

TEXTE

157/2025

Umweltzeichen Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte

Hintergrundbericht zur Überarbeitung der
Vergabekriterien DE-UZ 215, Ausgabe Juni 2024

von:

Jens Gröger, Felix Behrens
Öko-Institut e.V., Berlin

Stefan Naumann, Kira Obergöker, Achim Guldner, Marcel Andres
Umwelt-Campus Birkenfeld / Hochschule Trier, Birkenfeld

Herausgeber:

Umweltbundesamt

TEXTE 157/2025

Ressortforschungsplan des Bundesministeriums für
Umwelt, Naturschutz und nukleare Sicherheit

Forschungskennzahl 3720 37 308 0
FB001866

Umweltzeichen Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte

Hintergrundbericht zur Überarbeitung der
Vergabekriterien DE-UZ 215, Ausgabe Juni 2024

von

Jens Gröger, Felix Behrens
Öko-Institut e.V., Berlin

Stefan Naumann, Kira Obergöker, Achim Guldner, Marcel
Andres
Umwelt-Campus Birkenfeld / Hochschule Trier,
Birkenfeld

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber

Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

Durchführung der Studie:

Öko-Institut e.V.
Merzhauser Str. 173
79100 Freiburg

Umweltcampus Birkenfeld / Hochschule Trier
Postfach 13 80
55761 Birkenfeld

Abschlussdatum:

Juni 2024

Redaktion:

Fachgebiet III 1.3 Ökodesign, Umweltkennzeichnung, umweltfreundliche Beschaffung
Anna Zagorski

DOI:

<https://doi.org/10.60810/openumwelt-7650>

ISSN 1862-4804

Dessau-Roßlau, Dezember 2025

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen und Autoren.

Kurzbeschreibung: Umweltzeichen Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte

Die Digitalisierung dringt ungebrems in alle Lebensbereiche vor. Nach einem weiteren Hype-Cycle in der Künstlichen Intelligenz um Transformer-Architekturen und durch diese getriebene generative KI-Anwendungen wie Chatbots, Bild- und Musikgeneratoren nehmen maschinell unterstützte Datenverarbeitungsprogramme Einzug in die Softwareentwicklung und digitale Infrastruktur. In der öffentlichen Debatte und in der Gesetzgebung sind die negativen Effekte auf die Umwelt der digitalen Infrastruktur und Rechenzentren bereits angekommen.

Das Bewusstsein über die Umweltwirkungen von Software führt noch nicht in erkennbare Änderungen der Produkte oder ihres Konsums. Dabei gibt es erhebliche Unterschiede zwischen den Umweltwirkungen von Softwareprodukten mit vergleichbaren Funktionen. Software ist also ein Treiber der Umweltwirkungen der Hardware. Denn Software gibt maßgeblich vor, wie viele Hardwarekapazitäten zum Einsatz kommen. Dadurch beeinflusst Software indirekt, wie viel Energie und natürliche Ressourcen von Informations- und Kommunikationstechnik insgesamt verbraucht werden.

Um den Blauen Engel für energie- und ressourceneffiziente Softwareprodukte an aktuelle Software-Trends anzupassen und die Anschlussfähigkeit und Verbreitung zu verbessern, soll der bestehende Geltungsbereich erweitert und die Methode zur Bestimmung der Energie- und Ressourceninanspruchnahme überarbeitet werden. Dieser Hintergrundbericht stellt die Überlegungen und dahinterstehenden Ansätze der Überarbeitung des Umweltzeichens DE-UZ 215 vor und leitet daraus überarbeitete Vergabekriterien ab.

Abstract: Blue Angel eco-label for resource- and energy-efficient software products

Digitalisation is steadily advancing into all areas of life. Following the recent artificial intelligence hype cycle surrounding Transformer architectures and generative AI applications driven by them, such as chatbots, image, and music generators, machine-assisted data processing programmes are finding their way into software and digital infrastructure. The negative effects on the environment of digital infrastructure and data centres have already become part of the public debate and legislation.

Awareness of the environmental impacts of software has not yet led to any recognisable changes in the products or their consumption. Yet there are significant differences between the environmental impacts of software products with comparable functions. Software is therefore a driver of the environmental impacts of hardware. This is because software largely determines how much hardware capacity is used. As a result, software indirectly influences how much energy and natural resources are consumed by information and communication technology overall.

In order to adapt the Blue Angel for energy- and resource-efficient software products to current software trends and to improve its connectivity and dissemination, the existing scope should be expanded and the method for determining energy and resource consumption should be revised. This background report presents the considerations and underlying approaches for the revision of the DE-UZ 215 eco-label and derives revised award criteria from this.

Inhaltsverzeichnis

Inhaltsverzeichnis	6
Abbildungsverzeichnis.....	8
Tabellenverzeichnis	8
Abkürzungsverzeichnis	9
Zusammenfassung.....	10
Summary	15
1 Hintergrund und Zielsetzung.....	20
2 Untersuchungsgegenstand.....	22
3 Markt- und Umfeldanalyse.....	23
3.1 Marktanalyse	23
3.2 Hemmnisse der Zeichenvergabe der ersten Version des Blauen Engels DE-UZ 215	25
3.2.1 Systemische Hemmnisse	25
3.2.2 Hemmnisse der Kriterien.....	26
3.3 Regulatives Umfeld	27
3.3.1 EU-Verordnung über digitale Inhalte	28
3.3.2 EU-Ökodesign Verordnungen.....	28
3.3.3 EU-Richtlinie über unlautere Geschäftspraktiken	29
3.3.4 EU-Gesetz über künstliche Intelligenz.....	29
3.3.5 Internationale Normen.....	30
4 Umweltwirkungen von Software.....	31
4.1 Identifikation relevanter Umweltwirkungen.....	31
4.1.1 Energiebedarf / Leistungsaufnahme	31
4.1.2 Hardwareinanspruchnahme.....	32
4.1.3 Auswirkungen auf die Hardware-Lebensdauer	34
4.1.4 Indirekte Umweltwirkungen durch Nutzungsautonomie	34
4.2 Handreichungen und Leitfäden zu ressourcen- und energieeffizienter Software.....	34
5 Verfahren und Werkzeuge zur Messung der Hardwareinanspruchnahme und des Energiebedarfes	36
5.1 Lasterzeugung während des Messvorgangs.....	37
5.2 Exemplarische Messaufbauten	39
5.3 Exemplarische Auswertungen von Messungen	40
6 Ableitung der Vergabekriterien für ein Umweltzeichen	47

6.1 Geltungsbereich	47
6.2 Ressourcen- und Energieeffizienz	48
6.2.1 Erforderliche minimale Systemvoraussetzungen.....	48
6.2.2 Dokumentation des Messaufbaus.....	49
6.2.3 Messung der Grundauslastung und der Last des Softwareprodukts im Leerlaufzustand	49
6.2.4 Messung der Energie- und Hardwareinanspruchnahme während der Nutzung	50
6.3 Potenzielle Hardware-Nutzungsdauer	50
6.3.1 Abwärtskompatibilität.....	50
6.4 Nutzungsautonomie	51
6.4.1 Datenformate	51
6.4.2 Transparenz des Softwareproduktes	51
6.4.3 Kontinuität des Softwareproduktes	51
6.4.4 Deinstallierbarkeit	52
6.4.5 Modularität	52
6.4.6 Werbefreiheit und Vermeidung von Tracking.....	52
6.4.7 Dokumentation des Softwareproduktes.....	53
6.4.8 Offlinefähigkeit.....	53
6.5 Anforderungen an Weiterentwicklung und Updates des Produktes	53
6.6 Empfohlene Veränderungen gegenüber der zuletzt gültigen Fassung des Umweltzeichens DE-UZ 215, Ausgabe Januar 2020 (Version 3).....	54
6.7 Ausblick	55
7 Quellenverzeichnis	57
A Beispielhafte Methoden zur Bestimmung der Energie- und Hardwareinanspruchnahme	63
A.1 Messmethode SERENA.....	63
A.1.1 Software Energy and Resource Efficiency Analysis (SERENA)	63
A.1.2 SERENA für Client-Server Systeme.....	65
A.2 Weitere Verfahren zur Messdatenerfassung.....	65
A.2.1 AppChecker (mobilsicher).....	66
A.2.2 Green Software Measurement Process - GSMP	66
A.2.3 Erfassung des Datenverkehrs mobiler Geräte	66
A.2.4 Green Metrics Tool	67
A.2.5 Logging der CPU Process ID Rechenzeit.....	67
B Anforderungen an Auditor*innen	69
B.1 Qualifizierung zur Auditor*in	69

B.2 Aufgaben	70
B.3 Pflichten	70
B.4 Arbeit als Auditor*in	71
B.5 Kontrolle.....	71

Abbildungsverzeichnis

Abbildung 1:	Exemplarischer Ablauf eines Szenario-Tests	39
Abbildung 2:	Exemplarischer Messaufbau zur Bestimmung der Hardwareinanspruchnahme und des Energiebedarfes.....	40
Abbildung 3:	Leistungsaufnahme eines Client-Server-Softwareprodukts (Server-System)	42
Abbildung 4:	Leistungsaufnahme eines Client-Server-Softwareprodukts (Client-System)	43
Abbildung 5:	Prozessorauslastung eines Client-Server-Softwareprodukts (Server-System)	43
Abbildung 6:	Arbeitsspeicherauslastung eines Client-Server-Softwareprodukts (Client-System)	44
Abbildung 7:	Gesendeter Netzwerkverkehr eines Client-Server-Softwareprodukts (Client-System)	44
Abbildung 8:	Empfangener Netzwerkverkehr eines Client-Server-Softwareprodukts (Server-System)	45
Abbildung 9:	Leistungsaufnahme eines Softwareprodukts im 6-minütigen Messvorgang	45
Abbildung 10:	Prozessorauslastung eines Softwareprodukts im 6-minütigen Messvorgang	46
Abbildung 11:	Schematischer Messaufbau Client-Messung bei SERENA	64
Abbildung 12:	Schematischer Messaufbau zu Messung des Datenverkehrs ..	67

Tabellenverzeichnis

Tabelle 1:	Exemplarische Auswertungsübersicht eines Client-Server-Softwareprodukts im 8-minütigen Messvorgang des Servers ..	41
Tabelle 2:	Exemplarische Auswertungsübersicht eines Client-Server-Softwareprodukts im 8-minütigen Messvorgang des Clients ..	42
Tabelle 3:	Aufgerufene IP-Adressen und URLs eines Softwareprodukts im 5-minütigen Messvorgang	46

Abkürzungsverzeichnis

API	Application Programming Interface (Softwareschnittstelle, welche einer anderen Software die Nutzung einer bestimmten Funktionalität ermöglicht)
BE	Blauer Engel
CPU	Central Processing Unit (zentrale Recheneinheit)
DUT	Device under Test, dt. Hardware deren Ressourceninanspruchnahme gemessen wird. Für PC und Server ein physisches Gerät oder ein virtualisierter Softwarestack in einem Container. Für mobile device ein physisches, mobiles Endgerät.
FOSS	Free and Open-Source Software
HDD	Hard Drive Disk (rotierender Permanentspeicher)
IKT	Informations- und Kommunikationstechnik
IT	Informationstechnik
ISO	Internationale Organisation für Normung
KI	Künstliche Intelligenz
PKR	Produktkategorieregel
RAM	Random Access Memory (flüchtiger Arbeitsspeicher)
SERENA	Software Energy and Resource Efficiency Analysis (Prozess zur Messung und Analyse der Energie- und Hardwarenutzung von Software)
SNS	Standardnutzungsszenario
SSD	Solid State Drive (Permanentspeicher basierend auf Festkörpertechnologien)
SUT	System under test - Hardware, deren Ressourceninanspruchnahme gemessen wird
VRAM	Video Random Access Memory (flüchtiger Arbeitsspeicher der Grafikkarte)

Zusammenfassung

Dieser Bericht begründet die Überarbeitung der Vergabekriterien des Umweltzeichens Blauer Engel für Softwareprodukte. Aufgrund der Natur des Untersuchungsgegenstands der Produktgruppe „Ressourceneffiziente Software“, ist eine andere Vorgehensweise als bei den klassischen materiellen Produktgruppen notwendig. Hierzu wurden Literaturrecherchen, Interviews mit Expert*innen, eigene Experimente und Messungen sowie ein Beteiligungsverfahren mit drei Workshops durchgeführt, die im Ergebnis zu Überarbeitungsempfehlungen des bisherigen Blauen Engels für Softwareprodukte geführt haben.

Software ist ein **zentraler Baustein der Digitalisierung**, der Hardware und Daten zu einer produktiven und zweckdienlichen Weise in Synthese bringt. „Durch Geschäftsmodelle, die Software und Dienstleistungen „in der Cloud“ anbieten, werden sowohl Rechenleistung als auch Energiebedarf in Zukunft immer stärker auf Server in Rechenzentren verlagert.“ (Naumann et al. 2020). „Software ist ein maßgeblicher Treiber des **Energie- und Ressourcenbedarfs** von digitalen Infrastrukturen. Ihre Nutzung ist die Ursache dafür, dass digitale Endgeräte angeschafft, Rechenzentrumskapazitäten immer weiter ausgebaut und Kommunikationsnetzwerke immer leistungstärker werden müssen. Seit dem Jahr 2020 haben die Treibhausgasemissionen der Digitaltechnik in Deutschland nach einer aktuellen Berechnung des Fraunhofer IZM (Stobbe und Nissen 2023) kontinuierlich zugenommen. Von ursprünglich rund 20 Mio. Tonnen CO₂-Äquivalenten im Jahr 2020 (3 % der nationalen Emissionen) wurden diese bis zum Jahr 2023 auf rund 24 Mio. Tonnen gesteigert. Eine weitere Zunahme bis zu 30 Mio. Tonnen im Jahr 2030 wird prognostiziert. Steigende Hardware-Effizienz führt offensichtlich nicht dazu, dass Software sparsamer wird, sondern geht mit einem kontinuierlich steigenden Energie- und Hardwarebedarf einher. Außerdem führt immer anspruchsvollere Software dazu, dass funktionierende Hardware obsolet werden kann und frühzeitig ersetzt werden muss.“ (Blauer Engel 2024)

Seit 2020 existiert der **Blaue Engel für Ressourcen- und energieeffiziente Softwareprodukte DE-UZ 215**. Das Umweltzeichen legt derzeit seinen Schwerpunkt auf die Transparenz von Softwareprodukten und weniger auf die Einhaltung von Grenzwerten. Im Februar 2022 wurde das Umweltzeichen für Software erstmalig an den Dokumentenbetrachter „Okular“ verliehen (Umweltbundesamt 2020).

Die hier dokumentierte Überarbeitung des Umweltzeichens greift die unterschiedlichen Initiativen und Messmethoden zur Bestimmung des Ressourcen- und Energiebedarfs auf und integriert die neuen Möglichkeiten als Optionen zur Nachweisführung, sofern diese wissenschaftlichen Mindestanforderungen entsprechen. Dadurch bleibt das Umweltzeichen anschlussfähig an die laufende Diskussion innerhalb der Software-Community und verknüpft diese mit einer richtungssicheren und qualitativ hochwertigen Umweltaussage.

Mit der **Erweiterung des Geltungsbereichs** auf Softwareprodukte, die auf Servern, Desktop-Computern und mobilen Endgeräten lauffähig sind, kommen weitere Komplexitätsstufen hinzu, da die Referenzsysteme nicht mehr festgelegt werden können, der Datenverkehr berücksichtigt werden muss und eine Verschiebung der Rechenlast zwischen unterschiedlichen Computersystemen ermöglicht wird. Um die wichtigsten Auswirkungen des softwaregetriebenen Energie- und Ressourcenbedarfs zu erfassen, wurde daher die Methodik zur Bewertung der Ressourceneffizienz von Softwareprodukten weiterentwickelt und für weitere Software-Anwendungen anwendbar gemacht.

Software ist ohne die darunterliegende Hardware nicht funktionstüchtig. Je nach **Gerätetyp** unterscheiden sich wesentliche Parameter, z.B. wie viele Hardwarekapazitäten eine Software in Anspruch nehmen kann, ihre Konnektivität, oder verfügbare Sensoren und Aktoren. Zur Kategorisierung der Software im Untersuchungsbereich des Blauen Engels werden drei Gerätetypen unterschieden (Blauer Engel 2024):

- Personal Computer zum überwiegend stationären Einsatz wie beispielsweise Desktop-PC, Workstation, Laptop oder Convertible jeweils mit einem Desktop-Betriebssystem (Windows, MacOS, Linux oder vergleichbar), nachfolgend bezeichnet als „**pc device**“.
- Batteriebetriebenes tragbares Endgerät wie beispielsweise Smartphone, Tablet, Smartwatch mit einem mobilen Betriebssystem (Android, iOS oder vergleichbar), nachfolgend bezeichnet als „**mobile device**“.
- Server-Hardware zur zentralen Datenverarbeitung, betrieben in der Regel in einem Rechenzentrum oder Serverraum, nachfolgend bezeichnet als „**server device**“. Der Server verfügt über eine Netzwerkverbindung und kann mit mehreren Clients gleichzeitig Daten austauschen. Ein Server verfügt in der Regel nicht direkt über Eingabe- oder Ausgabegeräte (Human Interface Devices). Die Kommunikation mit dem Server erfolgt stattdessen softwaregetrieben über die Netzwerkschnittstelle.

