eine *andere Desktopumgebung* angegeben werden.

## 4 Die Funktionsweise

X2Go wird nach dem altbewährten Linux-Prinzip entwickelt: Wenn es eine Funktionalität im System bereits gibt, nutze sie. Bewährte Dienste werden gebündelt und verknüpft, so daß etwas neues, eigenständiges entsteht. Die Basis bilden die FreeNX-Libraries, um die verschlüsselte Client-Serververbindung kümmert sich *openSSH*, *SSHFS* übernimmt die Ordnerfreigaben, *Cups* ist die Druckumgebung, bei der Terminalserver - Thinclient-Umgebung kommen *TFTP*, *NFS* und *PXE* zum Einsatz u.s.w.

### 4.1 openSSH

X2Go nutzt zur Authentifizierung, Verbindungsverschlüsselung und Tunnelung weiterer Dienste openSSH. Dadurch muss in Firewalls lediglich der SSH-Port (22) freigegeben werden, um eine Verbindung zwischen Client und Server herzustellen. Zum passwortlosen Login wird ganz normal über SSH ein Schlüsselpaar generiert und der öffentliche Schlüssel am Server hinterlegt (Siehe weiter unten).

### 4.2 freeNX

Obwohl X2Go die FreeNX-Bibliotheken als Basis nutzt, war es zu keinem Zeitpunkt zu NX- oder FreeNX-Anwendungen kompatibel. FreeNX war bis zur Version 3 die Opensource-Variante des Terminalservers der Firma NoMachine, bis diese ihre Software ab der Version 4 unter eine proprietäre Lizenz stellte. Die *freexlibs* ermöglichen X2Go unter anderem die grafische Kompression der Remoteverbindungen. Auch im lokalen LAN kann so durch entsprechendes Tuning die Netzauslastung deutlich reduziert werden. Je nach Kompressionsstufe leidet natürlich mitunter die Darstellungsqualität, aber so wird auch das Arbeiten über schmalbandige Netzverbindungen möglich. Man muß Prioritäten setzen.

Mitlerweile pflegt das X2Go-Team die Bibliotheken selbstständig weiter und gilt bei Debian als FreeNX-Upstream. Im Zuge von Bereinigungen und Entfernung nicht benötigter Abhängigkeiten konnte das Entwicklerteam zum Beispiel die Paketgröße der *nx-libs* von ursprünglich 21 MB auf inzwischen 15 MB reduzieren.

### 4.3 Funktionsweise von X2Go

Bei einer Clientverbindung mit dem X2Go-Server wird ähnlich zu XNest vom `nxagent` eine eigene X-Session gestartet, die von der lokalen Bildschirmausgabe unabhängig ist. Die NX-Funktionalität beinhaltet das Anlegen eines Caches bereits übertragener Daten und Komprimierung der Netzwerkdaten<sup>4</sup>. Die Paketumlaufzeit (Round Trip Time) zwischen X-Client und X-Server wird deutlich verringert und somit die Antwortzeit des Desktops erhöht.

### 4.4 Sitzungen anhalten und wieder aufnehmen

Durch die Verwendung des `nxagent` und die von ihm gestarteten Xserver kann eine laufende Sitzung entweder durch einfaches Beenden des X2Go-Clients und späteres Neuverbinden einfach auf dem Server weiterlaufen. Dies ist auch nützlich, wenn unerwartet die Netzverbindung abbricht. Im Normalfall sollte die Sitzung ordentlich per Menü gesperrt, oder per Button im Client vorübergehend angehalten werden.

## 5 X2Go im Browser für Firefox, Chromium und Chrome

Ein sehr stark nachgefragtes Feature ist der Zugriff auf den X2Go-Desktop mit dem X2Go-Browserplugin. Dies ist ein abgespeckter X2Go-Client, der im Browserfenster gekapselt den Desktop quasi als Webseite zur Verfügung stellt. Die Session läuft - direkt im Browser, kann aber auch abgekoppelt oder in den Vollbildmodus geschaltet werden.

Serverseitig setzt dies einen installierten und funktionsfähigen Webserver wie z.B. Apache voraus. Direkt nach der Installation vom `x2goplugin-provider`<sup>5</sup> kann die X2Go-Loginseite über `http://server/x2goplugin.html` aufgerufen werden. Eventuell muss auf dem Client das Browserplugin noch installiert werden<sup>6</sup>, bei manchen Distributionen ist es bereits im X2Go-Client enthalten<sup>7</sup>.

---

<sup>4</sup>Auf dem Server kümmert sich der `nxagent` um die Bereitstellung und Komprimierung der X-Funktionalität, ihm nachgeschaltet ist der `nxproxy` client und auf z.b. dem lokalen Laptop der `nxproxyserver`. Dieser cached übertragene Daten ähnlich wie es auch jeder normale Internetproxy für http kann und tunnelt die Übertragung der Xserverdaten vom X2Go-Server zum X2Go-Client.

<sup>5</sup>`aptitude install x2goplugin-provider`

danach sind in der Datei `/usr/share/x2go/plugin/x2goplugin.html` die Werte für `server=localhost` und `command=XFCE` den eigenen Gegebenheiten anzupassen.

<sup>6</sup>`aptitude install x2goplugin`

<sup>7</sup>Die automatische Konfiguration des Webservers für `servername/x2goplugin.html` beeinträchtigt nicht die normale Webroot `/var/www`, natürlich kann `x2goplugin.html` auch dort oder in einem anderen Verzeichnis abgelegt werden. Dann sollte aber auch der Link in `/etc/apache2/conf.d` und die Datei in `/etc/apache2/conf-available` gelöscht werden.

## 6 Zusatzfunktionalitäten

Eine der Stärken von X2Go sind Features wie die Ordnerfreigabe, lokaler Druck oder die gemeinsame Desktopnutzung. Je nach verwendeter Desktopumgebung sollte eines der folgenden Pakete installiert werden, um die Integration von X2Go in den jeweiligen Desktop zu erhöhen: *plasma-widget-x2go* (KDE4), *x2golxdebindings* (LXDE), *x2gognomebindings* (Gnome) oder als generische Lösung für freedesktop.org kompatiblen Umgebungen *x2goserver-fmbindings*<sup>8</sup>.

Unter KDE stellt z.B. das Plasma-Widget KDE ein Icon zur Verfügung, über das Dateifreigaben vom lokalen Computer per Mausklick im Standard-Dateimanager der Wahl geöffnet werden können.

### 6.1 Passwortloses Login

Eigentlich ist das passwortlose Login kein Feature von X2Go, sondern von openSSH. Mittels *ssh-keygen* wird ein Schlüsselpaar erzeugt und der öffentliche Schlüssel mit *ssh-copy-id user@remote-server* auf den Server kopiert. Der X2Go-Client benötigt normalerweise keine weitere Konfiguration, er probiert in der Standardeinstellung zunächst die Authentifizierung per Schlüssel. Erst wenn er hier nicht erfolgreich ist, wird der Dialog zur Passworтеingabe angezeigt.

### 6.2 Desktop Sharing

Desktop Sharing meint die gemeinsame Nutzung ein und derselben Desktopsitzung. Hierfür muss das Paket *x2godesktopsharing* installiert sein<sup>9</sup>, und die Benutzer der Gruppe *x2godesktopsharing* angehören<sup>10</sup>.

Nach Start des Programms *desktopsharing* z.B. über das Programmenü kann der User den geteilten Zugriff auf seinen Desktop per Mausklick auf das Icon in der Taskbar genehmigen.

Um auf die Sitzung eines anderen Users zugreifen, wählt man im Sitzungsprofil des X2Go-Clients *Zugriff auf lokalen Desktop*. Während dem Sitzungsstart werden offene X2Go-Sitzungsdaten angezeigt, die Sitzung des gewünschten Users kann mit den Rechten *nur betrachten*, oder *Vollzugriff* gestartet werden. Bevor die Verbindung hergestellt wird, muß der Zugriff gewährende User dies per Dialogabfrage per Mausklick genehmigen. Das Icon von *X2Go Desktopsharing* informiert über den Gastzugriff, Fortdauer der geteilten Sitzung und meldet, wenn der andere Gastuser die Sitzung beendet.

---

<sup>8</sup> `aptitude install plasma-widget-x2go` oder eines der anderen Pakete

<sup>9</sup> `aptitude install x2godesktopsharing`

<sup>10</sup> `adduser username x2godesktopsharing`, zur Aktivierung müssen sich die User evt. neu einloggen

### 6.3 Verbindung zum eigenen lokalen Desktop auf dem Server

X2Go kann auch die Verbindung zu einem lokal geöffneten Desktop herstellen. Die Vorgehensweise ist wie beim Desktop Sharing, nur daß bei identischem User die SSH-Authentifizierung ohne zusätzliche Genehmigung durchgereicht wird. Im Unterschied zu nativen X2Go-Sessions kann die Auflösung natürlich nicht automatisch angepasst werden, idealerweise sind die Auflösungen von Server und Client aufeinander abgestimmt, die Fenstergröße wird per Hand angepasst. Die Sitzung muß allerdings aktiv und sein, X2go kann keine neue lokale Sitzung starten. Außerdem gilt es zu bedenken, daß auf dem entfernten Rechner die benutzte Desktopsitzung aktiviert sein muß und somit am Bildschirm zu sehen ist. Ist z. B. eine Konsole offen, ist mit X2Go kein Zugriff möglich. Um eine Sitzung sicher z. B. am Arbeitsplatz und über das Internet zu nutzen, sollte evt. eine X2Go-Sitzung lokal gestartet werden, um sie später über das Internet wieder aufzunehmen.

### 6.4 Ordnerfreigabe zwischen den Systemen

Die Ordnerfreigabe vom lokalen System zum entfernten Desktop wird über SSHFS ermöglicht. Da dies eine Basisfunktion des X2Go-Server ist, ist keine weitere Paketinstallation nötig, allerdings sollten die beteiligten User Mitglieder der Gruppe *fuse*<sup>11</sup> sein. Die freizugebenden lokalen Ordner können über das Menü vom X2Go-Client ausgewählt oder die Sitzungseigenschaften werden. Automatische Freigaben sind möglich.

### 6.5 Lokaler Druck

Es ist über X2Go möglich, z.B. ein Textdokument auf dem entfernten Desktop zu bearbeiten, um es anschließend am lokalen Drucker auszugeben. Hierfür werden die Pakete *x2goserver-printing* und *cups-x2go* benötigt<sup>12</sup> und die User müssen Mitglieder der Gruppen *fuse* und *x2goprint* sein<sup>13</sup>. Über die normale Druckerkonfiguration wird nun ein neuer *generischer Drucker* hinzugefügt<sup>14</sup>. Nach einem Neulogin steht dem User der neue (virtuelle) Drucker zur Verfügung, bei dessen Verwendung am Clientsystem (auch unter Windows und Mac) ein Dialogfenster erscheint und entweder die Anzeige des Dokuments als PDF anbietet, oder den Druck mit einem lokal installierten Drucker ermöglicht.

