

Forschungskennzahl (UFOPLAN) 3710 95 302 3

Grüne Software
**Ermittlung und Erschließung von Umweltschutzpotenzialen der Informations- und
Kommunikationstechnik (Green IT), TV 3: Potenzialanalyse zur
Ressourcenschonung optimierter Softwareentwicklung und -einsatz**

von

Prof. Dr. Lorenz Hilty
Dr. Wolfgang Lohmann
Universität Zürich, Institut für Informatik, Forschungsgruppe Informatik und Nachhaltigkeit,
Binzmühlestrasse 14, CH-8050 Zürich

Dr. Siegfried Behrendt
Michaela Evers-Wölk
IZT Institut für Zukunftsstudien und Technologiebewertung gemeinnützige GmbH,
Schopenhauerstr. 26, 14129 Berlin

Prof. Dr. Klaus Fichter
Dr. Ralph Hintemann
Borderstep Institut für Innovation und Nachhaltigkeit gemeinnützige GmbH,
Clayallee 323, D-14169 Berlin

**IM AUFTRAG
DES UMWELTBUNDESAMTES**

November 2013

Berichtskennblatt

Berichtsnummer	UBA-FB
Titel des Berichts	Grüne Software, Ermittlung und Erschließung von Umweltschutzpotenzialen der Informations- und Kommunikationstechnik (Green IT), TV 3: Potenzialanalyse zur Ressourcenschonung optimierter Softwareentwicklung und -einsatz
Autor(en) (Name, Vorname)	Prof. Dr. Lorenz Hilty, Dr. Wolfgang Lohmann, Dr. Siegfried Behrendt, Michaela Evers-Wölk, Prof. Dr. Klaus Fichter, Dr. Ralph Hintemann
Durchführende Institutionen (Name, Anschrift)	Universität Zürich, Institut für Informatik, Forschungsgruppe Informatik und Nachhaltigkeit, Binzmühlestrasse 14, CH-8050 Zürich IZT Institut für Zukunftsstudien und Technologiebewertung gemeinnützige GmbH, Schopenhauerstr. 26, 14129 Berlin Borderstep Institut für Innovation und Nachhaltigkeit gemeinnützige GmbH, Clayallee 323, D-14169 Berlin
Fördernde Institution	Umweltbundesamt Postfach 14 06 06813 Dessau-Roßlau
Abschlussjahr	2013
Forschungskennzahl (FKZ)	3710 95 302 3
Seitenzahl des Berichts	69
Zusätzliche Angaben	Eine Kurzfassung des Berichtes (Executive Summary) ist in englischer Sprache verfügbar
Schlagwörter	Software, Softwarearchitekturen, Virtualisierung, Cloud Computing, Soziale Netze, Mobiles Internet, Ressourceneffizienz, Ressourcenschonung, Bedarfsgerecht gesteuerte Software, Open Source Software, Rechenzentren, Lastmanagement, Informations- und Datenmanagement, Datenkomprimierung, Energieverbrauch von Software

Report Cover Sheet

Report No.	UBA-FB 00
Report Title	Green Software Establishing and exploiting potentials for environmental protection in information and communication technology (Green IT), Subproject 3: Analysis of potentials for optimizing software development and deployment for resource conservation
Author(s) (Family Name, First Name)	Prof. Dr. Lorenz Hilty, Dr. Wolfgang Lohmann, Dr. Siegfried Behrendt, Michaela Evers-Wölk, Prof. Dr. Klaus Fichter, Dr. Ralph Hintemann
Performing Organisation (Name, Address)	University of Zurich, Department of Informatics, Informatics and Sustainability Research, Binzmühlestrasse 14, CH-8050 Zürich IZT Institute for Futures Studies and Technology Assessment, non-profit limited company, Schopenhauerstr. 26, 14129 Berlin Borderstep Institute for Innovation and Sustainability, non-profit limited company, Clayallee 323, D-14169 Berlin
Funding Agency	Umweltbundesamt Postfach 14 06 06813 Dessau-Roßlau
Report Date (Year)	2013
Project No. (FKZ)	3710 95 302 3
No. of Pages	69
Supplementary Notes	Executive summary available in english
Keywords	software, software architecture, virtualization, cloud computing, social networks, mobile web, resource efficiency, resource conservation, demand-adaptive software, open source software, data centers, load management, information and data management, data compression, energy consumption of software

Kurzbeschreibung

Obwohl Softwareprodukte immaterielle Güter sind, kann die Nutzung von Software erhebliche Stoff- und Energieströme auslösen. Eigenschaften der Software entscheiden, welche Hardwarekapazitäten vorgehalten werden und wieviel elektrische Energie in Endgeräten, Netzwerken und Rechenzentren verbraucht wird. Der Zusammenhang zwischen Softwareeigenschaften und dem Bedarf an natürlichen Ressourcen, der durch Herstellung und Betrieb von IKT-Systemen ausgelöst wird, ist bisher wissenschaftlich noch wenig untersucht. Die vorliegende Studie betritt Neuland, indem sie explorativ den Einfluss von Software auf die indirekte Inanspruchnahme natürlicher Ressourcen durch Hardware untersucht. Die Ressourceneffizienz von Software wird dabei im Kontext neuer Nutzungsformen wie mobiler Internetnutzung und neuer Software-Architekturmuster wie Cloud Computing betrachtet. Vor dem Hintergrund der Dynamik der aufgezeigten Trends identifiziert die Studie Ansatzpunkte im Softwarebereich, die zur Schonung natürlicher Ressourcen beitragen oder zumindest ein weiteres Wachstum ihrer Inanspruchnahme durch IKT-Systeme bremsen können. Sie geht dabei insbesondere auf methodische Probleme ein, die sich bei der Beurteilung der Ressourceninanspruchnahme von Softwareprodukten stellen. Zu diesen Problemen gehören Schwierigkeiten bei der Definition funktioneller Einheiten, Messprobleme und Allokationsprobleme. Ansätze wie die Standardisierung von Nutzungsmustern und Benchmarks sowie die Definition und Umsetzung von Nachhaltigkeitsanforderungen im Softwareentwicklungsprozess werden als mögliche Lösungswege aufgezeigt. Basierend auf diesen Überlegungen formuliert die Studie erste Handlungsempfehlungen in den Bereichen Forschung und Standardisierung, Produktkennzeichnung, Konfigurationsempfehlungen, Best-Practice-Leitfäden sowie Aus- und Weiterbildung im Bereich der ressourcen-effizienten Software und ihrer Entwicklung.

Abstract

Although software products are immaterial goods, their use can bring about significant materials and energy flows. Software characteristics determine which hardware capacities are made available and how much electric energy is used by end-user devices, networks, and data centers. The connection between software characteristics and the demand for natural resources caused by the manufacture and use of ICT systems has been the object of little scientific study to date. The present study breaks new ground by exploring the effects of software on the indirect use of natural resources by hardware. The study identifies starting points in the realm of software that can contribute to conserving natural resources or at least to slowing further growth of their use by ICT systems. A particular focus of the study is on methodological problems arising when assessing the resource use of software products. Such problems include difficulties in defining functional units as well as problems of measurement and allocation. Approaches such as standardizing patterns of use and benchmarks as well as defining and implementing sustainability requirements in the software development process are sketched out as possible solutions. Based on these considerations, the study formulates initial recommendations for action in the areas of research and standardization, product labeling, information for users concerning configuration, best practice guides as well as training and professional development.

Grüne Software

Schlussbericht zum Vorhaben: Ermittlung und Erschließung von Umweltschutzpotenzialen der Informations- und Kommunikationstechnik (Green IT), TV 3: Potenzialanalyse zur Ressourcenschonung optimierter Softwareentwicklung und -einsatz

Im Auftrag des Umweltbundesamtes

Förderkennzeichen 3710 95 302 3

Universität Zürich, Institut für Informatik, Forschungsgruppe Informatik und Nachhaltigkeit

Prof. Dr. Lorenz Hilty, Dr. Wolfgang Lohmann

IZT Institut für Zukunftsstudien und Technologiebewertung gemeinnützige GmbH

Dr. Siegfried Behrendt, Michaela Evers-Wölk

Borderstep Institut für Innovation und Nachhaltigkeit

Prof. Dr. Klaus Fichter, Dr. Ralph Hintemann

Danksagung

In diese Studie sind zahlreiche Anregungen von Fachkolleginnen und -kollegen eingeflossen. Die Autoren bedanken sich besonders bei Dr. Heidrun Moser, Maike Janßen, Marina Köhn und Stefan Schmitz vom Umweltbundesamt für die detaillierten Kommentare zu mehreren Versionen des Textes, bei Prof. Dr. Stefan Naumann, Umwelt-Campus Birkenfeld der Hochschule Trier, für Anregungen zur Methodendiskussion sowie bei Patrizia Huber, Universität Zürich, für redaktionelle Arbeiten.

Disclaimer

Die vorliegende Studie ist im Rahmen des Umweltforschungsplans im Auftrag des Umweltbundesamtes erarbeitet worden. Die Ergebnisse der Studie müssen nicht in allen Punkten die Auffassungen des Auftraggebers wiedergeben.

Inhalt

Abstract.....	13
Abkürzungsverzeichnis.....	14
1 Einleitung	15
1.1 Zielsetzung und Aufbau der Studie	15 -
1.2 Grundlegende Definitionen	16 -
1.3 Untersuchungsrahmen	17 -
2 Trendanalyse	21
2.1 Übergeordnete Trends.....	21 -
2.1.1 Mobiler Internetzugang	21 -
2.1.2 Foto und Video in immer höherer Auflösung.....	23 -
2.1.3 Spam	24 -
2.2 Trends im Bereich der Softwarearchitekturen und Nutzungsformen	25 -
2.2.1 Das „Prinzip App“.....	25 -
2.2.2 Webbasierte Anwendungssoftware	26 -
2.2.3 Virtualisierung und Cloud Computing.....	27 -
2.2.4 Soziale Netze.....	30 -
3 Ansatzpunkte für Ressourcenschonung.....	32
3.1 Ansatzpunkte im Bereich Anwendungssoftware.....	32 -
3.1.1 Wählbare Bildauflösung.....	32 -
3.1.2 Mobiles Internet bevorzugt über WLAN.....	32 -
3.1.3 Das „Prinzip App“ ausweiten	33 -
3.1.4 Webbasierte Anwendungen effizient implementieren.....	33 -
3.1.5 Bedarfsgerecht gesteuerte Software.....	34 -
3.1.6 Die Rolle von Open Source Software	35 -
3.2 Ansatzpunkte in Rechenzentren	38 -
3.2.1 Dynamisch prädiktives Lastmanagement	38 -
3.2.2 Informations- und Datenmanagement.....	39 -
3.2.3 Datenkomprimierung und Datendeduplizierung.....	40 -
3.2.4 Herausforderung heterogener Rechenzentrumsmarkt.....	41 -
3.3 Weitere Ansatzpunkte für Ressourceneffizienz.....	42 -
3.3.1 Nutzerverhalten in sozialen Netzen.....	42 -
3.3.2 Verursacherprinzip	43 -

4	Methodische Herausforderungen und Lösungsansätze	45
4.1	Methodische Herausforderungen	45 -
4.1.1	Definition funktioneller Einheiten	45 -
4.1.2	Messung des Energieverbrauchs von Software	46 -
4.1.3	Allokationsprobleme bei stark schwankender Auslastung	47 -
4.2	Existierende Lösungsansätze	50 -
4.2.1	Konzentration auf messbare Einzelaspekte	50 -
4.2.2	Standardisierungsbestrebungen	50 -
4.2.3	Vergleich funktionell ähnlicher Softwareprodukte	52 -
4.2.4	Vergleich eines Softwareprodukts mit sich selbst im Zeitverlauf	53 -
4.2.5	Green Software Engineering	53 -
5	Handlungsempfehlungen	56
5.1	Forschungs- und Standardisierungsbedarf	56 -
5.1.1	Entwicklung von Methoden und Standards	56 -
5.1.2	Regelmäßige Datenerhebung	57 -
5.2	Verbraucherbezogene Maßnahmen	57 -
5.2.1	Vergabe eines Blauen Engels für Software	57 -
5.2.2	Informationsmaßnahmen	58 -
5.3	Empfehlungen für Softwareentwickler	59 -
5.3.1	Leitfäden, Best-Practice-Richtlinien, Checklisten	59 -
5.4	Aus- und Weiterbildung	59 -
5.4.1	Lehrmittel zu Ressourcenaspekten von Softwarearchitekturen für die Informatikausbildung	59 -
5.4.2	Weiter- und Fortbildungsangebote zu Ressourcenaspekten der IKT-Nutzung für Unternehmen und öffentliche Einrichtungen	59 -
5.4.3	Angebote für den Informatikunterricht an Schulen	60 -
5.4.4	Wettbewerbe für ressourceneffiziente Software	61 -
	Literatur	62

Abstract

Obwohl Softwareprodukte immaterielle Güter sind, kann die Nutzung von Software erhebliche Stoff- und Energieströme auslösen. Eigenschaften der Software entscheiden, welche Hardwarekapazitäten vorgehalten werden und wieviel elektrische Energie in Endgeräten, Netzwerken und Rechenzentren verbraucht wird. Der Zusammenhang zwischen Softwareeigenschaften und dem Bedarf an natürlichen Ressourcen, der durch Herstellung und Betrieb von IKT-Systemen ausgelöst wird, ist bisher wissenschaftlich noch wenig untersucht. Die vorliegende Studie betritt Neuland, indem sie explorativ den Einfluss von Software auf die indirekte Inanspruchnahme natürlicher Ressourcen durch Hardware untersucht. Die Ressourceneffizienz von Software wird dabei im Kontext neuer Nutzungsformen wie mobiler Internetnutzung und neuer Software-Architekturmuster wie Cloud Computing betrachtet. Vor dem Hintergrund der Dynamik der aufgezeigten Trends identifiziert die Studie Ansatzpunkte im Softwarebereich, die zur Schonung natürlicher Ressourcen beitragen oder zumindest ein weiteres Wachstum ihrer Inanspruchnahme durch IKT-Systeme bremsen können. Sie geht dabei insbesondere auf methodische Probleme ein, die sich bei der Beurteilung der Ressourceninanspruchnahme von Softwareprodukten stellen. Zu diesen Problemen gehören Schwierigkeiten bei der Definition funktioneller Einheiten, Messprobleme und Allokationsprobleme. Ansätze wie die Standardisierung von Nutzungsmustern und Benchmarks sowie die Definition und Umsetzung von Nachhaltigkeitsanforderungen im Softwareentwicklungsprozess werden als mögliche Lösungswege aufgezeigt. Basierend auf diesen Überlegungen formuliert die Studie erste Handlungsempfehlungen in den Bereichen Forschung und Standardisierung, Produktkennzeichnung, Konfigurationsempfehlungen, Best-Practice-Leitfäden sowie Aus- und Weiterbildung im Bereich der ressourcen-effizienten Software und ihrer Entwicklung.

Abkürzungsverzeichnis

BITKOM	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.
BPMN	Business Process Model and Notation
CRM	Customer Relationship Management
DCIM	Data Center Infrastructure Management
DSL	Digital Subscriber Line
EASED	Energy Aware Software-Engineering and Development
EEG	Erneuerbare-Energien-Gesetz
GeSI	Global eSustainability Initiative
GHG	Green House Gas
GHGP	Green House Gas Protocol
GI	Gesellschaft für Informatik
GPL	General Public License, genauer: GNU General Public License
GPS	Global Positioning System
HD	High Density
HDTV	High Density Television
HSM	Hierarchisches Speichermanagement
IKT	Informations- und Kommunikationstechnik
IP	Internet Protocol
ISP	Internet Service Provider
IO	Input/Output
IT	Informationstechnik
ITU	International Telecommunications Union
LCA	Life Cycle Assessment
LTE	Long Term Evolution
MDD	Model Driven Development
OSS	Open Source Software
SUT	System Unter Test
WBCSD	World Business Council for Sustainable Development
WG	Workload Generator
WLAN	Wireless Local Area Network
WRI	World Resources Institute

1 Einleitung

1.1 Zielsetzung und Aufbau der Studie

Die Entwicklung und der Einsatz von Software bieten Optimierungspotenziale in Bezug auf die Schonung natürlicher Ressourcen. Obwohl Softwareprodukte immaterielle Güter sind, kann die Nutzung von Software erhebliche Stoff- und Energieströme auslösen. Eigenschaften der Software entscheiden, welche Hardwarekapazitäten vorgehalten werden und wieviel elektrische Energie in Endgeräten, Netzwerken und Rechenzentren verbraucht wird.

Damit wird Software zu einem wichtigen Ansatzpunkt für die Reduktion der Inanspruchnahme natürlicher Ressourcen durch heutige und zukünftige Systeme der Informations- und Kommunikationstechnik (IKT). Während bisher häufig die Ressourceneffizienz der Hardware im Fokus von „Green IT“ stand, verfolgt diese Studie das Ziel, Ansatzpunkte für Ressourceneffizienz im Softwarebereich zu identifizieren.

Der Zusammenhang zwischen Softwareeigenschaften und dem Bedarf an natürlichen Ressourcen,¹ der durch Herstellung und Betrieb von IKT-Systemen ausgelöst wird, ist bisher wissenschaftlich noch wenig untersucht. Das Thema ist auch im Bewusstsein von Entwicklern, Anwendern, wirtschaftlichen und politischen Entscheidungsträgern noch kaum verankert. Da die stete Weiterentwicklung der Hardware in der Vergangenheit immer genügend Leistungsreserven geschaffen hat, wurde der Effizienz in der Softwareentwicklung (außer bei mobilen Geräten) wenig Bedeutung beigemessen. Die Studie betritt Neuland, indem sie explorativ den Einfluss von Software auf die Inanspruchnahme natürlicher Ressourcen durch IKT-Systeme untersucht.

Diese Aufgabe erweist sich als methodische Herausforderung, weil jedes isoliert betrachtete Softwareprodukt immer nur als Teil eines komplexen IKT-Systems und damit in Wechselwirkung mit anderen Hard- und Softwarekomponenten (und dem Nutzer/der Nutzerin) seine Funktion erfüllt. Die insgesamt beanspruchten Hardwarekapazitäten sind es aber, die auf dem Weg über den Stromverbrauch und über den Hardware-Lebenszyklus die Inanspruchnahme natürlicher Ressourcen bewirken. Zudem sind die Innovationszyklen im IKT-Bereich so kurz, dass Ergebnisse, die sich auf Momentaufnahmen stützen, schnell veralten. Die Analyse legt den Fokus deshalb auf die qualitativen Kausalzusammenhänge und die dynamische Entwicklung.

Der vorliegende Schlussbericht fasst die Ergebnisse des Vorhabens in folgenden Kapiteln zusammen:

Kapitel 2 analysiert aktuelle Trends in den IKT-Nutzungsformen und Software-Architekturen. Der Fokus liegt dabei auf Entwicklungen, von denen nach einer ersten Einschätzung Effekte auf die Inanspruchnahme natürlicher Ressourcen zu erwarten sind. Die Ressourcenrelevanz jedes Trends wird aufgezeigt.

Kapitel 3 dokumentiert das Ergebnis einer Analyse der Potenziale zur Ressourcenschonung verschiedener Bereiche und Maßnahmen. Vor dem Hintergrund der Dynamik der aufgezeigten Trends werden Ansatzpunkte im Softwarebereich identifiziert, um natürliche Ressourcen zu schonen oder zumindest ein weiteres Wachstum ihrer Inanspruchnahme zu bremsen.

¹ Siehe die Definitionen in Tabelle 1, Seite 8

Kapitel 4 diskutiert methodische Probleme und Lösungsansätze – insbesondere in Hinblick auf eine operationalisierbare Beurteilung von Softwareprodukten als „grüne“ Software – und zeigt den weiteren Forschungsbedarf auf.

Kapitel 5 enthält Handlungsempfehlungen im Sinne einer politischen Prioritätensetzung: Wo müssten Maßnahmen ansetzen, um Anreize zu schaffen, IKT-Systeme auch auf Softwareebene ressourceneffizienter zu gestalten?

1.2 Grundlegende Definitionen

Tabelle 1 definiert grundlegende Begriffe dieser Studie.

Tabelle 1: Grundlegende Definitionen

Bezeichnung	Definition	Quelle
(Computer-) Software	Sammelbegriff für ausführbare Programme und die dazugehörigen Daten.	Wikipedia (2013a)
(Computer-) Hardware	Sammelbegriff für die materiellen Bestandteile eines Computersystems; die zur Ausführung von Software verwendeten Geräte.	eigene Formulierung
IKT-System	Sammelbegriff für Systeme der Informations- und Kommunikationstechnik. (In diesem Text keine Unterscheidung zu „IT-System“.) IKT-Systeme bestehen aus Hardware und Software.	eigene Formulierung
Ressource	Mittel, das in einem Prozess genutzt wird oder genutzt werden kann. Eine Ressource kann materieller oder immaterieller Art sein.	UBA (2012)
Ressource, Hardware-	Eine Hardwareressource ist eine Komponente der Computer-Hardware, z.B. ein Prozessor, Arbeitsspeicher, Hintergrundspeicher oder eine Netzwerkkomponente.	eigene Formulierung
Ressource, natürliche	Ressource, die Bestandteil der Natur ist. Hierzu zählen erneuerbare und nicht erneuerbare Primärrohstoffe, physischer Raum (Fläche), Umweltmedien (Wasser, Boden, Luft), strömende Ressourcen (z.B. Erdwärme, Wind-, Gezeiten- und Sonnenenergie) sowie die Biodiversität.	UBA (2012)
Ressourcen-effizienz	Verhältnis eines bestimmten Nutzens oder Ergebnisses zum dafür nötigen Ressourceneinsatz.	UBA (2012)

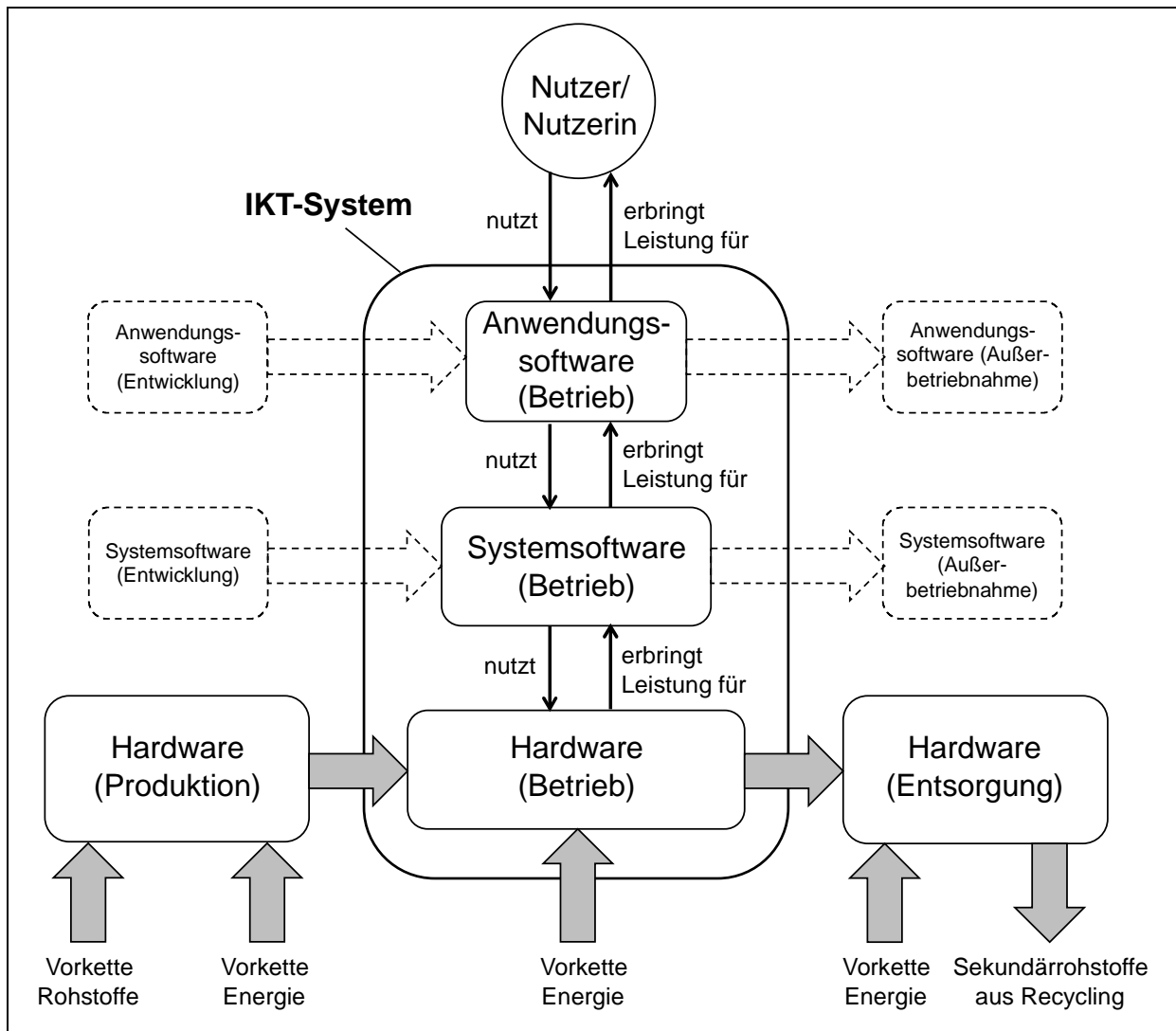
Rebound-Effekt	Der Effekt, dass als Folge erhöhter Ressourceneffizienz die Nachfrage nach einem Sachgut oder einer Dienstleistung zunimmt, so dass weniger Ressourcen eingespart werden, als dies unter der Annahme konstanter Nachfrage der Fall gewesen wäre, oder mehr Ressourcen verbraucht werden.	eigene Formulierung
----------------	--	---------------------

1.3 Untersuchungsrahmen

Gegenstand dieser Studie ist die Beanspruchung natürlicher Ressourcen, die bei der Nutzung von Funktionen von IKT-Systemen in Kauf genommen wird, und der Einfluss von Eigenschaften der Software auf das Ausmaß der Beanspruchung.