Der **Markt von Software** ist stark globalisiert und fragmentiert. Die gesamte digitale Lieferkette ist geprägt von Oligopolen. Gleichzeitig gibt es einen großen Markt an sehr kleinen und spezialisierten Softwareentwicklern und eine gut vernetzte Open Source Szene. Der Gesamtumsatz im Jahr 2022 wird weltweit auf 605,5 Mrd. € beziffert mit einem jährlichen Wachstum von 5,3 %. Gut 4 % des Umsatzes (24,5 Mrd. €) wird in Deutschland erzielt (Statista 2024e). Beispielsweise erzielte SAP im Jahr 2023 einen globalen Umsatz von 31,2 Mrd. € (Statista 2024d) und das Software-Beratungsunternehmen Capgemini¹ einen Umsatz von 22 Mrd. € im Jahr 2022. Seit 2010 ist das Umweltzeichen „Blauer Engel“ Bestandteil der Nachhaltigen Beschaffung, seit 2021 explizit der IT-Beschaffung des Bundes (IT-Beauftragter des Bundes 2021). IT-Dienstleistungen und IT Technik machten mit mehr als 11 Milliarden Euro im Jahr 2023 über 90 Prozent der Ausgaben des Beschaffungsamtes des Bundes aus (Beschaffungsamt des BMI 2024).

Die **Hemmnisse**, den Blauen Engel für Software zu beantragen, wurden mithilfe von Interviews ergründet. Es wurden sowohl systemische Hemmnisse benannt als auch solche, die in den Vergabekriterien begründet sind. Ein weit verbreitetes Paradigma bewertet die Zeit der Softwareentwickler*innen höher als die Rechendauer. Dies befördert im Zweifel ineffizienten Code. Das wichtigste Hemmnis der Kriterien ist die Beschränkung des Geltungsbereiches auf lokale Anwendungen. Diskussionsbedarf und Veränderungsbedarf wurde vor allem bei den Kriterien der Abwärtskompatibilität, Offlinefähigkeit, Modularität, Offenen Datenformate und (Sicherheits-)Updates gesehen.

Auf europäischer Ebene ist ein Trend zu erkennen, die Umweltwirkungen von Software im Sinne des Verbraucherschutzes zu regulieren und transparent zu machen. Die EU-Verordnung über digitale Inhalte ((EU) 2019/770, (Europäisches Parlament und Rat 2019), die aktuellen Entwürfe der EU-Ökodesign Verordnungen (Europäische Kommission 2019; 2023) und die EU-Richtlinie über unlautere Geschäftspraktiken /(EU) 2024/825, (Europäisches Parlament und Rat

¹ <https://www.capgemini.com/news/press-releases/capgemini-delivers-another-record-performance-in-2022/> (Capgemini (2023).

2024a) adressieren Updates, deren Verfügbarkeit und ihren Einfluss auf die Leistung des Produktes. Das EU-Gesetz über künstliche Intelligenz ((EU) 2024/1689, (Europäisches Parlament und Rat 2024b) verlangt, den Energiebedarf für das Trainieren von KI-Modellen zu dokumentieren. Neben den **gesetzlichen Anforderungen** gibt es darüber hinaus Internationale Normen (ITU 2014), die über einen ökobilanziellen Ansatz die Umweltwirkungen von digitalen Dienstleistungen bewerten.

Hinsichtlich der **Umweltwirkungen** werden die direkten Eigenschaften des Softwareprodukts als solches berücksichtigt (sog. “Green in Software”-Aspekte), und nicht Effekte, die durch die Nutzung des Softwareprodukts ausgelöst werden (nicht “Green by Software”, indirekte Wirkungen bzw. Effekte zweiter Ordnung wie bspw. Energieeinsparungen durch die Anwendung von Software) (Naumann et al. 2020).

„Zwei Einflüsse, die durch den Einsatz eines Softwareprodukts verursacht werden, sind wesentlich für die Bestimmung der Umweltwirkungen:

- der Energiefluss durch die Herstellung, Nutzung und Entsorgung der Hardware, auf der die Software läuft (Elektrische Energie wird zu Abwärme),
- der Materialfluss der Hardware durch die Organisation, die sie verwendet (neue Hardware wird zu Elektronikschrott).“ (Naumann et al. 2020)

Somit hat Software eine direkte Umweltwirkung auf den Energiebedarf und eine indirekte auf die Lebensdauer der Hardware (Gröger et al. 2018).

Software ist nicht nur für die Beanspruchung der Hardware, sondern auch für deren Funktionstüchtigkeit und (Un-)Brauchbarkeit verantwortlich. In beiden Aspekten können sich Softwareprodukte unterscheiden. Hardware und Software stehen in wechselseitiger Beziehung, deshalb sind auch ihre Umweltwirkungen eng miteinander verknüpft. Software nimmt während der Ausführung unterschiedliche Hardwarekomponenten wie Prozessor, Arbeits- und Permanentspeicher, Netzwerkkarten, Router, Grafikkarten, etc. in Anspruch, die wiederum im Betrieb elektrische Energie in Wärme umwandeln. Im Regelfall gilt, je passender die Hardware für die jeweilige Software-Last dimensioniert ist, desto effizienter. Beispielsweise ist die Rechenfähigkeit eines Laptops zu gering, um ein Sprachmodell wie GPT zu trainieren. Für Machine-Learning werden stattdessen Hochleistungscomputer mit dedizierter Hardware eingesetzt. Der gleiche Laptop ist dagegen zu leistungstark, um ein Heizkörperventil zu steuern. Stattdessen werden in Heizkörperregelungen energiesparende Mikroprozessoren eingesetzt.

Die Umweltbelastungen der Hardware werden von mehreren Softwareeigenschaften beeinflusst. Dazu gehören neben dem eigentlichen Energiebedarf während der Ausführung die Abwärtskompatibilität zu existierender Hardware, Datenformate, die Kontinuität der Software (Updates), Deinstallierbarkeit, Modularität und Dokumentation der Software (Naumann et al. 2020). Auch unerwünschte Werbung erzeugt einen vermeidbaren Datenverkehr (Hilty et al. 2017).

Der **Energiebedarf** ist das Integral der Leistungsaufnahme (Watt) über die Zeit (Sekunden). Je nach Computer-Plattform (pc device, server device, mobile device) variiert dabei die Höhe der Leistungsaufnahme. Die in diesem Vorhaben untersuchten Desktop- und Laptop-Computer (pc devices) wiesen eine Leistungsaufnahme zwischen 10 und ca. 50 Watt auf, wobei es sich um minimal ausgestattete PCs handelte. Gaming-PCs benötigen oft mehrere 100 Watt. Server sind noch leistungstärkere Geräte, die für den Einsatz in Rechenzentren bestimmt sind und deren Leistungsaufnahme mehrere 1.000 Watt erreichen kann. Zu der Unterschiedlichkeit der

Hardware kommt hinzu, dass unterschiedliche Softwareprodukte auf derselben Hardware in der Regel für dieselbe Aufgabe unterschiedlich viel Energie, Hardwarekapazitäten und damit natürliche Ressourcen benötigen, je nachdem wie effizient sie entwickelt und aufgebaut sind. So konnte in einem Projekt von Gröger et al. (2018) gezeigt werden, dass beim Vergleich zweier Textverarbeitungsprogramme bei der Erledigung der gleichen Aufgaben, eines der Softwareprodukte viermal so viel Energie benötigte wie das andere.

Die **Hardwareinanspruchnahme** ist das Integral der Auslastung der Hardwarekomponenten über die Zeit. Die Auslastungen der Komponenten sind dabei die Auslastung der CPU in %, die gespeicherte Datenmenge auf dem Permanent- und Arbeitsspeicher in GB und die Bandbreite bzw. Datenrate der Datenübertragung in Mbit/s. Da die Messung meist mit konstanten Zeitabständen, z.B. jede Sekunde, Messwerte liefert, lässt sich das Integral auch als Mittelwert der Messwerte multipliziert mit der Messdauer schreiben:

► Prozessorarbeit [Einheit: s*%] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} \text{Auslastung}(t)$$

wobei N die Anzahl an Messungen, Auslastung(t) die prozentuale Auslastung zum Zeitpunkt t und T die Messdauer in Sekunden sind.

► Permanent- und Arbeitsspeicherarbeit [Einheit: GB*s] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} \text{gespeicherte Datenmenge}(t)$$

wobei die gespeicherte Datenmenge(t) ebenso als Auslastung(t) [%] * Speichervolumen geschrieben werden kann.

► Netzwerkarbeit [Einheit: Mbit] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} \text{Datenrate}(t)$$

wobei die Datenrate(t) ebenso als Auslastung(t) [%] * Bandbreite geschrieben werden kann.

Eine hohe Hardwareinanspruchnahme bewirkt einerseits eine höhere Leistungsaufnahme der Hardware und damit einen höheren Energiebedarf. Andererseits kann sie dazu führen, dass die Hardware durch neue, leistungsstärkere Hardware ersetzt werden muss, obwohl die alte Hardware noch funktionsfähig ist. Durch die Verkürzung der Lebensdauer der Hardware verursacht Software daher auch einen Rohstoffbedarf und die Notwendigkeit der Entsorgung der alten Hardware.

Diverse Akteure in Wirtschaft (SDIA 2024; Bitkom e.V. 2021) und Wissenschaft (Calero und Piattini 2015; Hindle 2016; Hilty et al. 2017; Bonamy et al. 2023) veröffentlichen Best Practice Guidelines und ähnliche Dokumente, die dabei unterstützen sollen, die Ressourcen- und Energieeffizienz von Software zu verbessern.

Zum jetzigen Zeitpunkt existieren noch keine standardisierten oder normierten Verfahren zur messtechnischen Erfassung der Energie- und Ressourceneffizienz von Softwareprodukten. Es existieren jedoch bereits zahlreiche Vorgehensvorschläge zur Messung der Energie- und Hardwareinanspruchnahme durch Software. Gleichzeitig nimmt die Anzahl an verfügbaren Eigenentwicklungen und Beiträgen zu Messmethoden verschiedener Akteure aus Wirtschaft und Wissenschaft weiterhin zu. Dabei wird häufig zwischen softwarebasierten Messverfahren und direkter Messung der Hardwarekapazitäten unterschieden (Jay et al. 2023; Guldner et al. 2024). Anhang A gibt einen Überblick über einige der den Autor*innen bekannte Verfahren.

Typischerweise werden bei den Messverfahren die für die Energie- und Hardwarenutzung des Gerätes die Inanspruchnahme der relevanten Komponenten wie CPU, GPU, Arbeits- und Permanentspeicher, Datenverkehr und teilweise auch die elektrische Leistungsaufnahme des Gesamtsystems oder einzelner Komponenten gemessen. Dabei sind verschiedene Messverfahren jeweils für verschiedene Hardware-Software-Konstellationen ausgelegt und optimiert. Einige Messgrößen sind auf manchen Hardwareplattformen kaum zugänglich oder die Aussagekraft der Messung ist beschränkt. Dies betrifft insbesondere den Energiebedarf von mobilen Endgeräten. Hinzu kommt, dass die Umweltwirkungen bei mobilen Endgeräten eher vom Herstellungsaufwand der Hardware dominiert werden, bei Servern hingegen deutlich vom Energiebedarf in der Nutzungsphase.

Die Bandbreite an verfügbaren Softwareprodukten, die auf unterschiedlichen Computerplattformen lauffähig sind, erfordert unterschiedliche Messaufbauten. Dies ist bedingt durch die Vielfalt von Software-Architekturen, Hardwaresystemen und Anwendungsszenarien. Verschiedene Methoden können teilweise in einem Messaufbau kombiniert werden oder sich in unterschiedlichen Messaufbauten ergänzen.

Aus dieser Fülle schöpfend, wird für die Überarbeitung der Vergabekriterien empfohlen, das Konzept des Referenzsystems (vgl. (Umweltbundesamt 2020)) zu überarbeiten, für eine Vielzahl von Messmethoden und Messtechnik zu öffnen und somit an oben beschriebenen Gegebenheiten anzupassen.

Darüber hinaus sollen die Anforderungen an die Messung den messtechnischen Möglichkeiten und unterschiedlichen Umweltwirkungen der drei Hardware-Plattformen mobile Endgeräte (mobile device), Arbeitsplatz-Computer (pc device) und Server (server device) gerecht werden (siehe Abschnitt 6.2).

Um die Qualität und die Aussagekraft der Messung der Ressourcen- und Energieeffizienz sicherzustellen, wird empfohlen, Anforderungen an die Messmethode zu stellen und deren Einhaltung durch Auditor*innen überprüfen zu lassen.

Summary

This report documents the revision of the award criteria for the Blue Angel environmental label for software products. Due to the nature of the subject of the product group 'Resource-efficient software', a different approach is necessary than for traditional material product groups. To this end, literature research, interviews with experts, our own experiments and measurements, and a participatory process involving three workshops were carried out, which resulted in recommendations for the revision of the existing Blue Angel criteria for software products.

Software is a **central building block of digitalisation**, combining hardware and data in a productive and purposeful way. In the future, business models that offer software and services 'in the cloud' will increasingly shift both computing power and energy requirements to servers in data centres. Software is a key driver of the **energy and resource requirements** of digital infrastructures. Its use is the reason why digital end devices have to be purchased, data centre capacities have to be continuously expanded and communication networks have to become more and more powerful. According to a recent calculation by the Fraunhofer IZM (Stobbe und Nissen 2023), greenhouse gas emissions from digital technology in Germany have increased continuously since 2020. From around 20 million tonnes of CO₂ equivalents in 2020 (3% of national emissions), they increased to around 24 million tonnes by 2023. A further increase to 30 million tonnes by 2030 is predicted. Increasing hardware efficiency obviously does not lead to software becoming more economical but is accompanied by a continuously increasing demand for energy and hardware. In addition, increasingly sophisticated software can make functioning hardware obsolete, requiring it to be replaced prematurely.

The **Blue Angel for resource- and energy-efficient software products DE-UZ 215** has existed since 2020. The environmental label currently focuses more on the transparency of software products and less on compliance with limit values. In February 2022, the environmental label for software was awarded for the first time to the 'Okular' document viewer (Umweltbundesamt 2020).

The revision of the environmental label documented here takes up the various initiatives and measurement methods for determining resource and energy requirements and integrates the new possibilities as options for verification, provided that they meet the minimum scientific requirements. This allows the environmental label to remain compatible with the ongoing discussion within the software community and links it to a reliable and high-quality environmental claim.

With the **extension of the scope** to software products that can run on servers, desktop computers and mobile devices, further levels of complexity are added, since the reference systems can no longer be defined, data traffic must be taken into account and a shift in the computing load between different computer systems is made possible. In order to capture the most important effects of software-driven energy and resource requirements, the methodology for evaluating the resource efficiency of software products was further developed and made applicable to other software applications.

Software cannot function without the underlying hardware. Depending on the **device type**, essential parameters differ, e.g., how much hardware capacity a software can utilise, its connectivity, or available sensors and actuators. To categorise the software in the Blue Angel's area of investigation, three device types are distinguished:

- Personal computers for predominantly stationary use, such as desktop PCs, workstations, laptops or convertibles, each with a desktop operating system (Windows, MacOS, Linux or comparable), hereinafter referred to as '**pc device**'.
- Battery-powered portable devices such as smartphones, tablets or smartwatches with a mobile operating system (Android, iOS or comparable), hereinafter referred to as '**mobile device**'.
- Server hardware for centralised data processing, usually operated in a data centre or server room, hereinafter referred to as a '**server device**'. The server has a network connection and can exchange data with multiple clients simultaneously. A server usually does not have direct access to input or output devices (human interface devices). Instead, communication with the server is software-driven via the network interface.

The **market for software** is highly globalised and fragmented. The entire digital supply chain is characterised by oligopolies. At the same time, there is a large market of very small and specialised software developers and a well-connected open-source scene. Total global sales in 2022 are estimated at €605.5 billion, with an annual growth rate of 5.3%. A good 4% of sales (€24.5 billion) will be generated in Germany (Statista 2024e). For example, SAP generated global revenue of €31.2 billion in 2023 (Statista 2024d) and the software consultancy Capgemini² generated revenue of €22 billion in 2022. The Blue Angel ecolabel has been part of sustainable procurement since 2010, and since 2021 it has explicitly been part of the federal government's IT procurement (IT-Beauftragter des Bundes 2021). IT services and IT technology together account for more than half of the more than €5 billion in expenditures of the Federal Procurement Office in 2019 (Beschaffungssamt des BMI 2024)..