---

<sup>11</sup>adduser username fuse

<sup>12</sup>aptitude install x2goserver-printing cups-x2go

<sup>13</sup>adduser username fuse ; adduser username x2goprint

<sup>14</sup>z.B. via Webinterface am Server: localhost:631 => Drucker hinzufügen => CUPS-X2Go (Virtual X2Go Printer) => Generic Printer => Generic Cups-X2Go Printer

## 6.6 Anwendungen veröffentlichen

X2Go bietet nicht nur die Möglichkeit, den gesamten Desktop auf Clientsystemen zu nutzen, sondern auch einzelne Anwendungen freizugeben. Diese Funktionalität benötigt keine Installation weiterer Pakete, sondern wird über Symlinks zu *.desktop-Dateien* in `/etc/x2go/applications` auf dem Server gesteuert<sup>15</sup>. Im X2Go-Client bleibt der Verbindungsaufbau zum Desktop scheinbar stehen, aber über den ganz linken Button werden die veröffentlichten Anwendungen aufgerufen.

## 6.7 Anwendungen vom Server transparent per Desktopicon auf dem Client starten

Der Unterschied zu veröffentlichten Anwendungen ist, daß dem User kein Menü präsentiert wird, sondern der Programmstart vollkommen transparent per Iconklick geschehen kann. Voraussetzung ist die Einrichtung des passwortlosen Logins (s.o.). Zusätzlich sollte im X2Go-Client für jedes Programm eine neue Sitzung eingerichtet werden, in deren Eigenschaften der Punkt *Anwendung* ausgewählt und im dafür vorgesehenen Feld der komplette Pfad und Name der auszuführenden Datei angegeben wird. Das wäre z.B. `/usr/bin/xterm`. Das dazugehörige Desktopicon wird über das Menü des X2Go-Clients erzeugt. Nur bei Bestätigung des transparenten Startmodus wird das Clientfenster nicht mitgestartet.

## 7 Stable, LTS und die nightly builds

Von X2Go gibt es insgesamt drei parallel gepflegte Versionen: Stable, LTS (Long Time Support, Codename "Baikal") und das Entwicklungsrepository für nightly builds (Codename "Heuler"). Die hier verwendete Version ist Stable. Baikal oder auch LTS soll in die Debian-Distribution aufgenommen werden. Hier liegt der Schwerpunkt mehr auf Stabilität denn auf neuen Features.

Wann das Ziel der offiziellen Debianintegration erreicht wird ist momentan noch nicht klar. Aktuell ist nur der Client verfügbar.

## 8 Über das Private hinaus: X2Go in Produktivumgebungen

X2Go eignet sich nicht nur für die private Nutzung, sondern auch für den Einsatz in Produktivumgebungen. Sowohl Bildungseinrichtungen, Bibliotheken, Internetcafes als auch Firmen, Stadtverwaltungen und Konzernen sind denkbare Umfelder. X2go ist skalierbar, verfügt über Features zur Lastverteilung (der X2Go-Session Broker), ist redundanzfähig und spricht selbstverständlich auch mit einem LDAP-Server. Das Projekt Linux4Afrika verwendet zum Beispiel X2Go, um auch von relativen Laien

<sup>15</sup>In `-s /usr/share/applications/anwendung.desktop /etc/x2go/applications/`

wartbare Thinclient-Umgebungen an afrikanischen Schulen einzurichten. Ein Lehrer kann hier zum Beispiel per einfachem Shellscript den Userstamm in der LDAP-Datenbank komplett neu aufbauen.

Zur weiteren Absicherung der Authentifizierung kann beispielsweise über USB-Sticks oder GPG-Karten mit entsprechenden Lesegeräten zurückgegriffen werden. Weitere Informationen über die unten genannten Weblinks.

## 9 Ressourcen

<http://www.x2go.org>

Projekthomepage, inklusive Wiki

<http://www.x2go-community.org>

Vorwiegend deutschsprachiges Supportforum

## 10 Partnerprojekte

<http://www.linux4afrika.de>

Linux4Afrika richtet mit gespendeter Hardware Terminalserver - Thinclient Netzwerke in afrikanischen Schulen ein, natürlich mit X2Go

<http://www.skolelinux.de>

Debianbasierte Distribution für Schulen und Klassenräume

## 11 Lizenz dieses Textes

Creative Commons: Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International (CC BY-SA 4.0)

<http://creativecommons.org/licenses/by-sa/4.0/deed.de>



## 3 Zusammenfassungen der weiteren Vorträge

### 3.1 20 Dinge über Verschlüsselung, die Sie schon immer wissen (w|s)ollten

*Peer Heinlein, Heinlein Support GmbH, p.heinlein@heinlein-support.de*

Von Verschlüsselung wird häufig geredet, und so grob haben die meisten auch eine Ahnung, wie sie funktioniert. Aber es lohnt sich, das ganze Thema mal im Detail zu betrachten: Was funktioniert wie und warum?

Dieser Vortrag klärt alle möglichen bunten Fragen rund um das Thema «Verschlüsselung»: Echte Fragen, historische Fragen, Scherzfragen, ernstgemeinte Fragen, technische Fragen, Glaubensfragen, Prüfungsfragen, Sicherheitsfragen und und und. Halt all das, was man schon immer mal zu diesem Thema wissen wollte – oder eben auch sollte.

Weitere Informationen: <https://www.heinlein-support.de/vortrag/dinge-ueber-verschluesselung-die-sie-schon-immer-mal-wissen-wollten>

### 3.2 Aktuelle Entwicklungen beim Linux-Kernel

*Thorsten Leenhuys, c't / heise online / heise open, thl@ct.de*

Der Vortrag gibt einen Überblick über die jüngsten Verbesserungen beim Linux-Kernel, denn die sind oft auch für Allerwelts-PCs oder Server von Belang; mit Distributionen wie Ubuntu 14.04 erreicht der aktuelle Kernel in Kürze auch eine breite Anwenderschar.

Der Vortrag geht auch auf einige Neuerungen bei Kernel-naher Software ein – etwa bei den Open-Source-3D-Grafiktreibern. Angerissen werden auch einige noch in Vorbereitung befindliche Änderungen, der Entwicklungsprozess sowie andere Aspekte rund um den Kernel, die für die kurz- und langfristige Entwicklung von Linux und Linux-Distributionen wichtig sind.

Weitere Informationen: <http://www.heise.de/open/kernel-log-3007.html>

### 3.3 AMaViS und Kaspersky vs. Malware

*Roland Imme*

Wo geht die Reise hin? Wir stellen die Trends der Malware-Strategien vor. Es gibt unterschiedliche Infektionswege, die von Malware genutzt werden. Was steht hinter dem AMaViS-Projekt und dem Unternehmen Kaspersky Lab? Wir geben Antworten auf die Frage, wie AMaViS und Kaspersky gemeinsam erfolgreich im Kampf gegen Malware eingesetzt werden können.

### **3.4 Amazon Linux – Betriebssystem für die Cloud**

*Chris Schlaeger, Amazon Web Services*

Amazon Web Services (AWS) ist der weltweit führende Anbieter von Cloud-Diensten. Amazon Linux ist das Betriebssystem, welches zum einen die Grundlage für viele dieser Dienste darstellt, zum anderen die optimale Basis für eigene Cloud-Dienste bildet. Dieser Vortrag wird einen kurzen Überblick über die Geschichte von Amazon Linux geben und erläutern, wie man es als Grundlage für eigene Cloud-Dienste und -Anwendungen verwenden kann. Dieser Vortrag richtet sich an alle, die einen Blick hinter die Kulissen von AWS werfen möchten oder eigene Anwendungen oder Dienste in der Cloud anbieten wollen.

Weitere Informationen: <http://aws.amazon.com/de/amazon-linux-ami/>

### **3.5 Anwendung, Implementierung und Sicherheit von Kryptografie zur Datei- und Sprachverschlüsselung**

*Hartmut Luge, Hochschule Mittweida*

Eine leicht verständliche Kurzeinführung vermittelt die Verfahren der Kryptologie anhand der historischen Entwicklung. Aufbauend auf aktuellen Verschlüsselungsverfahren der symmetrischen Kryptographie werden Grundsätze zum Aufbau einer sicheren Datei- und Datenverschlüsselung dargestellt und eine praktische Realisierung erläutert. Dabei werden insbesondere Sicherheitslücken aufgrund praktischer Implementierung in Gerätetechnik oder Software aufgezeigt. Für die Übertragung von Sprachsignalen werden Grundkonzepte zur analogen und digitalen Sprachsignalverschlüsselung und mögliche Realisierungsansätze vorgestellt.

### **3.6 Automatisierte Systemkonfiguration mit Capistrano und Puppet**

*Christian Patsch, GONICUS GmbH*

Puppet bietet auch einige Jahre nach seiner Veröffentlichung verschiedene Wege zur Umsetzung in RZ-Umgebungen. Im Gegensatz zum zentralisierten Ansatz durch Verwendung des «puppetmaster» können technische oder organisatorische Anforderungen auch eine dezentrale Lösung erfordern. Ein Ansatz ist die Verwendung von Puppet mit Capistrano, welcher durch Beispiele aus einem umgesetzten Projekt präsentiert wird und aufzeigt, welche Vor- und Nachteile zu beachten sind. Ein Vergleich mit anderen etablierten Ansätzen soll zusätzlich Empfehlungen für zukünftige Einsatzszenarien geben.

### **3.7 B.A.T.M.A.N. Beginners: WLAN-Meshing für Einsteiger**

*Amadeus Alfa, Freifunk Chemnitz e.V., amadeus@chemnitz.freifunk.net*

Statisches Routing, harte Verdrahtung? Nicht mit B.A.T.M.A.N.! Wir stellen eine Software vor, die den Aufbau dynamischer Netze revolutioniert hat. Aber es geht nicht nur um kabellose Vernetzung. Auch auf die Kombination mit drahtgebundenen Zugangsmedien wird im Vortrag eingegangen. Es werden die theoretischen Grundlagen vorgestellt und in einer Live-Demo in die Praxis umgesetzt.

### **3.8 Best of Geany-FAQ – alles was ihr schon immer über den Editor wissen wolltet**

*Frank Lanitz, frank@geany.org*

Geany ist seit Jahren regelmäßig mit einem Stand bei verschiedenen Linux-Tagen vertreten. Dabei stellen die Besucher oftmals die gleichen Fragen und sind von den Antworten überrascht.

Dieser Vortrag soll eine Übersicht über die wichtigsten Funktionen des Editors geben und dabei die eine oder andere regelmäßig gestellte Frage beantworten. Ferner soll ein kleiner Ausblick in die Zukunft gewagt werden.

Weitere Informationen: <http://geany.org>

### **3.9 Ceph und Gluster im Vergleich**

*Robert Sander, Heinlein Support GmbH, r.sander@heinlein-support.de*

Ceph und Gluster sind zwei verteilte Storage-Systeme. Der Vortrag zeigt Ähnlichkeiten und Unterschiede in Konzepten und technischer Umsetzung. Es wird ein Performance-Vergleich vorgestellt und auf die Handhabung in der Administration eingegangen.