Als Untersuchungsmethode bietet sich eine Lebenszyklusanalyse (Life Cycle Assessment, LCA) für Softwareprodukte bzw. durch IKT-Systeme erbrachte Dienstleistungen an. Dies ist jedoch aus zwei Gründen nicht möglich: Erstens würde der Aufwand einer Serie von LCA-Studien den Umfang des Projekts weit überschreiten, und zweitens stoßen LCA-Studien für Software an ungelöste methodische Probleme. Diese reichen von der Schwierigkeit, funktionelle Einheiten und Nutzungsmuster für Software zu definieren, über die Dynamik des IKT-Marktes bis zu dem Problem, dass das gleiche Softwareprodukt in unterschiedlichen Hardware- und Softwareumgebungen unterschiedliche Ressourcennutzungen auslöst.

Abbildung 1: Vereinfachte Darstellung des IKT-Produktsystems. Graue Blockpfeile stehen für die wichtigsten Material- und Energieflüsse. Emissionen in Luft und Wasser sowie Abfälle und Materialflüsse für den Betrieb von Hardware (z.B. Tonerkartuschen) sind der Einfachheit halber nicht dargestellt.



Quelle: Eigene Darstellung

Abbildung 1 zeigt das IKT-Produktsystem, das die Lebenszyklen von Anwendungssoftware, Systemsoftware und Hardware enthält. In der Nutzungsphase fügen sich diese drei Produktarten zu einem IKT-System zusammen, das die gewünschte Leistung erbringt. Diese Darstellung ist insofern stark vereinfacht, als ein IKT-System heute in der Regel ein verteiltes System ist, in dem mehrere über ein Netzwerk verbundene Hardware- und Softwarekomponenten zusammenwirken.

Aufgrund der ungelösten methodischen Probleme bei der Anwendung der LCA-Methodik auf IKT-Systeme werden wir zwar von einer Lebenszyklusperspektive ausgehen, aber keine LCA-Studie im üblichen Sinn durchführen. Vielmehr untersuchen wir den Einfluss von Software auf die Inanspruchnahme natürlicher Ressourcen rein qualitativ und mit folgenden Einschränkungen:

1. - Wir betrachten ausschließlich System- und Anwendungssoftware für den Massenmarkt. Bei diesen Softwareprodukten steht der einmalige Aufwand für die Entwicklung einer Version einer sehr großen Zahl von Installationen und einer noch größeren Zahl von Ausführungen der

Programme gegenüber. Durch den Multiplikationseffekt in der Nutzungsphase erscheint es gerechtfertigt, die Phasen der Produktion und der Außerbetriebnahme der Software *a priori* zu vernachlässigen, die in einer LCA-Studie berücksichtigt würden (gestrichelte Lebenszyklusphasen der Software in Abbildung 1).

2. In der Nutzungsphase der Software betrachten wir ausschließlich die folgenden vier Wirkungsmechanismen, durch die ein Softwareprodukt natürliche Ressourcen beanspruchen kann:
 - a) *Stromverbrauch*: Software verursacht während ihrer Ausführung Stromverbrauch durch die Hardware; dies gilt nicht nur für das lokale Endgerät, sondern auch für Netzwerkkomponenten, Server und weitere Geräte, die am Prozess mitwirken.² Die Bereitstellung des elektrischen Stroms beansprucht natürliche Ressourcen.
 - b) *Hardware-Auslastung*: Software belegt während ihrer Ausführung Hardwarekapazitäten, die während dieser Zeit nicht anderweitig genutzt werden können. Deshalb ist ein entsprechender Anteil am gesamten Lebenszyklus der Hardware der Software zuzurechnen. Besonders die Produktion von Hardware beansprucht natürliche Ressourcen.
 - c) *Hardware-Management*: Software kann Betriebszustände der Hardware beeinflussen, insbesondere Energiesparmodi nutzen oder verhindern und die Rechen- oder Speicherlast in Netzwerken verteilen. Dies beeinflusst das Ausmaß und den Zeitpunkt des Stromverbrauchs durch die Hardware und auch die Effizienz der Hardware-Nutzung.
 - d) - *Hardware-Nutzungsdauer*: Softwareprodukte können den Zeitpunkt der Außerbetriebnahme (Obsoleszenz) von Hardwareprodukten beeinflussen, indem z. B. neue Versionen eines Softwareprodukts den Ersatz funktionierender Hardware durch leistungsfähigere Hardware nahelegen oder indem schlankere Installationen oder effizientere Versionen eines Softwarepakets die Weiternutzung älterer Hardware ermöglichen.
3. - Die Produktion und Entsorgung der Hardware sowie die Vorketten der Bereitstellung elektrischer Energie werden nicht im Detail betrachtet. Software ist zwar die Ursache eines Massenflusses (Stoffflusses) in Form von Hardware von einer Versorgungs- in eine Entsorgungskette und eines Energieflusses von einem Stromnetzanschluss durch die Hardware in die Umwelt (in Form von Abwärme), aber sie ändert diese Flüsse im wesentlichen quantitativ, nicht qualitativ. Abhängig von der eingesetzten Software können unterschiedliche Hardwarekapazitäten (Rechenleistung, Speicherplatz, Bandbreite von Netzwerken) benötigt werden, aber die Hardware wird – von Ausnahmen abgesehen – aus den gleichen Rohstoffen hergestellt.³ Ähnliches gilt für die Energiebereitstellungskette, die von der Software nicht beeinflusst werden kann, es sei denn sehr indirekt, indem die Software eine räumliche oder zeitliche Verlagerung der Ausführung von Aufgaben zulassen kann und damit auch den jeweiligen Strommix beeinflusst.⁴ Von diesen Ausnahmen abgesehen, gehen wir davon aus,

²- Auch bei Geräten, die im Dauerbetrieb laufen und deren Stromverbrauch nicht von der aktuellen Last abhängig ist, kann ein Anteil des Stromverbrauchs der Software zugerechnet werden. Dieses Allokationsproblem wird in Abschnitt 4.1.3 diskutiert.

³- Die Ausnahmen können im Einzelfall relevant sein. Beispielsweise kann die Verlagerung von Aufgaben vom Endgerät in ein Rechenzentrum dazu führen, so dass mehr Rechenzentrumskapazität und dadurch mehr Kältemittel für die aktive Kühlung benötigt wird. Auch können sich verschiedenen Größenklassen von Harddisks in ihrer Materialzusammensetzung unterscheiden, so dass ihre Herstellung andere Rohstoffe erfordert.

⁴- Dies ist insofern relevant, als sich dadurch z. B. im Cloud Computing gezieltes Demand Shaping betreiben lässt (als Beitrag zur Integration stark fluktuierender erneuerbarer Energiequellen wie Solar- und Windkraft in die Stromversorgung, siehe hierzu Abschnitt 3.1.5).

dass Software nur den Umfang, nicht aber die Art der Hardwareproduktion und-entsorgung beeinflusst, und dass sie auch keinen Einfluss auf die Art der Stromerzeugung ausübt. Die Systemgrenze für unsere Untersuchung entspricht somit der Grenze des IKT-Systems, wie in Abbildung 1 gezeigt.

4. - Wir ermitteln keine Umweltbelastungsindikatoren wie z. B. Treibhausgasemissionen, da diese weitgehend von den Annahmen über die Energiebereitstellungsketten, die Hardwareproduktion und -entsorgung beeinflusst wären und keinerlei Differenzierung innerhalb des von uns betrachteten Systems ermöglichen würden. Unsere Analysen zur Inanspruchnahme von Ressourcen enden also mit Aussagen zu den beanspruchten Hardwarekapazitäten, dem dadurch verursachten Stromverbrauch und der eventuell ausgelösten oder verzögerten Hardware-Obsoleszenz.
5. - Auch diese Aussagen können nicht als exakte und absolute, sondern nur als ungenaue und relative Angaben erarbeitet werden. Dies ist darauf zurückzuführen, dass diese Studie die prinzipiellen kausalen Zusammenhänge und Entwicklungstrends anstelle von Momentaufnahmen untersucht. Weder aus den in der Literatur vorhandenen Daten noch aus eigenen Erhebungen könnten verlässliche Trendaussagen abgeleitet werden, da sich sowohl Hardware als auch Software durch die kurzen Innovationszyklen im IKT-Sektor schnell verändern.

Aus den genannten Gründen kann diese Studie nicht direkt als Grundlage zum Vergleich von Softwareprodukten unter dem Aspekt ihrer Ressourceneffizienz (etwa in Hinblick auf eine Produktkennzeichnung) herangezogen werden. Nach unserer Auffassung ist das derzeit auch prinzipiell nicht leistbar, da entscheidende methodische Probleme ungelöst sind. Diese Probleme und mögliche Lösungsansätze werden deshalb in einer Methodendiskussion thematisiert (Kapitel 4).

2 Trendanalyse

Die Eigenschaften und Einsatzformen von Software für den Massenmarkt verändern sich schnell. Wichtiger als eine Analyse des momentanen Zustands in Bezug auf die beanspruchten Hardwarekapazitäten (und damit indirekt natürlicher Ressourcen) ist daher eine Erfassung der wichtigsten Trends im Softwarebereich.

Wir gehen davon aus, dass eine Wechselwirkung zwischen den im Markt dominierenden Softwarearchitekturen und den Nutzungsformen der Software besteht: Bestimmte Architekturmuster (z. B. Cloud Computing) ermöglichen und begünstigen bestimmte Nutzungsformen von Software (z. B. mobile Anwendungen), die dadurch wiederum stärker nachgefragt werden und die Etablierung des Architekturmusters fördern. Beide Seiten einer solchen Entwicklung, das Architekturmuster und die Nutzungsform, können hinsichtlich der benötigten Hardwarekapazitäten eine relevante Veränderung darstellen.⁵

Die im Folgenden beschriebenen Entwicklungstrends wurden nach ihrer voraussichtlichen Relevanz für die Beanspruchung von Hardwarekapazitäten (und damit indirekt natürlicher Ressourcen) ausgewählt: Welche heute erkennbaren Entwicklungen lassen erwarten, dass sie erhebliche (positive oder negative) Auswirkungen in Bezug auf die Inanspruchnahme natürlicher Ressourcen haben? Die Ressourcenrelevanz wird im Anschluss an jede Trendbeschreibung begründet. Die Trends sind in zwei Gruppen eingeteilt:

- Abschnitt 2.1: übergeordnete Trends, welche die *Rahmenbedingungen* für Softwarearchitekturen und Nutzungsformen von Software verändern;
- Abschnitt 2.2: Trends im Bereich der Softwarearchitekturen und Nutzungsformen selbst.

Diese Trendanalyse wird als Grundlage für die Analyse von Ansatzpunkten zur Ressourcenschonung in Kapitel 3 dienen.

2.1 Übergeordnete Trends

2.1.1 Mobiler Internetzugang

Die Nutzung des Internet über mobile Endgeräte wie Laptops, Tablets oder Smartphones nimmt zu. Bereits in den frühen Phasen dieser Entwicklung haben LCA-Studien auf die Energieintensität (Energie pro Datenmenge) der Mobilfunknetze als Übertragungsweg hingewiesen (Scharnhorst et al. 2006, Emmenegger et al. 2006). Neuere Studien bestätigen dies und benennen den mobilen Zugang über WLAN als Alternative mit wesentlich geringerer Energieintensität (CEET 2013).

⁵ Beispiel: Cloud Computing ist ein Architekturmuster, das u. a. aus der Absicht entstanden ist, Hardwarekapazitäten besser auszulasten und dadurch ressourceneffizienter zu nutzen (vgl. die Definition von Ressourceneffizienz in Abschnitt 1.2); Cloud Computing begünstigt zugleich die Nutzungsform „Mobiler Internetzugang“, die durch den entstehenden Datenverkehr in Mobilfunknetzen zusätzlichen Energieverbrauch verursacht. Nähere Angaben hierzu in den Abschnitten 2.1.1 und 2.2.3.

Gleichzeitig nimmt die Bildauflösung der Displays mobiler Geräte zu. Es wird erwartet, dass Internetfernsehen und andere Videodienste auch im mobilen Bereich eine wachsende Rolle spielen werden (Lücke 2011; Cisco 2013a).

Smartphones bekommen zusätzliche Funktionalitäten wie Bezahlen (Sawall 2011), mobile Tickets (z.B. Heise Mobil 2011c), Navigation oder Augmented Reality.

Funktionalität, die bisher durch Spezialgeräte angeboten wurde, wird durch Smartphones und zunehmend auch Tablets ersetzt. Beispiele:

- Fotografieren mit anschließendem Upload der Bilder.
- Navigationsgeräte sind auf dem Rückzug, da GPS-Unterstützung praktisch in jedem Smartphone und in vielen Tablets vorhanden ist und mit Zugriff auf das Internet kleine Programme vergleichbare Funktionalität auf der Basis von Internet-Diensten wie Google Maps oder Bing Maps bieten, Apps sorgen für Sprachausgabe (Graf 2012; Kossel 2013).

Der Einsparung eines separaten Gerätes (Digitalkamera, Navigationsgerät) steht hier der zusätzlich erzeugte Datenverkehr durch mobilen Internetzugang gegenüber.

Nach einer aktuellen Studie des Centre for Energy-Efficient Telecommunications (CEET) der University of Melbourne ist der Mobilfunk das energieintensivste Zugangsnetz zum Internet, wobei die Zugangsnetze generell den energieintensivsten Teil des Internet darstellen (nur auf die Nutzungsphase bezogen; CEET 2013).

Es ist zu beachten, dass in Berechnungen der Energieintensität von Datenverkehr notwendigerweise Annahmen über mittlere Auslastungen von Geräten einfließen, so dass eine Hochrechnung auf hypothetische Szenarien (wie beispielsweise einen flächendeckenden Einsatz öffentlicher WLAN-Hotspots) nicht auf einfache Weise möglich ist und Modellrechnungen erfordert (siehe z. B. Schien et al. 2013)

Ressourcenrelevanz:

Der Übergang vom leitungsgebundenen zum mobilen Internetzugang hat mehrere Auswirkungen, die sich teils positiv und teils negativ auf die Ressourcenbeanspruchung durch Internetnutzung auswirken.

- Mobile Endgeräte sind im Vergleich zu den üblichen stationären Geräten hinsichtlich Energieverbrauch optimiert; mobile Apps minimieren sowohl die lokale Rechenleistung als auch den Datenverkehr der einzelnen Anwendung (der allerdings beim Zugang über Mobilfunknetze sehr energieintensiv ist).
- Der mobile Zugang schafft einen Anreiz zur intensiveren Nutzung des Internet und insbesondere zur Nutzung von Cloud-Diensten (zur Ressourcenrelevanz von Cloud Computing siehe Abschnitt 2.2.3)
- Die Nutzung von Mobilfunknetzen für den Internetzugang ist mit einem relativ hohen Energieaufwand verbunden; die vorhandenen Daten und Prognosen legen die Schlussfolgerung nahe, dass die Energieintensität des Zugangs (Energie pro Datenmenge) deutlich höher liegt als beim leitungsgebundenen Zugang.
- Der Übergang von stationären zu mobilen Endgeräten führt zu einer Verkürzung der durchschnittlichen Nutzungsdauer der Hardware, da mobile Geräte weniger lang im Gebrauch sind, u.a. weil es schwerer ist, Einzelkomponenten auszutauschen. Bei Mobiltelefonen kommt hinzu,

dass der Kunde durch die Preisgestaltung animiert wird, bei Vertragsverlängerung das Gerät zu ersetzen.

2.1.2 Foto und Video in immer höherer Auflösung

Die Ansprüche an die Bildqualität bei Kommunikationsdiensten, Videoangeboten und Computerspielen steigen laufend und reizen jeweils die aktuellen technischen Möglichkeiten aus. Beispiele:

- Fotos und Videos werden in hoher Auflösung auf Austauschplattformen wie Flickr, Facebook⁶ und Youtube hochgeladen; seit 2010 unterstützt Youtube auch Video im sog. 4k-Format (bis zu 13 Millionen Pixel pro Bild, Kinoformat).
- Computerspiele gelten als der stärkste Treiber für wachsende Leistungsanforderungen an die Endgeräte; realitätsnahe Bilddarstellungen erfordern extreme Rechenleistungen.
- IP-Fernsehen, also über das Internet⁷ bereitgestellte Fernsehhalte, verbreiten sich zunehmend; durch den damit verbundenen Übergang von *Broadcast* zu *Unicast* (der Zuschauer bestimmt den Zeitpunkt selbst) vervielfachen sich die Datenströme.

Die Speicherung, Darstellung, Manipulation und Übertragung von Bilddaten setzt aufgrund der hohen Datenmenge leistungsfähige Hardware voraus. Für jeden dargestellten Bildpunkt (jedes Pixel) in einem Farbbild wird im unkomprimierten Zustand die Information zu 3 Farben gespeichert. Bei einem Video in HD Auflösung beispielsweise besteht jedes einzelne Bild aus 1920 x 1080, also rund 2 Millionen Pixeln. Bei Spielen mit realitätsnahen Bilddarstellungen kommt hinzu, dass die Information teilweise erst noch in einem Renderingprozess ermittelt wird.⁸

Die Speicherung und Übertragung von Videodaten erfolgt oft in einem komprimierten Format. Die Kodierung und Dekodierung wird durch spezielle Programme (sog. Codecs) durchgeführt. Je nach Umfang der Daten (z. B. Videoauflösung) kann die Wiedergabe über Software-Dekodierung auf einem PC hohe Anforderungen an die Rechenleistung stellen. In vielen Grafikkarten ist Hardware integriert, die das Kodieren und Dekodieren unterstützt.

Auch komprimierte Datenströme von Videos erzeugen im Vergleich zu anderen Datenarten relativ hohen Datenverkehr im Internet. Ein Internet-fähiger HD-Fernseher, der 30 Minuten Inhalte am Tag aus dem Internet bezieht, erzeugt gleich viel Datenverkehr wie ein durchschnittlicher Haushalt insgesamt (Cisco 2012a). Im Bereich der privaten Internet-Nutzung ist die audiovisuelle Unterhaltung deshalb heute der wichtigste Treiber der Nachfrage nach Bandbreite.

Als Nebentrend verlagert sich ein Teil des Fernsehkonsums auf mobile Geräte (Heise Online 2011a). In Zukunft werden ferner interaktive Angebote zunehmen (der Zuschauer wird ins Programm einbezogen, Unterhaltungen mit anderen Zuschauern werden möglich). Der Trend zu 3D-Angeboten könnte die Datenintensität ein weiteres Mal verdoppeln, weil dann zwei Bilder gleichzeitig übertragen werden (eines für jedes Auge). Hersteller für mobile Geräte stellen sich auf diese

⁶ Facebook ist zu einem der größten Foto-Archive des Internet. Laut Angaben von Pixable werden monatlich rund 6 Milliarden Fotos hochgeladen (Pixable 2011).

⁷ Anstelle des öffentlichen Internet kann auch ein geschlossenes Netz zu Übertragung verwendet werden. Wir verwenden „IP-Fernsehen“ als Oberbegriff für die Übertragung von Fernsehprogrammen auf Basis des Internet Protokolls (IP), unabhängig davon, ob die Übertragung über das Internet („Web TV“) oder ein anderes Netz geschieht (vgl. auch Deutscher IPTV-Verband 2010).

⁸ Schatten und Materialeigenschaften werden pixelweise hinsichtlich des Grades der Reflexion und Absorption der Lichtstrahlen berechnet, die bei der virtuellen Kamera ankommen.

Entwicklung ein und bieten 3D-fähige Geräte an, beispielsweise LG und HTC (Heise Mobil 2011a; Heise Mobil 2011b). Der vorzeitige Ersatz funktionierender TV-Geräte durch 3D-fähige bewirkt außerdem zusätzlichen Material- und Energieverbrauch.

Ressourcenrelevanz:

- Die Kodierung und Dekodierung (audio-)visueller Daten in ausreichender Geschwindigkeit führt auf dem Endgerät zu Energieverbrauch und schafft eine Nachfrage nach entsprechend leistungsfähigen Geräten, teilweise auch nach Zusatzgeräten für die Nutzung spezifischer Angebote (wie Set-top-Boxen).
- Die Übertragung und Speicherung audiovisueller Daten gehört zu den ressourcenintensivsten Massen Anwendungen des Internet; das Wachstum des Datenvolumens führt zum Ausbau von Netz- und Serverkapazitäten mit entsprechendem Material- und Energieverbrauch.

2.1.3 Spam

In der Konkurrenz um das knappe Gut Aufmerksamkeit wird jeder Kommunikationskanal, der sich automatisch beschicken lässt, für ungebetene Werbung oder andere unerwünschte Mitteilungen ausgenutzt.

Das Versenden unerwünschter Nachrichten lässt sich leicht automatisieren, das Herausfiltern dieser Nachrichten ist dagegen nur schwer und nicht perfekt automatisierbar. Die Übertragung und Bekämpfung von Spam beansprucht Hardwarekapazitäten von Netzwerkkomponenten, auf Servern und Endgeräten.

Einer Studie von McAfee (2009) zufolge verursacht der weltweite E-Mail-Spam einen jährlichen Energieverbrauch von 33 Milliarden Kilowattstunden. Das entspräche dem Stromverbrauch von 2,4 Millionen US-Haushalten.

Neben E-Mail-Spam gibt es auch zunehmend Spam in sozialen Netzwerken ("social spam"). Es wird geschätzt, dass 4 % des geteilten Inhalts auf Facebook und 1.5 % der Tweets bei Twitter in diese Kategorie fallen; bei Facebook werden rund 400 Personen eingesetzt, die nur die Aufgabe haben, dieses Problem einzugrenzen (Fowler et al. 2012).⁹

Spam könnte in Zukunft vermehrt auf andere Medien übergreifen, z.B. automatisierte Telefonanrufe. Auch Anzeigetafeln im öffentlichen Raum oder in Schaufenstern, die sich auf die anwesenden Personen einstellen, etwa in Verbindung mit Video- oder Ortungstechnologien (Hilty et al. 2012) kann als unerwünschte Kommunikation empfunden werden. Ähnliches gilt für interaktive Verkaufsautomaten, die Passanten individuell ansprechen („smart vending machine“, Keller 2010; Hilty 2012).

E-Mail ist nach wie vor das Hauptmedium für den Versand von Spam. Laut Radicati Group (2011) gab es im Jahr 2011 weltweit über 3,1 Milliarden E-Mail-Accounts (davon private Accounts: 75 Prozent),

⁹ Facebook-Spammer nutzen Cloud Dienste wie Amazons S3, da unerwünschte Werbung dann nicht anhand der URL zu erkennen ist und die Kosten für den Speicherplatz bei Amazon gering sind (Sebayang 2012). Außerdem arbeiten die Spammer mit Browserweichein und Positionserkennung anhand von IP-Adressen. Durch Installation von dem Nutzer als vertrauenswürdig erscheinenden YouTube-Plugins wird es dann möglich, weiteren Spam über Facebook-Pinwände zu verbreiten mit Links, die wieder auf Amazons S3-Dienst verweisen (Sebayang 2012).

die Anzahl täglich versendeter und empfangener E-Mails eines typischen beruflichen Nutzers wird mit 105 angegeben. Bis Ende 2015 werden 4,1 Milliarden E-Mail-Accounts und pro beruflichem Nutzer jeweils 125 E-Mails pro Tag erwartet. E-Mail ist damit das am häufigsten genutzte Kommunikationsmedium im Internet, auch wenn soziale Netze und moderne Collaboration-Tools zunehmend in Konkurrenz treten.

Nach einer Schätzung von Symantec sind 72.9 % aller versendeten E-Mails Spam (Symantec Corporation 2011). Der Cisco-Sicherheitsreport beziffert das weltweit auftretende Spam-Volumen für das Jahr 2011 mit täglich 124 Milliarden E-Mails (Cisco 2011) und verzeichnet einen Rückgang von 18 % in den letzten zwei Jahren (Cisco 2013b), der durch die Verfolgung und Stilllegung sogenannter Botnetze zu erklären ist.

Trotz verbesserter Spamfilter ist nach einer Prognose 2012-2016 von Radicati Group (2011) damit zu rechnen, dass weiterhin rund 15 % der empfangenen E-Mails Spam beinhalten. Gleichzeitig mit der kontinuierlichen Weiterentwicklung der Anti-Spam-Software entwickeln die Spammer immer neue und bessere Methoden, um installierte Spam-Filter zu umgehen.

Ressourcenrelevanz:

- Die Versendung und Behandlung von Spam beansprucht Hardwarekapazität in Netzen und Endgeräten.
- Neue Formen von Spam, z. B. multimedialer Spam, könnten die Entwicklung noch ressourcenintensiverer Spamfilter erfordern.
- Langfristig könnte das Spam-Problem, wenn es nicht technisch gelöst wird, durch Internet-Governance-Strukturen gelöst werden, die eine Durchsetzung des Verursacherprinzips im Internet ermöglichen würden. Eine solche Entwicklung, die vielfältige gesellschaftliche Chancen und Risiken in sich bergen würde, könnte theoretisch auch für eine verursachergerechte Zurechnung der Ressourceninanspruchnahme genutzt werden.

2.2 Trends im Bereich der Softwarearchitekturen und Nutzungsformen

2.2.1 Das „Prinzip App“

Als „App“ (Kurzform von „application software“) bezeichnet man im gegenwärtigen Sprachgebrauch meist mobile Anwendungsprogramme, die über einen Online-Shop zu erwerben und leicht zu installieren sind. Oft haben Apps nur eine oder wenige Funktionen, die dafür aber eine klar definierte Aufgabe gut unterstützen (BITKOM 2011b).

Apps müssen für die verschiedenen mobilen Betriebssysteme jeweils neu erstellt oder angepasst werden. Die meist verbreiteten, untereinander weitgehend inkompatiblen Systeme sind iOS, Android, Symbian, WindowsPhone, RIM OS und WebOS. Zwei asiatische Betriebssysteme von Baidu Inc. und Alibaba Group sind in Entwicklung (Fletcher 2011).

Die Zahl der Apps wird auch weiterhin stark anwachsen, da sich hier ein neues Geschäftsfeld darbietet (BITKOM 2011b). Dabei werden sich die Bestrebungen verstärken, gleiche Apps auf verschiedenen Geräten ohne Neuprogrammierung zu erstellen.¹⁰

Daneben werden vermehrt Web-Apps (Apps, die ohne Installation durch das Web bereitgestellt werden und gewöhnlich innerhalb eines Browsers ablaufen) auf der Grundlage von HTML5 entwickelt.

Apps nutzen in der Regel den Prozessor und Speicher des Endgerätes äußerst effizient und versuchen den Datenverkehr zu minimieren, um Energie zu sparen. Dies geschieht im Interesse der Akkulaufzeit der mobilen Geräte. Relevant werden sie durch die Vielzahl der Installationen und den damit verbundenen Übertragungs- und Speicherbedarf.