The **barriers to** applying for the Blue Angel for software were explored with the help of interviews. Both systemic barriers and those rooted in the award criteria were identified. The main reason for the low number of applicants for the environmental label is the low level of awareness of both the concepts of green software and the Blue Angel for software. A widespread paradigm values software developers' time higher than computing time. In case of doubt, this promotes inefficient code. The most important barrier to the criteria is the limitation of the scope to local applications. There is a need for discussion and change, especially with regard to the criteria of downward compatibility, offline capability, modularity, open data formats and (security) updates.

At the European level, a trend can be seen towards regulating and making transparent the environmental impacts of software in the interest of consumer protection. The EU Directive on certain aspects concerning contracts for the supply of digital content and digital services ((EU) 2023/1670, (Europäisches Parlament und Rat 2019), the EU Ecodesign Regulations (Europäische Kommission 2019; 2023) and the EU Unfair Commercial Practices Directive ((EU) 2024/825, (Europäisches Parlament und Rat 2024a) address updates, their availability and their influence on the performance of the product. The EU Artificial Intelligence Act ((EU) 2024/1689, (Europäisches Parlament und Rat 2024b) requires the energy required for training AI models to be documented. In addition to the legal requirements, there is also an international standard (ITU 2014), that uses a life cycle assessment approach to evaluate the environmental impacts of digital services.

² <https://www.capgemini.com/news/press-releases/capgemini-delivers-another-record-performance-in-2022/> (Capgemini (2023).

With regard to **environmental impacts**, the direct properties of the software product as such are considered (so-called ‘green in software’ aspects), and not effects that are triggered by the use of the software product (not ‘green by software’, indirect effects or second-order effects such as energy savings through the application of software) (Naumann et al. 2020).

Two influences caused by the use of a software product are essential for determining the environmental impacts:

- ▶ the energy flow through the production, use and disposal of the hardware on which the software runs (electrical energy becomes waste heat),
- ▶ the material flow of the hardware through the organisation that uses it (new hardware becomes electronic waste).

Thus, software has a direct environmental impact on energy consumption and an indirect impact on the lifespan of hardware (Gröger et al. 2018).

Software is not only responsible for the demands placed on hardware, but also for its functionality and (in)usability. Software products can differ in both aspects. Hardware and software are interrelated, which is why their environmental impacts are also closely linked. During execution, software uses various hardware components such as processors, main memory and permanent storage, network cards, routers, graphics cards, etc., which in turn convert electrical energy into heat during operation. As a rule, the more appropriately the hardware is dimensioned for the respective software load, the more efficient it is. For example, a laptop's computing capacity is too low to train a language model like GPT. Instead, high-performance computers with dedicated hardware are used for machine learning. By contrast, the same laptop is too powerful to control a radiator valve. Instead, energy-saving microprocessors are used in radiator control systems.

The environmental impact of hardware is influenced by several software properties. In addition to the actual energy demand during execution, these include backwards compatibility with existing hardware, data formats, software continuity (updates), uninstallability, modularity and software documentation (Naumann et al. 2020). Undesired advertising also generates avoidable data traffic (Hilty et al. 2017).

The **energy demand** is the integral of the power consumption (watts) over time (seconds). The level of power consumption varies depending on the computer platform (pc device, server device, mobile device). The desktop and laptop computers (pc devices) examined in this preliminary study had a power consumption of between 10 and approx. 50watts, whereby these were PCs with minimal equipment. Gaming PCs often require several 100 watts. Servers are even more powerful devices that are intended for use in data centres and whose power consumption can reach several 1,000 watts. In addition to the differences in hardware, different software products on the same hardware generally require different amounts of energy, hardware capacities and thus natural resources for the same task, depending on how efficiently they are developed and constructed. In a project of Gröger et al. (2018), it was shown that when comparing two text processing programs performing the same tasks, one of the software products required four times as much energy as the other.

The **hardware utilisation** is the integral of the utilisation of the hardware components over time, measured in %*s for processors, GB*s for permanent and main memory and Mbit/s*s (= Mbit) for the data transfer. The measurement is done at a constant time interval e.g., every

second. Thus, the integral can also be written as the mean of the measured value times the measurement time:

► processor work [s*%] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} utilisation(t)$$

with N the number of measurements, utilisation(t) the utilisation in percent at the time t and the measurement duration T.

► Storage and memory work [GB*s] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} amount\ of\ stored\ data(t)$$

where the amount of stored data (t) can also be written as utilisation(t)[%] * volume of the storage (or memory)..

► Network work [Mbit] =

$$T \cdot \frac{1}{N} \sum_{t=0}^{t=T} data\ rate(t)$$

where the data rate (t) can also be written as utilisation(t) [%] * bandwidth[Mbit/s].

On the one hand, high hardware utilisation results in higher hardware power consumption and thus higher energy consumption. On the other hand, it can also lead to the hardware having to be replaced by new, more powerful hardware, even though the old hardware is still functional. By shortening the lifespan of the hardware, software therefore also creates a demand for raw materials and necessitates the disposal of the old hardware.

Various stakeholders in industry (SDIA 2024; Bitkom e.V. 2021) and science (Calero und Piattini 2015; Hindle 2016; Hilty et al. 2017; Bonamy et al. 2023) publish best practice guidelines and similar documents to help improve the resource and energy efficiency of software.

At present, there are no standardised or normalised procedures for measuring the energy and resource efficiency of software products. However, there are already numerous proposals for measuring the energy and hardware consumption of software. At the same time, the number of available in-house developments and contributions to measurement methods from various stakeholders in business and science continues to increase. A distinction is often made between software-based measurement methods and direct measurement of hardware capacities (Jay et al. 2023; Guldner et al. 2024). Appendix A provides an overview of some of the methods known to the authors.

Typically, the measurement methods measure the energy and hardware utilisation of the device, the utilisation of the relevant components such as CPU, GPU, main memory and permanent memory, data traffic and, in some cases, the electrical power consumption of the entire system or individual components. Different measurement methods are designed and optimised for different hardware-software constellations. Some measured variables are difficult to access on

some hardware platforms or the informative value of the measurement is limited. This particularly applies to the energy consumption of mobile devices. In addition, the environmental impacts of mobile devices tend to be dominated by the manufacturing costs of the hardware, whereas for servers, on the other hand, the energy consumption during the use phase is significantly higher.

The range of available software products that can run on different computer platforms requires different measurement setups. This is due to the diversity of software architectures, hardware systems and application scenarios. Different methods can be combined to some extent in a single measurement set-up or can complement each other in different measurement set-ups.

Drawing on this amount of information, it is recommended that the concept of the reference system (see (Umweltbundesamt 2020)) be revised for the revision of the Award Criteria, that it be opened up to a variety of measurement methods and technology, and thus adapted to the circumstances described above.

Furthermore, the requirements for the measurement should do justice to the metrological possibilities and different environmental impacts of the three hardware platforms mobile device, pc device and server device (see section 6.2).

To ensure the quality and validity of the measurement of resource and energy efficiency, it is recommended that requirements will be set for the measurement method and that auditors verify compliance with them.

1 Hintergrund und Zielsetzung

Der **Blaue Engel** ist seit dem Jahr 1978 das Umweltzeichen der Bundesregierung. Er kennzeichnet umweltschonende Produkte und Dienstleistungen. Zweck des Umweltzeichens ist es, privaten Verbraucherinnen und Verbrauchern, institutionellen Großverbrauchern und öffentlichen Einrichtungen eine verlässliche Orientierung beim umweltbewussten Einkauf zu geben.

Software ist heutzutage in allen Wirtschafts- und Lebensbereichen allgegenwärtig. Sie ist ein **zentraler Baustein der Digitalisierung**, der Hardware und Daten zu einer produktiven und zweckdienlichen Weise zusammenbringt. „In privaten Haushalten sind die Ausstattung mit Informationstechnik (z.B. Smartphones, Computer, Internet-fähige Fernsehgeräte, Smart-Home-Geräte) und die Nutzung von digitalen Diensten (z.B. Voice-over-IP-Telefonie, Videostreaming, Cloud-Storage, Internet) selbstverständlich geworden. Im gewerblichen und industriellen Bereich ist die Nutzung von Digitaltechnologien (für Buchhaltung, Zahlungsverkehr, Simulation, computergestütztes Design und Fertigung, Logistik, Künstliche Intelligenz usw.) fester Bestandteil einer zukunftsfähigen Wirtschaftsweise geworden. Zusätzlich wird ein starkes Wachstum an Datenübertragung und -verarbeitung von Echtzeitdaten durch Mobilität (autonome Fahrzeuge) und zunehmende Sensorik (Internet-of-Things) erwartet“ (Naumann et al. 2020). Auch in sozialen Medien setzt sich der Trend von Kurznachrichten hin zu Bild- und Videodateien oder Computer-generierten Inhalten fort, lineares Fernsehen verlagert sich weiter auf Streaming- und On-demand Dienste. „Maschinelles Lernen und Künstliche Intelligenz ermöglichen und erfordern die Analyse großer Datenmengen. Diese werden für Modellierungen aller Art verwendet, insbesondere für die Personalisierung von digitalen Dienstleistungen und Werbung“ (Naumann et al. 2020). Seit der Veröffentlichung von Transformer-Modellen und den dadurch ermöglichten generativen KI-Systemen werden Anwendungen künstlicher Intelligenz auch vermehrt von Verbraucher*innen genutzt. „Die Verarbeitung dieser digitalen Daten findet nur zu einem Bruchteil lokal, d.h. im Computer der Nutzenden statt. In der Regel werden Daten über das Internet übertragen und in zentralen Rechenzentren verarbeitet und gespeichert. Durch Geschäftsmodelle, die Software und Dienstleistungen „in der Cloud“ anbieten, werden sowohl Rechenleistung als auch Energieverbrauch in Zukunft immer stärker auf Server in Rechenzentren verlagert.“ (Naumann et al. 2020)

„Software ist ein maßgeblicher Treiber des **Energie- und Ressourcenbedarfs** von digitalen Infrastrukturen. Ihre Nutzung ist die Ursache dafür, dass digitale Endgeräte angeschafft, Rechenzentrumskapazitäten immer weiter ausgebaut und Kommunikationsnetzwerke immer leistungstärker werden müssen. Seit dem Jahr 2020 haben die Treibhausgasemissionen der Digitaltechnik in Deutschland nach einer aktuellen Berechnung des Fraunhofer IZM (Stobbe und Nissen 2023) kontinuierlich zugenommen. Von ursprünglich rund 20 Mio. Tonnen CO₂-Äquivalenten im Jahr 2020 (3 % der nationalen Emissionen) wurden diese bis zum Jahr 2023 auf rund 24 Mio. Tonnen gesteigert. Eine weitere Zunahme bis zu 30 Mio. Tonnen im Jahr 2030 wird prognostiziert. Steigende Hardware-Effizienz führt offensichtlich nicht dazu, dass Software sparsamer wird, sondern geht mit einem kontinuierlich steigenden Energie- und Hardwarebedarf einher. Bei Serverherstellern ist ein neuer Trend zu beobachten, dass die elektrischen Leistungen von CPU und GPU, insbesondere bei Geräten für Maschinelles Lernen, stark zunehmen. Außerdem führt immer anspruchsvollere Software dazu, dass funktionierende Hardware obsolet werden kann und vorzeitig ersetzt werden muss.

Das Umweltzeichen Blauer Engel setzt daher direkt an der Software an und formuliert Anforderungen an diese bezüglich Transparenz und Genügsamkeit. Software wird dabei in der

Tradition des Umweltzeichens als ein Produkt betrachtet, für das umweltbezogene Exzellenzkriterien festgelegt werden können“ (Blauer Engel 2024). Seit 2020 existiert der **Blauer Engel für Ressourcen- und energieeffiziente Softwareprodukte DE-UZ 215**. Im Februar 2022 wurde das Umweltzeichen für Software erstmalig an den PDF-Dokumentenbetrachter „Okular“ verliehen (Umweltbundesamt 2020).

Das Umweltzeichen Blauer Engel für Software legt derzeit seinen Schwerpunkt auf die Transparenz von Softwareprodukten und weniger auf die Einhaltung von Grenzwerten. Softwareprodukte mit dem Blauen Engel müssen außerdem zu einer Erhöhung der Ressourcen- und Energieeffizienz beitragen, indem sie Hardwareressourcen genügsam in Anspruch nehmen. „Es gibt mittlerweile eine Vielzahl an Initiativen der Digitalwirtschaft sowie weitergehende Forschungsaktivitäten, die auf den Zusammenhang von Software und Umweltwirkungen aufmerksam machen (z.B. Umweltbundesamt³, Umwelt-Campus Birkenfeld⁴, KDE Eco⁵, Green Software Foundation⁶, SDIA⁷, Green Coding Berlin⁸, Ecodigit⁹, Carbon Tracker¹⁰)“ (Blauer Engel 2024). In einer empirischen Untersuchung von (Jay et al. 2023) werden einige verbreitete Software-basierte Messmethoden, wie zum Beispiel *Carbon Tracker*, *Code Carbon* oder *Perf* miteinander verglichen. Diese weisen je nach Einsatz unterschiedliche Vor- und Nachteile auf. Die hier dokumentierte Überarbeitung des Umweltzeichens greift die unterschiedlichen Initiativen und Messmethoden auf und integriert die neuen Möglichkeiten als Optionen zur Nachweisführung, sofern diese wissenschaftlichen Mindestanforderungen entsprechen. Dadurch bleibt das Umweltzeichen anschlussfähig an die laufende Diskussion innerhalb der Software-Community und verknüpft diese mit einer richtungssicheren und qualitativ hochwertigen Umweltaussage.

Die Architektur von Software hat sich in den letzten Jahren deutlich verändert. Während das Umweltzeichen Blauer Engel für Software DE-UZ 2015, Version 1 aus dem Jahr 2020 ausschließlich Anwendungssoftware für lokale Computer (PCs und Laptops) ausgezeichnet hat, sollte der Geltungsbereich bei der Überarbeitung auf verteilte Systeme ausgeweitet werden. Mit der **Erweiterung des Geltungsbereichs** auf Softwareprodukte, die auf Servern, Desktop-Computern und mobilen Endgeräten lauffähig sind, kommen auch weitere Komplexitätsstufen hinzu, da die Referenzsysteme nicht mehr festgelegt werden können, der Datenverkehr berücksichtigt werden muss und eine Verschiebung der Rechenlast zwischen unterschiedlichen Computersystemen ermöglicht wird. Um die wichtigsten Auswirkungen des softwaregetriebenen Energie- und Ressourcenbedarfs zu erfassen, wurde daher die Methodik zur Bewertung der Ressourceneffizienz von Softwareprodukten weiterentwickelt und für weitere Software-Anwendungen verfügbar gemacht.

Mit dem vorliegenden Hintergrundbericht wird die Herleitung der aktualisierten Vergabekriterien eines Umweltzeichens Blauer Engel für Software begründet und dokumentiert.

³ <https://www.umweltbundesamt.de/themen/digitalisierung> Umweltbundesamt (2024).

⁴ <https://www.umwelt-campus.de/forschung/projekte/green-software-engineering/home> Green Soft Engineering (o.J.).

⁵ <https://eco.kde.org/> KDE e.V. (o.J.).

⁶ <https://greensoftware.foundation/> Green Software Foundation (o.J.).

⁷ <https://sdialliance.org/> SDIA (o.J.).

⁸ <https://www.green-coding.berlin/> Green Coding Solutions GmbH (o.J.).

⁹ <https://www.ecodigit.de/> Eco:Digit (2024).

¹⁰ <https://github.com/lfwa/carbontracker> Anthony et al. (2020).

2 Untersuchungsgegenstand

Der Untersuchungsgegenstand ist allgemein „Software“, d.h. „Programme und Daten in digitaler Form“ (Naumann et al. 2020). Die Vergabekriterien beziehen sich auf Softwareprodukte, d.h. abgrenzbare Einheiten von Programmen und Daten, „die zur Ausführung und Verarbeitung auf einem Hardwaresystem bestimmt sind“ (ebd.). Die Definitionen und Grundverständnisse sind aus dem Hintergrundbericht zur Entwicklung der Kriterien des Umweltzeichens in seiner ersten Version (Naumann et al. 2020) entnommen. Der Kriterienkatalog für nachhaltige Software (Hilty et al. 2017), der im Rahmen eines vorangehenden Forschungsprojektes entwickelt wurde, ist weiterhin aktuell.

Wie bereits von Naumann et al. 2020 dargestellt, wurde „für die Erarbeitung der Vergabekriterien für einen Blauen Engel für energie- und ressourceneffiziente Software folgende Definitionen zugrunde gelegt:

- Software: Programme und zugehörige Daten in digitaler Form.
- Softwareprodukt: Eine lizenzierbare Einheit von Software und die zur Ausführung notwendigen Dienste, die zur Ausführung und Verarbeitung auf einem Hardwaresystem bestimmt sind.
- Softwarestack: Softwarekomponenten, die aufeinander aufbauen und gemeinsam das Softwareprodukt bilden bzw. zu seiner Ausführung notwendig sind.