### **3.10 CloudStack – Aufbau und Struktur**

*Stephan Seitz, Heinlein Support GmbH, s.seitz@heinlein-support.de*

Der Vortrag stellt den Aufbau und die Möglichkeiten beim Betrieb einer privaten IaaS-Plattform mit Apache CloudStack vor.

Weitere Informationen: <http://cloudstack.apache.org/>

### **3.11 Das Debian-LAN Projekt: Installation eines Debian-Netzwerks einfach gemacht**

*Andreas Mundt, Debian-LAN Project, andi@debian.org*

Der Vortrag präsentiert den Status, Konzepte und Techniken des Debian-LAN(Local Area Network)-Projekts.

Debian-LAN vereinfacht das Ausrollen einer kompletten Systemumgebung im lokalen Netzwerk erheblich, ohne dabei die Flexibilität einzuschränken. Es erlaubt ein komplettes Debian-basiertes Netzwerk mit Kerberos, zentraler Nutzerverwaltung, Diskless- und Roaming-Clients u. v. m. direkt zu installieren.

Das System findet Verwendung in Schulen, Arbeitsgruppen, Vereinen und kleinen Unternehmen oder bei der Installation von komplexen Testumgebungen.

Weitere Informationen: <https://wiki.debian.org/DebianLAN>

### **3.12 Data Leakage Protection: Zukünftige Herausforderungen zur Sicherung von Vertraulichkeit**

*Steffen Wendzel, Fraunhofer FKIE, Bonn*

Der Vortrag stellt das Problem «Data Leakage» vor und erläutert seine Ursachen. Dabei wird verdeutlicht, dass derzeit kaum hinreichende Schutzmaßnahmen gegen Datenexfiltration in Unternehmen vorhanden sind. Der Vortrag endet mit einem Ausblick auf zukünftige Technologien der Datenexfiltration, die derzeit erst erforscht werden, und legt dar, weshalb akuter Handlungsbedarf bei der Erforschung von Gegenmaßnahmen gegeben ist.

Weitere Informationen: <http://www.wendzel.de>

### **3.13 Datenintegration im Service – mit OTRS die passenden Daten für den Service-Prozess liefern!**

*Rico Barth, c.a.p.e. IT GmbH, [info@cape-it.de](mailto:info@cape-it.de)*

Der Vortrag betrachtet das Kunden- und IT Service Management (ITSM) sowie die Unterstützung dieser Prozesse durch die gezielte Integration zusätzlicher Datenquellen. Der Fokus liegt dabei auf der Erläuterung praktischer Projektbeispiele, die zeigen, wie flexibel mit der Open-Source-Applikation OTRS die Service-Abteilung mit den passenden Daten in mittelgroßen und großen Umgebungen unterstützt werden kann.

Weitere Informationen: <http://www.kix4otrs.de>

### **3.14 Den Schlapphüten die Ohren verstopfen – Transportverschlüsselung für alle**

*Alexander Schreiber, Google Switzerland GmbH, [als@thangorodrim.ch](mailto:als@thangorodrim.ch)*

Wer im Klartext digital kommuniziert, weiß spätestens seit 2013, dass die einschlägigen Dienste im Rahmen des technisch Möglichen mitlesen.

Verschlüsselung schützt.

Der Vortrag gibt einen Überblick und eine Einführung in die Möglichkeiten, Webtraffic und Mailtransport zu sichern, sowie zum einfachen Aufbau eines verschlüsselten Virtual Private Networks zum sicheren Zugriff auf eigene Systeme aus der Ferne. Angesprochen werden dabei sowohl Theorie als auch praktischer Einsatz an konkreten Beispielen.

### **3.15 Der Linux-Multimediastack**

*Lucas Stach, Pengutronix, [info@pengutronix.de](mailto:info@pengutronix.de)*

Der Linux-Multimediastack wirkt von außen betrachtet sehr komplex, ist allerdings auch sehr mächtig und flexibel. Gerade bei eingebetteten Systemen mit relativ geringer CPU-Leistung ist es wichtig, Hardwarebeschleunigungseinheiten wie De- und Encoder sowie GPUs effizient einzubinden. Dafür wurde in den letzten Jahren sowohl im Userspace als auch im Kernel viel neue Infrastruktur geschaffen.

Wer schon immer einmal wissen wollte, wie GStreamer, Video4Linux, Mesa3D und andere Komponenten zusammenarbeiten, um sowohl Laptop wie auch kleinsten Systemen FullHD Video zu entlocken, ist hier genau richtig.

### 3.16 Deutschlands Sicherheitspolitik im Cyberspace

*Jakob Kullik, Lehrstuhl für internationale Politik der TU Chemnitz*

Zahlreiche neue Phänomene wie Cyberkriminalität, Cyberspionage und Cybersabotage sind mittlerweile zu ernsthaften Bedrohungen für Wirtschaft, Politik und kritische Infrastrukturen geworden. Konkurrenten im Cyberspace sind nicht nur China und Russland, sondern auch unsere transatlantischen Partner USA und Großbritannien. Der Beitrag will zeigen, welchen Stellenwert Cybersicherheit in der Sicherheitspolitik Deutschlands hat, welche Ministerien und Behörden in der IT-Sicherheit beteiligt sind und welche IT-Fähigkeiten Bundeswehr und Nachrichtendienste besitzen, um Deutschland vor IT-Bedrohungen schützen zu können.

### 3.17 Die «Deutsche Wolke»: Open Source Cloud für den Mittelstand

*Kerstin Mende-Stief, Open Source Business Alliance*

Die Working Group «DEUTSCHE WOLKE» unterstützt den Aufbau einer föderalen Cloud-Infrastruktur in Deutschland mit offenen Standards und Schnittstellen nach den geltenden Datenschutzrichtlinien (BDSG). Das Fundament bilden Rechenzentren in ganz Deutschland. Ein Showcase veranschaulicht die Architektur der kompletten Cloud-Infrastruktur (XaaS). Innovatives Communitypayment und eigene Apps grenzen die Lösung von anderen Anbietern ab. 2013 wurde die «Secure Phone Alliance» mit dem Ziel, das erste sichere Open Source Smartphone zu entwickeln, gegründet. Der Vortrag enthält Anwendungsbeispiele und richtet sich an mittelständische Unternehmen.

Weitere Informationen: <http://www.deutsche-wolke.de>

### 3.18 Die private Cloud mit ownCloud

*Klaas Freitag, ownCloud Projekt, [freitag@owncloud.org](mailto:freitag@owncloud.org)*

Hinter dem Stichwort «Cloud» steckt mehr, als das Buzzword auf den ersten Blick vermittelt. Besonders das Synchronisieren und Teilen von Daten in der Cloud ist Standard geworden. Doch seit der NSA-Affäre ist klar, dass die Datenablage in der Cloud neu bewertet werden muss.

Das ownCloud-Projekt arbeitet an der private Cloud Software ownCloud, die den Komfort der Cloud für Anwender unter kontrollierten und sicheren Bedingungen ermöglicht.

Die Präsentation berichtet über den Stand des Projektes und erläutert Features und Möglichkeiten für Privat- und Firmenanwender.

### 3.19 Die Technik des elektronischen Personalausweises

*Jörg Schilling, Fraunhofer FOKUS*

Der Vortrag erklärt die Kommunikation mit dem Ausweis mit PACE (einem modifizierten Diffie-Hellman-Schlüsseltausch), die Berechtigungsprüfung mit Hilfe der Zertifikate bei der Terminal-Authentifikation, die Chip-Authentifikation zum Schutz vor gefälschten Chips sowie die Kommunikation mit einem eID-Server bei Benutzung des Ausweises im Netz.

Sicherheitsaspekte der verwendeten Kryptografie, der Kommunikation und Nutzung werden diskutiert.

### 3.20 E-Mail Made in Germany – steckt da was dahinter?

*Peer Heinlein, Heinlein Support GmbH, p.heinlein@heinlein-support.de*

«E-Mail Made in Germany» war vergangenen Sommer die Marketing-Antwort der Branchengrößen T-Online, GMX und web.de auf die NSA-Affäre. Doch was steckt dahinter? Oder anders: Steckt da was dahinter?

Dieser Vortrag beleuchtet, was «E-Mail Made in Germany» eigentlich bedeutet, wer welche Interessen dabei verfolgt und was das ganze auf technischer Ebene bedeutet. Überraschenderweise ist eine TLS-Verschlüsselung zwischen Providern gar nicht mal so einfach sicherzustellen. Wir werden nicht umhin kommen, uns hier mit etlichen technischen Fragen rund um TLS, DANE, DNS und DNSSec zu beschäftigen.

### 3.21 Effiziente Kommunikation und Arbeit im IT-Team

*Peer Heinlein, Heinlein Support GmbH, p.heinlein@heinlein-support.de*

In jedem Team knackt es, richtig rund läuft es selten. Gute Arbeit der Administratoren wird durch Ineffizienz und Reibereien wieder kaputt gemacht. Oft wird geschimpft, öfters noch resigniert. Administratoren ersticken in Arbeit und haben das Gefühl, nie fertig zu werden (werden Sie ja auch nicht!).

Dieser Vortrag zieht ein Resümee aus 20 Jahren Consulting-Arbeit und Team-Führung. Er liefert mehr Anregungen als ultimativ gültige Patentrezepte. Aber er bringt konkrete Beispiele, wie man Aufgaben, Absprachen und Vorgehensweisen anders und effizienter organisieren könnte.

Weitere Informationen: <http://www.heinlein-support.de>

### 3.22 Einführung in SSL mit Wireshark

*Martin Kaiser*

SSL ist das «s» in https und damit das wohl am häufigsten verwendete Security-Protokoll im Internet.

Dieser Vortrag bietet eine praxisorientierte Einführung in SSL und stellt die grundlegenden Eigenschaften und Konfigurationsmöglichkeiten vor. Ein Schwerpunkt liegt

auf der Verwendung von Wireshark zur Analyse von SSL-Verbindungen. Es wird gezeigt, wie man den Verbindungsaufbau, die Datenübertragung und die verwendeten Parameter ausliest und wie sich aufgezeichnete SSL-Verbindungen nachträglich entschlüsseln lassen, um Verbindungsprobleme zu untersuchen.

### **3.23 Entfernt einloggen – Grundlagen der SSH-Nutzung**

*Axel Beckert, ETH Zürich, [abe@debian.org](mailto:abe@debian.org)*

Viele Endbenutzer kennen SSH nicht, aber für den Unix-Server-Administrator ist es das wichtigste Arbeitsmittel schlechthin, kommt er doch meist ausschließlich über diesen Weg auf den Server, um ihn vom Arbeitsplatz aus zu administrieren. Auch wenn man einen Root-Server oder virtuellen Server gemietet hat, ist SSH meistens die einzige Administrationsmöglichkeit.

Der Vortrag soll eine Übersicht geben, wie SSH generell arbeitet, wie SSH-Authentifizierung mit digitalen Schlüsseln funktioniert, wie man einzelne Netzwerk-Ports per SSH weiterleiten kann, sowie kurz aufzeigen, was man damit noch alles machen kann.