Apps unterstützen den Trend zum mobilen Internetzugang (Abschnitt 2.1.1); viele Apps stellen lediglich Dienste aus dem Internet in vereinfachter Form bereit, z. B. die Apps für Twitter und Facebook.

Solche Apps übertragen möglichst wenig Daten (z.B. weniger Bilder der anbietenden Webseite), damit die vom Nutzer wahrgenommene Geschwindigkeit möglichst hoch bleibt. Oft vermeidet man dabei auch die Ausführung rechenintensiver Skripte.¹¹

Ressourcenrelevanz:

- Mobile Apps sind in der Regel effizient programmiert, da man den Akku der mobilen Geräte schonen möchte; dies könnte sich effizienzfördernd auch auf die Anwendungssoftware für stationäre Geräte auswirken.
- Der mobile Internetzugang wird durch Apps gefördert (vgl. Abschnitt 2.1.1).
- Für den mobilen Internetzugang sind maßgeschneiderte Apps potenziell effizienter als ein Webbrowser, weil sie Daten selektiver übertragen.

2.2.2 Webbasierte Anwendungssoftware

Als webbasierte Anwendungssoftware oder Webanwendungen bezeichnet man Programme, die mit Hilfe von Webbrowsern ausgeführt werden. Solche Software hat den Vorteil, dass sie nicht installiert und lokal gepflegt werden muss und nur bei Bedarf überhaupt auf das Endgerät heruntergeladen wird (Software-as-a-Service).¹²

Mit dem HTML5-Standard kann Audio, Video und Grafikerunterstützung für Webseiten bereitgestellt werden. HTML5 ist ebenfalls die Grundlage des sozialen Netzes Google+.

Neben den vielen Apps, die auf mobilen Geräten installiert werden, werden vermehrt Web-Apps angeboten, also webbasierte Varianten von Apps, die eine vergleichbare Funktionalität wie eigenständige Apps anbieten. Einerseits wird so der Mehraufwand der Entwicklung für verschiedene Plattformen gering gehalten; andererseits werden die Nutzer ab einer bestimmten Anzahl von

¹⁰ Beispiel: die Sprache Mobl zur plattformunabhängigen Programmierung auf mobilen Geräten. Model Driven Development (MDD) wird ebenfalls schrittweise für mobile Plattformen eingesetzt (Holstein 2011).

¹¹ Das Nichtausführen von Flash-Werbung kann beispielsweise auf einem PC durchschnittlich 3 Watt Leistung einsparen (Simons & Pras 2010).

¹² Beispiele: Die Büro-Anwendungen Google Docs von Google und Office 365 von Microsoft.

Anwendungen nicht für jede weitere Aufgabe separat Software installieren wollen. Damit wird sich der Ansatz des Software-as-a-Service auf dem Umweg über das „Prinzip App“ weiter durchsetzen.

Aus Sicht der Webanwendung stellt der Browser das Betriebssystem dar. Die Integration von Microsofts Internet Explorer mit Windows deutete die Entwicklung zum Browser als Betriebssystem in der Vergangenheit schon an, das Chromebook von Google geht heute diesen Weg weiter. Das Chromebook ist ein einfacher Laptop, bei dem der Webbrowser (Chrome) die einzige Oberfläche ist und alle Aktivitäten – notwendigerweise online – mit Webanwendungen abgedeckt werden.

Begünstigende Faktoren sind eine “always on“-Mentalität, die starke Verbreitung von Mobilfunknetzen und bezahlbare Internet-to-Go-Tarife.

Der Anteil dynamischer (interpretierter) Programmiersprachen in Webbasierter Anwendungssoftware ist stark angestiegen.¹³ Da dynamische Sprachen relativ ineffizient mit Hardwareressourcen umgehen, besteht hier ein hohes Optimierungspotenzial in der Softwareentwicklung.

Ressourcenrelevanz:

- Webbasierte Anwendungssoftware ermöglicht die Verwendung schlanker Endgeräte, insbesondere mit wenig Plattenspeicher.
- Die Ausführung der Software ist für den Prozessor potenziell ineffizient, es gibt aber Ansatzpunkte für weitreichende Optimierungen.
- Webanwendungen fördern den Trend, immer online zu sein.
- Bei mobilen Webanwendungen sind – wie beim mobilen Internetzugang generell – die Mobilfunketze als besonders ressourcenintensive Zugangsnetze zu berücksichtigen.

2.2.3 Virtualisierung und Cloud Computing

Die Virtualisierung von Servern ist das derzeit stärkste Mittel, um eine steigende Nachfrage nach IKT-Leistungen zu befriedigen, ohne die Hardware-Infrastruktur im gleichen Ausmaß auszubauen. Mehrere virtuelle Server können auf einem physischen Server laufen, der dadurch besser ausgelastet ist. Insgesamt werden weniger physische Server benötigt, und ein Teil der vorhandenen kann vorübergehend abgeschaltet werden, wenn weniger Leistung benötigt wird. Diese Server-Konsolidierung verbessert die Energie- und Materialeffizienz pro Einheit IKT-Leistung, die bezogen wird.

Cloud Computing¹⁴ (deutsch etwa „Rechnen in der Wolke“) ist die Fortsetzung dieser Abstraktion von Hardware-Ressourcen in einem größeren Netzwerk (normalerweise dem Internet). Hardware-Ressourcen wie z. B. Rechenkapazität oder Speicherplatz, Betriebssystemplattformen oder Anwendungsprogramme werden dynamisch – an den momentanen Bedarf angepasst – über das Netzwerk zur Verfügung gestellt, ohne dass der Ort der tatsächlichen Leistungserbringung durch physische Hardware eine Rolle spielt. Bei geeigneter Lastverteilung sind erhebliche Konsolidierungseffekte

¹³ Dynamische Sprachen sind gerade im Web-Umfeld stark vertreten, z. B. PHP, Perl, JavaScript. Es gibt inzwischen leistungsfähige Frameworks, die Entwurfsmuster bereitstellen, wie Grails, Rails und CakePHP. -

¹⁴ Eine weithin verwendete Definition stammt vom National Institute of Standards and Technology, - Information Technology Laboratory: “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” (Mell & Grance 2009)

möglich. Sekundär geht es bei Cloud Computing auch darum, eine konsequente Dienstleistungsicht in der IKT umzusetzen (BITKOM 2010a).

Zu beachten ist, dass mit dem Begriff Cloud Computing in der aktuellen Diskussion sehr verschiedene Szenarien bezeichnet werden. Damit kann die gemeinsame Nutzung eines oder mehrerer Rechenzentren durch verschiedene Teile eines Unternehmens gemeint sein ("private cloud") oder in der "public cloud" der vollkommene Verzicht auf eigene Rechenzentren, wenn stattdessen Speicher- und Rechenleistung in weltweit platzierten Großrechenzentren genutzt wird; sowie sämtliche Mischformen.

Diese Szenarien können in der Gesamtsumme sehr unterschiedliche Auswirkungen auf den Energie- und Ressourcenverbrauch haben. Es liegen noch keine Daten vor, die eine vollständige quantitative Abschätzung erlauben. Die zur Verfügung stehenden Studien haben sehr unterschiedliche Untersuchungsrahmen und kommen entsprechend zu sehr unterschiedlichen Ergebnissen.

Die Hoffnung ist, dass durch die Cloud-Architektur die Nutzung von Hardwareressourcen weiter optimiert werden kann als durch rechnzentrumsinterne Virtualisierung. Es werden nur die Hardwareressourcen eingesetzt, die für einen Service benötigt werden. Eine Teilmenge der physischen Server kann ausgeschaltet werden, wenn geringe Lastanforderungen vorliegen. Überkapazität von laufenden Servern kann flexibel anderen Diensten zur Verfügung gestellt werden. Damit sind erhebliche Energieeinsparungen möglich. Ein Teil dieser Effekte ist auch durch Virtualisierung der betriebseigenen Server erreichbar.

Firmenrechenzentren werden ausgegliedert, da sich damit das Verhältnis von Wartungsaufwand zu Serveranzahl wesentlich verbessern lässt. Damit fallen viele Kosten für den Betrieb eines Rechenzentrums weg (das bezieht sich auch auf den Platzbedarf und die für den Betrieb notwendige Infrastruktur).¹⁵ Unter dem Strich beruht das Effizienzpotenzial des Cloud Computing darauf, dass die zur Abdeckung von Lastspitzen notwendigen Reservekapazitäten sehr vieler Nutzer zu Pools zusammengefasst werden, wodurch sich die Last verstetigt und sich somit die durchschnittliche Auslastung der physischen Kapazitäten verbessern kann. Der Umfang der tatsächlich vorgehaltenen Hardwarekapazitäten und die Ressourcenbeanspruchung durch den zusätzlich ausgelösten Datenverkehr müssen bei einer Gesamtbeurteilung jedoch berücksichtigt werden. Eine vollständige quantitative Abschätzung oder die Bildung von Szenarien ist auf Grundlage der vorhandenen Daten nicht möglich.

Erste empirische Untersuchungen zur Energieintensität von Cloud Computing haben ergeben, dass Cloud-Anwendungen im 1:1-Vergleich mit Offline-Anwendungen dann besser abschneiden, wenn die Aufgaben rechen- und speicherintensiv sind oder konstante Verfügbarkeit erfordern und mit relativ wenig Datentransfer auskommen. Für Aufgaben wie Textverarbeitung mit häufigen Benutzerinteraktionen und geringem Rechenaufwand und Speicherbedarf kann Cloud Computing dagegen mehr Energieaufwand verursachen (Williams & Tang 2013). Modellbasierte Abschätzungen sagen bei geschäftlichen Anwendungen signifikante Einsparungen für CRM-Software und Groupware (Williams et al. 2013) sowie für E-Mail, ebenfalls CRM-Software und weitere alltägliche Bürosoftware (Masanet et al. 2013) voraus.

¹⁵ Beispiel: Die US-Regierung schließt 800 Rechenzentren, weil die entsprechenden Dienste effizienter über Cloud-Dienste bereit gestellt werden können (Lohr 2011).

Masanet et al. untersuchen das technische Potenzial¹⁶ von Cloud Computing für Business-Anwendungen und berücksichtigen dabei auch die graue Energie der Hardware. Sie schätzen ein technisches Einsparpotenzial von 87 % des Energieverbrauchs bei einer Verlagerung der typischen Büro-Anwendungen in die Cloud: „If all U.S. business users shifted their email, productivity software, and CRM software to the cloud, the primary energy footprint of these software applications might be reduced by as much as 87 % or 326 Petajoules.“ (Masanet et al. 2013; S. 1). Den größten Anteil an dieser Einsparung hat die angenommene Konsolidierung im Serverbereich, da in die betrachteten Anwendungen in der Regel betriebseigene Server oder Rechenzentren involviert sind, die dann ganz entfallen. Beispielsweise geht die Studie davon aus, dass in US-Firmen über 3,5 Millionen E-Mail-Server in Betrieb sind, deren Leistung von weniger als 50.000 Servern in Cloud-Rechenzentren erbracht werden könnte. Die Client-Endgeräte wurden als unverändert angenommen. Die Auswirkungen der Datenübertragung im Internet waren in dieser Studie gering gegenüber den Auswirkungen des Serverbetriebs.

Diese Abschätzungen zur Energieeffizienz des Cloud Computing sagen aber nichts über die absoluten Effekte dieses Trends auf die Ressourcenbeanspruchung durch IKT aus. Denn Cloud Computing macht IKT-Leistungen verfügbar, die vorher nicht (oder nicht mobil) in diesem Umfang und zu diesen geringen Kosten zugänglich waren. Da in der Geschichte der IKT stets Rebound-Effekte aufgetreten sind (Hilty et al. 2006), ist mit diesen auch im Falle von Cloud Computing zu rechnen.

Die meisten mobil genutzten Dienste sind heute als Cloud-Dienste realisiert. Nach einer Modellrechnung des Centre for Energy-Efficient Telecommunications (CEET) der University of Melbourne hat dabei der mobile Zugang (über Mobilfunk oder WLAN) den höchsten Anteil am Energieverbrauch der gesamten Nutzung des Cloud-Dienstes. Der weltweite geschätzte Energieverbrauch von 9173 GWh für mobile Nutzung von Cloud-Diensten im Jahr 2012 teilt sich nach der CEET-Studie wie folgt auf: 5307 GWh für den Zugang über Mobilfunk (LTE), 2413 GWh für den Zugang über WLAN, 1437 GWh für die eigentliche Erbringung der Dienste in Rechenzentren, 16 GWh für die leitungsgebundenen Teile des Internet. (CEET 2013)

Bis 2015 wird ein starkes Wachstum der mobilen Cloud-Dienste prognostiziert mit einem entsprechenden Energieverbrauch, der durch den mobilen Zugang dominiert wird: „The predicted large-scale take-up of wireless cloud services will consume 32 to 43 TWh by 2015. The energy consumption of wireless access dominates data centre consumption by a significant margin.“ (CEET 2013, S. 15).

Dass in der CEET-Studie die Datenübertragung den höchsten Anteil am Gesamtverbrauch hat, während sie in der oben zitierten Studie von Masanet et al. (2013) vernachlässigbar ist, ist hauptsächlich darauf zurückzuführen, dass die CEET-Studie allen Cloud-bedingten Datenverkehr abschätzt (einschließlich mobile Unterhaltung), während Masanet et al. sich auf wenige klar abgegrenzte Business-Anwendungen beziehen, deren Verlagerung in die Cloud nach eigener Annahme relativ wenig zusätzlichen Datenverkehr (+5 %) verursachen würde. Die CEET-Studie folgt einem Top-down-Ansatz (Gesamtschätzung minus geschätzte nicht-cloudbedingte Anteile), während

¹⁶ „It is critical to note that the results [...] represent our estimated technical potential for energy use and emissions savings associated with shifting from present day systems to cloud-based systems for software provision. Estimates of technical potential provide illustrative upper bounds on potential savings but do not take into account economic, infrastructure, temporal, institutional, or policy barriers that might limit the savings that can be achieved in real-world systems.“ (Masanet et al. 2013, S. 30).

Masanet et al. einen Bottom-up-Ansatz verfolgen (definierte Anwendungen und Aggregation ihrer Ressourcenbedarfe). Für eine Diskussion von Top-down- vs. Bottom-up-Ansätzen zur Abschätzung von Datenverkehr im Internet und dessen Energieintensität siehe auch Cororama et al. (2013).

Ressourcenrelevanz:

- Virtualisierung und Cloud Computing sind technisch-organisatorische Maßnahmen mit dem Potenzial, Hardware besser auszulasten bzw. nicht benötigte Hardware abzuschalten.
- Cloud Computing fördert die Verwendung von Endgeräten mit geringer Hardwarekapazität (Rechenleistung, Speicherplatz).
- Durch Virtualisierung können betriebseigene Rechenzentren konsolidiert, durch Cloud Computing in vielen Fällen ganz eliminiert werden.
- Cloud Computing erzeugt zusätzlichen Datenverkehr im Internet, was insbesondere bei mobiler Nutzung ressourcenrelevant ist; eine starke Zunahme der mobilen Nutzung ist zu erwarten.
- Es ist damit zu rechnen, dass Cloud Computing dort eine Erhöhung des Energie- und Ressourcenverbrauches durch Rebound-Effekte auslöst, wo IKT-Leistungen für den Nutzer dank Cloud Computing billiger, schneller oder bequemer verfügbar werden oder neue Anwendungsmöglichkeiten entstehen.

2.2.4 Soziale Netze

Im Kontext des Internet versteht man unter einem sozialen Netz (auch: soziales Netzwerk) eine Plattform auf der Basis von webbasierter Anwendungssoftware, über die die Nutzer persönliche Daten austauschen und Beziehungen untereinander herstellen und vertiefen. Aufgrund der Bündelung sehr großer Zahlen von Nutzern spielt die Datenhaltung und die Datenübertragung durch Plattformen für soziale Netze eine große Rolle. Das derzeitige Wachstum der Nutzerzahlen ist für die Rechenzentren der Plattformen sehr relevant.

Boyd und Ellison (2007) definieren eine Plattform für soziale Netze (social network site) “[...] als web-basierten Dienst, der es Individuen erlaubt, ein öffentliches oder halböffentliches Profil in einem beschränkten System zu erstellen, eine Liste anderer Nutzer zu benennen, mit denen sie eine Verbindung teilen, und ihre Liste der Verbindungen und die, die andere in dem System erstellt haben, anzusehen und zu durchlaufen. Die Natur und Bezeichnung dieser Verbindungen kann sich von Angebot zu Angebot unterscheiden.“

Facebook ist die weltweit größte Community mit – nach eigenen Angaben – über 1,111 Milliarden Mitgliedern (Facebook 2013). LinkedIn ist eine Online-Plattform zur Pflege und zum Knüpfen von Geschäftskontakten und hat nach eigenen Angaben weltweit mehr als 100 Millionen Mitglieder in über 200 Ländern und Regionen, davon über eine Million Mitglieder in der DACH-Region (Deutschland, Österreich, Schweiz) (LinkedIn Blog 2013). Twitter hat weltweit 554,75 Millionen Mitglieder (Statistic Brain 2013). Google hat 2011 Google+ als Konkurrenzprodukt zu Facebook gestartet. Es verfügt laut Google über mehr als 500 Millionen Nutzer (Google Blog 2012).

Bei der Software für eine soziale Netzplattform sind mehrere Ebenen zu unterscheiden. Auf Anwendungsebene werden Dienste bereitgestellt, die direkt für die Nutzer Leistungen erbringen wie die Erstellung von Profilsseiten, Formularen oder Fotosammlungen, Chat-Server, Spiele usw. Auf

systemnaher Ebene wird Software zur Verwaltung der Daten eingesetzt. Bei den großen Anbietern werden verteilte Dateisysteme benötigt.¹⁷

Die Hardware muss mit möglichst kurzer Latenzzeit auf diese Daten zugreifen können. Die Hauptprobleme sind die Energieversorgung der Server, die Kühlung und die Energieversorgung der Kühlung.

Das Zusammenwachsen verschiedener Dienste in diese Plattformen wird weitergehen. In Facebook ist beispielsweise seit Juli 2011 auch Skype-Videotelefonie integriert (Klaß 2011). Videotelefonie ist in Google+ seit Beginn enthalten. Mittelfristig ist eine Konzentration auf die derzeit dominanten Plattformen zu erwarten. Zusätzlich werden sich kulturell bedingt Plattformen in Asien und im arabischen Raum entwickeln. Es werden kleinere interessensgebundene Plattformen entstehen. Möglicherweise kommt es auch zu einer Vernetzung zwischen den Netzen.

Als Spezialfall des allgemeinen Trends zum mobilen Internetzugang (Abschnitt 2.1.1) ist die zunehmend mobile Nutzung der sozialen Netze zu beobachten. Dieser Trend geht einher mit der Integration standortbezogener Dienste (engl. location-based services) in diese Plattformen. GPS-fähige Smartphones ermöglichen dann beispielsweise, dass automatisch Freunde lokalisiert und kontaktiert werden können, die sich gerade in der Nähe befinden. Aber auch Fotos, Videos und andere Daten können mittels Geotagging mit Ortsinformation assoziiert und darüber wieder von anderen Netzteilnehmern gefunden werden. Neben dem von Nutzern erzeugten Content werden Ortungsfunktionen auch für standortbasierte Werbung und Mikromarketing genutzt. (Hilty et al. 2012)

Ressourcenrelevanz:

- Die Übertragung, Speicherung und Verarbeitung der Massendaten bindet hohe Hardwarekapazitäten, insbesondere wenn Fotos oder Videos im Spiel sind.
- Die häufigen Aktualisierungen verursachen Datenverkehr im Internet; dies gilt in besonderem Maße bei der Integration standortbezogener Funktionalitäten.

¹⁷ Beispiel: Die von Facebook verwendete Software basiert auf CentOS, eigener HTTP Software, PHP, und dem eigenen DBMS Cassandra für NoSQL (Lakshman & Prashand 2010).

3 Ansatzpunkte für Ressourcenschonung

Dieses Kapitel stellt die wesentlichen Ansatzpunkte und Potenziale zur indirekten Schonung natürlicher Ressourcen vor, die im Softwarebereich erkennbar sind.

Entsprechend Abbildung 1 in Abschnitt 1.3 (siehe Seite 18) gehen wir davon aus, dass Anwendungs- und Systemsoftware über die Nutzung von Hardwarekapazitäten natürliche Ressourcen in Anspruch nehmen. Software kann diese Ressourcen schonen, indem sie pro Leistungseinheit weniger Hardwarekapazität in Anspruch nimmt, den Stromverbrauch der Hardware minimiert oder die Nutzungsdauer von Hardwareprodukten nicht unter ihre technische Lebensdauer verkürzt.

Ausgehend von den in Kapitel 2 besprochenen Trends und deren ressourcenrelevanten Aspekten lassen sich Anhaltspunkte zur Ressourcenschonung identifizieren. Diese beziehen sich auf Anwendungssoftware (Abschnitt 3.1) und auf Rechenzentren (Abschnitt 3.2). Ansatzpunkte außerhalb des Softwarebereichs werden im Sinne eines Ausblicks auf Aspekte, die über die hier untersuchte Thematik hinausreichen, kurz angesprochen (Abschnitt 3.3).

3.1 Ansatzpunkte im Bereich Anwendungssoftware

3.1.1 Wählbare Bildauflösung

Die Verarbeitung von Fotos und Videos in hoher Auflösung stellt generell hohe Anforderungen an die Hardware. Multimediale Kommunikationsdienste (wie Skype) und multimediale Unterhaltungsangebote (wie Internet-TV und Computerspiele) können heute nur deshalb in der gewohnten Qualität angeboten werden, weil entsprechende Bandbreiten und Rechenleistungen verfügbar sind. Die Ansprüche halten mit der technischen Entwicklung Schritt: Der Unterschied zwischen der ursprünglich einzig angebotenen Auflösung von Youtube-Videos und der heute maximalen Auflösung macht einen Faktor 164 im Datenvolumen pro Zeiteinheit Film aus.

Es ist deshalb wichtig, dass Softwareprodukte dem Nutzer zumindest die Freiheit lassen, niedrigere als die jeweils technisch möglichen Auflösungen zu verwenden (z. B. bei Video Calls und Spielen) und die Bildauflösung auf einfache Weise zu reduzieren (z. B. Fotos beim Einfügen in Präsentationen automatisch oder defaultmäßig herunterzuskalieren), wo eine hohe Auflösung nicht benötigt wird.

Ebenso relevant wie die Frage der Auflösung ist allerdings das Problem, dass durch die Zunahme des IP-Fernsehens und den damit verbundenen Trend von *Broadcast* zu *Unicast* (jeder Nutzer bestimmt den Zeitpunkt selbst) eine Zunahme des Datenverkehrs bewirkt wird, die erheblichen Material- und Energieverbrauch in der benötigten Infrastruktur auslöst. Die entstehende Redundanz wieder zu reduzieren, ist allerdings keine Frage von Software, sondern von Geschäftsmodellen und Netzmanagement und fällt deshalb nicht in den Themenbereich dieser Studie.

3.1.2 Mobiles Internet bevorzugt über WLAN

Die Dichte von öffentlich zugänglichen WLAN-Zugangspunkten ist in vielen Ballungszentren schon so hoch, dass WLAN als Internet-Zugangsnetz auch unterwegs eine zumindest temporäre Alternative zum Mobilfunknetz darstellt. Wir erwarten, dass Softwareprodukte, die auf mobilen Internetzugang angewiesen sind, ressourceneffizienter arbeiten, wenn dies über WLAN und nicht über ein

Mobilfunknetz geschieht. Die Mobilfunknetze haben nicht nur den höchsten Energieaufwand pro übertragener Datenmenge,¹⁸ sondern gelten auch als sehr materialintensiv (Scharnhorst et al. 2006; Emmenegger et al. 2006). Dies wird sich aus technischer Sicht auch durch den Aufbau der LTE-Netze nicht grundlegend ändern.

Software hat Einfluss auf die aktuelle Wahl des Kommunikationsweges und sollte im Interesse der Energieeffizienz entscheiden bzw. die Entscheidungsfreiheit des Nutzers wahren. Softwareprodukte sollten nicht den Nutzer zwingen, über Mobilfunk zu kommunizieren, auch wenn ein Internetzugang über WLAN verfügbar ist.

3.1.3 Das „Prinzip App“ ausweiten

Die Reduktion auf die wichtigsten Funktionen und die begrenzten Hardwareressourcen führen bei mobilen Apps zu einer hohen Ressourceneffizienz auf dem Endgerät.

Mobile Apps könnten die Welt der stationären Anwendungssoftware befruchten. Die Einfachheit der Installation, Deinstallation und die Reduktion auf die jeweils wesentlichen Aufgaben haben hohes Potenzial auch im stationären Bereich.

3.1.4 Webbasierte Anwendungen effizient implementieren

Webbasierte Anwendungssoftware hat hohes Ressourceneffizienzpotenzial. Es handelt sich um eine niederschwellige Form von Software-as-a-Service, da Webbrowser praktisch überall verfügbar sind. Diese Entwicklung bietet die Chance, dass Endgeräte mit geringen Speicher- und Rechenkapazitäten auf breiter Basis attraktiv werden. Rechenaufwendige Operationen können auf dem Webserver durchgeführt werden und belasten den Webclient nicht, und dies wiederum typischerweise als Cloud-Dienst.

Beispiele dafür sind Thin Clients oder das Chromebook (siehe Abschnitt 2.2.2). Die Verbreitung solcher Geräte kann zur Ressourceneinsparung beitragen, wenn sich Tätigkeiten von klassischen stationären PCs auf diese verlagern lassen. Ob tatsächlich eine Ressourceneinsparung eintritt, hängt von folgenden Bedingungen ab:

1. Der Internetzugang sollte über LAN oder WLAN erfolgen, also *nicht* über das Mobilfunknetz.
2. Die Server müssen eine hohe mittlere Auslastung erreichen, d. h. der Pooling-Effekt muss groß genug sein und das tatsächliche Lastmanagement muss davon Gebrauch machen.
3. Es sollte überwiegend Anwendungssoftware eingesetzt werden, die die von Williams und Tang (2013) genannten Voraussetzungen erfüllt (siehe auch Abschnitt 2.2.3).