Informationstechnik (IT) – wie Computer, Laptops, Tablets und Smartphones – und Software bilden eine Einheit, die nur als Gesamtsystem eine Arbeitsleistung bzw. Rechenleistung erbringen kann. Im mobilen Bereich (Smartphones etc.) werden die Softwareprodukte i.d.R. als „Apps“ bezeichnet (Softwareapplikationen). Darüber hinaus gibt es eine Vielzahl von Geräten, die durch Software gesteuert werden, wie Haushaltsgroßgeräte, Unterhaltungselektronik, Gebäudetechnik, Mess- und Steuerungssysteme bis hin zu Kleingeräten. Diese Produkte sind mit Prozessoren, Arbeits- und Permanentspeicher ausgestattet, verrichten softwaregesteuert ihre Arbeitsleistung und sind häufig mit dem Internet verbunden („Internet of Things“). In diesem Fall spricht man von Produkten mit eingebettetem System (englisch embedded system).

- Systemsoftware: Im Gegensatz zur Anwendungssoftware regelt die Systemsoftware den Betrieb und die Wartung der Hardware. Hierzu zählt das Betriebssystem i.d.R. auch Compiler und spezielle Dienstprogramme. Der Geltungsbereich des entwickelten Blauen Engels umfasst diese Art der Software nicht.
- Anwendungssoftware: Softwareprogramme, die einen endbenutzerbezogenen 'Nutzen' erbringen. Beispiele für Anwendungssoftware sind: Bildbearbeitung, E-Mail-Programme, Webbrowser, Multimediaanwendungen, Textverarbeitung oder Tabellenkalkulation.

Wird ein Softwareprogramm gestartet, werden die jeweiligen Arbeitsanweisungen zur Erfüllung der Aufgabe von der Festplatte in den Arbeitsspeicher geladen und anschließend abgearbeitet. Für die Abarbeitung werden ggf. Daten benötigt, die entsprechend ihrem Bedarf geladen, bzw. über Sensoren oder Nutzereingaben erfasst werden. Zur Interaktion mit dem Nutzenden werden Daten oder Ergebnisse auf dem Bildschirm angezeigt und ggf. wieder auf die Festplatte geschrieben. Bei Beendigung des Programms werden die Daten in der Regel aus dem Arbeitsspeicher gelöscht. Auf diese Weise verbraucht das Softwareprogramm indirekt durch die

Nutzung der Hardware natürliche Ressourcen. Hat die Software einen hohen Bedarf an Hardwareressourcen, kann dies dazu führen, dass Hardware ersetzt werden muss, wenn diese die Ansprüche nicht erfüllen kann, obwohl sie – für andere Softwareprodukte – noch voll funktionsfähig wäre. Man spricht von Obsoleszenz durch Software.“ (Naumann et al. 2020)

Hardwarekapazitäten bezeichnen den Umfang an Grundfunktionen, die jedes Gerät bereitstellt, das Software ausführen und/oder Daten übertragen kann. Dazu gehört das Verarbeiten von Instruktionen (CPU, GPU), das Laden und Speichern von Daten im Arbeits- (RAM, VRAM) oder Permanentenspeicher (SSD, HDD) und das Übertragen der Daten über das Netzwerk (Netzwerkadapter, Router, etc.).

Software ist ohne die darunterliegende Hardware nicht nutzbar. Je nach **Gerätetyp** unterscheiden sich wesentliche Parameter, z.B. wie viele Hardwarekapazitäten eine Software in Anspruch nehmen kann. Es werden drei Gerätetypen unterscheiden (Blauer Engel 2024):

- „Personal Computer zum überwiegend stationären Einsatz wie beispielsweise Desktop-PCs, Workstations, Laptops oder Convertibles, jeweils mit einem Desktop-Betriebssystem (Windows, MacOS, Linux oder vergleichbar), nachfolgend bezeichnet als „pc device“.
- Batteriebetriebene tragbare Endgeräte wie beispielsweise Smartphones, Tablets, Smartwatches mit einem mobilen Betriebssystem (Android, iOS oder vergleichbar), nachfolgend bezeichnet als „mobile device“.
- Server-Hardware zur zentralen Datenverarbeitung betrieben in der Regel in einem Rechenzentrum oder Serverraum, nachfolgend bezeichnet als „server device“. Der Server verfügt über eine Netzwerkverbindung und kann mit mehreren Clients gleichzeitig Daten austauschen. Ein Server verfügt in der Regel nicht direkt über Eingabe- und Ausgabegeräte (Human Interface Devices). Die Kommunikation mit dem Server erfolgt stattdessen softwaregetrieben über die Netzwerkschnittstelle.“

Des Weiteren gibt es noch Geräte, die zur Funktionsfähigkeit von Software in verteilten Systemen notwendig sind, beispielsweise Storage Systeme und die gesamte Netzwerkinfrastruktur, insbesondere Router und Netzwerkgeräte, sog. Switches. Diese weiteren Geräte sind nicht explizit Teil des Untersuchungsgegenstandes. „Allen genannten Produkttypen ist gemeinsam, dass die Hardware ohne die Software nicht arbeitsfähig ist und dass die Software die Funktionalität und die Art der Funktionsausführung einer Hardware bestimmt.“ (Naumann et al. 2020)

3 Markt- und Umfeldanalyse

Nachfolgend wird ein Überblick über den Markt an Softwareprodukten gegeben (3.1 Marktanalyse), die Ergebnisse der Analyse geringer Zeichennutzung des BE-UZ 215, Version 3 präsentiert (3.2 Hemmnisse der Zeichenvergabe der ersten Version des Blauen Engels DE-UZ 215) und die Entwicklung der regulativen Anforderungen an die Umweltwirkungen von Software skizziert (3.3 Regulatives Umfeld).

3.1 Marktanalyse

Der Markt an Software ist stark globalisiert und fragmentiert. Die weltweit größten Akteure, sortiert nach Marktanteil des Umsatzes sind AWS (9 %), Oracle (7 %), Microsoft (5 %), Cisco (4 %), IBM (4 %), Salesforce (4 %), SAP (4 %), Adobe (2 %), Google (2 %), VMware (1 %)

(Statista 2024f). Die gesamte digitale Lieferkette ist geprägt von Oligopolen: von den Herstellern der Hardware (Intel, AMD) über die Anbieter von IT-Kapazitäten (AWS, Microsoft, etc.), Softwareherstellern (z.B. SAP), digitalen Diensten (z.B. Google) und Beratungsunternehmen (z.B. Capgemini). Gleichzeitig gibt es einen großen Markt an sehr kleinen und spezialisierten Softwareentwicklern und eine gut vernetzte Open Source Szene. Die Statistik-Plattform Statista unterscheidet die vier Kategorien Anwendungsentwicklungssoftware (29 %), Produktivitätssoftware (11 %), Systeminfrastruktursoftware (21 %) und Unternehmenssoftware (39 %) (ebd.). Der Gesamtumsatz im Jahr 2022 wird weltweit auf 605,5 Mrd. € beziffert mit einem jährlichen Wachstum von 5,3 %. Die Hälfte des Umsatzes wird in den USA erzielt, gut 4 % (24,5 Mrd. €) in Deutschland (Statista 2024e).

Freie und offene Software (FOSS) wird oft als kostenlose Starthilfe für profitorientierte Geschäftsmodelle genutzt. Beispielsweise ist Linux in das Betriebssystem von Apple integriert und die Basis von Android. Freie Software lizenziert alle Nutzungsrechte, inklusive dem Recht, sie verändern zu dürfen. Das geänderte Ergebnis muss bei den meisten Lizenzen aber nicht kostenfrei sein. Open Source heißt wörtlich „quelloffen“ und meint den Quellcode. Kommerzielle Produkte und Dienstleistungen können auf freier Software aufbauen, beispielsweise in Form von Beratung oder Integration in die jeweilige Kunden-Hardware und -Software. Deshalb beteiligen sich auch große Firmen und Stiftungen an Open Source Projekten (z.B. Intel, Alfresco, Adobe, Oracle, Canonical, Transcend, Redpill Linpro, IBM, Compiere, Epson, OpenText, Cleversafe, ClearCenter, FOSSID, Automattic, Acquia, Continuent, Comiit, RethinkDB, Red Hat, Astaro, SUSE etc.) (HTF Market Intelligence Consulting Pvt. Ltd. 04.10.2021). Die Bedeutung von FOSS mit wirtschaftlichen Indikatoren (z.B. dem Umsatz) zu messen, kann nur einen Hinweis auf ihre gesamtgesellschaftliche Relevanz liefern, da offene Software selbst meistens keinen Umsatz generiert.

Zu Marktdaten über Software für die öffentliche Hand liegen den Autor*innen keine Informationen vor.

Knapp 350 Unternehmen sind mit dem Siegel „Software made in Germany“ ausgezeichnet¹¹, die Softwareallianz Deutschland¹² besteht aus 120 Unternehmen. Die Wikipedia-Kategorie „Softwarehersteller (Deutschland)“ beinhaltet 155 Einträge. Die dort gelisteten Unternehmen sind meist mittelständische Unternehmen, die B2B Software oder Software-nahe Dienstleistungen anbieten. B2B Software besteht meist aus Serveranwendungen für die Verwaltung, Management, Analyse, Planung, Steuerung und Online-Services.

Es gibt über 42.000 Softwarefirmen in Deutschland (Listflix 2023), von denen jedoch nur die wenigsten eigenständige Softwareprodukte entwickeln. Zu den größten deutschen Softwareunternehmen gehören u.a. TeamViewer, SAP, avira, Plancraft, Bechtle, Arvato, TechData, Trados, AnNoText, BlueSystems, Suse, Chimera Entertainment (Hersteller von Angry Birds), Eyeo (Hersteller von Adblock Plus), G Data Cyberdefense, meetyoo, Mimimi Games, Ahead Software (Hersteller von nero), sixteentons (Jašarević 2022). Beispielsweise erzielte SAP im Jahr 2023 einen globalen Umsatz von 31,2 Mrd. € (Statista 2024d), das Software-Beratungsunternehmen Capgemini einen Umsatz von 22 Mrd. € im Jahr 2022.¹³

Der Umsatz mit mobilen Apps hat sich in den letzten drei Jahren verdoppelt und betrug 2022 in Deutschland ca. 3,4 Mrd. € (Statista 2024c) und global ca. 399 Mrd. € in 2022 (Statista 2024a). Damit hat dieser wachsende Markt in Deutschland einen Anteil von rund 14 %. Der größte Teil

¹¹ <https://www.software-made-in-germany.org/zertifizierte-unternehmen/> BITMI (2023).

¹² <https://softwareallianz.de/> Softwareallianz Deutschland (2023).

¹³ <https://www.capgemini.com/news/press-releases/capgemini-delivers-another-record-performance-in-2022/> Capgemini (2023).

des Umsatzes wird mit Apps für soziale Netzwerke und Spiele erzielt. Der durchschnittliche Umsatz pro Download beträgt ca. 1,70 €, wobei dieser hauptsächlich durch Werbung und In-App-Käufe generiert wird (Statista 2024c). Der Umsatz durch den Verkauf von Apps ist vernachlässigbar klein (ebd.).

Der Umsatz von lokaler und Cloud-basierter „Office Software“ in Deutschland wird für 2024 auf rund 800 Mio. € geschätzt (Statista 2024b), was etwa 3 % Marktanteil ausmacht.

Inzwischen ist das Umweltzeichen „Blauer Engel“ Bestandteil der Nachhaltigen IT-Beschaffung des Bundes (IT-Beauftragter des Bundes 2021). IT-Dienstleistungen und IT Technik zusammengekommen machen mehr als die Hälfte der Ausgaben des Beschaffungsamtes des Bundes von mehr als 5 Mrd. € im Jahr 2019 aus (Beschaffungsamt des BMI 2024).

3.2 Hemmnisse der Zeichenvergabe der ersten Version des Blauen Engels DE-UZ 215

Zur Analyse der Hemmnisse der Beantragung des Umweltzeichens „Blauer Engel“ wurden potenziell interessierte Softwareunternehmen, Verbände und Expert*innen recherchiert. Dabei ging sowohl ein kurzer Überblick über den nationalen und internationalen Markt ein als auch die Erfahrungen mit der Branche aus dem vorangegangenen Projekt zur Erstellung des Blauen Engels für ressourceneffiziente Software DE-UZ 215. Bei der Auswahl von Expert*innen für die Hemmnisanalyse konnte auf bestehende Kontakte aus dem Forschungsprojekt von Gröger et al. (2018) zurückgegriffen werden. Es wurden 13 Expert*innen befragt. Von diesen haben sich neun detailliert mit den Vergabekriterien beschäftigt und lediglich drei in der Absicht, den Blauen Engel (BE) zu beantragen.

3.2.1 Systemische Hemmnisse

Green Software unbekannt

Hauptursache für die geringe Bewerberzahl auf das Umweltzeichen sei die geringe Bekanntheit sowohl der Konzepte von Green IT als auch des Blauen Engels. Die Bekanntheit des Blauen Engels ist eher gering. Diese Beobachtung wird durch eine Studie von te Brake bestätigt. *„Selbst bei interessierten Menschen der Branche, ist das Bewusstsein für Green Software weniger als 70 %. Umweltkriterien werden bei 20 % der Teilnehmenden angewandt... Das Monitoring des Energiebedarfes der Software während der Entwicklung findet kaum (ca. 10 %) statt.“* (te Brake 2020).

Das Bewusstsein für Knappheit von Ressourcen ist in den letzten Jahren auch in der Softwareentwicklung angekommen. Jedoch stellen García-Berna et al. (2019) eine Diskrepanz zwischen dem Bewusstsein für die Umweltwirkung von Software und der praktischen Softwareentwicklung fest. Diese Abweichung zwischen gewünschtem und eigentlichem Tun beschreibt auch (te Brake 2020). Leistung, Benutzerfreundlichkeit und Design werden dem Energiebedarf übergeordnet (ebd.). Dies ergibt sich insbesondere aus einem der 17 Leitsätzen der Unix-Programmierung, die 2003 von Eric S. Raymond aufgestellt wurden und seither weltweit Beachtung finden. Der 13. Leitsatz besagt:

„Value developer time over machine time“ (Raymond 2003).

Die Zeit der Softwareentwickler*innen als wichtiger zu erachten als die Rechendauer, führt im Zweifel zu ineffizientem Code. In einer Umfrage der Gesellschaft für Informatik gab etwa die Hälfte der befragten Softwareunternehmen an, „Green Coding als strategisches Ziel“ anzusehen, jedoch messen aktuell nur „etwa 18% der Firmen die Umweltauswirkungen ihrer Software aktiv“ (Gesellschaft für Informatik e.V. 2023).

Keine Vergleichbarkeit zwischen Produkten

Ein komparativer Marktvorteil durch die Auszeichnung mit dem Umweltzeichen wird allgemein als gering eingeschätzt. Dies liege unter anderem an der geringen Vergleichbarkeit von verschiedenen Softwareprodukten. Im Gegensatz zu physischen Produkten ist die Vergleichbarkeit verschiedener Softwareprodukte nur sehr eingeschränkt möglich, da selbst Produkte der gleichen Kategorie (z.B. Mail-Programm) einen sehr unterschiedlichen Umfang an Funktionalitäten bereitstellen.

Bestehende Vendor-Lock-ins

Es besteht allgemein ein großes Problem darin, dass viele Softwareprodukte in der Beschaffung gar nicht ausgeschrieben werden können, da zu große Pfadabhängigkeiten von bestimmten Anbietern existieren, sogenannte „Lock-in Effekte“. Beispielsweise wird nach Ablauf der Lizenz von MS-Word eine neue Lizenz desselben Herstellers beschafft, da die Makros und Layouts aller bisherigen Dokumente mit konkurrierenden Produkten nicht fehlerfrei kompatibel sind.

3.2.2 Hemmnisse der Kriterien

Geltungsbereich zu begrenzt

Die Marktdynamik von Desktopanwendungen sei gering, diese würden kaum beworben und weitgehend durch webbasierte Software ersetzt. Der Markt um Apps für mobile Geräte sei sehr viel aktiver, aber auch sehr viel kurzlebiger. Der BE sei vorrangig für langlebige Software und die Entwicklung von Behördensoftware von Interesse. Es sei unklar, ob sich der Anwendungsbereich auf Desktopsoftware beschränkt.

Mangelnde Erfahrung mit der Messmethode

Da das Standardnutzungsszenario von den Antragsteller*innen festzulegen ist, sehen diese sich großen Unsicherheiten bezüglich der Messmethode ausgesetzt. Es besteht ein großer Wunsch nach einem einfach zu bedienenden Tool, das die Messungen und eventuell auch weitere Anforderungen automatisiert überprüft. Für die Dokumentation sollten mehr Standardformate und Templates zur Verfügung gestellt werden. Diese sollten sich an etablierten Dateiformaten (z.B. json, csv) orientieren.