Weitere Informationen: <http://noone.org/talks/ssh-tricks/>

### **3.24 Geotagging: Fotos mit Geoinformationen verknüpfen**

*Frank Hofmann, Hofmann EDV, [frank.hofmann@efho.de](mailto:frank.hofmann@efho.de)*

Nach jeder Reise gleicht das Sortieren der Mitbringsel und Zuordnen der Fotos häufig einem anspruchsvollen Puzzlespiel. Auch unsere Mitmenschen danken es uns, wenn sie später nicht nur endlose Pixelberge gezeigt bekommen, sondern die Fotos auf einer Landkarte mitverfolgen und geographisch einsortieren können. Im Mittelpunkt stehen Linux-Bordmittel und frei verfügbare Dienste wie OpenStreetMap, Nominatim, bbbike und GottenGeography.

### **3.25 Graylog 2 – Log-Management einfach gemacht**

*Klaus Kruse, Corpex Internet GmbH, [mail@klaus-kruse.de](mailto:mail@klaus-kruse.de)*

Serverdienste sind geschwätzig – und sollen es auch sein. In Logs werden Startvorgänge, Anmeldeversuche, Fehler und Zugriffe protokolliert und stehen für spätere Analysen zur Verfügung. Mit einer wachsenden Anzahl von Diensten und Servern wächst die Schwierigkeit, darüber den Überblick zu behalten.

Mit Graylog 2 steht eine weitere Log-Management-Lösung zur Verfügung, die auch mit einer komfortablen Web-Oberfläche daher kommt. Im Vortrag werden die Einrichtung, die Anbindung an Syslog und schließlich auch die Möglichkeit der graphischen Auswertung präsentiert.

Weitere Informationen: <http://graylog2.org>

### 3.26 Hier geht nix rein! Storage Performance im Virtualisierungsumfeld

*Michael Ziegler, it-novum GmbH, michael@open-attic.org*

RAID scheint obsolet: Einerseits funktioniert es bei aktuellen Plattengrößen schlicht nicht mehr. Andererseits gibt es Tools wie Ceph, LVM und ZFS, die RAID ersetzen können. Bei Performance-Messungen reizt ZFS Storage Controller und Festplatten mühelos bis ans Limit aus. Oftmals wird dazu aber nur die Durchsatzrate betrachtet. Im Virtualisierungsumfeld ist das jedoch überhaupt nicht aussagekräftig. Dieser Vortrag benennt die wirklich wichtigen Kriterien und vergleicht die Performance von ZFS und RAID in verschiedenen Szenarien.

### 3.27 High Availability und Disaster Recovery: Metro Storage Cluster mit ZFS

*Peter Großöhme, MEGWARE Computer Vertrieb und Service GmbH, peter.grossoehme@megware.com*

Der Vortrag vermittelt die Grundlagen sowie Voraussetzungen zum Betrieb eines standortübergreifenden Storage Clusters auf Basis von Nexenta MetroHA. Im Anschluss daran werden das Dateisystem ZFS näher beleuchtet sowie verschiedene Konfigurationen und Einsatzmöglichkeiten zum Betrieb aufgezeigt. Zum Abschluss des Vortrags erfolgt eine Remote-Live-Demonstration des Systems, das physisch am MEGWARE-Stand besichtigt werden kann.

### 3.28 I got root – I can read your mail

*Martin Neitzel, Gaertner Datensysteme, neitzel@gaertner.de*

Alle Welt redet über die böse NSA, aber in wie weit sind eigentlich Ihre vertraulichen Daten vor Ihrem eigenen Administrator sicher, der *nur ein Zimmer weiter sitzt*? Oder der ein externer Berater ist?

Dieser Vortrag richtet sich an Endanwender, die Linux nur nutzen möchten (oder müssen) und mit dem ganzen Administrationskram dabei am liebsten gar nichts zu tun haben wollen. Es wird deshalb «so wenig wie möglich, aber so viel wie nötig» erklärt, um die Vertraulichkeit der Dateien und Mails gegenüber Administratoren richtig einschätzen und entsprechend organisieren zu können.

Weitere Informationen: <http://gaertner.de/neitzel/clt/root/>

### 3.29 Icinga 2 – Secure Cluster Stack Monitoring and More

*Bernd Erk, ICINGA – Open Source Monitoring, bernd.erk@icinga.org*

One of the best places to spy on an IT environment would be via its monitoring system. Beyond housing user, application and performance data, encryption in NSCA or NRPE distributed systems are often forgotten. This talk will present Icinga 2 and the security considerations made in its development. With SSL cluster communication by default, and an ACL (access control list), Icinga 2 makes secure distributed monitoring easy. In a live demo we'll set up clusters that can be managed per individual

domains, such that both users and checks can be restricted to specific systems. We'll flaunt Icinga 2's feature set and future development plans.

Weitere Informationen: <http://www.icinga.org>

### 3.30 Installation und Arbeiten mit einer (La)T<sub>E</sub>X-Distribution unter Linux

*Herbert Voß, Freie Universität Berlin, [herbert@dante.de](mailto:herbert@dante.de)*

Eine T<sub>E</sub>X-Distribution besteht aus Tausenden von Einzeldateien, die es für die standardmäßigen Paketmanager unter Linux fast unmöglich machen, es zum einen in sinnvolle kleine Einheiten zusammenzufassen und zum anderen Updates aktuell zu halten. In diesem Forumbeitrag wird gezeigt, wie man sich am aktuellen Paketmanager vorbei eine T<sub>E</sub>XLive-Installation erstellen kann, die tägliche Updates erlaubt (SUSE/Debian).

### 3.31 Introduction to Software Collections

*Miro Hrončok, Red Hat, [mhroncok@redhat.com](mailto:mhroncok@redhat.com)*

Software Collections is a way to concurrently install multiple versions of specific software on the same system without affecting standard software packages that are installed on the system with the classic RPM package manager. Software Collections consist of several components, which are connected and distributed together in order to provide their full functionality without conflicting or overwriting system files.

Weitere Informationen: <https://fedorahosted.org/SoftwareCollections/>

### 3.32 Keine Angst vor den Befehlen – die Welt der Linux-Kommandozeile

*Holger Trapp, TU Chemnitz, URZ, [hot@hrz.tu-chemnitz.de](mailto:hot@hrz.tu-chemnitz.de)*

Linux bietet neben modernen Desktops eine leistungsfähige Shell-Schnittstelle mit Unix-Werkzeugkasten, bei der man das System durch geschickt kombinierte textuelle Kommandos steuert. Sie wird wegen ihrer Mächtigkeit von erfahrenen Anwendern sowohl interaktiv als auch bei der Automatisierung von Abläufen durch Skripte rege genutzt, oft im Zusammenspiel mit der grafischen Oberfläche. Der Vortrag unternimmt einen Abstecher in die Welt der Befehlsschnittstelle und möchte anhand praktischer Beispiele ein Gefühl für deren Philosophie und sinnvolle Anwendungsgebiete vermitteln.

Weitere Informationen: <http://www-user.tu-chemnitz.de/hot/LT/2014/>

### 3.33 Kerberos – sichere Authentifizierung seit 30 Jahren

*Markus Schade, Hetzner Online AG*

Kerberos gibt es als Authentifizierungsinstanz seit rund 30 Jahren. Anders als bei der klassischen Passwortheingabe weist sich hier auch der Server bzw. Service gegenüber dem Nutzer aus. Ganz nebenbei erhält man als Zugabe ein System für Single Sign On. Der Vortrag stellt Kerberos vor und geht auf die Grundlagen zur Einrichtung ein.

### **3.34 KMS UXA DRM OMG WTF BBQ – Durchblick im Linux-Grafikdschungel**

*Martin Fiedler, Dream Chip Technologies GmbH*

Noch viel verwirrender als die viel gescholtene Kommandozeile ist inzwischen die Grafiklandschaft unter Linux: X11 kennt man ja noch, aber was hat es mit den designierten Nachfolgern Wayland und Mir auf sich? Und was bedeuten die ganzen kryptischen Abkürzungen wie KMS und DRM? Gerade letzteres klingt ja nun gar nicht so toll, das will man doch nicht auf seinem Linux-Rechner haben ... oder etwa doch? Dieser Vortrag versucht, ein wenig Licht ins Dunkel zu bringen.

### **3.35 KryptoRide – Kryptographie zum Mitmachen**

*Doris Behrendt, dorisbehrendt@kaltensondheim.net*

Holpriger Kopfrechenritt durch die Grundlagen moderner Kryptographie. Beispiele RSA und Diffie-Hellman. Zielgruppe: interessierte Laien. Zettel und Bleistift mitbringen! Weicheier bitte auch den Taschenrechner einpacken ;-)

### **3.36 Kryptoschlüssel, Zertifikate und Smartcards in der Praxis**

*Rolf Wald, LUG-Balista Hamburg e.V., rolf.wald@lug-balista.de*

Die Nutzung eines Kryptoschlüssels mit Zertifizierung erlaubt die Nutzung vieler Programme mit nur einer Passphrase, dabei kann mit der Speicherung von Schlüsseln und Zertifikaten auf einer Smartcard eine höhere Sicherheitsstufe erreicht werden. Im Vortrag werden sichere Kommunikation (verschlüsselte und signierte E-Mail, OpenVPN, WLAN WPA-Enterprise), verschlüsselte Datenspeicherung und Authentifikation (pam-auth, login, webclient-auth) live gezeigt und die nötigen Einstellungen und Anpassungen erläutert. Wer keine eigene PKI betreiben will, kann auf die kostenlosen Zertifikate von CAcert zurückgreifen.

### **3.37 Linux im Automotive-Umfeld – wie baue ich mir mein eigenes Fahrerassistenzsystem**

*Marcus Obst, BASELABS GmbH, marcus.obst@baselabs.de*

Linux hat sich im Bereich der eingebetteten Systeme etabliert. Neben der Verbreitung im Consumerbereich wird Linux zunehmend in der Automobilbranche eingesetzt. Im Vortrag werden aktuelle Entwicklungen von Embedded Linux im Fahrzeug vorgestellt. Weiterhin wird erklärt, wie Open Source zur effizienten Entwicklung von Fahrerassistenzsystemen genutzt wird. Dabei wird besonders auf Linux mit seinen Echtzeitfähigkeiten sowie die Mono-Runtime eingegangen. In einer Live-Demo wird gezeigt, wie ein komplexes Fahrerassistenzsystem entworfen und auf einer Embedded-Linux-Plattform ausgerollt werden kann.

### 3.38 Linux-Booten leicht gemacht: der Barebox Bootloader

*Robert Schwebel, Pengutronix, info@pengutronix.de*

Barebox ist ein Bootloader (Firmware) für den Einsatz in eingebetteten Systemen, vorzugsweise aufbauend auf der ARM-Architektur. Er ist durch den konsequenten Einsatz von bekannten Techniken aus dem Linux-Kernel einfach auf neue Hardware zu portieren. Durch Implementierung der Freedesktop.org Bootloader Specification ermöglicht er sowohl ein problemloses Booten von Standarddistributionen wie Debian, Fedora oder OpenSUSE als auch die Umsetzung von komplexen Bootszenarien mit Redundanz- und Update-Mechanismen.