Die Untersuchungen von Williams und Tang (2013) zum Vergleich von traditioneller und webbasierter Anwendungssoftware zeigen gemischte Ergebnisse. Konkret hat ein empirischer Vergleich zwischen Office 2010 und seiner „Cloud“-Variante Office 365 ergeben: Bei Outlook (–8 %) und Excel (–17 %) wurde insgesamt (in der Summe über Endgerät, Netzwerk und Rechenzentrum, jedoch ohne Herstellung und Entsorgung der Hardware) Energie eingespart, bei Word jedoch lag der Energieverbrauch der Cloud-Variante um 17 % höher (Williams & Tang 2013). Die insgesamt eher

¹⁸ Die grundsätzliche Überlegenheit von Netzen mit kleinen Funkzellen gegenüber Netzen mit größeren Funkzellen ist physikalisch dadurch begründet, dass die benötigte Sendeleistung mit der Entfernung zwischen Sender und Empfänger quadratisch zunimmt.

kleinen Unterschiede deuten darauf hin, dass der Nutzen der neuen Architektur eng begrenzt ist, wenn das Ziel darin besteht, die alten Konzepte zu reproduzieren.¹⁹

Dies zeigt, dass anstelle einer Imitation komplexer PC-Software durch Webanwendungen aus Ressourcensicht eher neue Nutzungsformen vorteilhaft sind, die sich an der Reduktion auf überschaubare, klar definierte Aufgaben orientieren. Entsprechend werden nur wenig Daten übertragen. Bei dieser Verbindung des „Prinzips App“ (siehe Abschnitt 2.2.1) mit webbasierten Architekturmustern zu „Web Apps“ werden die Vorteile der beiden Konzepte in Bezug auf Ressourceneffizienz verbunden:

- Zum einen werden lokal nur geringe Hardwarekapazitäten benötigt. Sofern die Datenübertragung und die Ausführung in Rechenzentren ressourceneffizient durchgeführt werden (jeweils über den gesamten Lebensweg betrachtet), wirkt sich das günstig auf die Inanspruchnahme natürlicher Ressourcen aus.
- Bei gut programmierten Web Apps in Verbindung mit effizienten Webservern ist die Datenübertragung beim Starten des Programms und auch während der Ausführung minimal, so dass das Risiko einer Überkompensation der Einsparungen durch den verursachten Datenverkehr sehr gering ist.

Durch die Kombination dieser beiden Vorteile zeichnen sich hier ressourceneffiziente Lösungen z.B. für Büroarbeitsplätze ab.

3.1.5 Bedarfsgerecht gesteuerte Software

Wir bezeichnen ein Softwareprodukt als bedarfsgerecht gesteuert, wenn es in der Lage ist, zu jedem Zeitpunkt nur die Hardwareressourcen (Rechenleistung, Speicherplatz, Bandbreite im Netzwerk) zu beanspruchen, die für die jeweilige Aufgabe benötigt werden. Davon sind die meisten Softwareprodukte weit entfernt.

Die Arbeitsgruppe „Software und Green IT“ der Green IT Allianz des BITKOM schätzt, dass der durchschnittliche Nutzer nur 7 % der Funktionalität von Standardsoftware ständig und 47 % aller Anwendungsmerkmale niemals nutzt. Die erreichbaren Potenziale zur Energieeinsparung werden auf 10 bis 20 % geschätzt (BITKOM 2010b).

Bedarfsgerechte Steuerung wäre außerdem ein Hebel, um der Hardware-Obsoleszenz vorzubeugen. Weniger leistungsfähige, weil ältere Hardware könnte länger genutzt werden, wenn von neuen Softwareversionen nur die wirklich benötigten Module installiert und betrieben werden müssten.

Voraussetzung wäre eine konsequent modulare Softwarearchitektur (und damit eine Rückbesinnung auf klassische Prinzipien der Informatik). Die Auswahl der Module kann bei der Installation und Konfiguration oder auch im laufenden Betrieb erfolgen (bei webbasierter Software entfällt die Installation ohnehin). Obwohl dies heute in vielen Fällen möglich ist, ist es für den Endbenutzer in der Regel zu schwer zu durchschauen, welche der ihm angebotenen Optionen zu welchen

¹⁹ Dies steht in einem latenten Widerspruch zur Studie von Masanet et al. (2013), wo erheblich höhere Energiesparpotenziale durch Cloud Computing gerade auch für „productivity software“, also alltägliche Bürosoftware, geschätzt wurden. Die beiden Studien sind jedoch nicht direkt vergleichbar, da Williams und Tang einen speziellen Fall isoliert untersucht haben, während Masanet et al. die technisch möglichen Konsolidierungseffekte durch eine flächendeckende Einführung abgeschätzt und außerdem graue Energie mitberücksichtigt haben.

Konsequenzen führen und ob er seine Entscheidungen später revidieren kann, so dass er im Zweifel stets das Maximum wählt.

Ein zentraler Aspekt der bedarfsgerechten Steuerung ist es deshalb, wie leicht das Produkt es dem Nutzer macht, ressourcenschonende Einstellungen vorzunehmen. Von einem "grünen" Softwareprodukt sollte man erwarten, dass ressourcenrelevante Einstellungen an zentraler Stelle vorgenommen werden können (Naumann 2013). Dies kann im einfachsten Fall durch Setzen eines Hakens ("ressourcenschonender Modus") geschehen. Bei einem Trade-Off zwischen verschiedenen Ressourcen und Gebrauchseigenschaften kann die Kontrolle in die Hand des Nutzers gelegt werden, indem er seine Präferenzen durch Einstellen von Schiebereglern ausdrückt (ob er z.B. lieber auf Geschwindigkeit oder auf Bildauflösung verzichtet).

Es ist auch vorstellbar, dass Anwendungssoftware in Zukunft eine Balance zwischen verschiedenen Ressourcen dynamisch selbst einstellt (Naumann 2013). Wenn elektrischer Strom aus erneuerbaren Quellen gerade im Überfluss vorhanden ist, werden rechenintensive Aufgaben durchgeführt. Indexierung, Kompression, Deduplizierung sind Beispiele für rechenintensive Aufgaben, die teilweise zeitlich verschiebbar sind. Leistet man sich zu Zeiten günstiger Energie solchen Aufwand, schont dies zu einem späteren Zeitpunkt Netz- oder Speicherressourcen und spart damit dann die wertvollere Energie. Durch dieses kontextsensitive Verhalten können auch IKT-Endgeräte am Demand Shaping im Smart Grid mitwirken (siehe auch Abschnitt 3.2.1).

Insgesamt hat bedarfsgerechte Steuerung also einen statischen Aspekt (modulare Installation, Konfiguration) und einen dynamischen Aspekt (Ressourcenmanagement zur Laufzeit, Kontextsensitivität). Diese Aspekte verschmelzen jedoch durch den Trend zu Software-as-a-Service, z. B. webbasierten Anwendungen (Abschnitt 2.2.2). Dadurch eröffnen sich Entwicklungsperspektiven mit hohem Potenzial zur Ressourcenschonung.

3.1.6 Die Rolle von Open Source Software

Open Source Software (OSS)²⁰ hat in den letzten Jahren an Bedeutung gewonnen. Ernst&Young (2011) nennen die Zeit für OSS ab 2004 „Mainstream Epoche“. In einer Trendstudie (Diedrich 2009) wird der Einsatzgrad als abhängig von der Branche festgestellt: „Besonders häufig wird OSS eingesetzt als Server-Betriebssystem (84 Prozent), auch weitere klassische Open-Source-Einsatzbereiche wie Webserver (81 Prozent), Datenbank (79 Prozent) und Netzwerk-Infrastruktur (73 Prozent) kommen auf einen hohen Einsatzgrad.“ Auch in öffentlichen Einrichtungen wird verstärkt auf OSS gesetzt. Wikipedia (2013b) listet z.B. 18 Migrationsprojekte der letzten Jahre in Deutschland auf, z.B. in der Stadtverwaltungen München und Leipzig und der Agentur für Arbeit. Consulting-Unternehmen wie Accenture (o. J.) bieten inzwischen umfangreiche Dienstleistungen für den Open Source Bereich an. Das offene Betriebssystem Android ist derzeit auf Tablet-Computern und Smartphones am stärksten verbreitet (Gartner 2013).

²⁰ "Open Source [...] und quelloffen ist eine Palette von Lizenzen für Software, deren Quelltext öffentlich zugänglich ist und durch die Lizenz Weiterentwicklungen fördert. Open-Source-Software (kurz OSS) steht unter einer von der Open Source Initiative (OSI) anerkannten Lizenz. Diese Organisation stützt sich bei ihrer Bewertung auf die Kriterien der Open Source Definition, die weit über die Verfügbarkeit des Quelltexts hinausgeht. Sie ist fast deckungsgleich mit der Definition freier Software." (Wikipedia 2012a). "Freie Software [...] ist Software, die für jeden Zweck ausgeführt, untersucht, modifiziert und in ursprünglicher oder modifizierter Form weiterverbreitet werden darf. Das schließt auch die kommerzielle Nutzung ein. Freie-

Im Desktop-Bereich ist Linux noch nicht wesentlich vertreten (Linux erreichte im Dezember 2011 einen Marktanteil von 1,41 %; Heise Open Source 2012a), dafür aber im Serverbereich und auf Großrechnern.

Daher bietet sich eine gesonderte Betrachtung von OSS unter dem Aspekt der Ressourceneffizienz an. Softwaretechnisch gesehen ist OSS Software wie jede andere und kann theoretisch gleich ressourceneffizient oder -ineffizient erstellt werden. Aufgrund der ihrer Offenheit hat sie jedoch Vorteile, die im Interesse der Ressourceneffizienz genutzt werden können.

Als Vorteile werden die Anpassbarkeit, die Wiederverwendbarkeit von Code, die höhere Produktqualität, höhere Sicherheit, offene Standards und nicht-notwendige Lizenzkosten genannt (z.B. Renner et al. 2005). Demgegenüber werden die Nachteile fehlender Gewährleistungsrechte, geringer Support durch Entwickler, ungewisse Weiterentwicklung der Software, fehlende Anwendungen oder mangelnde Interoperabilität mit kommerzieller Software aufgeführt. Diese Nachteile werden jedoch immer mehr abgebaut.

Trotz des offenen Modells (oder gerade deshalb) wird der OSS im Vergleich zu proprietärer Software gleichwertige Qualität („jedoch nicht als grundsätzlich besser“, Heinrich et al. 2006) oder höhere Qualität bescheinigt (Renner et al. 2005, Heise Online, 2012).

Bei Server-Betriebssystemen, Webservern und Datenbanken gilt OSS als effizienter und kann mit gleicher Hardware mehr leisten (Heinrich et al. 2006, unter Bezug auf Creber 2004 und Enterprise Management Associates, 2006) und ist daher in diesen Bereichen auch stärker verbreitet (Diedrich 2009, Heise Open Source 2012a).

Heinrich et al. (2006) führen aus, dass „beim Einsatz von Open-Source-Software vielfach die Anforderungen an die verwendete Hardware geringer sind, weshalb die bereits vorhandenen Computersysteme über die steuerliche Nutzungsdauer hinaus eingesetzt und somit Neu-Investitionen in Hardware als Kostenfaktor hinausgezögert werden können“. Die Autoren beziehen sich dabei auf Bräuner (2005), Bokhari & Rehman (1999) und Enterprise Management Associates (2006).

Durch die freie Verfügbarkeit der Software und der Unabhängigkeit von einem Hersteller kann eine längere Nutzungsdauer der Hardware ebenfalls erreicht werden, weil die Nutzer sich dem Trend zu immer höherer Speicherplatz- und Rechenkapazität nicht im gleichen Ausmaß anschließen (müssen) wie die Nutzer proprietärer Software. Andererseits zwingen Sicherheitslücken und bekanntgewordene Fehler zumindest im Betriebssystem zu ständigen Aktualisierungen, was auf Dauer zu Inkonsistenzen in den Schnittstellen und unhandlichen Konfigurationen insbesondere bei komplexen Systemen wie einer Linux-Distribution führt.²¹

OSS profitiert von schnelleren Entwicklungsprozessen (Heise Open Source, 2008) und ermöglicht die rasche Verbreitung von Software (Ernst&Young, 2011). Maßnahmen zur Erhöhung von Ressourceneffizienz können daher sehr schnell skalieren.²² Paulson et al. (2004) verglichen die Quell-

Software-Lizenzen können eine Copyleft-Klausel enthalten, die besagt, dass bearbeitete und wieder-veröffentlichte Versionen der Software ebenfalls frei sein müssen." (Wikipedia 2012b).

²¹ Auch bei proprietärer Software kann Hardware teilweise lange genutzt werden. Beispielsweise wurde Windows XP 2001 eingeführt. Bis 2014 unterstützt Microsoft die Nutzer noch mit Sicherheitsaktualisierungen.

²² Etwaige verspätete Auslieferungen neuer Software unter Linux sind oft auch durch Hersteller von Hardware verursacht. Beispielsweise ließ sich Sandy-Bridge-Hersteller Intel bei der Fehlersuche in den Linux-Treibern

textentwicklung verschiedener Projekte OSS und Nicht-OSS und zeigten, dass Probleme bei OSS Projekten mit höherer Wahrscheinlichkeit behoben werden.

Der Möglichkeit, jederzeit an dem System Änderungen für eine erhöhte Energieeffizienz vorzunehmen, steht die noch geringe Anzahl von daran interessierten Spezialisten gegenüber, die in der Lage sind, diese Verbesserungen fachgemäß einzubringen. Dies gilt insbesondere für Änderungen an Quelltexten.²³ Dieses Problem könnte durch Kampagnen zur Bewusstseinsbildung für Ressourceneffizienz in der Entwicklergemeinde behoben werden. Bei Verbesserungen, die auf der Ausnutzung von Hardwareeigenschaften beruhen, fehlt zudem oft das Wissen über die Interna der Hardware.²⁴ Hier wäre nach Anreizsystemen zu suchen, die bewirken, dass Hardwarehersteller die Schnittstellen offenlegen, damit die Community mit diesem Wissen die OSS verbessern kann. (Es gibt OpenSource-Hardware-Projekte. Auch diese sind aus Ressourcensicht wichtig und förderungswürdig, um der OSS mehr Möglichkeiten zu ressourceneffizienten Verbesserungen zu geben).

Während Einstellungen für eine gute Energieeffizienz bei Herstellern proprietärer Software oft bei Auslieferung bzw. via Updates bereitgestellt werden, erfordern energiesparende Einstellungen unter Linux in der Regel manuelle Nacharbeit (Thoma, 2012b). Dieses Problem könnte durch Kampagnen zur Bewusstseinsbildung für Ressourceneffizienz in der Entwicklergemeinde behoben werden.

Ein potenzieller Nachteil in der Open-Source-Welt kann darin bestehen, dass man in vielen Bereichen oft noch auf freiwilliges Engagement angewiesen ist und die Umsetzung von Richtlinien nicht erzwungen werden kann. Mehr und mehr werden allerdings Entwickler für ihre Arbeit an OSS angestellt (Heise Open Source 2012b).

Auch für freie Software und OSS gelten verschiedene Lizenzen, die zu rechtlichen und finanziellen Risiken für Firmen führen können (BITKOM o. J.). Dies hat Konsequenzen für Entscheidungen, wie Unterstützungsmaßnahmen für OSS strategisch gestaltet werden.

Für viele Werkzeuge von Linuxdistributionen, den Linuxkernel selber und eine Vielzahl von Projekten auf dem OpenSource-Hoster SourceForge (SourceForge o. J.) gilt die GPL, die bekannteste Lizenz für freie Software mit einem Copy-Left Prinzip: In GPL-Software eingebrachte Verbesserungen hinsichtlich Ressourceneffizienz dürften danach nicht in proprietäre Software übernommen werden, ohne dass diese dann ebenfalls unter GPL veröffentlicht werden müsste. Einerseits müsste man Maßnahmen zur Erhöhung der Ressourceneffizienz in freie OSS einbringen, um die schnelle Verbreitung von Verbesserungen auch in proprietärer Software zu ermöglichen. Andererseits nehmen immer mehr Linuxdistributionen zunehmend auch proprietäre Software auf und untergraben das Prinzip der freien Software. Die Unterstützung GPL-lizenzierter Software würde zur Verbreitung des

Zeit. Die darauf basierenden Patches wurden erst verspätet in offizielle Distributionen aufgenommen: erst in den Kernel 3.4, und im Falle von Ubuntu zwar schon in den alten Kernel eingebracht, aber auch erst mit der neuen Version Ubuntu 12.04 verteilt (Thoma, 2012a).

²³ Für die Messung des Energiebedarfes von einzelnen Programmen steht unter Linux beispielsweise ein recht gutes Analysewerkzeug Powertop (o. J.) zur Verfügung. Eine damit untersuchte experimentelle CPU-lastabhängige Änderung der Framerate bei der Wiedergabe von Videos und damit Einsparung von Energie durch mplayer während einer Vorlesung an der ETH Zürich hat allerdings bisher immer noch nicht den Weg in die offiziellen Quellen gefunden.

²⁴ Grafikkarten-Hersteller beispielsweise stellen meist nur Gerätetreiber für Windows (in nichtlesbarer Form) bereit und weigern sich, notwendige Informationen über die Hardware an Entwickler von Open-Source-Treibern zu geben, um der Konkurrenz über die dann offenen Quellen keine Einblicke in ihre Technologie zu erlauben. Im Gegensatz dazu arbeiten Hardwarehersteller enger mit Microsoft zusammen und geben Einblick in proprietäre Treiber.

Wissens führen und dafür sorgen, dass sämtliche Verbesserungen beim Endnutzer ankommen²⁵. Gleichzeitig würde freie (und Open Source) Software weiter gestärkt werden. Rechtliche Aufklärungskampagnen für Firmen könnten die Hürden zum Einstieg in OSS absenken.

Das Wissen, wie energieeffiziente Software gestaltet und konfiguriert werden muss, ist heute sowohl für OSS als auch für prioritäre Software unzureichend, und es besteht ein genereller Mangel an Fachkräften, die dieses Wissen praktisch anwenden. Es ist deshalb wichtig, dass das vorhandene Wissen z.B. in Workshopserien wie „Energy Aware Software-Engineering and Development“ (EASED), „Green and Sustainable Software“ (GREENS), „Software Engineering Aspects of Green Computing“ (SEGC) sowie Konferenzen (wie ICT4S, ICT for Sustainability) und Zeitschriften verfügbar gemacht wird. Das Problem ist in der Forschung erkannt und wird angegangen. Verfügbares Wissen kann gerade in OSS schnell eingebracht und verbreitet werden. Entsprechende Forschungsinitiativen, die Wissen zu Ressourcen- und Energieeffizienz von Software erarbeiten, sammeln oder verbreiten, tragen mittelfristig zur Ressourcenschonung durch Software bei.

3.2 Ansatzpunkte in Rechenzentren

Rechenzentren gehören zu den größten Stromverbrauchern in Deutschland. Große Rechenzentren haben eine Anschlussleistung von mehreren Megawatt und liegen damit in der gleichen Größenordnung wie energieintensive Industrieunternehmen, die von der EEG-Umlage ausgenommen sind. Der Anteil aller Server und Rechenzentren am Stromverbrauch in Deutschland beträgt 1,8 % (Hintemann & Fichter 2013). Stobbe et al. (2009) gehen davon aus, dass der Stromverbrauch der Server und Rechenzentren zwischen 2010 und 2020 im Basisszenario um knapp 30 % ansteigen wird. Selbst in einem „Green IT“ Szenario rechnen sie mit einem Anstieg des Stromverbrauchs um mehr als 10 % bis 2020.

Während die Möglichkeiten der Ressourceneinsparung in Rechenzentren durch effizientere Hardware-Lösungen seit Jahren diskutiert und teilweise auch eingesetzt werden, ist das Potenzial zur Ressourcenschonung durch Software in Rechenzentren – mit Ausnahme der Servervirtualisierung – bislang noch wenig untersucht. Wir diskutieren im Folgenden drei Ansätze, die ein relativ hohes Ressourceneffizienzpotenzial versprechen.

3.2.1 Dynamisch prädiktives Lastmanagement

Mit einem dynamisch prädiktiven Lastmanagement werden die IT-Hardware und die Infrastruktur eines Rechenzentrums so gemanagt, dass sie möglichst optimal ausgelastet werden. Zeitlich verschiebbare Aufgaben werden in Phasen niedriger Auslastung der Hardware verschoben bzw. Hardware wird abgeschaltet, wenn diese nicht benötigt wird. Ein übergreifendes dynamisches Lastmanagement ist dabei nicht auf IT-Dienste und die IT-Hardware beschränkt, sondern umfasst auch die Infrastrukturkomponenten des Rechenzentrums. Beispielsweise können Teile einer modularen unterbrechungsfreien Stromversorgung (USV) abgeschaltet oder Bereiche des Rechenzentrums, in denen die Hardware abgeschaltet wurde, auch von der Klimatisierung ausgenommen werden (Nebel et al. 2009).

²⁵ In Kombination von Open Source und proprietärer Software ist dies nicht garantiert

Bislang wurden die verschiedenen Ebenen im Rechenzentrum in der Regel sowohl von unterschiedlichen Personen bzw. Abteilungen im Rechenzentrum als auch mit unterschiedlichen Tools gemanagt – auf der einen Seite die System- und Netzwerkmanager und auf der anderen Seite die Facility-Manager. Unter dem Begriff Data Center Infrastructure Management (DCIM) wird aktuell eine Vielzahl von Lösungen angeboten, die anstreben, die beiden „Welten“ zu vereinen und damit die Energieeffizienz im Rechenzentrum zu erhöhen. Der Begriff DCIM wird aber von Herstellern und von Marktforschungs- bzw. Beratungsunternehmen unterschiedlich verwendet. Die Lösungen ermöglichen in der Regel zumindest ein übergreifendes Monitoring und Management der IT-Hardware und der Rechenzentrumsinfrastruktur. Ein vollständiges integratives Lastmanagement inklusive der Ebenen der Hardwarevirtualisierung und der Dienste wird aber oft noch nicht erreicht (Reder 2012).

Bei Spitzenlasten oder ungünstigen Rahmenbedingungen, wie z.B. sehr hohen Außentemperaturen, ist auch eine Verschiebung von IT-Lasten in andere Rechenzentren möglich.

Erfahrungen zeigen, dass oft *schon mit den im Unternehmen prinzipiell vorhandenen Software-Systemen* (z.B. Server Management Suites der Hersteller; Baumeister 2012) zum Energiemanagement Einsparungen in Größenordnungen von ca. 30 % realisierbar wären, diese aber meist aufgrund des erforderlichen Aufwandes zur Umsetzung nicht realisiert werden (Müller 2013). Die Anbieter von DCIM-Lösungen stellen ebenfalls Einsparungen in der Größenordnung von 25 % in Aussicht (Lanline 2012).

Diese Einschätzungen werden auch durch Forschungsprojekte wie das Projekt GAMES (Green Active Management of Energy in IT Service centres - www.green-datacenters.eu) bestätigt, in dem Einsparungen des CO₂-Footprints von ca. 25 % ermittelt wurden. Ähnliche Größenordnungen sollen nach Auskunft der Projektverantwortlichen auch im Projekt „Cool em All“ (www.coolmall.eu) und im Projekt „Adaptive Computing for Green Data Centers“ (www.ac4dc.de) erreicht werden.

Ein rechenzentrumsübergreifendes Lastmanagement kann für den Rechenzentrumsbetreiber neben der Verbesserung der Auslastung seiner Systeme auch Kostenvorteile durch die Nutzung regional unterschiedlicher dynamischer Strompreise haben. Eine Untersuchung des Offis-Instituts hat hier ein zusätzliches (finanzielles) Einsparpotenzial von ca. 5-10 % je nach Strompreisdynamik berechnet (Nebel et al. 2009). Aus Sicht des gesamten Energiesystems bedeutet die Anwendung dieser Lösung eine bessere Anpassbarkeit an eine dynamische Stromerzeugung durch regenerative Energien (demand shaping). Die Anpassungsfähigkeit an Preissignale ist aus Ressourcensicht ein wichtiger Vorteil des dynamisch-prädiktiven Lastmanagements, weil sich bei entsprechender Preisgestaltung fossile Energieträger einsparen und Stromnetzkapazitäten begrenzen lassen. Dieser Vorteil ergibt sich selbst unter der pessimistischen Annahme, dass die gleiche IKT-Leistung mit der gleichen elektrischen Leistung erbracht wird.

Auf Basis der genannten Untersuchungen und Quellen lässt sich feststellen, dass Softwarelösungen zum dynamisch-prädiktiven Lastmanagement in Rechenzentren Energieeinsparpotenziale von 25 % bis 30 % versprechen. Durch die Verbesserung der durchschnittlichen Auslastung wird auch deutlich weniger Hardware benötigt, so dass auch hohe Potenziale zur Verbesserung der Materialeffizienz bestehen.

3.2.2 Informations- und Datenmanagement

Nach Berechnungen des Borderstep-Instituts im Rahmen des Forschungsprojektes AC4DC ist die Datenspeicherung aktuell für mehr als 10 % des Energieverbrauchs der Rechenzentren verantwortlich, bei steigender Tendenz. Dies ist unter anderem dadurch begründet, dass der Umgang mit

Daten in Rechenzentren oft durch die Tatsache geprägt ist, dass sich die Speicherkosten pro Gigabyte ständig verringern. Die Kapazitäten von Festplatten haben sich in der Vergangenheit ähnlich dem Mooreschen Gesetz ca. alle 18 Monate verdoppelt.²⁶ Nach Angaben der Analysten von Experton (zit. nach Bayer 2009) reduzieren sich die Kosten für die Speicherhardware jährlich um 30 %. Die Folge ist, dass für Unternehmen kaum Anreize bestehen, effizient mit Speicherplatz umzugehen (Bayer 2009).

Ein Ansatzpunkt zum effizienteren Datenmanagement liegt darin, die zu speichernde Datenmenge zu verringern. Beispielsweise können unnötige Kopien von Daten vermieden und nicht mehr benötigte Daten gelöscht werden. Dies kann zum Beispiel dadurch geschehen, dass Beschäftigte die eingehenden E-Mails, die dauerhaft gespeichert werden sollen, explizit markieren müssen. Alle anderen E-Mails werden nach Ablauf einer gewissen Frist automatisch gelöscht (Rüdiger 2011).