Niedriges Ambitionsniveau

Einige Experten betonen, dass die Kriterien allesamt nicht sehr schwierig zu erfüllen seien, ein zu hohes Ambitionsniveau also nicht der Grund für die geringe Anzahl an Zeichennehmer*innen sein könne.

Abwärtskompatibilität

Dieses Kriterium könne ein Hindernis darstellen, wenn die Tools zum Kompilieren der Software von Gatekeepern bereitgestellt werden, sodass es gar nicht möglich ist, Maschinencode für 5 Jahre alte Hardware zu generieren. Die Abwärtskompatibilität wurde dahingehend kritisiert, dass dieses Kriterium ggf. kontraproduktiv für die Sicherheit sei, da veraltete Softwareumgebungen häufiger Sicherheitslücken aufweisen. Die Prüfung dieses Kriteriums würde darüber hinaus einen sehr hohen Aufwand darstellen, da sich die Software sehr schnell entwickelt.

Offlinefähigkeit

Die Offlinefähigkeit wurde als nicht mehr zeitgemäß eingestuft. Es wurde zudem darauf hingewiesen, dass die meisten Kunden Online-Software bevorzugen würden. Stattdessen könnte

der Fokus der Anforderungen auf den Netzwerk-Traffic verschoben werden, der bei der Gegenüberstellung vergleichbarer Software möglichst gering sein sollte.

Modularität

Derzeit werden Projekte möglichst schnell umgesetzt und im Nachhinein auf nicht-funktionale Anforderungen (wie z.B. Nachhaltigkeitsaspekte) hin untersucht. Software, die schon (fast) fertig ist, könne gewisse Vorgaben des BE nicht einhalten. Dies betreffe insbesondere die Modularität, da im Design die entsprechenden Schnittstellen von Beginn an mitgeplant werden müssten. Es sei schlecht definiert, was Modularität bedeutet.

Offene Datenformate

Dieses Kriterium stehe nicht offensichtlich im Zusammenhang mit Ressourceneffizienz, sondern adressiere Lock-in Effekte. Er könnte dahingehend erleichtert werden, dass es eine Exportfunktion zu einem offenen Standard geben muss oder das Datenformat am Ende des Supportdauer offengelegt werden muss (escrow Vertrag).

Die Offenlegung wurde als eines der stärker hemmenden Kriterien identifiziert. Es gibt einige Unternehmen, die zur Geheimhaltung der Daten verpflichtet sind, da die Offenlegung eine Sicherheitslücke darstellen könnte. Dies ist z.B. der Fall bei medizinischer Software. Im bisherigen Kriterium ist auch die Möglichkeit eines Escrows gegeben. Die Offenlegung in Form von Escrows müsste im neuen Blauen Engel Kriterium stärker hervorgehoben werden.

Sicherheitsupdates

Sicherheitsupdates seien nicht zu trennen von Fehlerkorrekturen (Bugfixes) für stabilere Versionen. Kontinuierliche Releases würden den Bedarf von Nutzer*innen am besten decken. Auf diese Weise würden keine großen Brüche in der Software entstehen und die Anforderungen an die Hardware blieben nahezu konstant.

Updates des Produkts

Dieses Kriterium verhindere große Funktionserweiterungen. In der Praxis würden Softwareprodukte in Unternehmen agil entwickelt, sodass in regelmäßigen Abständen eine funktionsfähige Version veröffentlicht würde, die sich nach dem Bedarf der Kunden und Auftraggeber richte. Bei Erstauslieferung sei das Produkt also noch nicht fertig und enthalte auch noch nicht alle geplanten Funktionalitäten. Daher werde Software im zeitlichen Verlauf mit jedem Update wachsen.

Bei FOSS würden Softwareupdates mehrfach pro Woche veröffentlicht und der Quellcode würde über verschiedene Kanäle verteilt werden, die nicht alle von den Entwickler*innen kontrolliert würden. Daher sei eine Überprüfung jedes Updates mit den jetzigen Energiemessverfahren unverhältnismäßig, zumal es sich bei den Updates um graduelle Änderungen oder kleine Bugfixes handele. Sinnvoll sei eine Überprüfung nach einem Quartal.

3.3 Regulatives Umfeld

Auf europäischer Ebene ist ein Trend zu erkennen, die Umweltwirkungen von Software im Sinne des Verbraucherschutzes zu regulieren und transparent zu machen.

Die EU-Verordnung über digitale Inhalte, die aktuellen Entwürfe der EU-Ökodesign Verordnungen und die EU-Richtlinie über unlautere Geschäftspraktiken adressieren Updates, deren Verfügbarkeit und ihren Einfluss auf die Leistung des Produktes. Das EU-Gesetz über künstliche Intelligenz verlangt, den Energiebedarf für das Trainieren von KI-Modellen zu dokumentieren.

Neben den gesetzlichen Anforderungen gibt es darüber hinaus auch Internationale Normen, die über einen ökobilanziellen Ansatz die Umweltwirkungen von digitalen Dienstleistungen bewerten.

3.3.1 EU-Verordnung über digitale Inhalte

Die EU-Verordnung über digitale Inhalte ((EU) 2019/770, (Europäisches Parlament und Rat 2019) führt das Konzept der Konformität einer Verkaufsware zum Kaufvertrag ein und überträgt es auf Software. Demnach muss Software

(a) der Beschreibung, der Menge und der Qualität entsprechen und die im Vertrag geforderte Funktionalität, Kompatibilität, Interoperabilität und andere Merkmale aufweisen;

[...]

(d) wie im Vertrag vorgesehen aktualisiert werden.

Der Händler oder Hersteller der Software muss also am Übergabepunkt der Software benennen, wie lange die Software mit Updates versorgt wird.

3.3.2 EU-Ökodesign Verordnungen

Für Softwareprodukte gibt es im Rahmen der aktuellen Ökodesign-Verordnungen keine direkten Anforderungen an die umweltgerechte Gestaltung von Software oder deren Energieverbrauchs-kennzeichnungen. Die Anforderungen in diesem Bereich beziehen sich nur auf Hardware-produkte, wie bspw. die Ökodesign-Verordnung für Server und Datenspeicherprodukte (Europäische Kommission 2019) oder die Ökodesign-Verordnung für Smartphones (Europäische Kommission 2023).

Die bestehende Ökodesign-Richtlinie 2009/125/EG (Europäisches Parlament und Rat 2009) wird derzeit überarbeitet und durch die Ökodesign-Verordnung für nachhaltige Produkte (Ecodesign for Sustainable Products Regulation – ESPR) ersetzt ((EU) 2024/1781, (Europäisches Parlament und Rat 2024c). Zusätzlich zu Anforderungen an die Energieeffizienz werden Aspekte der Haltbarkeit, Reparierbarkeit und Zirkularität reguliert werden. Software wird dort in Bezug auf die Leistungsfähigkeit der Hardware reguliert:

„Artikel 33 Umgehung

(4) Bei Messung mit der für die Konformitätsbewertung verwendeten Prüfmethode führen **Software- oder Firmware-Aktualisierungen zu keiner Verschlechterung der Produktleistung** in Bezug auf einen oder mehrere der Produktparameter, die in den gemäß Artikel 4 erlassenen delegierten Rechtsakten geregelt sind, unter die die Produkte fallen, oder der Funktionsfähigkeit aus Sicht des Nutzers, es sei denn, der Endnutzer hat vor der Aktualisierung seine ausdrückliche Zustimmung erteilt. Die Ablehnung der Aktualisierung führt nicht dazu, dass sich die Leistungsmerkmale ändern.

Software- oder Firmware-Aktualisierungen führen zu keiner Verschlechterung der in Unterabsatz 1 genannten **Leistung** in einem Maße, dass das Produkt den Anforderungen der gemäß Artikel 4 erlassenen delegierten Rechtsakte nicht mehr entspricht, die zum Zeitpunkt des Inverkehrbringens oder der Inbetriebnahme des Produkts galten.“

Die Ökodesign-Verordnung für Smartphones (Europäische Kommission 2023) enthält Anforderungen an Betriebssysteme von Smartphones und Tablets:

„ANHANG II: A 1.2. Auslegung auf Zuverlässigkeit

Ab dem 20 Juni 2025 gilt:

(6) Aktualisierungen des Betriebssystems:

(a) Ab dem Datum der Beendigung des Inverkehrbringens und für einen **Mindestzeitraum von fünf Jahren** sind von Herstellern, Importeuren oder Bevollmächtigten zur Verfügung gestellte **Sicherheits-, Korrektur- oder Funktionsaktualisierungen für ein Betriebssystem** von diesen **kostenlos** für alle Einheiten eines Produktmodells mit demselben Betriebssystem zugänglich zu machen;“

Betriebssysteme sind zwar nicht im Geltungsbereich des Blauen Engels, die Ökodesign-Verordnung für Smartphones stellt jedoch sicher, dass fünf Jahre alte Betriebssysteme noch verfügbar sind und ermöglicht somit die Abwärtskompatibilität von Anwendungssoftware (Apps) von mindestens fünf Jahren.

3.3.3 EU-Richtlinie über unlautere Geschäftspraktiken

Die Änderung der Richtlinie über unlautere Geschäftspraktiken ((EU) 2024/825, (Europäisches Parlament und Rat 2024a) beinhaltet eine Liste mit Tätigkeiten, die verboten sind.

„Anhang I der Richtlinie 2005/29/EG wird wie folgt geändert:

[...]

(4) Es werden folgende Nummern 23d bis 23i eingefügt:

23d. Unterlassung der Information des Verbrauchers, dass sich eine **Software-Aktualisierung** negativ auf die Verwendung von Waren mit digitalen Elementen oder bestimmte Merkmale dieser Waren auswirkt, selbst wenn die Software-Aktualisierung die Funktionsweise anderer Merkmale verbessert.

[...]“

3.3.4 EU-Gesetz über künstliche Intelligenz

Das Gesetzes über künstliche Intelligenz ((EU) 2024/1689, (Europäisches Parlament und Rat 2024b) wurde am 13.06.2024 verabschiedet.

In den Anhängen XI und XIII des Gesetzes wird erstmals eine Information über den Energiebedarf einer Software gesetzlich eingefordert:

„ANHANG XI

Technische Dokumentation gemäß Artikel 53 Absatz 1 Buchstabe a – technische Dokumentation für Anbieter von **KI-Modellen mit allgemeinem Verwendungszweck**

Abschnitt 1

Von allen Anbietern von KI-Modellen mit allgemeinem Verwendungszweck bereitzustellende Informationen

[...]

2. e) **bekannter oder geschätzter Energieverbrauch des Modells.**

Wenn der Energieverbrauch des Modells nicht bekannt ist, kann für Buchstabe e der Energieverbrauch auf Informationen über die eingesetzten Rechenressourcen gestützt werden.

[...]

ANHANG XIII

Kriterien für die Benennung von **KI-Modellen mit allgemeinem Verwendungszweck mit systemischem Risiko** gemäß Artikel 51

Um festzustellen, ob ein KI-Modell mit allgemeinem Verwendungszweck über Fähigkeiten oder eine Wirkung verfügt, die den in Artikel 51 Absatz 1 Buchstaben a und b genannten gleichwertig sind, berücksichtigt die Kommission folgende Kriterien:

- a) die Anzahl der Parameter des Modells;
- b) die Qualität oder Größe des Datensatzes, zum Beispiel durch Tokens gemessen;
- c) die Menge der für das Trainieren des Modells verwendeten Berechnungen, gemessen in FLOPs oder anhand einer Kombination anderer Variablen, wie geschätzte Trainingskosten, geschätzter Zeitaufwand für das Trainieren oder **geschätzter Energieverbrauch für das Trainieren**;

[...]“

3.3.5 Internationale Normen

Die gemeinsam von der ITU und dem ETSI entwickelte Norm ITU L.1410 „Methodology for environmental life cycle assessments of information and communication technology goods, networks and services“ (ITU 2014) ist derzeit die einzige internationale Norm für die Ökobilanzierung digitaler Dienstleistungen.

Die französische Umweltbehörde ADEME hat zur einheitlichen Bewertung von digitalen Dienstleistungen zudem Product Category Rules (PCR) entwickelt und veröffentlicht (ADEME 2021). Darin werden Parameter formuliert, die helfen sollen, die funktionelle Einheit zu definieren, die Grundlage für jede Ökobilanz ist.

Die beschreibenden Parameter sind:

- die erbrachte(n) Funktion(en) / die erbrachte(n) Dienstleistung(en): "Was?";
- den Umfang der Funktion oder Dienstleistung: "Wie viel?";
- das gewünschte Qualitätsniveau: "Was?";
- die (Lebens-)Dauer des Produkts: "Wie lange?"; (ebd.)

Davon verspricht sich der die Umweltbehörde, dass die Definition der funktionellen Einheit, „den Vergleich der Umweltauswirkungen von zwei Produkten auf der Grundlage einer gemeinsamen Einheit ermöglicht“ (ebd.).

4 Umweltwirkungen von Software

Hinsichtlich der Umweltwirkungen werden die direkten Eigenschaften des Softwareprodukts als solches berücksichtigt (sog. “Green in Software”-Aspekte), und nicht Effekte, die durch die Nutzung des Softwareprodukts ausgelöst werden (“Green by Software”, indirekte Wirkungen bzw. Effekte zweiter und dritter Ordnung wie bspw. Energieeinsparungen durch die Anwendung von Software) (Naumann et al. 2020).

„Zwei Einflüsse, die durch den Einsatz eines Softwareprodukts verursacht werden, sind wesentlich für die Bestimmung der Umweltwirkungen:

- ▶ den Energiefluss durch die Herstellung, Nutzung und Entsorgung der Hardware, auf der die Software läuft (elektrische Energie wird zu Abwärme),
- ▶ der Materialfluss der Hardware durch die Organisation, die sie verwendet (neue Hardware wird zu Elektronikschrott).“ (Naumann et al. 2020)

Somit hat Software eine direkte Umweltwirkung auf den Energiebedarf und eine indirekte auf die Lebensdauer der Hardware (Gröger et al. 2018).

4.1 Identifikation relevanter Umweltwirkungen

Software ist nicht nur für die Beanspruchung der Hardware, sondern auch für deren Funktionsfähigkeit und (Un-)Brauchbarkeit verantwortlich. In beiden Aspekten können sich Softwareprodukte unterscheiden. Hardware und Software stehen in wechselseitiger Beziehung, deshalb sind auch ihre Umweltwirkungen eng miteinander verknüpft. Software nimmt entlang ihres Lebenszyklus unterschiedliche Hardwarekomponenten wie Prozessor, Arbeits- und Permanent-speicher, Netzwerkkarten, Router, Grafikkarten, etc. in Anspruch, die wiederum im Betrieb elektrische Energie in Wärme umwandeln. Im Regelfall gilt, je passender die Hardware für die jeweilige Software-Last dimensioniert ist, desto effizienter ist das Gesamtsystem. Beispielsweise ist die Rechenfähigkeit eines Laptops unterdimensioniert, um große Sprachmodelle zu trainieren. Für Machine-Learning-Modelle werden stattdessen eher Hochleistungscomputer eingesetzt. Der gleiche Laptop ist dagegen überdimensioniert, um ein Heizkörperventil zu steuern. Stattdessen lassen sich für Heizkörperregelungen Mikroprozessoren mit geringer Leistungsaufnahme gegenüber dem Laptop einsetzen.

Die Umweltwirkung der Hardware wird von mehreren Softwareeigenschaften beeinflusst. Dazu gehören neben des eigentlichen Energiebedarfs während der Ausführung die Abwärtskompatibilität zu existierender Hardware, Datenformate, die Kontinuität der Software (Updates), Deinstallierbarkeit, Modularität und Dokumentation der Software (Naumann et al. 2020). Auch unerwünschte Werbung erzeugt im Regelfall vermeidbaren Datenverkehr (Hilty et al. 2017).

4.1.1 Energiebedarf / Leistungsaufnahme

Die elektrische Energie berechnet sich als Integral der Leistungsaufnahme (Watt) über die Zeit (Sekunden). Je nach Computer-Plattform variiert dabei die Höhe der Leistungsaufnahme. Desktop-Computer (PCs), die im Rahmen dieses Projektes gemessen wurden, wiesen eine Leistungsaufnahme zwischen 10 und ca. 50 Watt auf, wobei es sich um minimal ausgestattete PCs handelte. Gaming-PCs erreichen oft mehrere 100 Watt. Noch leistungsstärkere Geräte sind

Server innerhalb von Rechenzentren, deren Leistungsaufnahme mehrere 1.000 Watt erreichen kann.

Generell ist wesentlich, dass unterschiedliche Softwareprodukte für dieselbe Aufgabe unterschiedlich viel Energie und Ressourcen benötigen können, je nachdem wie effizient sie entwickelt und aufgebaut sind. So konnte in einem Projekt von Gröger et al. (2018) gezeigt werden, dass beim Vergleich zweier Textverarbeitungsprogramme bei der Erledigung der gleichen Aufgaben, eines der Softwareprodukte viermal so viel Energie benötigte wie das andere.