Weitere Informationen: <http://www.barebox.org>

### 3.39 Linux-Dienstleister stellen sich vor (Business-Forum)

*Björn Krellner*

Bei den Chemnitzer Linux-Tagen sind Unternehmen seit Jahren ein fester Bestandteil. Sie entsenden ihre Mitarbeiter, um Vorträge zu halten, sie präsentieren sich bei Linux-Live und unterstützen uns als Sponsoren. Auch unter den Gästen sind viele, die beruflich mit Freier Software arbeiten. Umrahmt von speziellen Business-Vorträgen bieten wir hier eine Stunde lang die Möglichkeit, dass sich Unternehmen in wenigen Worten vorstellen, ihre Wünsche für gemeinsame Projekte äußern und Arbeitsmöglichkeiten mit sowie in verschiedenen Branchen zeigen können.

Weitere Informationen: <http://chemnitzer.linux-tage.de/2014/de/addons/business-forum>

### 3.40 LPI-Zertifizierung, aber wie?

*Fabian Thorns, LPI e.V., fthorns@lpi-german.de*

Das Linux Professional Institute (LPI) bietet Zertifizierungen für Linux-Experten. Der Vortrag stellt das LPI vor, gibt eine Übersicht über die Prüfungen und Zertifikate des LPI und zeigt Möglichkeiten der Vorbereitung auf eine LPI-Prüfung. Er richtet sich an alle Interessierten, die noch keine Erfahrungen mit LPI-Prüfungen gesammelt haben.

### 3.41 Medienalphabetismus – heilbar?

*Wolf-Dieter Zimmermann, zimmermann@netzwerk-bildung.net*

Unstrittig ist, dass eine wirklich kritische Mediennutzung und gar -produktion Wissen und Kenntnisse von Prozessen «hintendran» erfordern, unstrittig auch, dass derlei ausschließlich mit quelloffener Software vorstellbar ist. Sind Kriterien wie Transparenz und Partizipation denn nicht Erkennungsmerkmale demokratischer Teilhabe? Die Unkenntnis über Rechner und ihre Arbeit hat inzwischen eine Dimension erreicht, die man als Medienalphabetismus bezeichnen könnte. Er betrifft Jung und Alt im Umfang wohl gleichermaßen, wenn auch bei unterschiedlichen Nutzungsszenarien.

Weitere Informationen: <http://netzwerk-bildung.net>

### 3.42 Methoden zur Gewinnung neuer Teammitglieder

*Andreas Tille, Debian, tille@debian.org*

Der Vortrag beschreibt die praxiserprobten Konzepte «Mentoring of the Month» und «Sponsoring of Blends», die innerhalb von Debian-Teams erfolgreich waren, um neue Mitstreiter zu gewinnen. Dabei werden gewonnene Erkenntnisse verallgemeinert und Debian-spezifische Fragen in den Hintergrund gestellt.

### 3.43 Nachrichtenverschlüsselung im Alltag

*Tommy Sauer, B1 Systems GmbH, sauer@b1-systems.de*

Warum sollte ich meine Kommunikation verschlüsseln? Was «kostet» mich Verschlüsselung, und kann ich sie einfach in meinen Kommunikationsalltag integrieren? Der Vortrag gibt Antwort auf diese grundlegenden Fragen und stellt mit OTR (Off-the-Record) Messaging ein Protokoll im Detail vor, mit dessen Hilfe man seine Kommunikation vor den Blicken anderer verbirgt. Außerdem vorgestellt wird die PGP-Verschlüsselung für E-Mails und Dateien.

### 3.44 NeDi – Network Discovery that Really Works

*Michael Schwartzkopff, sys4 AG, ms@sys4.de*

Automatic network discovery and documentation is a long wanted feature of admins. NeDi is one of the first programs that really delivers this promises. NeDi gives a nearly instantaneous overview over all devices and hosts in your network, but can also be utilized as a nearly zero-config monitoring tool.

Weitere Informationen: <http://www.nedi.ch>

### 3.45 nftables – der neue Paketfilter im Linux-Kernel

*Michael Steinfurth, B1 Systems GmbH, steinfurth@b1-systems.de*

nftables steht in den Startlöchern, um mittelfristig bestehende Paketfilter wie iptables, ip6tables, arptables und ebtables abzulösen. Dieser Vortrag geht auf das Design des neuen Filter-Frameworks ein und erläutert dessen Gemeinsamkeiten und Unterschiede zu iptables. Im Praxisteil des Vortrags lernen Sie mehrere Anwendungsfälle und eine beispielhafte Migration von iptables zu nftables kennen.

### 3.46 Open Source in der brasilianischen Regierung

*Lars Kneschke, Metaways Infosystems GmbH, l.kneschke@metaways.de*

Schon vor der NSA-Affäre hat Brasilien sehr stark auf Open Source gesetzt. Seit Oktober 2013 gibt es aber ein Dekret der Präsidentin Dilma Rousseff, dass das bisherige E-Mail-System, basierend auf Microsoft Exchange, durch eine Eigenentwicklung zu ersetzen ist. Da man so etwas nicht komplett neu entwickeln kann, setzt die brasilianische Verwaltung auf Open Source, unter anderem auch aus Deutschland. Neben dem aktuellen Stand der Umstellung wird in diesem Vortrag auch die Infrastruktur vorgestellt, die nötig ist, um die Regierung des fünftgrößten Landes der Welt auch weiterhin arbeitsfähig zu halten.

### 3.47 Open-Source-Lizenzen in der kommerziellen Praxis

*Andre Ziemann, msg systems ag*

Die Verwendung von Open-Source-Komponenten ist auch aus der kommerziellen Softwareentwicklung nicht mehr wegzudenken. Der Vortrag versucht sich an Antworten zu den dabei entstehenden Fragen: Welche Lizenzen sind geeignet, die dabei entstehenden Anforderungen zu erfüllen? Wo liegen die Risiken? Welche Auswirkungen gibt es für die Vertragsgestaltung mit dem Kunden?

### 3.48 PDF-KungFoo mit Ghostscript & Co.

*Kurt Pfeifle, Kurt Pfeifle IT-Beratung, kurt.pfeifle@gmail.com*

Der Vortrag demonstriert einige der Top-10-Probleme, die bei der Verarbeitung und Erstellung von PDF-Dokumenten in der Praxis auftreten. Die Besucher lernen, wie man diese Schwierigkeiten lösen oder umgehen kann oder sie besser von vornherein vermeidet. Es werden verschiedene Tools zur Analyse und Reparatur typischer Probleme wie qpdf, pdftk, Poppler-Tools, pdfresurrect, pdf-parser.py, pdfid.py, mutool und GhostScript vorgeführt und erläutert.

Weitere Informationen: <http://leanpub.com/pdfkungfoo>

### 3.49 Perfekte Silbentrennung in E-Books mit präreformatrischen Texten

*Georg Pfeiffer, gp@praetor.de*

«Präreformatrisch» bezieht sich nicht etwa auf die «Rechtschreibreform» von 1996, sondern auf die von 1901. Es geht also um Texte aus dem 19. Jahrhundert mit wenig standardisierten Schreibweisen. Kein Rechtschreibprogramm der Welt kann diese zuverlässig korrekt trennen. Dabei ist in E-Books auf der kleinen Lesefläche eine durchgehend richtige Trennung besonders wichtig. Aus offenen Quellen kann man eine Lösung basteln.

### 3.50 Pond – E-Mail sicher und vertraulich

*Jens Kubieziel, jens@kubieziel.de*

Es ist bekannt, dass E-Mails für jeden les- und veränderbar sind. Die Veröffentlichungen von Edward Snowden führten uns schmerzhaft vor Augen, dass dies nicht nur Theorie ist. Mittels GnuPG bzw. PGP lassen sich E-Mails digital unterschreiben oder verschlüsseln. Dadurch sind die Inhalte nicht mehr änderbar oder lesbar. Dennoch bleiben die Metadaten erhalten. Pond ist ein neuer Ansatz zum Austausch von Nachrichten. Es basiert auf dem OTR-Protokoll und nutzt Tor zur Übertragung. Der Vortrag erläutert die theoretischen Hintergründe und zeigt, was das Programm leistet.

Weitere Informationen: <https://pond.imperialviolet.org/>

### 3.51 PostgreSQL: Killing NoSQL

*Hans-Jürgen Schönig, Cybertec Schönig & Schönig GmbH*

NoSQL ist in den letzten Jahren immer beliebter geworden – Grund genug, dieses Phänomen einmal unter die Lupe zu nehmen. Doch auch die relationale Welt hält mittlerweile jede Menge Erweiterungen bereit, die nicht von schlechten Eltern sind. Neben Unterstützung für JSON, XML, Analytics und unscharfer Suche bietet PostgreSQL horizontale Skalierung, Customer Worker Processes und vieles mehr – allesamt Features, die es der NoSQL-Konkurrenz sehr schwer machen werden.

### 3.52 PREEMPT-RT – More than just a kernel

*Steven Rostedt, Red Hat Inc., rostedt@goodmis.org*

The Real Time Patch (also known as the -rt patch) enables the user with the PREEMPT\_RT option for his kernel. This converts the Linux native kernel into a hard real-time designed kernel, that gives users sub 100 microsecond response times. This patch set has been critical for users of Jack for audio recording. It can even benefit hard core gamers. As with all real-time work, one can not just flip a switch and expect everything to just work. The kernel gives you the ability for quick reaction times, but the user must still understand how to tune it for best results.

Weitere Informationen: <https://www.rt.wiki.kernel.org/>

### 3.53 Quelle: Internet? Das können wir besser! – Mit Metadaten Ordnung ins Chaos bringen

*Leena Simon, Commons Machinery, Leena@commonsmachinery.se*

Das Urheberrecht wurde einst mit Blick auf den Gebrauch durch große Firmen geschrieben und ist entsprechend komplex. Dagegen bildeten sich alternative Lizenzmodelle. Doch es fehlt Software, die uns hilft, die unterschiedlichen Werke mit ihren Lizenzen und Quellen automatisch zu organisieren. Dem stellt sich Commons Machinery entgegen. Wir möchten, dass ein Werk mit seiner Herkunft verbunden bleibt. Mit Hilfe von Metadaten sollen Lizenzen und Quellen technisch organisiert werden. Dafür prüfen wir verschiedene Modelle, die im Vortrag neben kulturphilosophischen Betrachtungen besprochen werden.

Weitere Informationen: <http://www.commonsmachinery.se>

### 3.54 Samba 4 und OpenLDAP als Home Server mit UCS

*Ben Haberhauer, Univention GmbH, haberhauer@univention.de*

OpenLDAP und Samba 4 sind in professionellen Umgebungen fest etabliert. Im Vortrag wird gezeigt, wie diese Komponenten auch einfach und sinnvoll in einem Heimnetzwerk konfiguriert und genutzt werden können. Neben der Vorstellung der wichtigsten Funktionen eines Home Servers wird die Einrichtung im Rahmen einer kurzen Live-Demonstration mit Univention Corporate Server (UCS) gezeigt.