Außerdem kann der Stromverbrauch verringert werden, indem die Daten jeweils auf dem energieeffizientesten Medium gespeichert werden. Archivdaten können z. B. automatisch auf energieeffiziente Bandsysteme (Tapes) ausgelagert werden. Solche Softwaretools des Hierarchischen Speichermanagements (HSM) sind schon länger am Markt verfügbar. Sie verschieben Daten, auf die nur selten zugegriffen wird, auf kostengünstige Speichersysteme, die zumeist auch die ressourceneffizientesten sind. Da HSM automatisch und ohne erforderliche Eingriffe des Nutzers erfolgt, ist die Anwendung im Rechenzentrum durchaus üblich (Müller 2013). Mit der zunehmenden Einführung von Solid State Disks in Rechenzentren ist die Korrelation zwischen kostengünstigen und energieeffizienten Speichersystemen so aber nicht mehr gegeben. Solid State Disk zählen aktuell zu den kostenintensiven, aber energieeffizienten Speichersystemen (Wilde 2013).

Mit Information Lifecycle Management Lösungen (Ehmann & Hintemann 2004) und Enterprise Content Management Lösungen stehen über HSM hinausgehende Softwaretools und Konzepte zum Informations- und Datenmanagement zur Verfügung. Sie werden allerdings in der Praxis als Technologie zur Ressourcenschonung nur wenig eingesetzt, da sie einen hohen Aufwand für die Kategorisierung der Informationen und Daten erfordern (Müller 2013).

Aufgrund der ständig steigenden Speicherkapazitäten und sinkenden Speicherpreise der Festplatten besteht wenig Anreiz, Datenmengen zu reduzieren. Der Vorteil des günstigen Speicherplatzes hat für das Unternehmen allerdings auch den Nachteil, dass das Identifizieren und Finden der relevanten Daten immer aufwendiger wird (Vilsbeck 2012). Softwarelösungen, die die Datenflut *vermeiden* helfen, würden mehr zur Ressourceneffizienz beitragen als „Big Data“ Lösungen, die die Auswertung sehr großer Mengen an unstrukturierten Daten erleichtern.

3.2.3 Datenkomprimierung und Dateneduplizierung

Eine weitere Möglichkeit zur Verringerung der zu speichernden Datenmenge sind Datenkomprimierung und -deduplizierung. Mit Datenkomprimierung sind Verfahren gemeint, mit deren Hilfe digitale Daten so verändert werden, dass der benötigte Speicherplatz geringer wird bzw. die Übertragungszeit der Daten zwischen zwei IT-Systemen verkürzt wird. Die Komprimierung von Daten wird auch als Kodierung, die spätere Dekomprimierung als Dekodierung bezeichnet.

Eine verlustfreie Komprimierung liegt vor, wenn die Daten nach Kodierung und Dekodierung exakt dem Original entsprechen. Bei einer verlustbehafteten Komprimierung lassen sich die Daten in der

²⁶ 1983 waren 10-MB-Festplatten von IBM üblich, heute (2013) gibt es die ersten 8-TB-Festplatten, was ungefähr einer 20maligen Verdoppelung entspricht.

Regel nicht fehlerfrei wiederherstellen. Dafür werden höhere Kompressionsraten erreicht. Während die verlustfreie Komprimierung vor allem bei Programmdateien, Unternehmensdatenbanken, etc. erforderlich ist, wird die verlustbehaftete Komprimierung insbesondere bei Multimediadateien wie Bildern, Videos oder Audiodateien eingesetzt.

Das Verfahren der Deduplizierung von Daten ist ein spezielles Verfahren der Komprimierung. Dabei werden die vorhandenen Daten auf Redundanz analysiert. Redundante (mehrfach vorhandene) Daten werden identifiziert und beseitigt (Geer 2008; Pelkmann 2010). Die Effizienz der Deduplizierung hängt von den konkret vorhandenen Daten ab. Liegen viele ähnliche Daten vor – wie z. B. bei vielen geringfügig variierten Versionen einer Powerpoint-Präsentation – so sind sehr hohe Komprimierungsraten möglich. Der Hersteller EMC gibt den Faktor, um den die Datenmenge dedupliziert werden kann, mit 10 bis 30 an (EMC 2013).

Andere Hersteller (Unterseher 2008) und Berater (Wilde 2013) gehen von in der Praxis üblichen Komprimierungsraten bei Backup und Archivierung von 1:10 aus. Diese Größenordnungen werden durch Messungen an realen Systemen bestätigt (Meyer & Bolosky 2012). Erste Anwendungen in einem im Rahmen des Umweltinnovationsprogramms vom BMU geförderten Projekt bei der Erecon AG zeigen, dass die Dateneduplizierung nicht nur für Backup-Systeme, sondern auch zum direkten Einsatz „on the fly“ im Rechenzentrum geeignet sein kann (BMU 2012). Hier sind dann allerdings nur geringere Deduplizierungsraten (z.B. 1:5) möglich (Wilde 2013).

Deduplizierung verringert sowohl den Bedarf an Energie als auch an Hardware deutlich (He et al. 2010). Nach Berechnungen des Borderstep Instituts im Rahmen des Projektes AC4DC (o.J.) waren in deutschen Rechenzentren im Jahr 2012 ca. 12,5 Mio. Festplatten installiert. Eine Überschlagsrechnung zeigt die Größenordnung von Einsparungspotenzialen: Nimmt man an, dass das Verfahren aufgrund technischer und organisatorischer Einschränkungen nur auf 20 % aller Systeme angewendet würde, so wäre über Inline-Deduplizierung bei einer einer Deduplizierungsrate von 1:5 eine Einsparung von ca. 2 Millionen Festplatten möglich. Der Energieverbrauch der Rechenzentren ließe sich so um insgesamt ca. 2 % verringern.²⁷ Außerdem wird auch weniger Hardware in der Infrastruktur (Klimatisierung etc.) benötigt.

Ob der Einsatz von Deduplizierungstechniken künftig überhaupt in größerem Maße erfolgen wird, ist fraglich. So wird in einem Artikel des Fachmagazins speicherguide.de der Einsatz von kostengünstigen Festplattensystemen mit SATA-Festplatten als wirtschaftlich häufig günstiger angesehen als der Kauf eines Deduplizierungssystems (Rieß 2012).

3.2.4 Herausforderung heterogener Rechenzentrumsmarkt

Ressourceneffiziente Softwarelösungen für Rechenzentren müssen sich auf einem sehr heterogenen Markt durchsetzen. Dies betrifft allein schon die Struktur der Rechenzentren selbst. So verfügen von den ca. 52.000 Rechenzentren in Deutschland die meisten über weniger als 10 physikalische Server. Auf der anderen Seite des Spektrums stehen ca. 60 Rechenzentren, die über mehr 5.000 Server verfügen. In diesen 60 Rechenzentren sind allerdings insgesamt mehr als doppelt so viele Server installiert wie in den 31.500 kleinen Lokationen (Hintemann & Fichter 2012, 2013).

²⁷ Nach Berechnungen des Borderstep Instituts im Rahmen des noch laufenden Projektes AC4DC (o.J.) waren in deutschen Rechenzentren im Jahr 2012 ca. 12,5 Mio. Festplatten installiert, deren Betrieb für knapp 12 % des Strombedarfs der Rechenzentren verantwortlich war. Eine Reduktion um 2 Millionen Festplatten entspricht daher einer Reduktion des Strombedarfs in der Größenordnung von 2 %.

Nach einer Erhebung des Borderstep Instituts²⁸ sind fast 50 % der großen Rechenzentren (mehr als 5000 Server) sogenannte Colocation-Rechenzentren, die IT-Flächen inklusive Infrastrukturleistungen als Dienstleistung zur Verfügung stellen. Da in diesen Fällen der Betrieb von IT-Hardware und Infrastruktur nicht in einer Hand sind, verringern sich die Potenziale für übergreifende effiziente Softwarelösungen erheblich. Rund ein Viertel der großen Rechenzentren sind ferner Hosting-Rechenzentren, in denen zwar auch die IT-Hardware vom Betreiber gemanagt wird, aber die Software in Hand der Kunden ist. Auch hier sind die Einsatzmöglichkeiten von ressourceneffizienten Software-Lösungen stark eingeschränkt.

Nicht nur die Rechenzentren selbst sind sehr heterogen strukturiert. Im Rechenzentrumsmarkt agiert auch eine Vielzahl von unterschiedlichen Akteuren. Neben den Anbietern von IT-Hard- und Software sind hier insbesondere die Anbieter der Infrastrukturlösungen, IT-Berater, Rechenzentrumplaner und Systemhäuser zu nennen. Die Akteure verfolgen teilweise Ziele, die einer Durchsetzung von ressourceneffizienten Software-Lösungen am Markt entgegenstehen. So sind z.B. Hardware-Anbieter primär am Verkauf der Hardware interessiert. Teilweise wird auch – zusammen mit entsprechenden Softwarelösungen – versucht, proprietäre Systeme am Markt zu etablieren, wodurch übergreifende Lösungen erschwert werden.

Der Endkunde hat aufgrund der Intransparenz dieses Marktes derzeit nur eng begrenzte Möglichkeiten, durch seine Entscheidungen zu einer ressourceneffizienten Leistungserstellung beizutragen.

3.3 Weitere Ansatzpunkte für Ressourceneffizienz

3.3.1 Nutzerverhalten in sozialen Netzen

Das ursprüngliche Leistungsportfolio beispielsweise von Facebook führte zu einem vergleichsweise geringen Rechenaufwand. Die Implementierung von Videotelefonie, Gesichtserkennung, Spielen und Ortungsfunktionen mit Umgebungsinformationen und Werbung hat den Rechenaufwand aber signifikant erhöht.

Bei den großen Plattformen für soziale Netze ist heute davon auszugehen, dass das *auf Softwareebene* vorhandene Ressourceneffizienzpotenzial weitgehend ausgeschöpft ist, einfach weil ohne innovative Maßnahmen die Verarbeitung der Daten für Hunderte Millionen Nutzer gar nicht bewältigt werden könnte. Da die Betreiber mit ihren Rechenzentren früher oder später an absolute Grenzen stoßen (beispielsweise bezüglich elektrischer Anschlussleistung), liegt es in ihrem ureigenen Interesse, keine Energie und keine Serverkapazität zu verschwenden. Investitionen selbst in kleine ressourcensparende Verbesserungen an der Software lohnen sich in der Regel aufgrund des Multiplikationseffekts: Der jeweilige Code wird für viele Millionen Nutzer jeweils tausendfach ausgeführt.

Ressourcenschonung bei der Teilnahme an sozialen Netzen ist deshalb keine Softwarefrage, sondern kann nur über folgende Wege erreicht werden:

1. Datenhaltungspolitik des Betreibers

²⁸ Dem Borderstep-Institut liegen die Daten zu Größe, Standort, Betreiber und Rechenzentrumszweck von ca. 80 % der großen Rechenzentren in Deutschland vor.

2. Ressourcenschonender Betrieb von Rechenzentren
3. Verhalten der Nutzerinnen und Nutzer

Punkt 1 betrifft Fragen wie: Ist es vorgesehen, dass Daten auch physisch gelöscht werden, wird dem Nutzer ein „Recht auf Vergessen“ zugestanden?²⁹ Punkt 2 betrifft in einer bereits optimierten Softwarelandschaft primär die Frage der Kühlung und der Beschaffung der elektrischen Energie (Ökostrom). Punkt 3 wird im Folgenden kurz besprochen, da es hier Wechselwirkungen mit der Softwaregestaltung gibt.

Die Software ist in der Regel so gestaltet, dass sie die Nutzer zu häufigen und lang andauernden Interaktionen mit der Plattform motiviert. Denn je intensiver Mitglieder einer Plattform interagieren, desto attraktiver ist sie für werbetreibende Unternehmen. Als Konsequenz werden Mittel zur Verlängerung der Verweildauer eingesetzt (u. a. Spiele). Dies führt zu einer Erhöhung des Datenverkehrs, was in Kombination mit dem energieintensiven mobilen Zugang durchaus ins Gewicht fällt.

Besonders ungünstig erscheint aus Ressourcensicht die Kombination sozialer Netze mit standortbezogenen Diensten, weil diese besonders den Zugang über Mobilfunknetze fördert. Dass gerade der Markt der mobilen sozialen Netze (wie z. B. Foursquare³⁰) als zukünftig relevant eingeschätzt wird, belegt das Engagement von großen Unternehmen wie Nokia, Intel oder Samsung. Alle diese Unternehmen arbeiten beispielsweise mit dem Anbieter der mobilen sozialen-Netz-Software GyPsii zusammen (Heise Online 2011b).

Die „Always On“-Mentalität erreicht mit den mobilen sozialen Netzen einen Höhepunkt. Im Interesse der Ressourcenschonung müssten Plattformen gefördert werden, die auf einen Gegentrend setzen, der sich etwa so charakterisieren lässt: Minimierung der Interaktion, Angebot nur der notwendigsten Funktionen, Verkleinerung von Bildern auf die jeweils notwendige Auflösung als Default, Angebot der physischen Löschung von Daten auf einfache Weise. Ein solcher Trend wäre nicht nur aus Gründen der Ressourcenschonung, sondern auch der informationellen Selbstbestimmung vorteilhaft.³¹

3.3.2 Verursacherprinzip

Wie das Beispiel Spam zeigt (Abschnitt 2.1.3) sind die Funktionsprinzipien des Internet denkbar weit vom Verursacherprinzip entfernt. Anderenfalls hätten die Spammer des Jahres 2008 die Stromrechnung aller 150 Millionen Einwohner des Landes Bangladesch begleichen müssen.³²

Ein Zuordnung der Beanspruchung technischer (und damit indirekt natürlicher) Ressourcen zum jeweiligen Verursacher ist im Internet extrem schwierig und selbst bei Straftatbeständen wie dem

²⁹ Zur Forderung eines "Rechts auf Vergessen" aus Sicht des Datenschutzes siehe auch Hilty et al (2012).

³⁰ Foursquare hat eigenen Angaben zufolge weltweit acht Millionen Nutzer die sich täglich 2,5 Millionen mal mit Check-Ins anmelden. Täglich sollen ca. 35.000 neue Mitglieder hinzu kommen. Die Händler-Plattform nutzen 250.000 Unternehmen (Stand: November 2011).

³¹ Die häufige mobile Interaktion mit Internet-Plattformen ermöglicht die Erstellung von Bewegungs- und Kontaktprofilen der Nutzer (Hilty et al. 2012)

³² Der von McAfee geschätzte spambedingte Stromverbrauch von 33 TWh entspricht ungefähr dem jährlichen Stromverbrauch von Bangladesch (Hilty & Lohmann 2011).

Betrieb von Botnetzen³³ fast nicht zu leisten. Auch beispielsweise die Abrechnung der für die Erbringung einer Leistung verbrauchten elektrischen Energie gegenüber dem Kunden eines Internet Service Providers (ISP) wäre nur begrenzt möglich, weil der ISP nicht nachverfolgen kann, was der von ihm weitergeleitete Datenverkehr auslöst.

Betrachtet man das Internet als öffentliches Gut und nicht als kommerziellen Dienst, so ist es ausgesprochen schwierig, Kosten für Ressourcenbeanspruchungen einem Verursacher zuzurechnen. Eine Änderung dieses Zustandes hätte allerdings weitreichende Folgen, etwa die Gefahr einer weitgehenden Überwachung von Aktivitäten im Internet. Hier zeichnet sich langfristig ein Zielkonflikt mit demokratischen Grundrechten ab. Aus technischer Sicht sind datenschutzgerechte Lösungen prinzipiell möglich.

Wenn auf Dauer keine Wege gefunden werden, um den latenten Zielkonflikt zwischen Verursacherprinzip und Grundrechten im Internet zu lösen, wird die Ressourcenlast des weiteren exponentiellen Wachstums des Datenverkehrs unausweichlich dort getragen werden, wo dem Verursacher keine Kosten entstehen.

³³ - Ein Botnetz oder Botnet ist eine Menge von Computern, die in der Regel ohne Wissen des Inhabers über das Internet ferngesteuert werden. Der Betreiber des Botnetzes kann die fremden Rechner für seine Zwecke nutzen, beispielsweise Spam-Email verschicken, Angriffe gegen Websites führen, illegale Inhalte hosten oder Daten ausspähen.

4 Methodische Herausforderungen und Lösungsansätze

Basierend auf der Trendanalyse und den identifizierten Ansatzpunkten soll nun die Frage konkretisiert werden, wie sich ein „ressourceneffizientes“ Softwareprodukt von einem „weniger ressourceneffizienten“ unterscheiden lässt. Diese Frage führt auf ungelöste methodische Probleme, da sich Software nicht durch routinemäßige Anwendung der standardisierten Methode der Produktökobilanz (engl. Life Cycle Assessment, LCA) untersuchen lässt. Auf dem Weg zu einem Label für „grüne Software“ sind noch erhebliche Schwierigkeiten zu überwinden.

Dieses Kapitel zeigt anhand der Fachliteratur auf, welche methodischen Herausforderungen sich hier konkret stellen und gibt einen Überblick über existierende Lösungsansätze.

4.1 Methodische Herausforderungen

4.1.1 Definition funktioneller Einheiten

Die Anwendung der LCA-Methodik beginnt mit der Definition einer funktionellen Einheit. Dies ist eine präzise und quantitative Definition der Dienstleistung, die vom betrachteten Produktsystem zu erbringen ist (Rebitzer et al. 2004). Auf diese Einheit werden alle Ressourceninanspruchnahmen bezogen. Sie ist die Basis für den Vergleich verschiedener Produktsysteme in Bezug auf die Effizienz, mit der sie die Leistung erbringen. Dabei wird funktionelle Äquivalenz vorausgesetzt, d.h. die funktionellen Einheiten müssen un-unterscheidbar oder zumindest gleichwertig sein.

Betrachten wir die in Tabelle 1 auf Seite 16 zitierte Definition von Ressourceneffizienz des Umweltbundesamtes, „Verhältnis eines bestimmten Nutzens oder Ergebnisses zum dafür nötigen Ressourceneinsatz“, so entspricht die funktionelle Einheit dem „bestimmten Nutzen oder Ergebnis“. Sie ist also die Einheit für den Zähler der Effizienz, während ein Maß für die Ressourceninanspruchnahme im Nenner steht:

$$\text{Ressourceneffizienz} = \frac{\text{Anzahl erbrachter funktioneller Einheiten}}{\text{Anzahl in Anspruch genommener Ressourceneinheiten}}$$

Die Definition funktioneller Einheiten für Software ist generell schwierig. Dies liegt zum einen daran, dass die meisten Softwareprodukte eine große Menge von Funktionen erfüllen, und nur selten zwei Produkte die gleiche Menge. Selbst verschiedene Versionen des gleichen Softwareprodukts unterscheiden sich häufig in ihrem Funktionsumfang.

Selbst wenn man sich auf eine Funktion konzentriert wie z.B.

„1 E-Mail erstellen und versenden“,

so kann die funktionelle Äquivalenz leicht in Frage gestellt werden; beispielsweise können Email-Programme unterschiedliche Unterstützung beim Erstellen der Emails bieten (Verwaltung der Kontakte, Tippfehlerkorrektur usw.). Außerdem ist diese Einheit noch nicht präzise definiert, die Länge in Zeichen, der Umfang von Attachments usw. müssten exakt festgelegt werden. Die funktionelle Einheit wird dadurch so speziell, dass es einen besonderen methodischen Aufwand erfordert, eine oder mehrere statistisch repräsentative funktionelle Einheiten festzulegen. Die

statistische Repräsentativität kann wiederum nur auf der Basis empirischer Daten über das Nutzungsverhalten begründet werden.

Soll eine Bewertung von Software nach Ressourceneffizienz durchgeführt werden, muss die Repräsentativität der gewählten funktionellen Einheit(en) gewahrt sein, was hohen Erhebungsaufwand für Daten verursachen würde. Weil sich sowohl die Nutzungsmuster als auch die Eigenschaften des Softwareprodukts (durch Updates) verändern können, können die Ergebnisse schnell veralten. Hinzu kommt die methodische Herausforderung der Multifunktionalität: Ein einzelnes Softwareprodukt kann mehrere Funktionen erfüllen; dies erfordert die Definition verschiedener funktioneller Einheiten, deren Erstellung dann aus Sicht einer Lebenszyklusanalyse als Kuppelproduktion zu behandeln ist.

4.1.2 Messung des Energieverbrauchs von Software

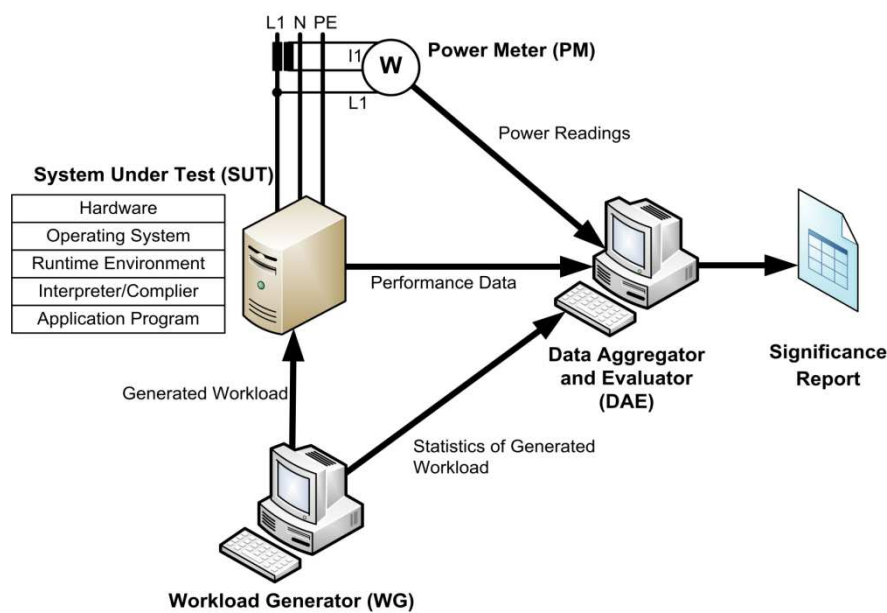
Der scheinbar einfachste Aspekt der Ressourceninanspruchnahme durch Software – der via Hardware verursachte Stromverbrauch – stellt eine methodische Herausforderung dar. Das ist darauf zurückzuführen, dass ein Softwareprodukt stets Teil eines komplexen IKT-Systems ist, das als Ganzes den Verbrauch verursacht, wobei jede einzelne Komponente des Systems Einfluss auf den resultierenden Stromverbrauch hat.³⁴ Existierende Kennzahlen für die Energieeffizienz von IKT-Systemen (Erek et al. 2013, Drenkelfort 2013) sehen in der Regel nicht vor, den Einfluss der Software isoliert zu betrachten (Van Bokhoven & Bloem 2013).

Abbildung 2 zeigt eine exemplarische Messanordnung (Kern et al. 2013). Das getestete System (System Under Test, SUT) wird hier in fünf Komponenten gegliedert: Hardware, Betriebssystem („Operating System“), Laufzeitumgebung („Runtime Environment“), Interpreter/Compiler und das Anwendungsprogramm („Application Program“). Möchte man beispielsweise verschiedene Anwendungsprogramme hinsichtlich ihres Stromverbrauchs vergleichen, so sind die übrigen vier Komponenten methodisch zwingend konstant zu halten. Das Ergebnis wird dadurch extrem kontextabhängig: Eine Aussage der Art „Programm A braucht mehr Energie als Programm B“ wäre nur mit dem Zusatz „ausgeführt von Hardware P, unter Betriebssystemversion Q, in Laufzeitumgebung R und mit Interpreter S“ gerechtfertigt.

Das Problem der Kontextabhängigkeit stellt sich zwar auch bei der Beurteilung anderer Produkte hinsichtlich ihres Ressourcenverbrauchs, jedoch stellt bei Softwareprodukten die Vielzahl der Einflussgrößen eine besondere Herausforderung dar.

³⁴ Bozzelli et al. (2013) geben einen Überblick über die Literatur zu entsprechenden Metriken. Um den Energieverbrauch in Cloud-Umgebungen abzuschätzen, sind komplexe Energieverbrauchsmodelle erforderlich (Chen et al., 2012).

Abbildung 2: Exemplarische Anordnung zur Messung des Energieverbrauchs von Software



Quelle: Kern et al. (2013; S. 91)

Das Problem verschärft sich, wenn die Interaktion der Komponenten in Netzwerken hinzukommt, was heute der Normalfall ist. Untersuchungen des Energieverbrauchs (oder allgemeiner des Ressourcenverbrauchs) von Software sind aus diesen Gründen methodisches Neuland und äußerst komplex.

In Abbildung 2 ist ein „Workload Generator (WG)“ zu sehen, der die Aufgabe hat, eine Arbeitslast für das SUT zu erzeugen, wie ein realer Nutzer das tun würde. Die Wahl der Nutzungsmuster oder Nutzungsszenarien, die durch den WG automatisiert werden, ist als Näherungslösung für das oben besprochene Problem der Definition funktioneller Einheiten zu sehen.

4.1.3 Allokationsprobleme bei stark schwankender Auslastung

Wenn mehrere Softwarekomponenten Kapazität der gleichen Hardwarekomponente nutzen, dann stellt sich das in LCA-Studien oft auftretende Problem der Allokation. Produktionswirtschaftlich betrachtet handelt es sich um Kuppelproduktion, da eine Hardwarekomponente gleichzeitig verschiedene Produkte herstellt: Leistung für Software 1, Leistung für Software 2 usw. Die Inanspruchnahme von Ressourcen durch die Hardware muss also nach einem geeigneten Schlüssel den verschiedenen sie nutzenden Softwareprodukten (genau genommen den von diesen erbrachten funktionellen Einheiten, wenn diese definiert sind) zugerechnet werden.

Besonders offensichtlich wird das Allokationsproblem bei Hardware in Netzknoten, z.B. Routern, und bei Servern. Hier hat ein einzelnes auf einem Endgerät laufendes Softwareprodukt in der Regel einen verschwindend kleinen Anteil an der gesamten Last, die der Router oder Server bewältigt.

Die Zurechnung kann entweder nach dem jeweiligen Anteil an der *momentan genutzten* oder an der *insgesamt vorgehaltenen* Hardwarekapazität vorgenommen werden. Normalerweise entscheidet man sich für das letztgenannte Schema, da ansonsten die vorgehaltenen Reservekapazitäten in Zeiten außerhalb der Lastspitzen nirgends zugerechnet würden.