Mobile Devices

Smartphones und Tablets, zusammengefasst bezeichnet als „Mobile Devices“, werden mit einem Akku betrieben, welcher in der Regel über mehr als einen Tag lang Energie bereitstellen soll, bis er erneut geladen werden muss. Mobile Endgeräte sind durch den Einsatz besonders energieeffizienter Bauteile auf Energiesparen ausgelegt und so auch die darauf betriebene Software. In der Literatur sind unterschiedliche Ansätze zu finden, die sich mit der Messung des Energiebedarfs von mobilen Apps beschäftigen (Ahmad et al. 2015). Im Feldtest wurde beobachtet, dass sich die Leistungsaufnahme während der Nutzung bei mobilen Geräten in einem schmalen Spektrum unter 4 Watt bewegt. Der Energiebedarf mobiler Endgeräte ist also im Vergleich zum Netzwerk und den Servern vernachlässigbar klein. Gleichzeitig wird deutlich, dass viele Einflussfaktoren auf ein mobiles Gerät einwirken, die die Messergebnisse beeinflussen. Der Strombedarf ist dabei bspw. von Faktoren wie CPU-Frequenz und deren Auslastungsgrad, Größe des Bildschirms und Helligkeitsstufe sowie der Internetanbindung abhängig (Ahmad et al. 2015).

Ein wichtiger Umweltaspekt bei mobile devices ist der Herstellungsaufwand. Vergleicht man beispielsweise die CO₂-Emissionen der Herstellung eines Smartphones (ca. 100 Kilogramm) mit denen, die während der Nutzung durch den Energiebedarf auftreten (ca. 5 Kilogramm pro Jahr), so dominiert die Herstellungsphase deutlich (Gröger 2020). Daraus lässt sich ableiten, dass umweltgerechte Software für Mobile Devices vor allem zu deren Langlebigkeit beitragen sollte.

4.1.2 Hardwareinanspruchnahme

Als Daten-Hardwareinanspruchnahme lässt sich bspw. das Integral der Auslastung (%) der Hardwarekapazitäten über die Zeit (s), gemessen in %*s für Prozessoren, GB*s für Permanent- und Arbeitsspeicher und Mbit/s*s (= Mbit) für die Menge der übertragenen Daten anwenden. Je nach Kontext sind auch andere Präfixe (k, M, G, T, etc.) sinnvoll.

Eine hohe Hardwareinanspruchnahme bewirkt zum einen eine höhere Leistungsaufnahme der Hardware und damit einen höheren Energiebedarf und zum zweiten kann sie dazu führen, dass die Hardware durch neue, leistungsstärkere Hardware ersetzt werden muss, obwohl die alte Hardware noch funktionsfähig wäre. Durch die Verkürzung der Lebensdauer der Hardware verursacht Software daher auch indirekt einen Rohstoffbedarf.

Betrachtet man als Umweltwirkung von Software den Rohstoffbedarf, gemessen als Abiotic Depletion Potential (ADP), so dominiert hier ebenfalls die Herstellungsphase (ca. 90 %) gegenüber der Nutzungsphase (ca. 10 %), wie im Projekt „Green Cloud Computing“ am Beispiel von Servern und Speichersystemen in Rechenzentren aufgezeigt (Gröger et al. 2021).

Bei anderen Umweltwirkungskategorien, wie Treibhausgasemissionen (GWP) und Primärenergiebedarf (KEA), dominiert wegen des hohen Energiebedarfs von Servern und Speichersystemen jedoch die Nutzungsphase (ca. 80 %) (ebd.).

Da der **Prozessor** in konventioneller Computer-Hardware integriert ist und dort einen zentralen Baustein darstellt, ist er besonders relevant bei der Untersuchung der Hardwareinanspruchnahme. Je nach Gerätetyp haben Prozessoren sehr große elektrische Anschlussleistungen von bis zu mehreren 100 W bei aktuellen Modellen (Sun et al. 2019). Diese werden von Herstellern in der Regel als thermische Nennleistung (Thermal Design Power, TDP) angegeben. Je nach Software wird der Prozessor unterschiedlich ausgelastet und benötigt somit auch unterschiedlich viel Energie.

Grafikprozessoren sind bei Software, die ein Display nutzt, üblicherweise für die Bildverarbeitung verantwortlich. Zusätzlich stellen GPUs immer öfter zusätzliche Rechenkapazität bereit, da die GPU bei hochgradig parallelisierbaren Algorithmen signifikant schneller arbeitet als die CPU. Dies ist aktuell hauptsächlich für das Training und die Anwendung von Modellen des Maschinellen Lernens oder für technische oder wissenschaftliche Simulationen relevant. GPUs können im Hauptprozessor integriert oder als zusätzliche Hardwarekomponente verbaut sein. Je mehr Daten die Grafikkarte verarbeitet werden muss, desto mehr Energie benötigt diese. Aktuelle Grafikkarten weisen TDP von einigen Dutzend bis mehreren Hundert Watt auf, einzelne sogar über 2kW (Sun et al. 2019), weshalb auch GPUs ein relevanter Faktor im Energiebedarf ist.

Die Inanspruchnahme des **Arbeitsspeichers** durch die Software gibt einen Hinweis, ob viele oder wenige Daten bearbeitet werden. Ein großer Arbeitsspeicher führt zu einem erhöhten Energiebedarf und eine hohe Inanspruchnahme des Arbeitsspeichers reduziert die Anzahl parallel verarbeitbarer Prozesse, bzw. verlängert deren Laufzeit.

Die Belegung des **Permanentspeichers** verursacht einen Herstellungsaufwand. Rotierende Festplatten nehmen dauerhaft eine annähernd konstante Leistung auf, unabhängig von der Belegung und von Lese- und Schreibzyklen. Festkörperspeicher (SSD- Solid State Drive) haben einen wesentlich höheren Herstellungsaufwand, dafür aber einen geringeren Strombedarf im Leerlauf als rotierende Festplatten. Ihre Lebensdauer hängt von der Anzahl an Schreibzyklen ab, der Energiebedarf während der Nutzung von der Zeit, in der Daten geschrieben werden.

Der **Datenverkehr** verursacht einen Energiebedarf bei verteilten Hardwaresystemen (z.B. Client-Server Architekturen), der größtenteils für Nutzende nicht sichtbar ist. Die Teile des Systems kommunizieren über das Netzwerk, welches aus Routern, Switches, Gateways, Kabeln, Funkmasten und weiterer Infrastruktur besteht, die alle Energie benötigen. Es gibt einige Ansätze, die versuchen, diesen Strombedarf näherungsweise zu bestimmen (Europäische Kommission, Joint Research Centre et al. 2024). Da es sich jedoch um Näherungswerte für den gesamten Netzwerkverkehr handelt, ist es schwer zu sagen, wie viel Energie einer einzelnen Verbindung zugeordnet werden sollte. Hinzu kommt, dass der Datenverkehr zwischen den Netzwerkknoten üblicherweise nicht die gleichen Wege nimmt. Mit der übertragenen Datenmenge nimmt vermutlich auch der Energiebedarf im Netzwerk zu. Daher sollte die Datenmenge, z.B. durch eine effektive Datenkomprimierung, möglichst minimiert werden. Ob beispielsweise in einem Client-Server System der Energiebedarf, der der Software zugeordnet werden kann, im Netzwerk, Endgerät oder Server am größten ist, ist je nach Softwareprodukt unterschiedlich und ggf. sehr aufwendig zu bestimmen. Je nach Anwendungsfall kann es ressourcenschonender sein, rechenintensive Aufgaben auf leistungsstarken Servern durchzuführen und deren Ergebnisse über das Netzwerk zu versenden, als diese Aufgaben auf Endgeräten durchführen zu lassen.

In jedem Fall trägt der Datenverkehr über das Netzwerk zum Gesamt-Ressourcenbedarf während der Nutzung einer Software bei und sollte betrachtet werden. Insbesondere sollte dabei derjenige Datenverkehr identifiziert werden, der nicht zur Funktionalität einer Software

beiträgt, sondern z.B. für Werbung und Tracking genutzt und somit für die Nutzenden keinen direkten Mehrwert bietet, aber zu einem erhöhten Ressourcenbedarf beiträgt.

4.1.3 Auswirkungen auf die Hardware-Lebensdauer

Updates und neue Versionen für neue Anforderungen eines Softwareproduktes führen oft dazu, dass immer mehr Hardwarekapazitäten in Anspruch genommen oder vorausgesetzt werden. Gleichzeitig muss Software in Form von Sicherheitsupdates und Fehlerbehebung gewartet werden, um die weitere Nutzbarkeit sicherzustellen.

Die zunehmende Entwicklung von anspruchsvoller werdender Software und leistungsstärkerer Hardware führt dazu, dass Hardware aufgrund mangelnder Leistung ersetzt und aussortiert wird (Jaeger-Erben et al. 2023). Hardware altert in der Regel physisch nur sehr langsam, Batterien und Akkus ausgenommen. Sie wird in der Regel obsolet aufgrund von Software, die nach größeren Kapazitäten verlangt.

4.1.4 Indirekte Umweltwirkungen durch Nutzungsautonomie

Software ist in der Lage, Nutzungsbeschränkungen durchzusetzen und bestimmte Nutzung oder Inhalte aufzuzwingen. Damit greift Software tief in die Autonomie und das Nutzerverhalten ein. Über diesen Mechanismus kann Software indirekt negative Umweltwirkungen verursachen.

Lock-in Effekte, proprietäre Datenformate und Lizenzbeschränkungen können z.B. dafür sorgen, dass eine energie- und ressourcenschonendere alternative Software nicht genutzt werden kann. Werbung und Tracking verursachen einen vermeidbaren und von den Nutzenden häufig nicht gewollten Datenverkehr. Speicherplatz bleibt verloren, wenn die Software nach dem Lebensende nicht komplett deinstalliert werden kann. Andererseits können modulare Softwareprodukte, also solche die auf ihre Hauptfunktionalitäten beschränkt sind und die Aktivierung bzw. Deaktivierung entsprechender Installationsoptionen einzelner Module ermöglichen, Hardwarekapazitäten sparen, wodurch auch der Energiebedarf reduziert wird (Kern et al. 2018). Wenn energie- und ressourcensparsame Software mit freien Lizenzen ausgestattet ist, kann sie in andere Projekte übernommen werden und dort ineffizienten Code ersetzen und von anderen Programmierer*innen weiter verbessert werden.

4.2 Handreichungen und Leitfäden zu ressourcen- und energieeffizienter Software

Diverse Akteure in Wirtschaft (SDIA 2024; Bitkom e.V. 2021) und Wissenschaft (Calero und Piattini 2015; Hindle 2016; Hilty et al. 2017; Bonamy et al. 2023) veröffentlichen Best Practice Guidelines und ähnliche Dokumente, die bei der Verbesserung der Ressourcen- und Energieeffizienz von Software unterstützen sollen. Die hier genannten Auszüge sind nicht vollständig, aber weisen auf eine zunehmende Relevanz von Prinzipien ressourcen- und energieeffizienter Software in der Softwareentwicklung hin. Es lassen sich folgende inhaltliche Cluster bilden:

► Messung des Ressourcen- und Energiebedarfs

- „Mach Messen so einfach wie möglich“ (Bitkom e.V. 2021)
- „Verwenden Sie Messinstrumente, um die Leistung (performance) Ihres Codes zu bewerten. Ob es sich um die Menge der ausgetauschten Daten, die Ladezeit der Seite, die Ausführungszeit oder die Anzahl der Anfragen handelt – alles kann optimiert werden.“ (Ninassi und Benjamin 2021)

- „Ziehen Sie in Erwägung, den gesamten Ressourcenverbrauch als Teil Ihrer Integrationstests zu messen, um dem Produktmanagement und der Entwicklung Erkenntnisse über verbesserungswürdige Bereiche zu liefern.“ (SDIA 2024)
- „Melden Sie den Ressourcenverbrauch eines einzelnen Prozesses und machen Sie ihn auf der Client-Seite sichtbar.“ (SDIA 2024)

► Abwärtskompatibilität und Genügsamkeit

- „Die Entwicklung des digitalen Dienstes von morgen, der mit der Hardware von gestern funktioniert, reduziert nicht nur unzeitgemäße Hardwareänderungen, sondern ist auch einfacher zu warten und weiterzuentwickeln. Eine einfache SMS kann zur Übermittlung einer Nachricht verwendet werden, das Internet ist nicht die Antwort auf alles.“ (Ninassi und Benjamin 2021)
- „Anstatt die Leistung später zu optimieren, sollten Sie von Anfang an für jede Funktion eine minimale Ressourcennutzung vorschreiben.“ (SDIA 2024)

► Datensparsamkeit

- „Vermeiden Sie es, unnötige Daten zu speichern oder zu manipulieren. Es ist verlockend, Daten "für den Fall der Fälle" zu speichern. Dies ist jedoch bestenfalls schädlich für die Umwelt und schlimmstenfalls nach der Europäischen Datenschutzverordnung illegal. Der FAIR -Grundsatz beschreibt Möglichkeiten, Daten so zu verwalten, dass sie "leicht auffindbar, zugänglich, interoperabel und wiederverwendbar" sind. Die Anwendung dieser Methode ermöglicht es, die Auswirkungen zu verringern, indem eine bessere Interoperabilität und Wiederverwendung von Daten geschaffen wird.“ (Ninassi und Benjamin 2021)

► Offlinefähigkeit

- „Reduzieren Sie den von Ihrem digitalen Dienst erzeugten Datenverkehr auf ein Minimum: Sie verringern seinen ökologischen Fußabdruck und machen ihn für langsame Verbindungen zugänglich. Ideal ist es, ihn so zu gestalten, dass er gegen Verbindungsabbrüche resistent ist.“ (Ninassi und Benjamin 2021)

► Updates

- „Vermeiden Sie Komplexität. Code, der einfach zu verstehen und zu implementieren ist, ist auch langlebiger und lässt sich leichter weiterentwickeln. Dies ist die Grundlage des KISS-Prinzips: Keep It Simple and Stupid.“ (Ninassi und Benjamin 2021)
- „Setzen Sie auf Low-Tech: Die Entwicklung des digitalen Dienstes von morgen, der auf der Hardware von gestern funktioniert, verringert nicht nur die Zahl der unvorhergesehenen Hardwarewechsel, sondern ist auch einfacher zu warten und weiterzuentwickeln.“ (Ninassi und Benjamin 2021)

► Offene Lizenzen

- „Vermeiden Sie, dass andere das Rad neu erfinden: Wählen Sie, wann immer möglich, eine Open Source-Lizenz und tragen Sie so zur Welt der freien Software bei. Damit ermöglichen Sie nicht nur anderen, ihre eigenen Auswirkungen zu verringern, sondern Sie werden vielleicht auch von den Beiträgen der Gemeinschaft zu Ihrer Kreation angenehm überrascht sein!“ (Ninassi und Benjamin 2021)

Bei den Leitfäden fällt auf, dass die Empfehlungen größtenteils in dieselbe Richtung gehen. Zielkonflikte werden beispielsweise bei den Empfehlungen zu Messungen sichtbar. Einerseits sollen diese einfach sein, andererseits würden detailliertere Messungen für verschiedene Zielgruppen Handlungsoptionen zur Steigerung der Ressourcen- und Energieeffizienz ermöglichen.

Hinsichtlich der Abwärtskompatibilität sind die Empfehlungen unkonkret, es wird nicht genannt, wie viele Jahre eine Software mit Updates unterstützt werden sollte und wie weit die Abwärtskompatibilität zurückgehen sollte.

Zudem sind nicht alle Empfehlungen für sich genommen richtungssicher. Beispielsweise ist es denkbar, dass eine Software mit proprietärem Lizenzmodell weniger Energie benötigt als eine Software mit offenen Lizenzen. In ihrer Gesamtheit beschreiben die Empfehlungen jedoch ein gemeinsames Verständnis von Wirtschaft und Wissenschaft darüber, was ressourcen- und energieeffiziente Programmierung im Regelfall charakterisiert.

5 Verfahren und Werkzeuge zur Messung der Hardwareinanspruchnahme und des Energiebedarfes

Zum jetzigen Zeitpunkt existieren noch keine standardisierten oder normierten Verfahren zur messtechnischen Erfassung der Energie- und Ressourceneffizienz von Softwareprodukten. Es liegen jedoch bereits einige Vorgehensvorschläge zur Messung der Energie- und Ressourceninanspruchnahme von Software vor. Gleichzeitig nimmt die Anzahl an verfügbaren Eigenentwicklungen und Beiträgen zu Messmethoden verschiedener Akteure aus Wirtschaft und Wissenschaft kontinuierlich zu. Dabei wird üblicherweise zwischen softwarebasierten Messverfahren und direkter Messung der Hardware-Ressourcenbedarfe unterschieden (Jay et al. 2023; Guldner et al. 2024). Anhang A benennt einige der den Autor*innen bekannten Verfahren.