Weitere Informationen: <http://www.univention.de/download-und-support/lizenzmodelle/free-for-personal-use-lizenz/>

### 3.55 SCSI EH and the real world

*Hannes Reinecke, SUSE Linux Products GmbH, hare@suse.de*

The current SCSI error handling has been modeled for the now rather ancient SCSI-2 standard. When working with modern hardware more often than not the error handling will lead to unforeseen results, interrupting I/O and occasionally disable LUNs altogether. Having been involved with several customer calls complaining about EH gone wrong I've been working on a patchset resolving some of the most pressing issues. This presentation will give you some real-life examples on what can go wrong, and present you with the results from the new EH. Finally I'll give an overview on the layout of a new EH, based on current standards.

### 3.56 SELinux: Bitte nicht deaktivieren . . .

*Robert Scheck, Fedora Project, robert@fedoraproject.org*

Die vermutlich häufigste Reaktion von Benutzern und Administratoren zu SELinux ist: «Habe ich sofort deaktiviert!». Dieser Vortrag erklärt, warum SELinux, die Sicherheitserweiterung «Security Enhanced Linux» im Linux-Kernel, existiert, wie es funktioniert und aufgebaut ist. Selbstverständlich werden auch die täglichen Probleme und Lösungswege erläutert, die Konfiguration und Analyse bzw. Fehlersuche durchgesprochen. Ziel ist es, den verbreiteten Schrecken zu nehmen und ein Verständnis zu schaffen, warum man SELinux nicht sofort nach der Installation deaktivieren, sondern tatsächlich benutzen sollte.

### 3.57 Shell lernen und günstig tanken

*Harald König, science + computing ag, koenig@linux.de*

Im Vortrag wird «interaktiv» ganz langsam und in kleinen Schritten ein Shell-Skript entwickelt – um zu demonstrieren, wie einfach das mit ein wenig Übung sein kann (und wie nützlich zum Schluss!).

Mit der Shell (bash) und einer kleinen Zahl hilfreicher Kommandozeilen-Tools kann man sehr schön Daten bearbeiten, umwandeln, verarbeiten und auswerten. Einmal erlernt und geübt ist dies ein praktisches und mächtiges Werkzeug in UNIX, welches durch die grafischen Oberflächen vielfach zu Unrecht immer mehr in Vergessenheit gerät.

### 3.58 Sichere entfernte Rechnernutzung und Dateitransfer

*Holger Trapp, TU Chemnitz, URZ, hot@hrz.tu-chemnitz.de*

In verschiedenen Situationen besteht der Bedarf, sich auf der textuellen oder grafischen Oberfläche eines entfernten Rechners anzumelden, dort Kommandos auszuführen oder in einer interaktiven Sitzung zu arbeiten und Dateien zwischen den beteiligten Systemen auszutauschen.

Hierfür existieren diverse Verfahren mit differierendem Leistungsumfang und Sicherheitsniveau. Der Vortrag möchte mit SSH, NX und X2Go drei etablierte, leistungsfähige und nach bisherigem Kenntnisstand sichere Werkzeuge vorstellen, mit denen sich Linux-Systeme bequem aus der Ferne nutzen und administrieren lassen.

Weitere Informationen: [http://www-user.tu-chemnitz.de/hot/LT/2014\\_remote\\_access/](http://www-user.tu-chemnitz.de/hot/LT/2014_remote_access/)

### 3.59 Sichere Netze mit OpenVPN

*Roman Geber, B1 Systems GmbH, [geber@b1-systems.de](mailto:geber@b1-systems.de)*

VPN (Virtual Private Networking) sichert von Laptops in öffentlichen WLAN-Netzen über Home-Office-Workstations bis hin zur Kommunikation zwischen zwei Rechenzentren den Datenverkehr über ungesicherte Netze wie z.B. dem Internet ab. Mit OpenVPN steht ein vielseitiges, Enterprise-taugliches Open-Source-VPN-Protokoll zur Verfügung, das von seiner breiten Benutzerbasis profitiert. Lernen Sie die Einsatzmöglichkeiten kennen und erleben Sie live die Installation eines OpenVPN-Servers, die Anbindung von Clients und die Absicherung ungesicherter Dienste.

Weitere Informationen: <http://openvpn.net/>

### 3.60 Sicheres Anwendungsmonitoring mit SNMP

*Gerrit Beine, Gerrit Beine GmbH, [mail@gerritbeine.com](mailto:mail@gerritbeine.com)*

Der Vortrag gibt Unix-Nutzern einen Einblick, wie man Net-SNMP zum Monitoring und Steuern beliebiger Anwendungen nutzen kann. Der Schwerpunkt liegt dabei auf dem Thema der Absicherung des SNMP-Dienstes mit Hilfe von SSL/TLS und Authentifizierung. Als Beispiele dienen hierzu SNMP4J und Net-SNMP.

Weitere Informationen: <http://www.net-snmp.org/>, <http://www.snmp4j.org/>

### 3.61 Statische Codeanalyse – wo ist der Fehler in meinem Programm?

*Wolfgang Dautermann, FH Joanneum*

Für verschiedene Programmiersprachen werden Codeanalyse-Tools vorgestellt, um Fehler in eigenen und fremden Programmen zu finden. Compiler und Interpreter beanstanden zwar ungültigen Programmcode, in vielen Programmiersprachen sind aber selbst sehr ungewöhnliche Codezeilen noch gültig. Glücklicherweise gibt es Tools, die hier ansetzen und solche ungewöhnlichen Codekonstruktionen aufspüren. Damit helfen sie, mögliche Fehlerquellen zu beseitigen, und sorgen so für bessere Codequalität.

### 3.62 Systemmanagement mit Puppet und Foreman

*Mattias Giese, B1 Systems GmbH, [giese@b1-systems.de](mailto:giese@b1-systems.de)*

Im heutigen IT-Alltag müssen neue Systeme innerhalb kürzester Zeit konfiguriert und verfügbar sein. Immer mehr Systeme werden von immer weniger Administratoren betreut. Werkzeuge zum Deployment und zum Konfigurationsmanagement bieten die nötige Automatisierung. Dieser Vortrag führt in das Systemmanagement mit Puppet

und Foreman ein und zeigt beispielhaft, wie ein System frisch eingerichtet, konfiguriert und in eine Monitoring-Umgebung eingebunden wird.

### 3.63 Thin Clients von morgen, booten via WLAN

*Jörn Frenzel, openthinclient gmbh, j.frenzel@openthinclient.com*

Nach einer kurzen Vorstellung der openthinclient GmbH werden openthinclient und dessen Netzwerkstack sowie der Bootvorgang einer klassischen «diskless workstation» gezeigt. Beim kabellosen Bootvorgang treten aufgrund der PXE-Implementierung architekturbedingte Probleme auf, die behoben werden müssen. In Zukunft wird mit der Open-Source-Lösung iPXE ein Booten über WLAN deutlich einfacher und dank HTTP-Unterstützung und Skriptfähigkeit auch leichter konfigurierbar.

Weitere Informationen: <http://ipxe.org>, <http://openthinclient.org>

### 3.64 truecrypt.sh: Deniable File System with bash

*Hannes Reinecke, SUSE Linux Products GmbH, hare@suse.de*

truecrypt.sh is an approach to implement a deniable file system within an existing ext2/3 file system. The file system is created and assembled using existing tools, without the need to install additional packages. The file system itself can be assembled and removed without affecting the underlying ext2/3 file system, making it easy to implement a «big red switch» to wipe the deniable file system without any trace.

Weitere Informationen: <https://github.com/hreinecke/truecrypt.sh>

### 3.65 Tux im Passivhaus – Klimaschutz und Smarthome mit Freier Software

*Kurt Gramlich, VHS Ravensberg, kurt.gramlich@skolelinux.de*

In einem neu errichteten Mehrfamilien-Passivhaus werden Temperatur, Luftfeuchte, Kohlendioxid und Gerüche, die Menschen und Gegenstände (neue Teppiche, Einbauschränke, etc.) abgeben, mit Freier Software erfasst und ausgewertet. Kurt Gramlich, Bauherr und Bewohner des Passivhauses, berichtet von ersten Messergebnissen, die mit digitemp und osdomotics erfasst und mit gnuplot grafisch dargestellt werden. Goesta Smekal wird die Smarthome-Technik vorstellen und zeigen, wie mit 6LoWPAN und energiesparenden Funkknoten die Daten verarbeitet werden.

### 3.66 Vertrauen ist gut, Kontrolle ist besser: 7 Aspekte der Vertrauenswürdigkeit von freien Office-Suiten

*Lothar K. Becker, riess applications gmbh, app@riess-app.de*

Der Vortrag fragt nach Aspekten der Vertrauenswürdigkeit der freien Office-Suiten LibreOffice und Apache OpenOffice. Deren Sicherheit steht noch am Anfang. Es werden die Langlebigkeit und Funktionsvollständigkeit sowie die Anwenderakzeptanz

und die Unabhängigkeit betrachtet. Am Ende steht die Einordnung der Vertrauenswürdigkeit zu Kompatibilität und Kosten in Bezug auf proprietäre Lösungen. Es wundert nicht, dass freie Office-Suiten dem Motto genügen: Vertrauen ist gut, Kontrolle ist besser. Der Vortrag entstand aus der Themen- und Expertenplattform F-O-X.biz, Free Office eXperts.

Weitere Informationen: <http://www.f-o-x.biz>

### **3.67 Vollautomatische Betriebssystemtests mit openQA**

*Bernhard M. Wiedemann, SUSE Linux Products GmbH*

Betriebssysteme (wie z.B. die openSUSE-Linux-Distribution) sind große Sammlungen von Softwarepaketen, deren Abhängigkeiten bisweilen komplex sein können. Um dennoch eine gewisse Qualität zu gewährleisten, ohne Tester mit unbrauchbarer Software zu verärgern, benötigt man gute vollautomatische Tests. Dieser Kurzvortrag gibt einen Überblick über openQA und die OS-autoinst-Testsoftware, mit denen in den letzten 3,5 Jahren durchgehend die neuesten openSUSE-Entwicklerversionen getestet wurden.

Weitere Informationen: <http://openqa.opensuse.org/>

### **3.68 Vom Aussterben bedroht: die Universalmaschine Computer**

*Matthias Kirschner, Free Software Foundation Europe (FSFE), [mk@fsfe.org](mailto:mk@fsfe.org)*

Computer sind universelle Maschinen, die beliebig programmierbar sind und prinzipiell alles können. Vielen IT-Unternehmen ist dies mittlerweile ein Dorn im Auge. Sie wollen willkürlich beschränken, was wir als Gesellschaft mit dieser Maschine machen können. Sie ergreifen technische Maßnahmen, mit denen sie uns diese Möglichkeiten nehmen und uns Stück für Stück Rechte entziehen, die wir normalerweise haben, wenn wir ein Produkt kaufen. Die Industrie will entscheiden, was wir mit unseren Computern machen können und was mit unseren Daten passiert. Wollen wir Ihnen diese Macht einräumen?