Warum es sich dennoch um ein ungelöstes methodisches Problem handelt, sei am Beispiel des gemessenen Energieverbrauchs für eine Videokonferenzverbindung über das Internet zwischen der

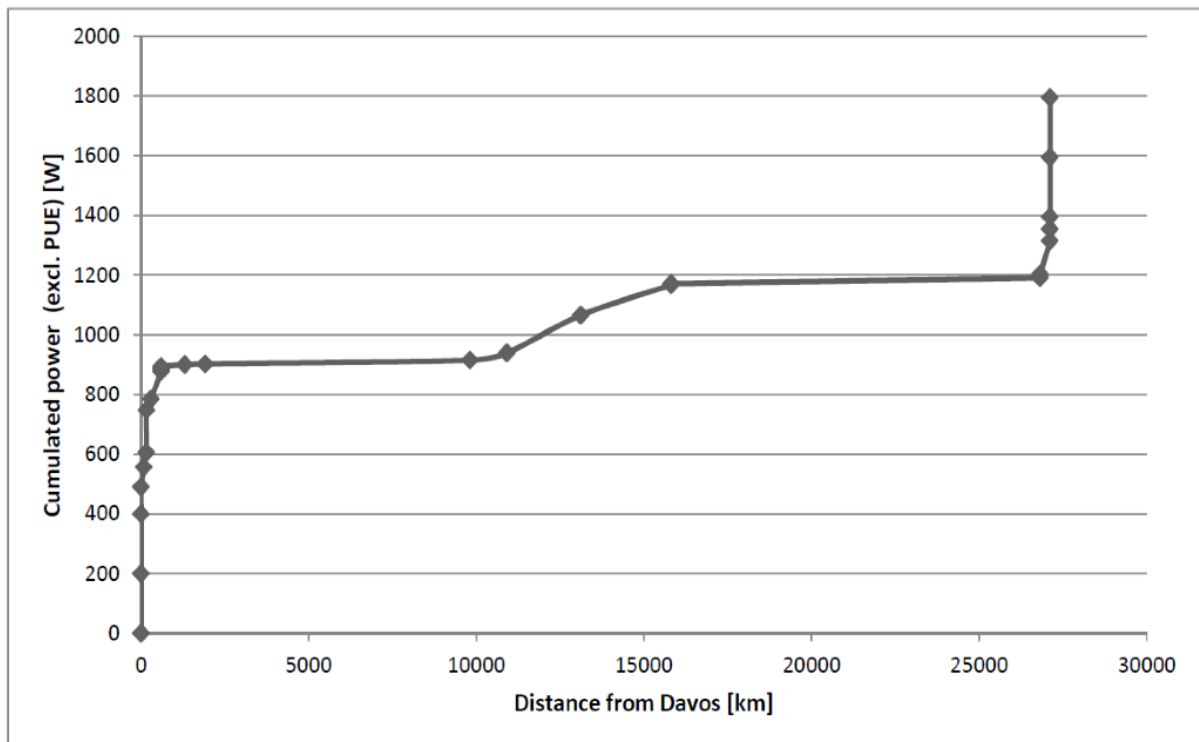
Schweiz und Japan verdeutlicht. Corama et al. (2013) haben für eine 40 Mbit/s Verbindung vom Kongresszentrum in Davos zur Universität Nagoya (4 bidirektionale Full-HD-Kanäle gleichzeitig) die Leistungsaufnahme und Auslastung aller beteiligten Netzknoten und -verbindungen ermittelt. Die Daten legten über 24 Hops eine Entfernung von 27.117 km zurück. Abbildung 3 zeigt die kumulierte elektrische Leistung über die gesamte Strecke, von Davos aus betrachtet. Das auffälligste Ergebnis ist, dass nahe an den beiden Endknoten die größten Verbräuche stattfinden und über die großen Entfernungen fast keine Energie verbraucht wird (die Tiefsee-Glasfaserverbindungen durch den Atlantik und durch den Pazifik sind als fast waagerechte Abschnitte zu erkennen).

Dieses Ergebnis ist zu einem Teil durch die Allokation zu erklären. Nahe an den Endknoten gibt es Knoten, die zur Zeit der Messung unter 3 % ausgelastet waren (zum Teil unter 1 %). Im Internet-Backbone vermischen sich dagegen sehr viele Verkehrsströme, was die Auslastung verstetigt und den Unterschied zwischen einer durchschnittlichen und der maximalen Auslastung verkleinert. Die gesamte, für seltene Lastspitzen vorgehaltene Reservekapazität in den schlecht ausgelasteten Knoten dagegen wird nach dem oben erwähnten Allokationsschema einer durchschnittlich sehr geringen Last zugerechnet.

Geht man von dieser Situation aus, kann man unter Berücksichtigung der übertragenen Datenmenge ableiten, dass die Übertragung von einem Gigabyte über diese Strecke weniger als 0,2 kWh erfordert, wobei hier ein geschätzter PUE-Wert für die Knoten und Verbindungen berücksichtigt wurde (Coroama et al., 2013). Dies ist im Vergleich zu früher publizierten Zahlen recht wenig, obwohl der größte Teil davon rechnerisch auf die extrem niedrige Auslastung einiger hausinterner und regionaler Netzknoten zurückzuführen ist.

Angenommen, man würde gleichzeitig 1000 solche Videokonferenzen in Davos abhalten: Dann würde die Zahl 0,2 kWh/GB nicht mehr gelten, denn die Auslastung der regionalen Knoten würde signifikant zunehmen. Die Energieintensität pro Konferenz würde dadurch signifikant niedriger, man würde also den Energieverbrauch *überschätzen*, wenn man weiterhin mit 0,2 kWh/GB rechnet. Erst an dem Punkt, an dem die Kapazitäten ausgebaut würden, würde real mehr Strom verbraucht.

Abbildung 3: Kumulierte elektrische Leistung einer HD-Videokonferenzverbindung



Quelle: Coroama et al. (2013)

Allokationsprobleme treten auch bei Peer-to-Peer (P2P)-Netzen auf, wo die einzelnen Nutzer dem Netz Hardwarekapazitäten zur Verfügung stellen. Beispielweise werden bei Skype Telefonate über einzelne Nutzer mit freier Kapazität (sog. Supernodes) weitergeleitet, die davon nichts bemerken. Dadurch steigt beim Supernode die Auslastung der eigenen Hardware, was bei entsprechender Allokation die Ressourceneffizienz *anderer* Programme, die gleichzeitig auf der eigenen Hardware laufen, rechnerisch verbessert. Unabhängig davon stellt sich die Frage, ob man die durch *fremde* Skype-Nutzung entstehende Last den *eigenen* Skype-Gesprächen zurechnen soll. Je nachdem, wie man diese Fragen entscheidet, kommt man zu anderen Resultaten.

Diese Beispiele sollen verdeutlichen, dass die Ergebnisse von Ressourcenabschätzungen, die (notwendigerweise) von Allokationsentscheidungen beeinflusst sind, nicht unabhängig vom Kontext interpretiert werden sollten.

Diese methodische Schwierigkeit macht u. a. auch die Frage der Beurteilung von Cloud Computing unter Ressourcenaspekten zu einem primären Problem der Abschätzung von Netzwerk- und Serverauslastungen und der Wahl von Allokationsregeln.

4.2 Existierende Lösungsansätze

4.2.1 Konzentration auf messbare Einzelaspekte

Einige Autoren gehen mit den methodischen Problemen um, indem sie sich auf messbare Einzelaspekte konzentrieren:

- Es gibt eine Reihe von Programmen, die mit standardisierten Benchmarks die durchschnittliche Prozessorauslastung durch PC-Software ermitteln und dem Nutzer helfen sollen, unter mehreren Softwareprodukten das energieeffizientere auszuwählen (siehe z. B. Amsel und Tomlinson, 2010.)
- Wilke (2012) misst den lokalen Energieverbrauch von Smartphone-Apps mit dem Ziel, die Grundlagen für ein Energie-Label zu schaffen. Er vergleicht Apps mit ähnlichen Funktionen (z.B. Email Clients) und definiert einfache Use Cases als funktionellen Einheiten.³⁵ Zwei wesentliche Beiträge zur Inanspruchnahme natürlicher Ressourcen – die Produktion des Smartphones und der im Mobilfunknetz verursachte Datenverkehr – werden nicht betrachtet, weil diese Arbeiten primär auf die Akkulaufzeit abzielen.
- Naumann et al. (2008) führten den Power Indicator ein (heute “Green Power Indicator”), ein Add-on für den Webbrowser Firefox, das anzeigt, ob eine Website auf einem Server läuft, der mit erneuerbarer Energie betrieben wird. Das Add-on greift auf eine zentrale Liste von “Öko-Providern” zu.
- Zapico et al. (2010) haben mit Greenalytics ein Verfahren zur automatischen Abschätzung des Energieverbrauchs von Websites auf Client-, Netz- und Serverseite geschaffen. Die Abschätzungen sind aber sehr grob und beruhen hauptsächlich auf Datenvolumen und Abrufstatistiken. Eine Messung des Energieverbrauchs ist nach Auffassung von Grosskop und Visser (2013) auf diese Weise nicht möglich.

4.2.2 Standardisierungsbestrebungen

Sowohl das Problem der Definition funktioneller Einheiten als auch die Mess- und Allokationsprobleme lassen sich durch Standardisierung teilweise entschärfen, weil dadurch zumindest eine Vergleichbarkeit von Ergebnissen erreicht wird.

Standardisierte Nutzungs-Szenarien könnten in Einzelbereichen für den Test von Softwareprodukten eine ähnliche Rolle übernehmen wie standardisierte Fahrzyklen für die Abgas- und Verbrauchstests von Kraftfahrzeugen. Deren Realitätsnähe ist allerdings umstritten³⁶, so dass aufgrund der noch höheren Komplexität von Softwareprodukten ein breiter Konsens über „Software-Fahrzyklen“ unter verschiedenen Herstellern, der akademischen Forschung, Behörden und Verbraucher- oder Umweltorganisationen eher schwer erreichbar erscheint. Während bei einem Kraftfahrzeug implizit vorausgesetzt werden kann, dass dessen Funktion im Zurücklegen einer Strecke besteht, ist bei Software schon die Wahl der dem Testzyklus zugrundeliegenden Funktion(en) eine normative Entscheidung mit entsprechendem Konfliktpotenzial.

³⁵ z. B. “Check Inbox”, “Read Mail”, “Open Attachment”. Ein signifikant höherer Energieverbrauch wurde bei jenem Produkt festgestellt, das Werbung anzeigt (Wilke 2012).

³⁶ Dies zeigt sich schon auf der Wikipedia-Seite zum Stichwort „Fahrzyklus“: „Die genormten Fahrzyklen stellen Durchschnittsprofile dar, um die Fahrzeuge untereinander vergleichen zu können. Sie stimmen oft nicht mit dem Nutzungsprofil des Kunden überein...“ (Wikipedia 2013c).

Dick et al. (2011) definieren Lastprofile für IKT-Systeme nach ISO/IEC 14756.³⁷ Diese Profile sind randomisiert, d.h. die Abfolge und Dauer von Nutzeraktionen wird in einem festgelegten Rahmen zufällig gewählt. Durch zahlreiche Wiederholungen des Profils kann der durchschnittliche Energieverbrauch für das Profil in einer gegebenen Konfiguration des IKT-Systems ermittelt werden. Damit solche Lastprofile realistisch sind, müssen sie sich an „typical workloads“ im jeweils intendierten Anwendungsgebiet orientieren (Dick et al. 2011; S. 292). Kern et al. (2013) fordern, Nutzungsszenarien für Standardsoftware zu definieren, damit auf diese Weise der Energieverbrauch verschiedener Softwareprodukte und -konfigurationen systematisch verglichen werden kann. Solche Nutzungsszenarien würden dann auch den Lastprofilen für Messungen zugrundeliegen. Standards für Nutzungsszenarien sind noch nicht in Sicht. Grundsätzlich können auch Geschäftsprozessmodelle (die z.B. in BPMN³⁸ definiert sind) als Grundlage für Lastprofile dienen.

Die International Telecommunications Union (ITU) hat ein „Assessment framework for environmental impact of ICT“ erarbeitet (ITU 2012a). Es ist Teil des „ITU Toolkit on Environmental Sustainability for the ICT sector“, welches durch ITU-T mit über 50 Organisationen und ICT Firmen entwickelt wurde, um für den IKT-Sektor die Anforderungen für Nachhaltigkeit aus Umweltperspektive zu definieren („Environmental Sustainability Requirements“; ITU 2012b).

Das „Assessment framework“ gibt einen Überblick über die vorhandenen Normen und Standards, darunter auch den ITU-T Standard L.1410, „Methodology for environmental impact assessment of ICT goods, networks and services“, der auf den ISO-Standards zu LCA aufbaut. Darin wird eine Methodik zur Abschätzung der Umweltauswirkungen von IKT-Systemen präzisiert (ITU 2012c).

In dem 85seitigen Dokument sind 2 Seiten im Haupttext und 2 Seiten im Anhang dem Thema Software gewidmet. Dies zeigt, dass Software als Gegenstand von umweltbezogenen Standards noch wenig etabliert ist. In der ITU-T Norm wird im Wesentlichen die Abschätzung der Wirkungen der *Produktion* von Software und die Allokation der Produktionsphase (Division durch die Anzahl der verkauften Lizenzen) geregelt.

Das Problem der Definition funktioneller Einheiten bei Software wird in einem Beispiel zur Textverarbeitung erwähnt: „The function experienced by a user of a word processor program is to deliver word processing of documents electronically. The corresponding functional unit could then be the number of pages processed per time unit (e.g., one hour) during the operational lifetime (e.g., three years). Finally, the reference flow is defined as one unit of word processing software (distributed e.g., in a CD with packaging)“ (ITU 2012c; S. 8). Auch in diesem Beispiel steht der Impact der Produktionsphase der Software im Vordergrund, sogar die traditionelle Auslieferung via CD wird betrachtet. Die Auswirkungen der Software über die Beanspruchung von Hardwarekapazitäten in der Nutzungsphase werden dagegen nicht thematisiert.

Im Zusammenhang mit IKT-Diensten wird als funktionelle Einheit generell die Nutzung während eines Jahres („annual service use“) vorgeschlagen. Es wird darauf hingewiesen, dass hierfür

³⁷ ISO/IEC 14756 beschreibt ein Vorgehensmodell zur Leistungsmessung für Software. Es werden klassische Leistungsmerkmale von IKT wie Ausführungszeit und Durchsatz für gegebene Lastprofile gemessen. Diese Norm hat keinen spezifischen Bezug zur Ressourceneffizienz von Software, die Vorgehensweise bei der Definition und Verwendung von Lastprofilen und bei der Durchführung von Messungen kann jedoch übertragen werden.

³⁸ Business Process Model and Notation, Standard der Object Management Group zur Beschreibung von Geschäftsprozessen (BPMN 2013).

realistische Nutzungs-Szenarien definiert werden müssen und dass eine Allokation von Netzwerk-Kapazität stattfinden müsse, ohne dass konkrete Lösungshinweise gegeben werden (ITU 2012c; S. 9).

Das GHG Protocol, eine 1998 von WRI und WBCSD gegründete Initiative, entwickelt Standards für die Erhebung und Berichterstattung zu Treibhausgasemissionen von Unternehmen. Auf Anstoß von IKT-Unternehmen entwickelt das GHG Protocol zusammen mit The Carbon Trust und der Global eSustainability Initiative (GeSI) spezielle Richtlinien für den IKT Sektor („ICT Sector Guidance“, GHGP 2013a, Stephens & Didden 2013).

Ingesamt werden in der „ICT Sector Guidance“ folgende Themen behandelt: Telekommunikationsdienste, Desktop Managed Services, Cloud- und Rechenzentrumsdienste, Hardware, Software, Transportvermeidung (GHGP 2013b). Das Kapitel zu Software liegt erst als Draft vor (Version 2.9, GHGP 2013c). Im Folgenden werden die für unseren Gegenstand wichtigsten Aspekte herausgegriffen.

Im Gegensatz zu ITU-T legt das GHG Protocol den Fokus auf die Auswirkungen von Software in der Nutzungsphase: „Up to 90 % of the energy used by ICT hardware can be attributed to the application software running on it, and the design of software can have a significant impact on the amount of energy used. It is therefore important that software designers carefully consider the energy use of the software, and design software efficiently to reduce the energy use. Examples of where better software design can reduce the energy use are: optimizing the CPU usage; optimizing the disk IO usage; optimizing remote calls such as database calls, and web accesses.“ (GHGP 2013c; S. 4)

Der zentrale Beitrag des Softwarekapitels sind die detaillierten Vorschläge zum Vorgehen bei der Vorbereitung, Durchführung und Auswertung von Stromverbrauchstests für System- und Anwendungssoftware einschließlich eingesetzter Softwarebibliotheken. Dabei wird großer Wert auf die Verwendung von Benchmarks und die korrekte Berücksichtigung des Energiemanagements durch das Betriebssystem oder die Anwendungssoftware gelegt. Auch das Testen entfernter Geräte und virtueller Maschinen wird berücksichtigt.

Weil dieser Standard die prinzipielle Vorgehensweise beschreibt, werden für die Benchmarks keine konkreten Vorschläge gemacht. Der Standard leistet somit zwar einen substantiellen Beitrag zur Lösung des Messproblems (Abschnitt 4.1.2), setzt aber implizit voraus, dass sich sinnvolle und akzeptierte funktionelle Einheiten definieren lassen (Abschnitt 4.1.1), denn auf diese muss sich die Definition von Benchmarks stützen. Wie weit der Standard zur Lösung von Allokationsproblemen bei niedrigen Auslastungen beitragen kann (Abschnitt 4.1.3), ist aus heutiger Sicht noch nicht festzustellen.

4.2.3 Vergleich funktionell ähnlicher Softwareprodukte

Da funktionelle Äquivalenz bei Softwareprodukten in der Regel nicht gegeben ist (vgl. Abschnitt 4.1.1), kann der Vergleich von Produkten, die einander funktionell lediglich ähnlich sind, einen Ausweg darstellen. Der Vergleich kann sich dann auf die gemeinsamen Funktionen beschränken, setzt für diese allerdings ein realistisches Nutzungsszenario voraus.

Beispiele für entsprechende Studien zum Energieverbrauch von Softwareprodukten in der Nutzungsphase:

- Kern et al. (2011) vergleichen den Webbrowser Mozilla Firefox mit dem Microsoft Internet Explorer sowie Open Office Writer mit Microsoft Word.

- Roth et al. (2013) vergleichen im Auftrag von Microsoft die Webbrowser Mozilla Firefox, Google Chrome und Microsoft Internet Explorer und kommen zu einem anderen Ergebnis als Kern et al. (2011).
- Williams und Tang (2013) vergleichen die Microsoft-Produkte Outlook, Excel und Word mit ihren webbasierten und in der Cloud ausgeführten Pendanten aus Office 365 (vgl. auch Abschnitt 3.1.4)

Die größte funktionelle Ähnlichkeit liegt vor, wenn unterschiedliche Konfigurationen des gleichen Produkts verglichen werden. Der reine Konfigurationsvergleich kann relevante Energiesparpotenziale aufdecken, wie Dick et al. (2011) am Beispiel eines Content Management Systems gezeigt haben: Mit Harddisk-Cache war der Energieverbrauch signifikant niedriger als ohne.

4.2.4 Vergleich eines Softwareprodukts mit sich selbst im Zeitverlauf

Eine andere Möglichkeit besteht darin, verschiedene Releases oder Versionen des gleichen Softwareprodukts miteinander unter dem Aspekt der Ressourceninanspruchnahme zu vergleichen.

Wenn eine neuere Version mehr Hardwarekapazität (z. B. Speicherplatz, Rechenleistung, Netzwerk-Bandbreite) benötigt als die ältere benötigt, so schafft sie einen Anreiz zum Ersatz funktionsfähiger Hardware. Wenn dieser Obsoleszenz-Effekt im Vordergrund der Betrachtung steht, sind die Mess- und Allokationsprobleme teilweise entschärft, da sich die Nutzung von Hardwarekapazitäten leichter erheben lässt als der daraus resultierende Anteil am Energieverbrauch der Hardware.

Die funktionelle Äquivalenz ist allerdings in der Regel nicht gegeben, da neue Versionen häufig zusätzliche funktionale Anforderungen erfüllen.

Hilty et al. (2006) haben beim Vergleich von drei Versionen des Betriebssystems Windows und den dazu passenden Versionen von Word festgestellt, dass das zum Zeitpunkt der Untersuchung noch neue Windows NT im Vergleich zu Windows 2000 mehr Rechenzeit zur Ausführung der gleichen Funktionen benötigte, selbst wenn es auf einer mehr als doppelt so leistungsfähigen Hardware ausgeführt wurde.³⁹

Grundlage des Tests waren Aufgaben zum Dateimanagement und zur Textverarbeitung, die von 42 Versuchspersonen ausgeführt wurden. Dabei wurden die Ausführungszeiten für die Aufgaben und die Beanspruchung des Prozessors gemessen. Die Überkompensation der zunehmenden Hardwareleistung durch die Ansprüche der Software an die Hardware wird von Herstellern generell als notwendige Folge der Erfüllung neuer Anforderungen gerechtfertigt. Allerdings ist die Notwendigkeit dieses Zusammenhangs nicht belegbar, und die Ressourceneffizienz selbst könnte als Anforderung mit hoher Priorität behandelt werden (siehe auch den folgenden Abschnitt 4.2.5).

4.2.5 Green Software Engineering

Ein weiterer Lösungsansatz besteht darin, die Perspektive von messbaren Eigenschaften des Softwareprodukts auf den Prozess auszuweiten, der dieses Produkt hervorbringt und pflegt. Die Frage lautet dann nicht mehr „Was ist grüne Software?“ sondern „Was ist grünes Software Engineering?“.

³⁹ Damit wurde an diesem Beispiel das so genannte Wirth'sche Gesetz bestätigt: „Software is getting slower more rapidly than Hardware becomes faster“ (zit. nach Grosskop & Visser 2013; S. 101).

Softwareentwickler sind es gewohnt, in Anforderungen zu denken und Systeme zu entwickeln, die diese erfüllen. Man unterscheidet funktionale Anforderungen (die beschreiben, was das zu entwickelnde System tun soll) und nichtfunktionale Anforderungen, die qualitative Aspekte des Systems beschreiben. Typische nichtfunktionale Anforderungen im Software Engineering sind Sicherheit, Wartbarkeit oder Erweiterbarkeit. Weil nichtfunktionale Anforderungen Eigenschaften der Softwarearchitektur sind, müssen sie schon in frühen Phasen der Softwareentwicklung berücksichtigt werden und prägen den gesamten Prozess. Ein nachträgliches Hinzufügen von Funktionen ist leichter möglich als die nachträgliche Erfüllung nichtfunktionaler Anforderungen.

Wenn es also gelingen soll, „grüne“ Softwareprodukte zu erarbeiten, dann sind zwei Probleme zu lösen:

1. Die Definition nichtfunktionaler Anforderungen, die präzisieren, was „grün“, „ressourcenschonend“ oder allgemeiner „nachhaltig“ als qualitative Eigenschaft eines Softwareprodukts bedeuten.
2. Die Berücksichtigung dieser Anforderungen über den ganzen Prozess der Softwareentwicklung durch die Entwickler.

Damit wäre es möglich, Softwareprodukte danach zu kennzeichnen, ob sie aus einem Prozess hervorgegangen sind, der die Anforderungen berücksichtigt, bzw. den Prozess als solchen zu zertifizieren.

1. Definition nichtfunktionaler Anforderungen für „grüne“ oder nachhaltige Software:

Basierend auf den vorausgegangenen Kapiteln und der zitierten Literatur werden hier erste Ideen für nichtfunktionale Anforderungen formuliert:

- Bedarfsgerechte Steuerbarkeit (wie in Abschnitt 3.1.5 beschrieben)
- Neue Versionen stellen bezüglich Speicherplatz, Rechenleistung und Bandbreite keine höheren Anforderungen, es sei denn, zusätzliche Funktionen erfordern dies zwingend.
- Kernfunktionen bleiben dauerhaft auf älterer Hardware ausführbar.
- Benutzergerechte Konfigurationsmöglichkeiten für energiesparende Modi.
- Power Awareness, optimales Management der Hardware in Bezug auf Energieverbrauch; Software für Server sollte auch die auf Client-Seite benötigte Energie berücksichtigen, keinesfalls das Abschalten der Endgeräte oder lokales Powermanagement behindern.
- „Abschaltfreundlichkeit“, Software sollte nicht zum Dauerbetrieb von Hardware animieren
- Unterstützung sparsamer Datenformate in Hinblick auf Bandbreite und Speicherplatz.
- Unterstützung offener Standards bei Datenformaten (kein „customer lock-in“ durch Formate).
- Flexibilität bezüglich verwendbarer Peripheriegeräte (Beschaffungszwänge minimieren).
- Unerwünschte Werbung lässt sich abschalten.

2. Prozessaspekte des Green Software Engineering:

Naumann et al. (2011, 2013) haben das Referenzmodell GREENSOFT für Green Software Engineering entwickelt. Es deckt den gesamten Software-Lebenszyklus ab und sieht in der Entwicklungsphase unter anderem vor, regelmäßige „Sustainability Reviews and Previews“ durchzuführen (Dick 2010a) oder den Ansatz der „Continuous Integration“ zu nutzen, um Energieeffizienzmessungen schon während des Entwicklungsprozesses durchzuführen (Dick et al, 2013).

Als Instrument für Green Software Engineering werden mehrfach Leitfäden und Checklisten für Entwickler genannt, es gibt aber bisher kaum konkrete Vorschläge. Dick et al. (2010b, 2010c)

formulieren Richtlinien für „Green Web Engineering“, Microsoft (2010) gibt praktische Hinweise für Anwendungsprogrammierer zum Power-Management auf Windows-Plattformen.

Eine Konsolidierung und Weiterentwicklung von Richtlinien für Entwickler auf Basis eines Katalogs nichtfunktionaler Anforderungen steht noch bevor. Die neue Workshopreihe „Energy Aware Software-Engineering and Development“ sammelt die existierenden Leitfäden und Checklisten, um sie der Allgemeinheit zur Verfügung zu stellen (EASED 2013).

5 Handlungsempfehlungen

Dieses Kapitel formuliert Handlungsempfehlungen, die darauf abzielen, die durch Softwareprodukte indirekt ausgelöste Inanspruchnahme natürlicher Ressourcen zu verringern. Die Empfehlungen stützen sich auf die in den vorausgegangenen Kapiteln dargestellten Ergebnisse zu Trends, Ansatzpunkten, methodischen Problemen und Lösungsansätzen. Wo es zum Verständnis der Empfehlungen erforderlich ist, wird auf entsprechende Abschnitte zurückverwiesen.