Typischerweise werden bei Messverfahren der Energie- und Ressourceneffizienz einer Software die Auslastung der relevanten Komponenten und teilweise auch die elektrische Leistungsaufnahme des Gesamtsystems oder einzelner Komponenten erhoben. Dabei sind verschiedene Messverfahren jeweils für verschiedene Hardware-konstellationen ausgelegt und optimiert. Einige Messgrößen sind auf manchen Hardwareplattformen kaum zugänglich oder die Aussagekraft der Messung ist beschränkt. Dies betrifft insbesondere den Energiebedarf von mobilen Endgeräten. Hier kommt hinzu, dass die Umweltwirkungen bei mobilen Endgeräten vom Herstellungsaufwand dominiert werden, bei Servern hingegen von dem Energiebedarf in der Nutzungsphase.

Die Bandbreite an verfügbaren Softwareprodukten, die auf unterschiedlichen Computerplattformen lauffähig sind, erfordert daher unterschiedliche Messaufbauten. Dies ist bedingt durch die Vielfalt von Software-Architekturen, Hardwaresystemen und Anwendungsszenarien. Verschiedene Methoden können teilweise in einem Messaufbau kombiniert werden oder sich in unterschiedlichen Messaufbauten ergänzen. Entsprechend wird für die Überarbeitung der Vergabekriterien empfohlen, das Konzept des Referenzsystems (vgl. (Umweltbundesamt 2020)) zu überarbeiten und auch für andere Messmethoden und Messtechnik zu öffnen. Darüber hinaus

sollen die Anforderungen an die Messung den messtechnischen Möglichkeiten und unterschiedlichen Umweltwirkungen der drei Hardwareplattformen mobile Endgeräte, Arbeitsplatz-Computer und Server gerecht werden (siehe Abschnitt 6.2), und gleichzeitig auch praxisnah und praktikabel sein.

Um die Qualität und die Aussagekraft der Messung der Ressourcen- und Energieeffizienz sicherzustellen, wird empfohlen, Anforderungen an die Messmethode zu stellen und diese bspw. durch Auditor*innen überprüfen zu lassen. Die Messungen sollen Transparenz über und einen Anreiz für ressourcenschonende Software schaffen. Dafür sollte die Messmethode nachvollziehbar, reproduzierbar, plausibel und richtungssicher sein.

Die Reproduzierbarkeit und Nachvollziehbarkeit könnte durch eine Dokumentation gewährleistet werden, welche die Messmethode inklusive genutzter Messinstrumente nachvollziehbar beschreibt.

Plausibilität und Richtungssicherheit der gemessenen Werte sollten zudem von neutralen Parteien geprüft werden. Auditor*innen sollen Unklarheiten identifizieren, damit diese von Antragstellenden bereinigt werden können. Die Plausibilitätsprüfung sollte sich dabei sowohl auf den Messaufbau als auch auf die Messergebnisse beziehen. So sollte bei einer Messung, deren Last mit einem repräsentativen Nutzungsszenario (Szenario-Test), das Softwareprodukt auf einer abgegrenzten Computerplattform ausführbar sein (vgl. Blauer Engel, (Umweltbundesamt 2020)). Zudem ist eine Grenze für die relative Standardabweichung der Ergebnisse wiederholter Messungen zu empfehlen, die bspw. 5 % nicht überschreiten sollte und ansonsten gesondert zu begründen wäre. Richtungssicherheit beschreibt in diesem Kontext die Eigenschaft der Messung, bei einer Optimierung des Software-Codes oder anderer technischer Rahmenbedingungen auch eine Verbesserung der Hardware-Inanspruchnahme sichtbar zu machen. Es sollte also einen erkennbaren Zusammenhang zwischen der Ausführung der Software und den gemessenen Werten für Energieverbrauch und Hardware-Inanspruchnahme geben. Es wird also empfohlen, aufgrund der Bandbreite an Softwareprodukten und Messverfahren neben der Messdatenerfassung per Standardnutzungsszenarien auch Langzeit-Tests im Live-Betrieb zuzulassen (vgl. Blauer Engel, (Umweltbundesamt 2020)). Gleichzeitig sollte das Verfahren für potenzielle Antragstellende des Umweltzeichens keinen übermäßigen Aufwand darstellen und praktikabel bleiben (Guldner et al. 2024; Kern et al. 2018).

Nachfolgend werden die wesentlichen empfohlenen Elemente einer Messmethode exemplarisch nachgezeichnet. Dazu gehören die Lasterzeugung, der Messaufbau (die Hardwarekonstellation), und die Datenauswertung und -visualisierung.

5.1 Lasterzeugung während des Messvorgangs

Um eine repräsentative Last des zu zertifizierenden Softwareprodukts zu erzeugen, gibt es unterschiedliche Möglichkeiten. Die Hardwareinanspruchnahme der Software kann entweder unter realen Bedingungen untersucht werden (in Folgenden als Messung im Live-Betrieb, oder Langzeit-Test bezeichnet) oder es wird wiederholt eine (realistische) Last simuliert (in Folgenden als „Szenario-Test“ bezeichnet). Da die Durchführung eines Szenario-Tests nicht immer praktikabel ist (siehe Abschnitt 5), wird empfohlen, zusätzlich zu den Szenario-Tests auch Langzeit-Tests zur Lasterfassung eines Softwareproduktes zuzulassen. Der Langzeit-Test bietet sich beispielsweise dann an, wenn sich eine Software (in der Regel auf einem Server device) im produktiven Einsatz befindet und eine repräsentative Anzahl an Anwender*innen diese Software nutzen. In diesem Fall kann die Ressourcen-Inanspruchnahme durch einzelne Nutzende nicht mehr separat erfasst werden. Statt einem generischen Standardnutzungs-

szenario findet eine reale Nutzung der Software im Live-Betrieb statt. Der Live-Betrieb kann in diesem Fall als repräsentative Nutzung angesehen werden.

Bei Client-Server Software kann es gegebenenfalls erforderlich sein, mehr als einen Server und/oder mehr als einen Client in die Messung mit einzubeziehen. Um hier den Aufwand möglichst gering zu halten, wird dazu geraten, nur die Daten und Messwerte eines einzelnen repräsentativen Clients (PC-device) und eines repräsentativen Servers (Server-device) zu erheben, sofern dies zum Softwareprodukt passt. Empfehlenswert ist jedoch die Angabe der Anzahl der (simulierten) Nutzenden während der Messung. Alle Geräte, die für eine repräsentative Nutzung benötigt werden, werden im Folgenden in diesem Dokument zusammen als System unter Test (SUT) bezeichnet.

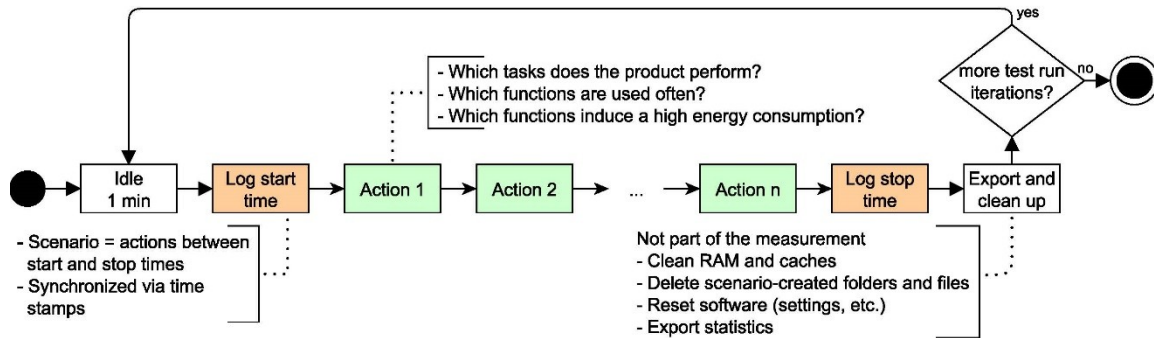
Bereits Hilty et al. (2017) schlagen vor, so ein reproduzierbares Standardnutzungsszenario zu entwickeln, das wiederholt ausgeführt werden kann. Ein solcher Szenario-Test könnte dann zum Einsatz kommen, wenn es möglich ist, die Hardware und das Softwareprodukt so zu isolieren, dass dessen Energieverbrauch bzw. Hardwareinanspruchnahme gemessen werden kann. Abbildung 1 zeigt eine Schema eines Szenario-Tests (Gröger et al. 2018). Ein Szenario bezeichnet dabei beliebig viele Aktionen zwischen Start- und Endzeit, dessen Synchronisation über Zeitstempel erfolgt. In Praxiserfahrungen hat sich folgendes Vorgehen bewährt. Es wird dabei dazu geraten, Werks- und Standardeinstellungen zu nutzen. Auf Nachfrage sollten Abweichungen von der Standardkonfiguration dem Auditor erklärt werden.

- ▶ Vor dem Start wird für eine Minute der Leerlauf Zustand aufrechterhalten, damit Hardwarekomponenten zu ihrem Ruhezustand zurückkehren und abkühlen können (die Temperatur der Hardware hat einen Einfluss auf den Energieverbrauch).
- ▶ Start des Logging (Startzeit) der Messwerte und die Ausführung der Aktionen. Die durchgeführten Aktionen ergeben sich aus den folgenden Fragen:
 - Welche Aufgaben werden mit dem Produkt erledigt?
 - Welche Funktionen werden häufig genutzt?
 - Welche Funktionen induzieren hohen Energiebedarf?
- ▶ Nachdem die letzte Aktion durchgeführt wurde, wird das Logging beendet (Endzeit).

Nicht Teil der Messung ist das darauffolgende Exportieren der Messwerte und ggf. das Aufräumen des Systems, sodass der Zustand vor der Szenario-Durchführung wiederhergestellt wird, um die folgenden Iterationen des Szenarios zu messen, z.B. werden RAM und Caches geleert, angelegte Ordner und Dateien gelöscht und ggf. Statistiken exportiert.

Dann beginnt das Szenario von vorne, zunächst mit der vorgelagerten Idle-Zeit. Um Schwankungen auszugleichen, die beispielsweise durch Hintergrundaktivitäten des Betriebssystems entstehen, und statistisch auswertbare Messwerte zu erhalten, hat es sich bewährt, die Messung ca. 30-mal zu wiederholen.

Abbildung 1: Exemplarischer Ablauf eines Szenario-Tests



Quelle: Guldner et al. (2024)

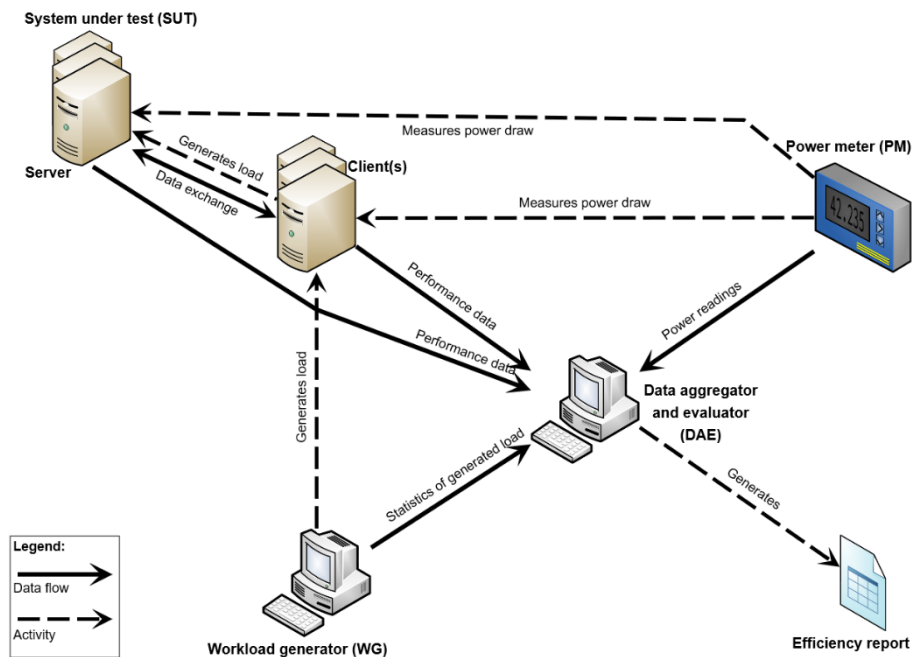
Bei verteilten Systemen ergeben sich Unterschiede im Vergleich zu lokaler Desktopsoftware. Die Hardwareinanspruchnahme und der Energiebedarf von Servern sind im Regelfall stark abhängig von der Last, die durch Client-seitige Anfragen (requests) erzeugt wird. Je mehr Geräte mit dem Server interagieren, desto mehr Last wird auf diesem erzeugt. Server sind dafür ausgelegt, Anfragen von mehreren Geräten zu bedienen, weshalb die Interaktion mit einem einzigen Gerät zu einer geringen, kaum messbaren Auslastung führen kann. Deshalb bietet es sich für Server an, die Leistungs- und Ressourceninanspruchnahme über einen längeren Zeitraum hinweg unter Realbedingungen im Live-Betrieb zu messen. Unter solchen Realbedingungen ist es für Server oft schwierig, ihren Dienst für eine Messung im Leerlauf zu pausieren. Stattdessen wird, wie bereits beschrieben, vorgeschlagen, dass ein typisches Verhalten von Nutzenden über einen längeren Zeitraum live aufgezeichnet wird, um eine repräsentative Ressourceninanspruchnahme zu ermitteln (Langzeit-Test).

5.2 Exemplarische Messaufbauten

Ein Überblick über ausgewählte Messmethoden des Ressourcenbedarfs von Softwareprodukten wird im Anhang A gegeben. Daraus geht hervor, dass verschiedene Methoden unterschiedliche Softwaretypen und Hardwarekomponenten messen und dazu unterschiedliche Messaufbauten nutzen.

Abbildung 2 zeigt einen exemplarischen Messaufbau. Im Zentrum steht der Data Aggregator and Evaluator (DAE), der alle zur Auswertung der Messung notwendigen Daten sammelt. Der Workload Generator generiert Last auf dem SUT und sendet die Statistik der generierten Last an den DAE. Das Power Meter (PM) misst die Leistungsaufnahme des System Under Test und sendet die Leistungsmesswerte an den DAE. Das System Under Test, in diesem Fall ein Client sendet seine Performance-Daten an den DAE. Der DAE generiert abschließend aus allen erhaltenen Daten einen Efficiency Report.

Abbildung 2: Exemplarischer Messaufbau zur Bestimmung der Hardwareinanspruchnahme und des Energiebedarfes



Quelle: eigene Darstellung basierend auf Gröger et al. (2018)

5.3 Exemplarische Auswertungen von Messungen

Die folgenden Abbildungen (Abbildung 3 bis Abbildung 10) zeigen die Auswertungen beispielhafter Messungen eines Client-Server-Systems (hier ein Content Collaboration System) sowie einer Videoplayer Desktop-Anwendung mit jeweils insgesamt 30 Messdurchläufen, die jeweils übereinandergelegt wurden. Die rote Linie stellt die Sekundenmittelwerte der Mess-Iterationen dar. Es wird deutlich, dass wiederholte Messungen des gleichen Produkts unterschiedliche Werte ergeben. Dennoch sind die Abweichungen vom Mittelwert deutlich kleiner als die lastbedingten Ausschläge. Statistische Auswertungen des Mittelwerts und der Standardabweichung geben Aufschluss über die Genauigkeit und Schwankungsbreite der Messung.

In der beispielhaften Auswertung werden zunächst Übersichten erzeugt, aus denen die erforderlichen gemittelten Werte hervorgehen. Anhand dieser können erste Aussagen getroffen werden, wie die Last des Softwaresystems verteilt ist und wie hoch die Hardwarekomponenten im Mittel ausgelastet sind.

Client-Server-Softwareprodukt

In diesem exemplarischen Szenario wurde zunächst testweise die Automatisierung gestartet, die Software auf dem Client im Browser geöffnet (Aktion 1+2), danach verschiedene Dateitypen (Aktion 3 bis 6) und anschließend eine größere Datei hochgeladen (Aktion 7). Diese Datei wurde im Anschluss mit verschiedenen Methoden online bearbeitet (Aktion 8 bis 12) und die hochgeladenen Daten gedownloadet (Aktion 13). Zum Schluss wurden sowohl die lokalen sowie die hochgeladenen Dateien gelöscht (Aktion 14-17) und das Programm geschlossen (Aktion 18). Die Laufzeit des Szenarios für das Client-Server-System beträgt 5:45 Minuten.

Nachfolgende Tabelle 1 und Tabelle 2 geben eine Übersicht über die Mittelwerte der elektrischen Leistung und Arbeit, CPU-Auslastung, RAM-Belegung, über das Netzwerk übertragenen Datenmenge sowie Permanentspeicherbelegung jeweils für den Server und den Client.