Weitere Informationen: <https://fsfe.org/campaigns/generalpurposecomputing/secure-boot-analysis.de.html>

### **3.69 Wald und Bäume – Log-Analyse für Serverparks**

*Jens Kühnel, [clt2014@jens.kuehnel.org](mailto:clt2014@jens.kuehnel.org)*

Die Log-Dateien in `/var/log/` sind für jeden Administrator die zentrale Anlaufstelle, wenn es Probleme gibt. Was auf ein, zwei Maschinen noch gut funktioniert, wird bei 100 oder 1000 Servern nicht mehr skalieren. Also müssen die Logs eingesammelt und verarbeitet werden. Dieser Vortrag stellt verschiedene Lösungen zur Sammlung, Verarbeitung und Analyse von Log-Dateien mit Open-Source- und Freien Software-Tools vor. Er basiert auf der Bachelorarbeit des Sprechers mit dem Titel: «Centralized and structured log file analysis with Open Source and Free Software tools».

Weitere Informationen: <http://www.it-hure.de/2013/10/bachelor-thesis-centralized-and-structured-log-file-analysis-with-open-source-and-free-software-tools/>

## 3.70 Wanderreise mit OpenStreetMap

*Thomas Bellmann, [osm@malenki.ch](mailto:osm@malenki.ch)*

Was tun, wenn man in GanzWeitWeg wandern will, es aber keine Karten gibt? Selbermachen! Der Vortrag zeigt, wie man mit Hilfe von OpenStreetMap und freien Werkzeugen eine Wanderreise plant und Kartenmaterial erzeugt. Das konkrete Beispiel ist eine 300-km-Wanderung durch einen abgelegenen Teil Albaniens mit einem Outdoor-GPS-Gerät von Garmin.

## 3.71 Warum Kinder eine Open-Source-Community brauchen

*Dominik George, Teckids e.V., [dominik.george@teckids.org](mailto:dominik.george@teckids.org)*

Der Teckids e. V. hat als Ziel, eine nachhaltige Open-Source-Community unter Kindern und Jugendlichen aufzubauen und die Benutzung von Freier Software und freien Standards unter den Jugendlichen zu verbreiten. Warum das wichtig ist, was wir alle zusammen tun können und an welchen Punkten unsere eigene Community bei der Verbreitung von sicherer Software bei jungen Menschen scheitert, und vor allem, was eine solche Community mit Vertrauen zu tun hat, möchten wir in dieser Vorstellung erörtern.

## 3.72 Was kommt nach SysVinit?

*Alexander Böhm*

Über den Nachfolger von SysVinit auf Linux-Systemen ist bisher nichts definitiv entschieden. Wie in der Open-Source-Welt üblich, gibt es zahlreiche Alternativen, `systemd` und `Upstart` dürften die bekanntesten sein. Dabei können sie heute wesentlich mehr, als lediglich das Verwalten und Starten von Diensten zu regeln. Anhand von `systemd` soll beispielhaft gezeigt werden, welche Fähigkeiten diese neuen Systeme mit sich bringen.

## 3.73 WebODF – gemeinsame Dokumentenbearbeitung in der eigenen Website

*Friedrich W. H. Kossebau*

WebODF ist eine JavaScript-Bibliothek, die das Betrachten und gemeinsame Editieren von Dokumenten im OpenDocument-Format in jeder Webseite oder Webanwendung ermöglicht. Sie arbeitet komplett im Browser, noch nicht unterstützte Formatierungen bleiben erhalten beim Laden und Speichern. Abstraktionsebenen erlauben die Anbindung an beliebige Messaging- und Speichersysteme. Benutzt wird WebODF derzeit in Tiki Wiki, Zarafa, Kolab/Roundcube Webmailer sowie in ownCloud Documents. Der Vortrag gibt eine Übersicht der Funktionsweise und zeigt, wie WebODF in eigenen Projekten eingesetzt werden kann.

Weitere Informationen: <http://webodf.org>

### **3.74 Wenn Geeks Langeweile haben – reloaded**

*Uwe Berger, [bergeruw@gmx.net](mailto:bergeruw@gmx.net)*

Im Mittelpunkt des Vortrages steht ein ungewöhnliches Ausgabemedium für eine Uhr, eine Kathodenstrahlröhre. Es werden die Funktionsweise und Realisierung einer solchen Mikrocontroller-gesteuerten «Scopeclock» detailliert erläutert. Dabei wird aber auch besprochen, wie man an solche oder ähnliche Projekte herangeht – wenn man mal wieder viel Langeweile hat.

Weitere Informationen: <http://bralug.de/wiki/Scopeclock>

### **3.75 Wie kann man Zertifikate von CAcert verwenden**

*Reinhard Mutz, [reinhard@cacert.org](mailto:reinhard@cacert.org)*

Die eigene Identität schützen und bewahren ist Bürgerpflicht. Auch wenn vielen Personen die Verwendung von digitalen Zertifikaten bekannt ist, finden diese in der täglichen Praxis keinesfalls die erforderliche Beachtung. Dieser Vortrag wendet sich an alle Personen, die auf digitalem Wege Informationen austauschen und nutzen. Neben der Erzeugung wird auch auf die sichere Aufbewahrung und die eigentliche Verwendung von Zertifikaten in Software- und Hardwarelösungen eingegangen.

### **3.76 Wie wir einmal 500 Server mit 150 Personen in 3 Tagen migriert haben und was wir alles gelernt haben**

*Ralph Angenendt, Immobilien Scout 24 GmbH, [ralph@strg-alt-entf.org](mailto:ralph@strg-alt-entf.org)*

Der Plan: Wir wechseln von einer «bereitgestellten» virtuellen Serverumgebung auf eine andere Virtualisierungslösung: selbstverwaltet, «on demand», mit Monitoring out of the box. Mit einem anderen Betriebssystem. Innerhalb kurzer Zeit. Mit ca. 1500 Servern. Ohne Downtime. So etwas startet man mit einem Big Bang: Wir migrieren mit der kompletten IT (Entwickler, QA, DBAs, Administratoren) 500 dieser Server innerhalb von 3 Tagen. In diesem Talk geht es um die Organisation und Durchführung eines solchen Events – und was man daraus lernen kann. Und welchen Spaß man dabei mit 150 Leuten haben kann.

### **3.77 Zur eigenen Linux-Distribution in 30 Minuten**

*Oliver Rath, GreenUnit UG, [rath@mglug.de](mailto:rath@mglug.de)*

Wenn man im Linux-Umfeld arbeitet, benötigt man des öfteren Distributionen mit besonderen Eigenschaften wie die Unterstützung einer speziellen Architektur oder eine bestimmte Paketauswahl für konkrete Anwendungsszenarien. Wir bauen hier live (mit Stoppuhr!) unsere eigene kleine Minidistribution (auf Ubuntu-Basis) für den USB-Stick, die als Vorlage für eigene Entwicklungen dienen soll. Wenn noch Zeit bleibt, gibt es einen kleinen Ausblick zur Nutzung via Netboot und zu Optimierungsmöglichkeiten.

### **3.78 Zur Geschichte der Verschlüsselung: Von der Kopfrasur zur Kopfakrobatik**

*Andreas Steil, B1 Systems GmbH, steil@b1-systems.de*

Der Wunsch nach Geheimhaltung ist nicht neu. Seit der Antike verschlüsseln Menschen ihre Kommunikation. Dieser Vortrag stellt die Geschichte und die Grundlagen der Verschlüsselung vor. Er geht ein auf die Funktionsweise der dabei entwickelten Verfahren von den Anfängen bis heute und bietet einen Überblick über deren Anwendung.



## 4 Zusammenfassungen der weiteren Workshops

### 4.1 darktable – die digitale Dunkelkammer

*Sirko Kemter, Fedora Project, gnokii@fedoraproject.org*

Wer mit digitalen Kameras aus dem mittleren bis hohen Preissegment arbeitet, hat die Möglichkeit, eigene Bilder als JPEG- oder als RAW-Datei zu speichern. Letztere enthalten die «rohen», von der Kamera unbearbeiteten Bildinformationen des Sensors. Für eine weitere Verwendung müssen diese Bilder also noch «entwickelt» werden. Seit 2009 macht hier das Programm darktable von sich reden, dass sich schnell zu einem Renner entwickelte. Wie es funktioniert, soll in diesem Workshop anhand eines wohl typischen Urlaubsfotos gezeigt werden.

Weitere Informationen: <http://www.darktable.org/>

### 4.2 Die Schale um den Kern – Einstieg in die Bash und den GNU-Werkzeugkasten

*Holger Trapp, TU Chemnitz, URZ, hot@hrz.tu-chemnitz.de*

Die Standard-Shell Bash sowie der GNU-Werkzeugkasten sind mächtige Tools, mit denen sich unterschiedliche Aufgaben sehr effizient und elegant lösen lassen. Allerdings stellen sie für Neulinge oft eine gewisse Hürde dar, da die Konzepte und konkreten Befehle zunächst wenig intuitiv erscheinen und man sich erst damit anfreunden muss.

Der Workshop möchte Einsteigern helfen, einen einfacheren Zugang zu dieser kommandoorientierten Computerbedienung zu finden, indem gemeinsam ausgewählte Problemstellungen besprochen und von den Teilnehmern am Rechner praktisch bearbeitet werden, wobei der Workshop-Leiter bei Bedarf Hilfestellung gibt.

Weitere Informationen: <http://www-user.tu-chemnitz.de/hot/LT/2014/>

### 4.3 Django: Schnell performante Web-Applikationen entwickeln

*Markus Zapke-Gründemann, Deutscher Django-Verein e. V.*

Ziel des Workshops ist die Erstellung einer Django-Applikation zur Verwaltung von Lesezeichen für beliebig viele Benutzer.

Nach einer kurzen Einführung muss jeder Teilnehmer Python und Django auf seinem Rechner installieren. Zunächst wird das Projekt erstellt, dann werden die Objekte (Models) zum Verwalten der Daten angelegt, für die im weiteren Verlauf Templates und Views zur Darstellung der Inhalte im Browser entwickelt werden. Dabei wird das in Django bereits enthaltene Admin-Backend vorgestellt und zum Anlegen und Bearbeiten der Datensätze genutzt.

Weitere Informationen: <https://www.djangoproject.com/>

## 4.4 Einführung in die 3D-Visualisierung mit Blender

*Erik Schufmann, Phase-10 Ingenieur- und Planungsgesellschaft mbH, info@erikschufmann.de*

In diesem Workshop sollen den Teilnehmern nach einem Überblick über die Open Source 3D Suite Blender folgende Themen praktisch vermittelt werden: Benutzeroberfläche, Modellierung (Object- und Edit-Mode), Materialien, Beleuchtung, Rendern und je nach Fortschritt auch die Animation der erstellten 3D-Szene.