5.1 Forschungs- und Standardisierungsbedarf

5.1.1 Entwicklung von Methoden und Standards

An der Schnittstelle zwischen unabhängiger anwendungsorientierter Forschung auf der einen und Standardisierung auf der anderen Seite besteht der größte Handlungsbedarf. Der Forschung fällt dabei die Aufgabe zu, methodische Probleme zu analysieren und zu lösen. Die Aufgabe der Standardisierung besteht dann darin, wissenschaftlich fundierte Methoden für die Praxis zu operationalisieren und die Grundlage für die Vergleichbarkeit in der Umsetzung zu schaffen. In diesem Sinne werden im Folgenden zwei Maßnahmen empfohlen.

Standardisierte Nutzungsszenarien als Grundlage von Softwaretests

Nutzungsszenarien beschreiben einen typischen Ablauf der Nutzung einer Softwarefunktion und bilden bei der Durchführung von Vergleichstests eine entscheidende Grundlage zur Definition von Lastprofilen und Benchmarks (siehe Abschnitt 4.2.2). Sie werden nicht nur für den Vergleich verschiedener Softwareprodukte, sondern auch verschiedener Versionen und Konfigurationen des gleichen Produkts benötigt.

Bei mangelnder Systematik der Vorgehensweise zur Entwicklung und Dokumentation von Nutzungsszenarien ist die Qualität der Ergebnisse nicht einschätzbar. Ohne einheitliche Nutzungsszenarien sind Testergebnisse untereinander nicht vergleichbar.

Bei der Entwicklung standardisierter Nutzungsszenarien kann auf ersten bestehenden Ansätzen aufgebaut werden (Abschnitt 4.2.2).

Definition nichtfunktionaler Anforderungen für nachhaltige Software

Forschungsbedarf besteht für die systematische Entwicklung von qualitativen Kriterien, die die Idee von nachhaltiger Software präzisieren. Diese Kriterien sind im Sinne des Software Engineering als nichtfunktionale Anforderungen zu verstehen.

Gleichzeitig mit der Entwicklung dieser Anforderungen sind Handreichungen oder Werkzeuge für die Praxis zu schaffen, die ihre Berücksichtigung im Prozess der Softwareentwicklung unterstützen. Hierzu zählen insbesondere Ergänzungen zu bestehenden Vorgehensmodellen, Best-Practice-Richtlinien und Checklisten.

5.1.2 Regelmäßige Datenerhebung

Im Sinne eines Monitorings werden regelmäßige Erhebungen in den folgenden zwei Bereichen empfohlen, solange die derzeit zu beobachtende Wachstumsdynamik anhält.

Beobachtung des Energieverbrauchs der Mobilfunknetze und seiner Ursachen

Wenn erwartungsgemäß ein starker Anstieg des Energieverbrauchs durch die Zunahme des mobilen Internetzugangs eintritt, sollten Maßnahmen zur Förderung einer verursachungsgerechten Preisgestaltung im IKT-Bereich generell geprüft werden (z. B. keine Flatrates für besonders energieintensiven oder besonders CO₂-intensiven Datenverkehr). Dadurch würden sich die Kunden bewusst, dass aus Ressourcensicht z. B. ein riesiger Unterschied zwischen einer SMS und einem Video besteht, der heute durch die Preisgestaltung verschleiert wird. Solche Maßnahmen könnten auch zusätzliche Anreize zur weiteren Verbesserung der Ressourceneffizienz in den Internet-Zugangsnetzen, einschließlich Mobilfunknetze, durch die Provider schaffen, weil diese unter Konkurrenzdruck zu niedrigen Preisen anbieten wollen. Gleichzeitig würde für die Entwickler von Web- und Cloud-basierten Anwendungen ein Anreiz entstehen, den von ihren Produkten verursachten Datenverkehr zu minimieren, da dies die Betriebskosten für die Nutzerinnen und Nutzer senkt.

Beobachtung des Marktes von web- und cloudbasierten Anwendungen unter dem Aspekt der Ressourceneffizienz

Durch ein regelmäßiges Screening können besonders günstige und besonders ungünstige Angebote und Konfigurationen identifiziert werden, wobei die Inanspruchnahme von Ressourcen durch das Gesamtsystem betrachtet werden muss, das die jeweilige Leistung erbringt. Dieses Screening dient

- als Vorbereitungs- und Lernphase für die Definition von Kriterien für die Vergabe eines „Blauen Engels“ für ressourcenschonende Webbasierte Anwendungen (siehe Abschnitt 5.2.1).
- zur laufenden Aktualisierung der Konfigurationsempfehlungen an die Nutzerinnen und Nutzer (siehe Abschnitt 5.2.2).

5.2 Verbraucherbezogene Maßnahmen

5.2.1 Vergabe eines Blauen Engels für Software

Auf dem Weg zu einem Blauen Engel für Software sind zwar noch erhebliche methodische Probleme zu lösen (siehe Forschungs- und Standardisierungsbedarf, Abschnitt 5.1), aber es lassen sich Teilbereiche identifizieren, die mittelfristig für eine solche Maßnahme geeignet wären. Diese sind:

- Traditionelle Websites
- Webbasierte Anwendungsprogramme

Diese Angebote erfordern die Abschätzung der netzwerk- und serverseitigen Ressourceninanspruchnahme für zu definierende Nutzungsszenarien. Anforderungen an die Energiebereitstellung (Anteil erneuerbarer Energie) auf Serverseite und andere Aspekte des Hostings (existierender Blauer Engel für Rechenzentren) sollen berücksichtigt werden. Die Hardwareprodukte, Betriebssysteme, Browser und Konfigurationen auf Clientseite werden nicht bewertet, aber sie

werden als Kontext für Messungen oder Abschätzungen benötigt (siehe Abschnitt 4.1.2), woraus als Nebeneffekt Informationen für Konfigurationsempfehlungen (Abschnitt 5.2.2) hervorgehen.

Für die Vergabe eines Blauen Engels für lokal installierte Softwareprodukte sind quantifizierbare Produkteigenschaften schwer zu definieren, da die Funktionsvielfalt und die Variabilität der Nutzungsformen zu hoch sind. Ergänzend sind hier deshalb qualitative produktbezogene Kriterien wie bedarfsgerechte Steuerbarkeit, Unterstützung ressourcensparender Datenformate und Energiemanagement auf dem Endgerät hinzuzuziehen (siehe auch Abschnitte 4.2.5 und 5.1.1).

Zusätzlich wird die Anwendung prozessbezogener Kriterien für die Softwareentwicklung empfohlen, etwa die Einhaltung von (noch zu entwickelnden) Best-Practice-Richtlinien für nachhaltige Softwareentwicklung (Abschnitt 4.2.5). Dabei ist insbesondere zu berücksichtigen, ob die Weiterentwicklung des Softwareprodukts zur Außerbetriebsetzung funktionsfähiger Hardware führt und welche Maßnahmen der Hersteller ergreift, um vorzeitige Obsoleszenz zu vermeiden bzw. nicht zu stimulieren.

5.2.2 Informationsmaßnahmen

Aufgrund der bestehenden methodischen Schwierigkeiten ist es auf dem heutigen Stand der Forschung sehr schwierig, objektive und belastbare Empfehlungen an die Endnutzerinnen und -nutzer von Software abzugeben; Empfehlungen für oder gegen bestimmte Produkte oder Produktarten wären nur unter stark einschränkenden, komplex zu kommunizierenden Bedingungen gültig.

Eine Ausnahme bilden Empfehlungen zur ressourceneffizienten Konfiguration verbreiteter Softwareprodukte. Es werden in diesem Fall nicht Produkte, sondern Konfigurationen des gleichen Produkts miteinander verglichen. Durch Erarbeitung und Publikation solcher Empfehlungen könnte das Umweltbundesamt die „low hanging fruits“ in der Ressourceneffizienz von IKT ernten.

Erarbeitung und regelmäßige Aktualisierung von Konfigurationsempfehlungen

Die folgende Vorgehensweise (z. B. durch das Umweltbundesamt oder durch Forschungsstellen) wird empfohlen:

1. - Auswahl eines verbreiteten Softwareprodukts (oder einer verbreiteten Kombination von Anwendungs- und Systemsoftware), das bedarfsgerecht konfiguriert werden kann.
2. - Durchführung von Vergleichstests der Konfigurationen des Softwareprodukts (gleiches Produkt, gleiche Version, verschiedene Konfigurationen) zur Messung der beanspruchten Hardwarekapazitäten und des verursachten Stromverbrauchs.
3. - Identifikation von Einstellungen, die signifikanten Einfluss auf die Messgrößen haben. Hier können eventuell bereits existierende Empfehlungen der Hersteller oder von User Communities überprüft werden.
4. - Erarbeitung bzw. Aktualisierung von Handreichungen, die den Nutzerinnen und Nutzern, System-Administratorinnen und -Administratoren z. B. energiesparende Einstellungen empfehlen.

Auf der gleichen Datengrundlage sollten gezielte Empfehlungen für die Computer-Arbeitsplätze der Bundesverwaltung erarbeitet und für möglichst viele Arbeitsplätze umgesetzt werden.

Diese Maßnahme zielt darauf ab, die von Softwareprodukten geforderten Eigenschaften wie bedarfsgerechte Steuerbarkeit und Energiemanagement in der Praxis auch zur tatsächlichen Ressourcenschonung auszunutzen.

5.3 Empfehlungen für Softwareentwickler

5.3.1 Leitfäden, Best-Practice-Richtlinien, Checklisten

Der Stand der Forschung lässt es – trotz vielversprechender Ansätze wie dem GREENSOFT-Modell (Naumann et al. 2011) – derzeit noch nicht zu, Instrumente wie Leitfäden, Best-Practice-Richtlinien oder Checklisten für die Software-Entwicklung zu formulieren.

Zumindest in einem speziellen Bereich gibt es jedoch eine Vielzahl von Hinweisen und Praktiken für energieeffizientes Programmieren. Dies ist die Entwicklung von Apps für mobile Endgeräte, da hier die Maximierung der Akkulaufzeit eine wichtige Anforderung darstellt. Entsprechend wird allerdings nur der lokale Energieverbrauch berücksichtigt. Die neue Workshopreihe EASED (Energy Aware Software-Engineering and Development) konsolidiert und publiziert das hier erarbeitete Wissen, das sehr begrenzt auch auf andere Gebiete der Softwareentwicklung übertragbar ist. Softwareentwicklern wird empfohlen, die Aktivitäten der EASED-Workshops und die Forschung im Bereich des „Green Software Engineering“ generell zu verfolgen und sich aktiv am Ausbau des Wissens- und Erfahrungsschatzes zu beteiligen.

5.4 Aus- und Weiterbildung

5.4.1 Lehrmittel zu Ressourcenaspekten von Softwarearchitekturen für die Informatikausbildung

Es wird empfohlen, Lehrmittel zum Thema der Ressourceninanspruchnahme durch Software zu entwickeln mit dem Ziel, die Inhalte in das Fach Informatik bzw. Wirtschaftsinformatik an Universitäten, Fachhochschulen und in die Ausbildung zum Fachinformatiker einzubringen. Weitere geeignete Studiengänge sind: Computervisualistik, Medieninformatik, Webdesign, Spielentwicklung und -design.

Ein möglicher Partner wäre die Gesellschaft für Informatik (GI). Beispiele für Themen:

- Nachhaltigkeit als nichtfunktionale Anforderung in der Softwareentwicklung
- Ressourceneffizienz im Webbased und Cloud Computing
- Energy awareness und Lastmanagement in verteilten Systemen
- Ressourceneffizientes Daten- und Informationsmanagement

5.4.2 Weiter- und Fortbildungsangebote zu Ressourcenaspekten der IKT-Nutzung für Unternehmen und öffentliche Einrichtungen

Unternehmen und öffentliche Einrichtungen können durch Maßnahmen im Softwarebereich erheblich zur Ressourcenschonung beitragen und Kosten sparen. Ein großes Potenzial besteht bei KMU und Schulen (hier als kleine bis mittlere IKT-betreibende Organisationen, siehe aber Abschnitt 5.4.3) aufgrund der großen Zahl von lokalen Servern, die von diesen Organisationen in der Summe betrieben werden.

Es wird empfohlen, entsprechende Lehrinhalte für die Weiterbildung in Unternehmen und öffentlichen Einrichtungen zu entwickeln. Diese sollten in bestehende Weiterbildungsangebote

integriert werden. Hier sollte insbesondere mit dem Deutschen Industrie- und Handelskammertag (DIHK) Kontakt aufgenommen werden und das Thema adressiert werden. Ziel könnte es sein, die Ressourceneffizienz von Software bei Aus- und Weiterbildungsmaßnahmen stärker in die Lerninhalte zu integrieren. Partner, mit denen gemeinsam entsprechende Weiterbildungsprogramme entwickelt und eingeführt werden könnten, wären softwarebezogene IKT-Verbände. Für die berufliche Aus- und Weiterbildung bieten sich insbesondere Trainingsseminare zum Thema Green IT an, wie sie beispielsweise vom Oberösterreichischen Energiesparverband (ESV 2013) angeboten werden. Zielgruppen sind vor allem IT- und Energieverantwortliche in Betrieben, Kommunen und öffentlichen Institutionen. Gerade kleine und mittlere Unternehmen nutzen die Angebote.

Aufgrund ihrer großen Zahl von Arbeitsplätzen und ihrer Nachfragemacht im IKT-Markt wird empfohlen, die Bundesverwaltung ebenfalls einzubeziehen. Als Partner für Fortbildung von Angehörigen der Bundesverwaltung bietet sich die Bundesakademie für öffentliche Verwaltung (BAkÖV) an.

5.4.3 Angebote für den Informatikunterricht an Schulen

Im Informatikunterricht an allgemeinbildenden Schulen ist es generell wichtig, den Fokus auf Prinzipien der Informatik zu legen und kein flüchtiges Spezialwissen zu vermitteln. Das muss auch bei der Integration umwelt- und ressourcenorientierter Themen in den Informatikunterricht (oder andere Fächer mit IKT-Bezug) berücksichtigt werden.

Beispiele:

1. Grundlagenorientierte Lerneinheiten sollen die Fähigkeit der Lernenden fördern, Sachinformationen auf dem Gebiet der IKT richtig zu verstehen und einzuschätzen. Dazu gehört es, Maßeinheiten zu verstehen (Was ist ein Byte, ein Gigabyte, ein Exabyte?), Größenordnungen zu kennen (Wie groß ist eine Email, ein Foto, ein digitaler Kinofilm? Was leistet mein Handy im Vergleich zu einem Supercomputer in meinem Geburtsjahr? Wie hat sich der Datenverkehr im Internet entwickelt? Wie lange würde es dauern, alle in diesem Moment existierenden Youtube-Videos anzuschauen?). Darauf aufbauend wird das technische Grundwissen dann mit naturwissenschaftlichen Fakten und aktuellen Daten verknüpft (Wieviel Energie benötigt heute ein Video-Chat im Vergleich zu einer Autofahrt, einer Bahn- oder Flugreise? Welche chemischen Elemente sind heute in einem Mikroprozessor enthalten, wo kommen sie her? Wie groß müsste ein Solarmodul sein, um die IT eines heutigen Privathaushalts zu versorgen? Wieviele Kraftwerke arbeiten heute für Google?). An der Erhebung bzw. Recherche und Aktualisierung von Daten können die Lernenden aktiv beteiligt werden. Auch ökonomische Zusammenhänge passen in eine solche Lerneinheit (Wer finanziert das Internet? Wie verdienen Firmen wie Google oder Facebook ihr Geld?).

2. Eine Möglichkeit der Wissensvermittlung für Schüler und Schülerinnen bieten Technikmuseen. Viele Technikmuseen thematisieren die wachsende Bedeutung der IKT in ihren Ausstellungen (Technische Sammlungen in Dresden, Museum für Kommunikation in Berlin u.a.). Hier wird empfohlen die Einrichtung von „School Labs“ für IKT thematisch dahingehend zu unterstützen, dass ökologische Aspekte integriert werden. Dies schließt auch weitere Formate wie Experimentierworkshops für Schüler und Schülerinnen mit ein, um die Jugendlichen für die Thematik „grüner“ Software zu sensibilisieren und Handlungsmöglichkeiten aufzuzeigen.

5.4.4 Wettbewerbe für ressourceneffiziente Software

Es wird empfohlen, jährliche Wettbewerbe zu veranstalten, bei denen Studenten und Entwickler aus der Praxis ressourcensparende Verbesserungen an existierenden Open-Source-Softwareprodukten einreichen können. Damit kann sich jeder Entwickler auch ohne umfangreiches Spezialwissen zur Anwendungsdomäne eines Softwareprodukts beteiligen und sich beispielsweise einzelnen Algorithmen in einem Modul widmen.

Neben dem bewusstseinsbildenden Effekt für Ressourcenfragen von Software auf die Teilnehmenden und auf die interessierte Öffentlichkeit können gute Lösungen sich aufgrund der Offenheit von OSS schnell verbreiten und auch in andere Softwareprodukte übernommen werden.

Der Gewinn in 3 Kategorien soll an die Projekte gehen, deren Änderung

- A. die meisten Ressourcen pro Nutzung einspart,
- B. die meisten Ressourcen über alle erwarteten Nutzungen des gleichen Softwareprodukts einspart
- C. die meisten Ressourcen über alle potenziellen Nutzungen der Idee (auch in anderen Softwareprodukten) einspart.

Die Dokumentation der Lösungen kann zu einem Katalog wachsen, der als Beispielsammlung für innovative, ressourceneffiziente Verbesserungen von Software in die Aus- und Weiterbildung von Softwareentwicklern eingebracht werden kann.

Literatur

- AC4DC (o.J.): Bisher unveröffentlichte Angaben aus dem Projekt AC4DC, an dem die Autoren der vorliegenden Studie beteiligt sind, <http://www.AC4DC.de>; abgerufen am 24.07.2013
- Accenture (o. J.): Application Services for Open Source Software: Service-Überblick. <http://www.accenture.com/de-de/Pages/service-technology-systems-integration-open-source-overview.aspx> .abgerufen am 29.6.2013
- Amsel, N., Tomlinson, B. (2010): Green Tracker: a tool for estimating the energy consumption of software. In: CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. ACM, New York, S. 3337–3342
- Bärwolff, M., Gehring, R. A., Lutterbeck, B. (Hrsg.) (2005): Open Source Jahrbuch 2005: Zwischen Softwareentwicklung und Gesellschaftsmodell, Berlin, Lehmanns Media – LOB.de, <http://www.opensourcejahrbuch.de/download/jb2005/>; abgerufen am 30.6.2013.
- Baumeister, J. (2012): Server-Management-Suites: Management-Software der Server-Hersteller im Vergleich. In Tecchannel vom 18.11.2009. http://www.tecchannel.de/server/hardware/2023817/server_management_software_das_bieten_die_besten_verwaltungsuites/index6.html; abgerufen am 31.05.2013.
- Bayer, M. (2009): Hardware, Software und Prozesse: 11 Ratschläge, Storage billiger zu machen. In: CIP vom 04.05.2009. <http://www.cio.de/knowledgecenter/storage/887974>; abgerufen am 17.12.2012.
- Berleur, J., Hercheui, M., Hilty, L. M. (2010): What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience. IFIP Advances in Information and Communication Technology 328, Springer, Berlin Heidelberg New York 2010.
- BITKOM (2010a): Cloud Computing – Was Entscheider wissen müssen. Ein ganzheitlicher Blick über die Technik hinaus. Positionierung, Vertragsrecht, Datenschutz, Informationssicherheit, Compliance. Leitfaden. In Homepage BITKOM http://www.bitkom.org/de/publikationen/38337_66148.aspx; PDF: http://www.bitkom.org/files/documents/BITKOM_Leitfaden_Cloud_Computing-Was_Entscheider_wissen_muessen.pdf; abgerufen am 2.7.2013.
- BITKOM (2010b): Green-IT-Allianz, Ergebnisstand AG3 - Software und Green IT. Internes Arbeitspapier.
- BITKOM (2011a): Das mobile Internet boomt. Presseerklärung. http://www.bitkom.org/66885_66799.aspx; abgerufen am 31.05.2013.
- BITKOM (2011b): Zahl der App-Downloads explodiert. Presseerklärung. http://www.bitkom.org/de/presse/8477_66877.aspx; abgerufen am 31.05.2013.
- BITKOM (o. J.): Open Source Software: Rechtliche Grundlagen und Hinweise. Leitfaden (Version 1.0). http://www.bitkom.org/files/documents/bitkom_publication_oss_version_1.0.pdf; abgerufen am 29.6.2013.
- BMU (2012): Ressourcen schonende Speicherlösung für Rechenzentren. <http://www.bmu.de/bmu/presse-reden/pressemittelungen/pm/artikel/ressourcen-schonende-speicherloesung-fuer-rechenzentren/>; abgerufen am 14.01.2013.
- Bokhari, S. H., Rehman, R. (1999): Linux and the Developing World, IEEE Software 16(1), S. 58-64.
- Boyd, D. M., Ellison, N. B. (2007): Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), article 11. <http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html>; abgerufen am 31.05.2013.
- Bozzelli, P., Gu, Q., Lago, P. (2013): A systematic literature review on green software metrics. Technical report.
- BPMN (2013): Business Process Modelling and Notation. Homepage. <http://www.bpmn.org/>; abgerufen am 20.6.2013.
- Bräuner, H. (2005): Linux im Rathaus – Ein Migrationsprojekt der Stadt Schwäbisch Hall, in (Bärwolff et al., 2005), S. 37-50, <http://www.opensourcejahrbuch.de/download/jb2005/>; abgerufen am 30.6.2013.
- CEET (2013): The Power of Wireless Cloud. An Analysis of the Energy Consumption of Wireless Cloud. http://www.ceet.unimelb.edu.au/pdfs/ceet_white_paper_wireless_cloud.pdf; abgerufen am 31.05.2013.

Chen, F., Schneider, J.-G., Yang, Y., Grundy, J., He, Q. (2012): An Energy Consumption Model and Analysis Tool for Cloud Computing Environments. 1st ICSE Workshop on Green and Sustainable Software (GREENS 2012), Zurich, Switzerland, 3rd June 2012, S. 45-50.

Cisco (2011): Annual Security Report.

http://www.cisco.com/en/US/prod/collateral/vpndevc/security_annual_report_2011.pdf; abgerufen am 31.05.2013.

Cisco (2012a): Entering the Zettabyte Era. White Paper.

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf; abgerufen am 31.05.2013.

Cisco (2012b): Cisco Visual Networking Index: Forecast and Methodology, 2011–2016. White Paper.

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf; abgerufen am 31.06.2012.

Cisco (2013a): Cisco Visual Networking Index: Forecast and Methodology, 2012–2017, White Paper.

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf; abgerufen am 30.6.2013

Cisco (2013b): Annual Security Report. https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2013_ASR.pdf; abgerufen am 31.05.2013.

Cisco (2013c): Entering the Zettabyte Era. White Paper.

http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf .; abgerufen am 2.7.2013.

Coomonte, R., Lastres, C., Feijóo, C., Martín, Á. (2012): A simplified energy consumption model for fiber-based Next Generation Access Networks. Journal of Telematics and Informatics, Volume 29, Issue 4, S. 375-386.

Coroama, V., Hilty, L. M., Heiri, E., Horn, F. (2013): The Direct Energy Demand of Internet Data Flows. Journal of Industrial Ecology. DOI: 10.1111/jiec.12048

Craig-Wood, K., Krause, P. (2013): Toward the Estimation of the Energy Cost of Internet Mediated Transactions. Preliminary version of a technical report of the EEC (Energy Efficient Computing) SIG of the ICT KTN (Knowledge Transfer Network).

Creber, C. (2004): Die Bedeutung von Open Source in der Geschäftsstrategie von IBM, in (Picot und Doebelin, 2004), S. 111-116.

Deutscher IPTV-Verband (2010): <http://www.diptv.org/verband/definitionen-und-statements-zu-iptv-und-web-tv.html>; abgerufen am 31.05.2013.

Dick, M., Drangmeister, J., Kern, E., Naumann, S. (2013): Green Software Engineering with Agile Methods. 2nd ICSE Workshop on Green and Sustainable Software (GREENS 2013), San Francisco, CA, USA, May 20, 2013.

Dick, M., Kern, E., Drangmeister, J., Naumann, S., Johann, T. (2011): Measurement and Rating of Softwareinduced Energy Consumption of Desktop PCs and Servers. In Innovations in sharing environmental observations and information. Proceedings of the 25th International Conference EnviroInfo October 5 - 7, 2011, Ispra, Italy, W. Pillmann, S. Schade and P. Smits, Eds. Shaker, Aachen, S. 290–299.

Dick, M., Naumann, S. (2010a): Enhancing software engineering processes towards sustainable software product design, in: K. Greve, A.B. Cremers (Eds.), EnviroInfo 2010, Integration of Environmental Information in Europe, Proceedings of the 24th International Conference on Informatics for Environmental Protection, October 6–8, 2010, Cologne/Bonn, Germany. Shaker, Aachen, 2010, S. 706–715.

Dick, M., Naumann, S., Held, A. (2010b): Green Web Engineering. A Set of Principles to Support the Development and Operation of "Green" Websites and their Utilization during a Website's Life Cycle. In: Filipe, J., Cordeiro, J. (Hrsg.). WEBIST 2010 - Proceedings of the Sixth International Conference on Web Information Systems and Technologies, Volume 1, Valencia, Spain, April 07-10, 2010, 2 volumes, INSTICC Press, Setúbal, S. 48–55.

Dick, M., Naumann, S., Kuhn, N. (2010c): A Model and Selected Instances of Green and Sustainable Software. In Berleur et al. (2010) S. 248-259.

Diedrich, O. (4.2.2009): Trendstudie Open Source. Wie Open-Source-Software in Deutschland eingesetzt wird. In Heise Open Source. <http://heise.de/-221696> .; abgerufen am 29.6.2013.

- Drenkelfort, G., Pröhl, T., EreK, K. (2013): Energiemonitoring von IKT-Systemen. Kennzahlen. Projektberichte IKM| 3. Universitätsverlag der TU Berlin.
- EASED (2013): 2nd Workshop EASED@BUIS 2013 – Energy Aware Software-Engineering and Development – Proceedings. Bunse, C., Gottschalk, M., Naumann, S., Winter, A. (Hrsg.). Carl von Ossietzky Universität Oldenburg. OLNSE Number 4/2013.
- Ehmann, S., Hintemann, R. (2004): Leitfaden zum Thema Information Lifecycle Management. In Competence Site. <http://www.competence-site.de/location-intelligence-geomarketing-gis/Leitfaden-zum-Thema-Information-Lifecycle-Management>; abgerufen am 17.12.2012.
- EMC (2013): Deduplication Solutions. <http://germany.emc.com/backup-and-recovery/deduplication.html>; abgerufen am 10.05.2013.
- Emmenegger, M. F., Frischknecht, R., Stutz, M. (2006): Life Cycle Assessment of the mobile communication system UMTS – towards eco-efficient systems, International Journal of Life Cycle Assessment, 11, S.265-276.
- Enterprise Management Associates: 2006, EMA study: Get the Truth on Linux Management, Website, http://www.thalix.com/files/EMA_Levanta-Linux_RR.pdf; abgerufen am 1.7.2013.
- EreK, K., Drenkelfort, G., Pröhl, T. (2013): Energiemonitoring von IKT-Systemen. State-of-the-Art von Energiemonitoringsystemen. Projektberichte IKM| 2. Universitätsverlag der TU Berlin.
- Ernst&Young (2011): Open Source Software im geschäftskritischen Einsatz. [http://www.ey.com/Publication/vwLUAssets/Open_Source_Software_im_geschaeftskritischen_Einsatz/\\$FILE/Open_Sourc_e_Software_DE.pdf](http://www.ey.com/Publication/vwLUAssets/Open_Source_Software_im_geschaeftskritischen_Einsatz/$FILE/Open_Sourc_e_Software_DE.pdf) ; abgerufen am 29.6.2013.
- ESV (2013): Oberösterreichischer Energiesparverband. Homepage. <http://www.esv.or.at/startseite/>; abgerufen am 28.06.2013.
- Facebook (2013): Newsroom. Key Facts. Mai 2013. <http://newsroom.fb.com/Key-Facts>; abgerufen am 31.05.2013.
- Fletcher, O. (4. Juli, 2011): Alibaba aims to launch Operating System for mobile phones. In Wall Street Journal. <http://online.wsj.com/article/BT-CO-20110704-702552.html>; abgerufen am 17.12.2011.
- Forrester Consulting (November 2008): Open Source Paves The Way For The Next Generation Of Enterprise IT - A European Survey Finds That Open Source Becomes The Hidden Backbone Of The Software Industry, And Leads To A Paradigm Change In Enterprise IT. <http://www.bull.com/p/register.php?id=172> ; abgerufen am 29.6.2013.
- Fowler, G. A., Raice, S., Efrati, A. (2012): Spam Finds New Target. In The Wall Street Journal, 4. Januar 2012. <http://online.wsj.com/article/SB10001424052970203686204577112942734977800.html>; abgerufen am 31.05.2013.
- Fraunhofer ESK (2010): Low-power DSL. http://www.esk.fraunhofer.de/content/dam/esk/en/documents/PDB_L2_en.pdf; abgerufen am 28.06.2013.
- Gartner (4. Mai, 2013): Gartner Says Asia/Pacific Led Worldwide Mobile Phone Sales to Growth in First Quarter of 2013. Pressemitteilung. <http://www.gartner.com/newsroom/id/2482816>; abgerufen am 18.07.2013.
- Geer, D. (2008): Reducing the Storage Burden via Data Deduplication. In Computer, Volume 41, Issue 12, December 2008, S. 15-17.
- GHGP (2013a): GHG Protocol Product Life Cycle Accounting and Reporting Standard ICT Sector Guidance, 26. Januar 2013, <http://www.ghgprotocol.org/feature/ghg-protocol-product-life-cycle-accounting-and-reporting-standard-ict-sector-guidance>
- GHGP (2013b): GHG Protocol Product Life Cycle Accounting and Reporting Standard ICT Sector Guidance, Chapter 1, Introduction and General Principles, Draft v1.3, 26. Januar 2013, <http://www.ghgprotocol.org/files/ghgp/GHGP-ICT-Introduction-Chapter-v1-3-26-JAN-2013.pdf>
- GHGP (2013c): GHG Protocol Product Life Cycle Accounting and Reporting Standard ICT Sector Guidance, Chapter 7, Software, Draft v2.9, <http://www.ghgprotocol.org/files/ghgp/GHGP-ICT-Software-v2-9-26JAN2013.pdf>
- Google Blog (2012): Google+: Communities and photos. Official Blog. 6. Dezember 2012. <http://googleblog.blogspot.ch/2012/12/google-communities-and-photos.html>; abgerufen am 14.05.2013.

- Graf, A. (2010): Navigationsgeräte bekommen Konkurrenz vom Handy. In Handelsblatt. 4.9.2010. <http://www.handelsblatt.com/technologie/it-tk/mobile-welt/smartphones-navigationsgeraete-bekommen-konkurrenz-vom-handy/3531484.html>; abgerufen am 27.6.2013.
- Grosskop, K., Visser, J. (2013): Identification of Application-level Energy Optimizations. In Hilty et al. (2013), S. 101-107.
- He, Q., Li, Z., Zhang, X. (2010): Data deduplication techniques. In Future Information Technology and Management Engineering (FITME), 2010 International Conference on (Volume 1). Date of Conference: 9-10 Oct. 2010, S. 430-433.
- Heinrich, H., Holl, F.-L., Menzel, K., Mühlberg, J. T., Schäfer, T., Schüngel, H. (2006): Metastudie: Open-Source-Software und ihre Bedeutung für Innovatives Handeln. In Holl, F.-L. (Hrsg.) Entwicklungen in den Informations- und Kommunikationstechnologien. Band 1. http://www.bmbf.de/pubRD/oss_studie.pdf ; abgerufen am 28.6.2013.
- Heise Developer (2011): Windows-Azure-Toolkit für Social Games. In Heise Developer <http://heise.de/-1283544>; abgerufen am 31.05.2013.
- Heise Mobil (2011a): 3D-Smartphone von LG kommt im Juli. In Heise Mobil, 28. Juni 2011. <http://heise.de/-1269160> abgerufen am 31.05.2013.
- Heise Mobil (2011b): 3D-Android-Smartphone von HTC. In Heise Mobile, 27. Juni 2011. <http://heise.de/-1268542> ; abgerufen am 31.05.2013.
- Heise Mobil (2011c): Berliner Handyticket ist nicht mehr Beta. In Heise Mobil, 14. Juli 2011. <http://heise.de/-1279543> ; abgerufen am 31.05.2013.
- Heise Online (2008) Studie: Open Source in Unternehmen ebnet den Weg für Innovation, in Heise Online, 3.12.2008. <http://heise.de/-219195>; abgerufen am 29.6.2013
- Heise Online (2011a): Samsung-Fernseher streamen aufs Smartphone. In Heise Online, 12. Juli 2011. <http://heise.de/-1277857>; abgerufen am 31.05.2013.
- Heise Online (2011b): Nokia forciert ortsbezogene Anwendungen für Windows Phone 7. In Heise Online. <http://heise.de/-1272416>; abgerufen am 31.05.2013.
- Heise Online (2012): Studie: Open-Source-Software qualitativ besser als proprietäre Entwicklungen. In HeiseOnline. <http://www.heise.de/developer/meldung/Studie-Open-Source-Software-qualitativ-besser-als-proprietare-Entwicklungen-1440788.html>; abgerufen am 4.3. 2012.
- Heise Open Source (2008): Open Source ist überall. In Heise Online. <http://heise.de/-217214>; abgerufen am 29.6.2013.
- Heise Open Source (2012a): Immer mehr Linux auf dem Desktop. In HeiseOnline. <http://www.heise.de/open/meldung/Immer-mehr-Linux-auf-dem-Desktop-1404775.html>; abgerufen am 3.3.2012.
- Heise Open Source (2012b): Anteil der Freizeit-Kernel-Hacker sinkt. In Heise Online. <http://heise.de/-1500629>; abgerufen am 29.6.2013.
- Hilty, L. M. (2012): Why Energy Efficiency is not Sufficient: Some Remarks on "Green by IT". In Arndt, H. K., Pillmann, W. (Hrsg.). EnviroInfo 2012, Proceedings of the 26th Environmental Informatics Conference, Shaker, Aachen 2012, S. 13-20.
- Hilty, L. M., Aebischer, B., Andersson, G., Lohmann, W. (Hrsg.) (2013): ICT4S – ICT for Sustainability. Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, February 14-16, 2013. ETH. <http://e-collection.library.ethz.ch/eserv/eth:6558/eth-6558-01.pdf>; abgerufen am 31.05.2013.
- Hilty, L. M., Köhler, A., von Schéele, F., Zah, R., Ruddy, T. (2006): Rebound Effects of Progress in Information Technology. Poiesis & Praxis: International Journal of Technology Assessment and Ethics of Science, 1 (4), S. 19-38.
- Hilty, L.M., Lohmann, W. (2011): The Five Most Neglected Issues in "Green IT" In CEPIS UPGRADE 12: 4, S. 12-15.
- Hilty, L.M., Oertel, B., Wölk, M., Pärli, K. (2012): Lokalisiert und identifiziert. Wie Ortungstechnologien unser Leben verändern. TA-Swiss. ISBN 978-3-7281-3460-8.
- Hintemann, R., Fichter, K. (2012): Energieverbrauch und Energiekosten von Servern und Rechenzentren in Deutschland - Aktuelle Trends und Einsparpotenziale bis 2015, Berlin. <http://www.borderstep.de>; abgerufen am 31.05.2013.
- Hintemann, R., Fichter, K. (2013): Server und Rechenzentren in Deutschland im Jahr 2012, Berlin 2013. http://www.borderstep.de/pdf/Kurzbericht_Rechenzentren_in_Deutschland_2012_09_04_2013.pdf; abgerufen am 14.05.2013.

- Holstein, B. D. (2011): Speed delivery of Android devices and applications with model-driven development. In IBM developerWorks. Technical Library. <https://www.ibm.com/developerworks/rational/library/model-driven-development-speed-delivery>; abgerufen am 31.05.2013.
- ITU (2012a): Assessment Framework for Environmental Impacts of the ICT Sector (September 2012), http://www.itu.int/dms_pub/itu-t/oth/4B/04/T4B0400000B0008PDFE.pdf; abgerufen am 12. Mai 2013.
- ITU (2012b): Toolkit on environmental sustainability for the ICT sector (September 2012), http://www.itu.int/dms_pub/itu-t/oth/4B/01/T4B010000060001PDFE.pdf; abgerufen am 12. Mai 2013.
- ITU (2012c): L.1410: Methodology for the assessment of the environmental impact of information and communication technology goods, networks and services, <http://www.itu.int/rec/T-REC-L.1410-201203-I/en>, März 2012; abgerufen am 16. Mai 2013.
- Keller, P. (2010): Smart vending machine: Culture & Society, Interaction Design, 11 August 2010.
- Kern, E., Dick, M., Johann, T., Naumann, S. (2011): Green Software and Green IT: An End Users Perspective. In Golinska, P., Fertsch, M., Marx-Gomez J. (Hrsg.). Information Technologies in Environmental Engineering Environmental Science and Engineering. New Trends and Challenges. Berlin, Heidelberg: Springer, S. 199-211. <http://link.springer.com/book/10.1007/978-3-642-19536-5>
- Kern, E., Dick, M., Naumann, S., Guldner, A., Johann, T. (2013): Green Software and Green Software Engineering – Definitions, Measurements, and Quality Aspects. In: Hilty et al. (2013), S. 87-94.
- Klaß, C. (6. Juli, 2011): Facebook mit Videochat: Soziales Netzwerk integriert Skype. In Golem.de. <http://www.golem.de/1107/84768.html>; abgerufen am 31.05.2013.
- Kossel, A. (2013): Gutes Geleit: Navigations-Apps für Android, iOS und Windows Phone 8. In c't 13/2013, 3.6. 2013. http://www.heise.de/artikel-archiv/ct/2013/13/088_Gutes-Geleit; abgerufen am 27.6.2013.
- Lakshman, A., Prashant, M. (2010): Newsletter. ACM SIGOPS Operating Volume 44 Issue 2. April 2010. ACM New York, NY, USA. Doi 10.1145/1773912.1773922.
- Lanline (2012): Ganzheitliches IT- und Facility-Management optimiert Leistung im RZ. In Lanline (online) vom 25.10.2012. <http://www.lanline.de/fachartikel/ganzheitliches-it-und-facility-management-optimiert-leistung-im-rz.html>; abgerufen am 18.12.2012.
- LinkedIn Blog (2013): 200 Million Members. Official LinkedIn Blog. 9. Januar 2013. <http://blog.linkedin.com/2013/01/09/linkedin-200-million/>; abgerufen am 14.05.2013
- Lohr, S. (20. Juli, 2011): U.S. to Close 800 Computer Data Centers. In New York Times. <http://www.nytimes.com/2011/07/20/technology/us-to-close-800-computer-data-centers.html>; abgerufen am 31.05.2013.
- Lücke, H. (2011): Zattoo wächst auf 8 Millionen Nutzer. In onlinekosten.de, 14. Juni 2011. <http://www.onlinekosten.de/news/artikel/43971/0/Zattoo-waechst-auf-8-Millionen-Nutzer>; abgerufen am 31.05.2013.
- Malmodin, J. (2010): The Energy and Carbon Footprint of ICT and Media Services and Lessons Learned so far in the EARTH Project for Wireless Access Networks. Accessnets 2010. 5th International ICST Conference on Access Networks. <http://www.accessnets.org/2010/keynote.shtml>; abgerufen am 24.07.2013.
- Masanet, E., Shehabi, A., Ramakrishnan, L., Liang, J., Ma, X., Walker, B., Hendrix, V., Mantha, P. (2013): The Energy Efficiency Potential of Cloud-Based Software: A U.S. Case Study. Lawrence Berkeley National Laboratory, Berkeley, California. http://crd.lbl.gov/assets/pubs_presos/ACS/cloud_efficiency_study.pdf;; abgerufen am 20.8.2013
- McAfee (2009): Die CO2-Bilanz von Spam-E-Mails. https://www.info-point-security.com/open_downloads/2009/McAfee_CO2_Spam_BRD.pdf; abgerufen am 31.05.13.
- Mell, P., Grance, T. (2009): The NIST Definition of Cloud Computing. <http://www.slideshare.net/crossgov/nist-definition-of-cloud-computing-v15>; abgerufen am 31.05.2013.
- Meyer, D.T., Bolosky, W.J. (2012): A study of practical deduplication. In Transactions on Storage (TOS), Volume 7 Issue 4. January 2012, Article No. 14. http://delivery.acm.org/10.1145/2080000/2078864/a14-meyer.pdf?ip=130.60.155.223&acc=ACTIVE%20SERVICE&key=C2716FEBFA981EF1DD5891E815377FF4A1D6D263689CD484&CFID=219314061&CFTOKEN=74874759&_acm_=_1369323403_e879f7afe03eda700773f4fa34cf0fce; abgerufen am 31.05.13.

- Microsoft (2010): Energy Smart Software. http://www.microsoft.com/whdc/system/pnppwr/powermgmt/Energy-Smart_SW.mspix; abgerufen am 20.6.2013.
- Microsoft (28. März, 2011): Browser Power Consumption – Leading the Industry with Internet Explorer 9. <http://blogs.msdn.com/b/ie/archive/2011/03/28/browser-power-consumption-leading-the-industry-with-internet-explorer-9.aspx>; abgerufen am 28.6.2013.
- Müller, K. (2013): Persönliches Gespräch von R. Hintemann mit K. Müller zu Softwarelösungen im Rechenzentrum am 11.01.2013.
- Naumann, S. (2013): Telefonisches Gespräch von L. M. Hilty mit Stefan Naumann, Professor an der Hochschule Trier, am 13.05.2013.
- Naumann, S., Dick, M., Kern, E., Johann, T. (2011): The GREENSOFT Model: A Reference Model for Green and Sustainable Software and Its Engineering. In Sustainable Computing: Informatics and Systems 1 (2011), S. 294-304.
- Naumann, S., Gresk, S., Schäfer, K. (2008): How green is the web? Visualizing the power quality of websites, in: A. Möller, B. Page, M. Schreiber (Hrsg.), Environmental Informatics and Industrial Ecology, 22nd International Conference on Informatics for Environmental Protection, Envirolnfo 2008, Proceedings of the 22nd International Conference Environmental Informatics – Informatics for Environmental Protection, Sustainable Development and Risk Management, September 10–12, 2008, Leuphana University Lueneburg, Germany, Shaker, Aachen, 2008, S. 62–65.
- Naumann, S., Kern, E., Dick, M. (2013): Classifying Green Software Engineering - The GREENSOFT Model. In Bunse, C., Gottschalk, M., Naumann, S., Winter, A. (Hrsg.): Proceedings of the 2nd Workshop Energy Aware Software-Engineering and Development (EASED@BUIS). OLNSE Number 4/2013, S. 13-14. <http://www.se.uni-oldenburg.de/documents/olnse-4-2013-eased.pdf>; abgerufen am 31.05.2013.
- Nebel, W., Hoyer, M., Schröder, K., Schlitt, D. (2009): Untersuchung des Potentials von rechenzentrenübergreifendem Lastmanagement zur Reduzierung des Energieverbrauchs in der IKT, Studie für das Bundesministerium für Wirtschaft und Technologie, OFFIS, Dezember 2009.
- Nouredine, A., Bourdon, A., Rouvoy, R., Seinturier, L. (2012): A Preliminary Study of the Impact of Software Engineering on GreenIT. 1st ICSE Workshop on Green and Sustainable Software (GREENS 2012), Zurich, Switzerland, 3rd June 2012, S. 21-27.
- Paulson, J. W., Succu, G., Eberlein, A. (2004): An Empirical Study of Open-Source and Closed-Source Software Products, IEEE Transactions on Software Engineering 30(4), S. 246-256.
- Pelkmann, T. (2010): Deduplizierung - wie geht das? In Computerwoche online vom 29.11.2010. <http://www.computerwoche.de/a/deduplizierung-wie-geht-das,2358569>; abgerufen am 17.12.2012.
- Picot, A., Doeblin, S. (Hrsg.) (2004): Münchner Kreis: Open Source-Tagungsband, Bonn, Hüthig Telekommunikation.
- Pixable (2011): Facebook Photo Trends [INFOGRAPHIC]. In Pixable Blog, 14. Februar 2011. <http://classic.pixable.com/blog/2011/02/14/facebook-photo-trends-infographic/>; abgerufen am 31.05.2013.
- PowerTop (o. J.): <https://01.org/powerTop/>; abgerufen am 18.07.2013.
- Radicati Group, Inc. (2011): Email Statistics Report, 2011-2015. Editor: Sara Radicati, PhD, Principal Analyst: Quoc Hoang
- Rebitzer, G., Ekvall, T., Frischknecht, R., Hunkeler, D., Norris, G., Rydberg, T., Schmidt, W.-P., Suh, S., Weidema, B.P., Pennington, D.W. (2004): Life cycle assessment Part 1: Framework, goal and scope definition, inventory analysis, and applications. Environment International. 30(2004), S. 701-720.
- Reder, B. (2012): Data Center Infrastructure Management - Mit DCIM das RZ in den Griff bekommen. In Computerwoche online vom 08.10.2012. <http://www.computerwoche.de/a/mit-dcim-das-rz-in-den-griff-bekommen,2514482>; abgerufen am 20.12.2012.
- Renner, T., Vetter, M., Rex, S., Kett, H. (2005): Open Source Software. Einsatzpotenziale und Wirtschaftlichkeit. Eine Studie der Fraunhofer-Gesellschaft. Fraunhofer IRB Verlag, Stuttgart und Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart. http://wiki.iao.fraunhofer.de/index.php/Open_Source_Software; abgerufen am 28.6.2013.
- Rieß, U (2012): Effizienz ohne Deduplizierung? In speicherguide.de vom 01.06.2012. <http://www.speicherguide.de/backup-recovery/disk-backup/effizienz-ohne-deduplizierung-15576.aspx>; abgerufen am 17.12.2012.
- Roth, K., Patel, S., Perkinson, J. (2013): The Impact of Internet Browsers on Computer Energy Consumption. Final Report to Microsoft. Fraunhofer Center for Sustainable Energy Systems.

- Rüdiger, A. (2011): Storage-Kosten: Zehn Tipps zum Sparen beim Speichern. In Computerwoche online vom 26.03.2011. <http://www.computerwoche.de/a/zehn-tipps-zum-sparen-beim-speichern,1911740>; abgerufen am 17.12.2012.
- Sawall, A. (2011): Googles Handy-Bezahlsystem wird morgen vorgestellt. In Golem.de, 25. Mai 2011. <http://www.golem.de/1105/83720.html>; abgerufen am 31.05.2013.
- Scharnhorst, W., Hilty, L. M, Jolliet, O. (2006): Life cycle assessment of second generation (2G) and third generation (3G) mobile phone networks ENVIRONMENT INTERNATIONAL 32: 5. 656-675 JUL
- Schien, D., Preist, C., Shabajee, P., Yearworth, M. (2013): Modeling and Assessing Variability in Use Phase Energy of Digital Services. Journal of Industrial Ecology. In press.
- Sebayang, A. (2012): Facebook-Spammer nutzen Amazons Dienste. In Golem.de, 20. Januar 2012. <http://www.golem.de/1201/89416.html>; abgerufen am 31.05.2013.
- Simons, R. J. G., Pras, A. (2010). The Hidden Energy Cost of Web Advertising. Technical Report TR-CTIT-10-24, Centre for Telematics and Information Technology University of Twente, Enschede. ISSN 1381-3625. <http://eprints.eemcs.utwente.nl/18066/>; abgerufen am 31.05.2013.
- SourceForge (o.J.): <http://sourceforge.net/>; abgerufen am 18.07.2013.
- Statistic Brain (2013): Twitter Statistics. Research Date: 7. Mai 2013. <http://www.statisticbrain.com/twitter-statistics/>; abgerufen am 31.05.2013.
- Stephens, A., Didden, M. (2013): The Development of ICT Sector Guidance: Rationale, Development and Outcomes. In Hilty et al. (2013), S. 8-11.
- Stobbe, L., Nissen, N. F., Proske, M., Middendorf, A., Schlomann, B., Friedewald, M., Georgieff, P., Leimbach, T. (2009): Abschätzung des Energiebedarfs der weiteren Entwicklung der Informationsgesellschaft Bearbeitungsnummer I D 4 - 02 08 15 - 43/08, Abschlussbericht an das Bundesministerium für Wirtschaft und Technologie. Berlin: Fraunhofer IZM, 2009, 164 pp. urn:nbn:de:0011-n-1102312. <http://publica.fraunhofer.de/eprints/urn:nbn:de:0011-n-1102312.pdf>; abgerufen am 31.05.2013.
- Symantec Corporation (2011): April 2011 Intelligence Report.
- Thoma, J. (2012a): Ubuntu 12.04 mit aktivierter Energiesparoption. In Golem.de. <http://www.golem.de/news/sandy-bridge-ubuntu-12-04-mit-aktivierter-energiesparoption-1202-89883.html>; abgerufen am 3.3.2012.
- Thoma, J. (2012b): Stromsparen nur mit Handarbeit. In Golem.de. <http://www.golem.de/news/test-asus-zenbook-mit-linux-stromsparen-nur-mit-handarbeit-1202-89934.html>; abgerufen am 3.3.2012.
- UBA (2012): Glossar zum Ressourcenschutz. Umweltbundesamt. <http://www.umweltdaten.de/publikationen/fpdf-4242.pdf>; abgerufen am 31.05.2013.
- Unterseher, D. (2008): Data-Deduplication, Vortrag auf dem BITKOM-Arbeitskreis Speichertechnologien am 27.11.2008. http://www.bitkom.org/files/documents/Unterseher_Data_Deduplication_2008_11_27d.pdf; abgerufen am 16.12.2012.
- Van Bokhoven, F., Bloem, J. (2013): Pilot result Monitoring Energy usage by Software. In Hilty et al. (2013), S. 108-115.
- Vilsbeck, C. (2012): Virtualisierung, Cloud, Big Data, SSD: Storage-Trends: Speichertechnologien für 2012. In Tecchannel vom 30.01.2012. http://www.tecchannel.de/storage/management/2038683/storage_trends_speichertechnologien_2012_virtualisierung_cloud_big_data_ssd/; abgerufen am 31.05.2013.
- Wikipedia (2012a): Seite „Open Source“. In: Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Open_Source; abgerufen am 30. März 2012
- Wikipedia (2012b): Seite „Freie Software“. In: Wikipedia, Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Freie_Software; abgerufen am 30. März 2012
- Wikipedia (2013a): Seite „Software“. In: Wikipedia, Die freie Enzyklopädie. <http://de.wikipedia.org/w/index.php?title=Software>; abgerufen am 30. Juni 2013.
- Wikipedia (2013b): Seite "Open Source Software in öffentlichen Einrichtungen". In Wikipedia. http://de.wikipedia.org/wiki/Open-Source-Software_in_%C3%B6ffentlichen_Einrichtungen; abgerufen am 29.6.2013.

Wikipedia (2013c): Seite „Fahrzyklus“. In: Wikipedia, Die freie Enzyklopädie.
<http://de.wikipedia.org/wiki/Fahrzyklus>; abgerufen am 30. Juni 2013

Wilde, H. (2013): Persönliches Gespräch von R. Hintemann mit H. Wilde zu Deduplizierungslösungen im Rechenzentrum am 14.01.2013.

Wilke, C. (2012): Energy Labels for Mobile Applications. Fakultät Informatik, Institut für Software- und Multimediatechnik.
<http://www.claaswilke.de/publications/workshops/EEbS2012.pdf>; abgerufen am 31.05.2013.

Williams, D. R., Tang, Y. (2013): Impact of Office Productivity Cloud Computing on Energy Consumption and Greenhouse Gas Emissions. In *Environmental Science & Technology* 47 (9), S. 4333-4340.

Williams, D.R., Thomond, P., Mackenzie, I. (2013): The Greenhouse Gas Abatement Potential of Enterprise Cloud Computing. In Hilty et al. (2013), S. 21-28.

YouTube (o.J.): YouTube Statistik. http://www.youtube.com/t/press_statistics; abgerufen am 03.08.2011.

Zapico, J. L., Turpeinen, M., Brandt, N. (2010): Greenalytics: a tool for mash-up life cycle assessment of websites. In: *Proceedings of the 24th International Conference on Informatics for Environmental Protection (EnvirolInfo 2010)*. Cologne/Bonn, Germany, Shaker, Aachen, 2010, S. 754-763. ISBN: 978-3-8322-9458-8.