## 4.5 Einführung in Python

*Stefan Schwarzer, SSchwarzer.com, ssschwarzer@sschwarzer.com*

Der Workshop bietet eine Einführung in die Programmiersprache Python. Diese ermöglicht kompakte, gut lesbare Programme für Systemadministration, Web, Wissenschaft und viele andere Gebiete. Sie ist auch eine ausgezeichnete Sprache zum Verbinden verschiedener Systeme (*glue language*). In Python lässt sich prozedural und objektorientiert programmieren.

Weitere Informationen: <http://www.python.org>

## 4.6 Elektronikbasteln für Kinder

*Detlef Heine, TU Chemnitz, Universitätsrechenzentrum, detlef.heine@hrz.tu-chemnitz.de*

«Etwas selbst Gebasteltes mit nach Hause nehmen» – unter diesem Motto haben wir einige Bausätze ausgesucht und möchten diese zusammen mit unseren jungen Besuchern zusammenbauen. Dabei reicht das Spektrum vom batteriebetriebenen Dinosaurier bis zum Mikrocontroller-basierten Memory-Spiel. Und passt mal ein Teil nicht, oder will eine Lötstelle nicht gelingen, so stehen Euch natürlich Helfer zur Seite.

Weitere Informationen: <http://chemnitzer.linux-tage.de/2014/de/vortraege/basteln>

## 4.7 Hardware-Workshop

*Philipp Seidel*

Im diesjährigen Hardware-Workshop wird eine Ansteuerung für eine 8×8-RGB-Matrix gebaut. Als Basis dient ein Mikrocontroller von Atmel, wie er auf Arduino-Boards zum Einsatz kommt. Die Ansteuerung erfolgt über ein Bluetooth-Modul. Alle benötigten Bauteile können vor Ort erworben werden. Im Workshop wird Unterstützung beim Zusammenbau geleistet. Der Preis pro Bausatz wird zwischen 25 und 30 Euro liegen.

## 4.8 KDE/Kubuntu-Grundeinstellungen

*Monika Eggers, kubuntu-de.org*

Dieser Workshop führt durch die ersten Einstellungen in KDE und Kubuntu: Wie aktiviert und deaktiviert man Fenstereffekte und welche gibt es? Welche Tipps und Tricks wie die Erhöhung der Animationsgeschwindigkeit gibt es noch? Was sind Aktivitäten und wie setzt man sie ein? Was sind Plasmoids? Welche interessanten Plasmoids

gibt es und wie fügt man sie zu Desktop und Kontrollleisten hinzu? Wie erstellt man zusätzliche Kontrollleisten und ändert die vorhandenen? Wie installiert man Pakete (Programme) und Hardwaretreiber?

## **4.9 Kreatives Programmieren mit Processing**

*Silvio Müller*

Processing ist eine einfache Programmiersprache und Entwicklungsumgebung, mit der man schnell verschiedenste multimediale Inhalte und Anwendungen entwickeln kann. Im Workshop werden die Freie Software und deren kreative Möglichkeiten kurz vorgestellt, bevor deren Grundlagen durch das Modifizieren und Programmieren von verschiedenen Beispielen vermittelt werden. Die Teilnehmer erstellen dabei einfache Grafiken und Animationen, die mit Processing in selbst generierten Videos und digitalen Büchern dokumentiert werden.

Weitere Informationen: <http://www.processing.org>

## **4.10 Open Knowledge: Dein Wissen als interaktiver Online-Kurs**

*Tibor Horvath, chemmedia AG, [info@openknowledgeworker.org](mailto:info@openknowledgeworker.org)*

Von Algorithmus bis Zettabyte – Du bist Experte auf Deinem Gebiet? Lass andere an Deinem Wissen teilhaben. Im Workshop stellen wir Dir die offene Plattform Open KnowledgeWorker vor, mit der Du ganz schnell und einfach interaktive Online-Kurse erstellen kannst, um Dein Wissen mit anderen kostenfrei zu teilen. In vielen Sprachen, gemeinsam mit Freunden, Kollegen und anderen Experten. Alle Inhalte in Open KnowledgeWorker stehen unter offenen Lizenzen.

Weitere Informationen: <http://www.openknowledgeworker.org/>

## **4.11 openATTIC – offenes Storage Management**

*David Breitung, it-novum GmbH / openATTIC, [david@open-attic.org](mailto:david@open-attic.org)*

openATTIC ist ein hochflexibles zentrales Framework für Storage Management, das unter einer grafischen Oberfläche verschiedene Open Source Tools vereint. Die Lösung stellt eine kostenoptimierte Alternative zu proprietären Storage-Systemen dar, die oft schon bei Basisanforderungen hohe Lizenzkosten mit sich bringen. Der Workshop stellt das Projekt vor, das Anfang 2012 veröffentlicht wurde. Bei der Entwicklung von openATTIC wird Wert auf einen modulbasierten Aufbau gelegt, so dass neue Technologien und Produkte anderer Hersteller nahtlos in das System integriert werden können.

Weitere Informationen: <http://openattic.org/>

## **4.12 PyMove3D – Vorbereitung zum Programmierwettbewerb**

*Peter Koppatz, Sudile GbR, [peter.koppatz@sudile.com](mailto:peter.koppatz@sudile.com)*

Der Python Software Verband e. V. veranstaltet mit den Ausrichtern der EuroPython Konferenz vom 21. bis 27.7.2014 einen Python-Programmierwettbewerb. Um sich dafür fit zu machen, kann dieses Kursangebot genutzt werden. Es werden Grundlagen

zur Programmierung, der Programmiersprache Python und deren Anwendung in Blender vermittelt.

Weitere Informationen: <http://pymove3d.sudile.com>

#### **4.13 Raspberry Pi zum Anfassen**

*Andreas Heik, TU Chemnitz, Universitätsrechenzentrum, [andreas.heik@linux-tage.de](mailto:andreas.heik@linux-tage.de)*

Aus der Vision, Computertechnik für den schmalen Geldbeutel technisch interessierten Jugendlichen verfügbar zu machen, entstand ein kreditkartengroßer Einplatinencomputer, der Raspberry Pi. Im Workshop wollen wir diesen Gedanken aufgreifen und zum Anfassen und Experimentieren anregen. An verschiedenen Arbeitsplätzen laden RasPis mit externer Hardware wie Tastern, LEDs und Sensoren zum Basteln und Programmieren ein. Die Unterhaltung kommt mit dem XBMC Media Center auf dem Raspberry Pi auch nicht zu kurz.

Weitere Informationen: <http://chemnitzer.linux-tage.de/2014/de/vortraege/basteln>

#### **4.14 SSL-gesicherte Web-Seiten – was ist da wie «sicher»?**

*Martin Neitzel, Gaertner Datensysteme, [neitzel@gaertner.de](mailto:neitzel@gaertner.de)*

Ob Online-Banking, Google, oder Twitter: Viele Web-Seiten sind mit einem schicken Schloss oder grünen Balken bei ihrer Adresse markiert. Aber was bedeutet dieses Extra-S bei den <https://>-Seiten? Was alles ist da «sicher»? Was nicht? Was ist so ein «SSL-Zertifikat», wo kommt das her, und wie überprüft man dessen Echtheit? Wie ist die Warnung zu bewerten, die man beim Zugriff auf <https://chemnitzer.linux-tage.de/> erhält, und wann muss man Angst bekommen, dass man gar nicht bei seiner eigenen Bank gelandet ist, sondern bei einem Betrüger?

Weitere Informationen: <http://gaertner.de/neitzel/ct/ssl/>

## Personen

Alfa, Amadeus, 132  
Angenendt, Ralph, 150

Barth, Rico, 134  
Becker, Lothar K., 147  
Beckert, Axel, 137  
Behrendt, Doris, 140  
Beine, Gerrit, 146  
Bellmann, Thomas, 149  
Berger, Uwe, 81, 150  
Böhm, Alexander, 149  
Breitung, David, 155

Courtenay, Mark, 13

Dautermann, Wolfgang, 146

Eggers, Monika, 154  
Erk, Bernd, 138

Fiedler, Martin, 140  
Findeisen, Ralf, 91  
Freitag, Klaas, 135  
Frenzel, Jörn, 147

Geber, Roman, 146  
George, Dominik, 149  
Giese, Mattias, 146  
Gramlich, Kurt, 147  
Großöhme, Peter, 138

Haberhauer, Ben, 144  
Heik, Andreas, 156  
Heine, Detlef, 154  
Heinlein, Peer, 131, 136  
Herms, Robert, 107  
Hofmann, Frank, 137  
Horváth, Tibor, 155  
Hrončok, Miro, 139

Imme, Roland, 131

Kaiser, Martin, 136  
Kemter, Sirko, 153  
Kirschner, Matthias, 148  
Kneschke, Lars, 142  
Kölbel, Cornelius, 19  
König, Harald, 145  
Koppatz, Peter, 155  
Kossebau, Friedrich W. H., 149  
Kramer, Frederik, 51  
Krellner, Björn, 141  
Kruse, Klaus, 137  
Kubieziel, Jens, 143  
Kühnel, Jens, 148  
Kullik, Jakob, 135

Lang, Jens, 25  
Lanitz, Frank, 133  
Leemhuis, Thorsten, 131  
Lohr, Christina, 107  
Luge, Hartmut, 132  
Luithardt, Wolfram, 33  
Luther, Tobias, 123

Mende-Stief, Kerstin, 135  
Müller, Silvio, 155  
Mundt, Andreas, 133  
Mutz, Reinhard, 150

Neitzel, Martin, 138, 156

Obst, Marcus, 140

Patsch, Christian, 132  
Pfeiffer, Georg, 143  
Pfeifle, Kurt, 61, 143

Rath, Oliver, 150  
Reinecke, Hannes, 145, 147  
Rostedt, Steven, 144

Sander, Robert, 133

Sauer, Tommy, 142  
Schade, Markus, 139  
Scheck, Robert, 145  
Schilling, Jörg, 136  
Schlaeger, Chris, 132  
Schneider, Markus, 51  
Schöner, Axel, 99  
Schönig, Hans-Jürgen, 144  
Schreiber, Alexander, 134  
Schütz, Georg, 117  
Schufmann, Erik, 154  
Schwartzkopff, Michael, 142  
Schwarzer, Stefan, 154  
Schwebel, Robert, 141  
Seidel, Philipp, 154  
Seitz, Stephan, 133  
Simon, Leena, 144  
Stach, Lucas, 134  
Steil, Andreas, 151  
Steinfurth, Michael, 142

Thorns, Fabian, 141  
Tille, Andreas, 142  
Trapp, Holger, 139, 145, 153

Voß, Herbert, 139

Wachtler, Axel, 91  
Wald, Rolf, 140  
Wendzel, Steffen, 134  
Wiedemann, Bernhard M., 148

Zapke-Gründemann, Markus, 153  
Ziegler, Michael, 138  
Ziemann, Andre, 143  
Zimmermann, Wolf-Dieter, 141  
Zscheile, Falk, 41