Tabelle 1: Exemplarische Auswertungsübersicht eines Client-Server-Softwareprodukts im 8-minütigen Messvorgang des Servers

Größe	Gemessener Wert (Szenario)	Gemessener Wert (Baseline)
Mittlere el. Leistung	29,34 W	26,78 W
Mittlere el. Arbeit	4,03 Wh	3,67 Wh
Mittlere CPU-Auslastung	2,90 %	0,15 %
Mittlere RAM-Belegung	0 %	0 %
Über Netzwerk übertragene Datenmenge	167,04 Mbit	0 Mbit
Permanentspeichernutzung	2 Mbit	1,68 Mbit

Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

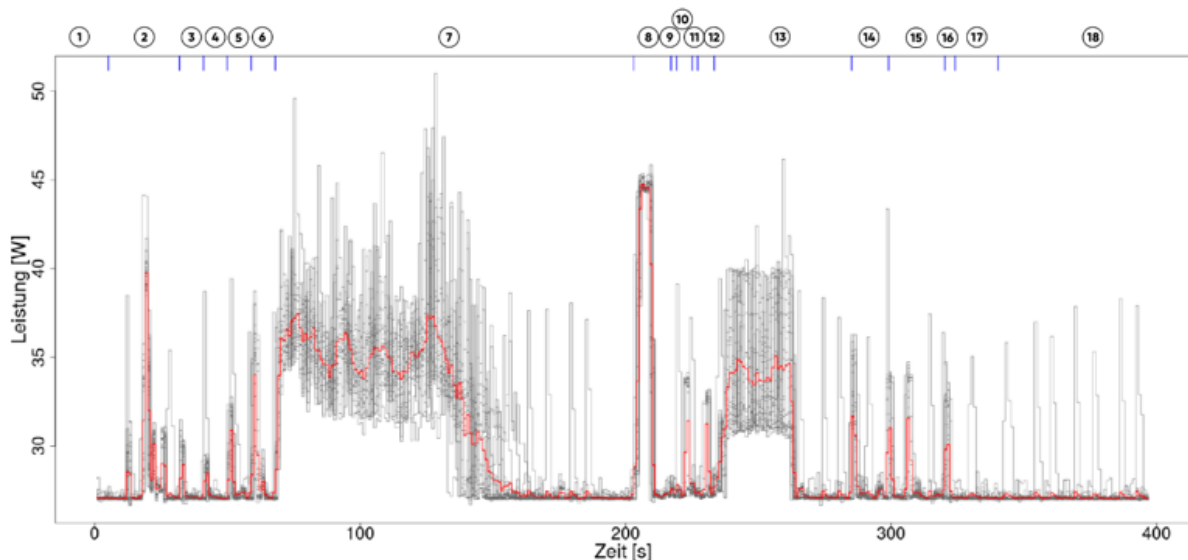
Tabelle 2: Exemplarische Auswertungsübersicht eines Client-Server-Softwareprodukts im 8-minütigen Messvorgang des Clients

Größe	Gemessener Wert (Szenario)	Gemessener Wert (Baseline)
Mittlere el. Leistung	14,63 W	11,59 W
Mittlere el. Arbeit	2,01 Wh	1,59 Wh
Mittlere CPU-Auslastung	6,57 %	0,05 %
Mittlere RAM-Belegung	0,06 %	0 %
Über Netzwerk übertragene Datenmenge	166,88 Mbit	0 Mbit
Permanentspeichernutzung	4 Mbit	0 Mbit

Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

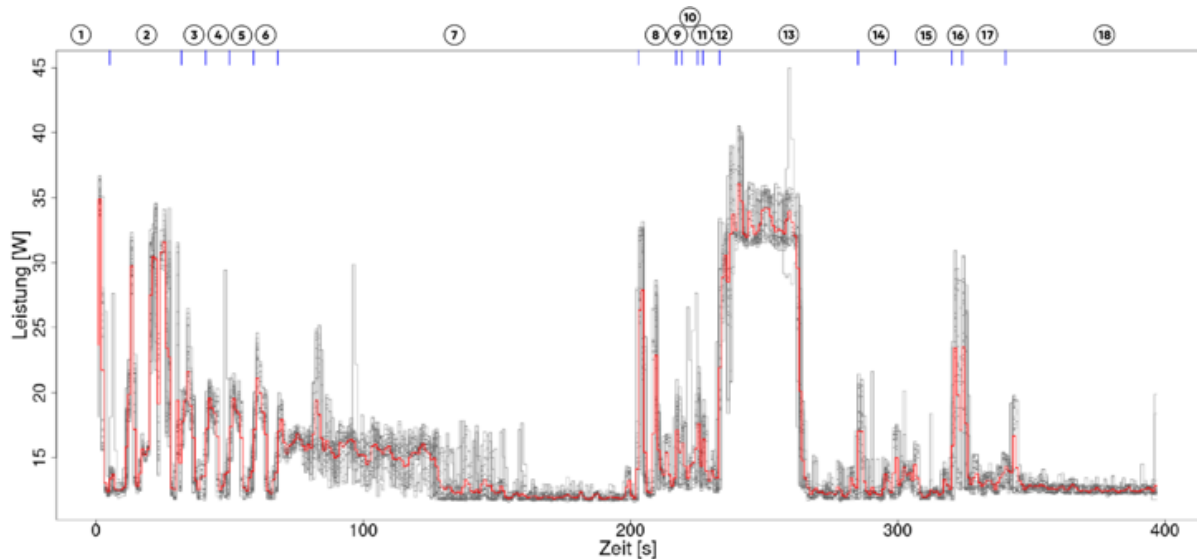
Neben der Übersicht der einzelnen Größen werden in diesem Fall verschiedene Grafiken erzeugt. Anhand dieser können veranschaulichte Aussagen über den Ablauf der Messungen über die Zeit getroffen werden. Beispielhaft wird dafür die Leistungsaufnahme des Serversystems in Abbildung 3 dem parallel gemessenen Client-System in Abbildung 4 gegenübergestellt. Bei der Betrachtung der gemittelten Werte ist ein ähnliches Muster bei der Ausführung bestimmter Operationen erkennbar. Sichtbar ist damit einhergehend auch der optisch einem Rechteck gleichende Ausschlag bei Aktion 13 um Sekunde 250. In diesem Zeitraum wird eine aus Aktion 7 hochgeladene Datei wieder heruntergeladen. Dieser Anstieg spiegelt sich in beiden Komponenten des verteilten Systems wider und ist ein Indiz dafür, dass beide Komponenten an der gleichen Aufgabe arbeiten.

Abbildung 3: Leistungsaufnahme eines Client-Server-Softwareprodukts (Server-System)



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

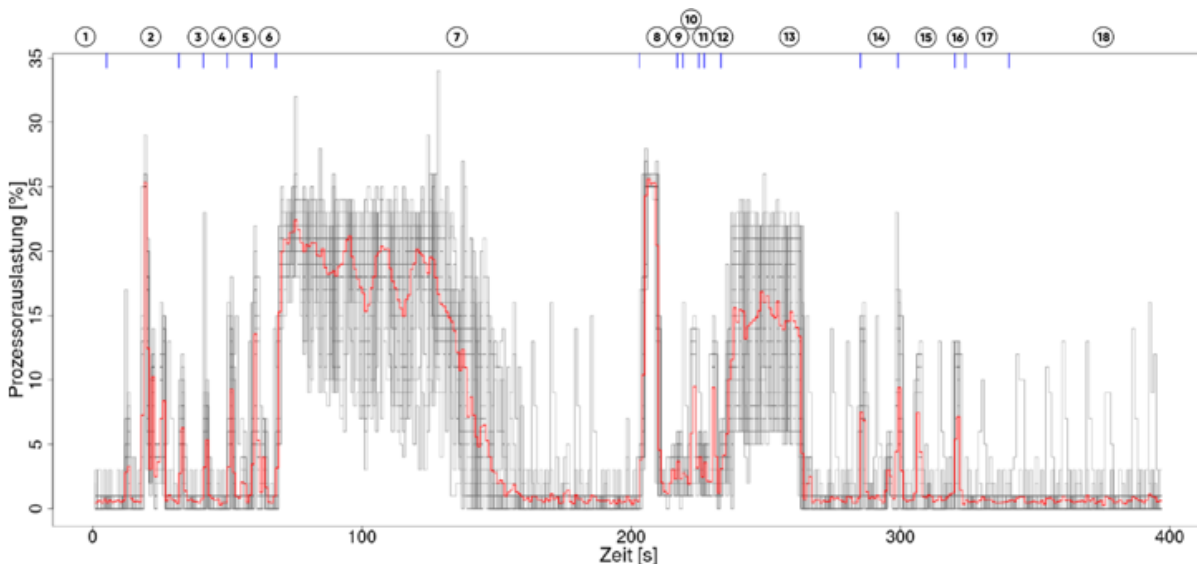
Abbildung 4: Leistungsaufnahme eines Client-Server-Softwareprodukts (Client-System)



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

Die Prozessorauslastung in Abbildung 5 spiegelt sich in der zuvor betrachteten Leistungsaufnahme wider. Sie ist eine der einflussreichsten Komponenten hinsichtlich des Energiebedarfs des Server-Systems. Die pro Sekunde gemittelte Auslastung der CPU des Servers beträgt im Maximum etwa 26 %.

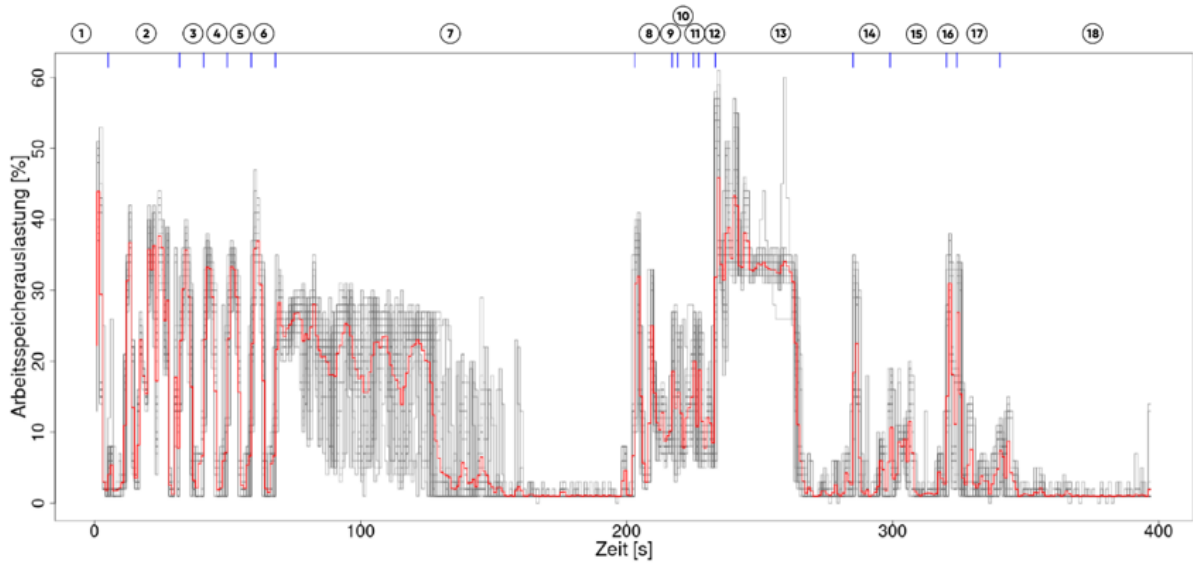
Abbildung 5: Prozessorauslastung eines Client-Server-Softwareprodukts (Server-System)



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

Ebenso ist bei der Betrachtung des Arbeitsspeichers des Clients in Abbildung 6 das gleiche Muster komponentenübergreifend erkennbar. Es kann also anhand der Grafiken davon ausgegangen werden, dass bei Aktion 13 m Zeitraum ab 250 Sekunden bis 280 Sekunden ein speicherintensiver Prozess durchgeführt wird. Dieser ist sowohl in Betrachtung der Leistungsaufnahme wie auch der Hardwareressourcennutzung erkennbar.

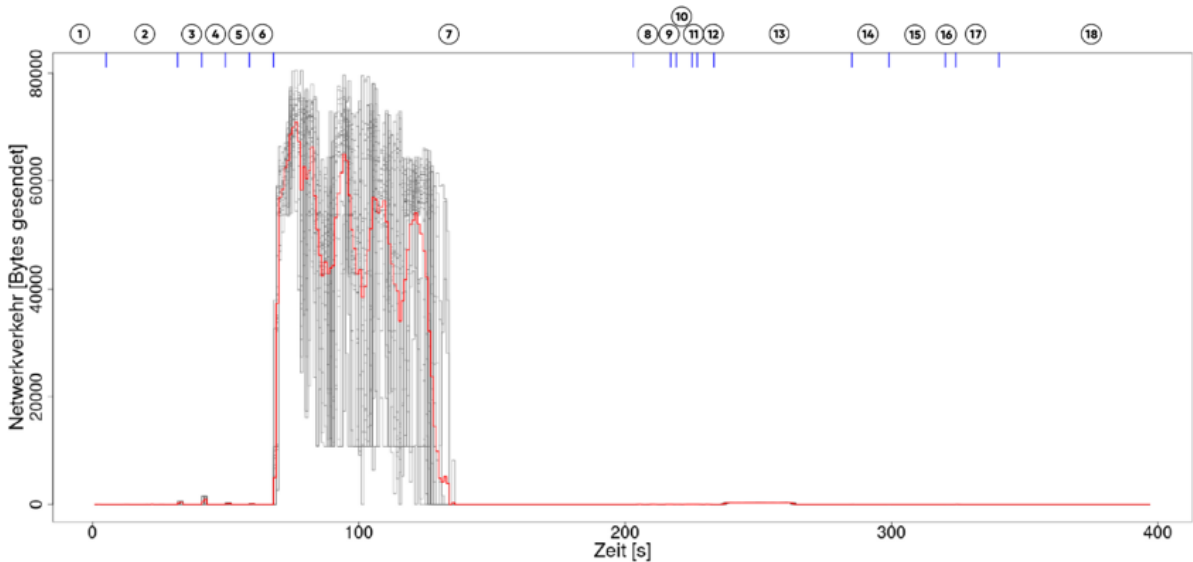
Abbildung 6: Arbeitsspeicherauslastung eines Client-Server-Softwareprodukts (Client-System)



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

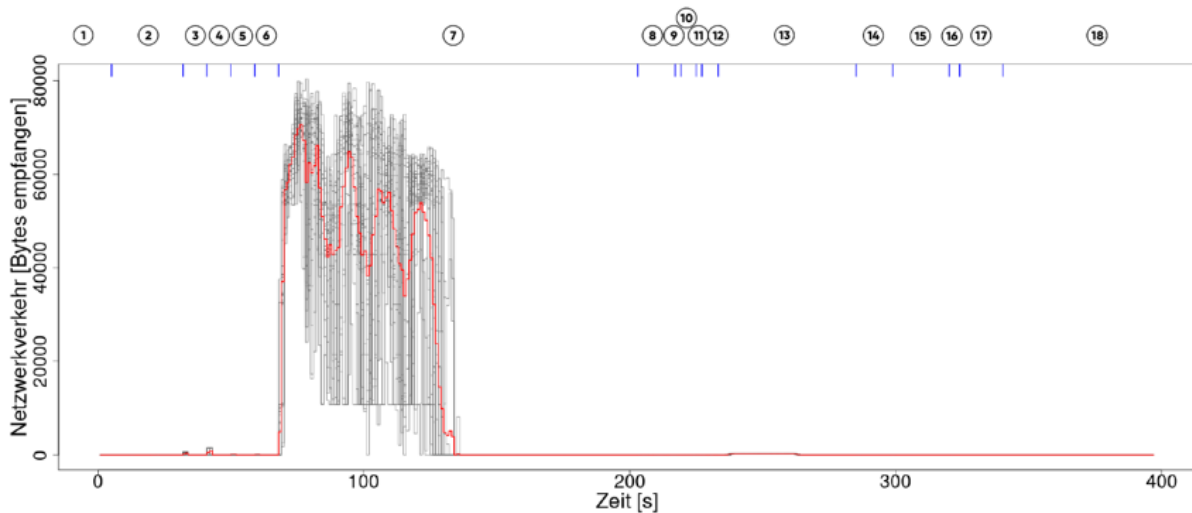
Die Aktion 7 des Nutzungsszenarios beinhaltet das Hochladen einer großen Datei auf das Server-System. Das Muster und die damit einhergehende Datenmenge der gesendeten Bytes durch das Client-System in Abbildung 7 entspricht nachvollziehbar dem Muster der empfangenen Datenbytes seitens des Server-Systems in Abbildung 8.

Abbildung 7: Gesendeter Netzwerkverkehr eines Client-Server-Softwareprodukts (Client-System)



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

Abbildung 8: Empfangener Netzwerkverkehr eines Client-Server-Softwareprodukts (Server-System)

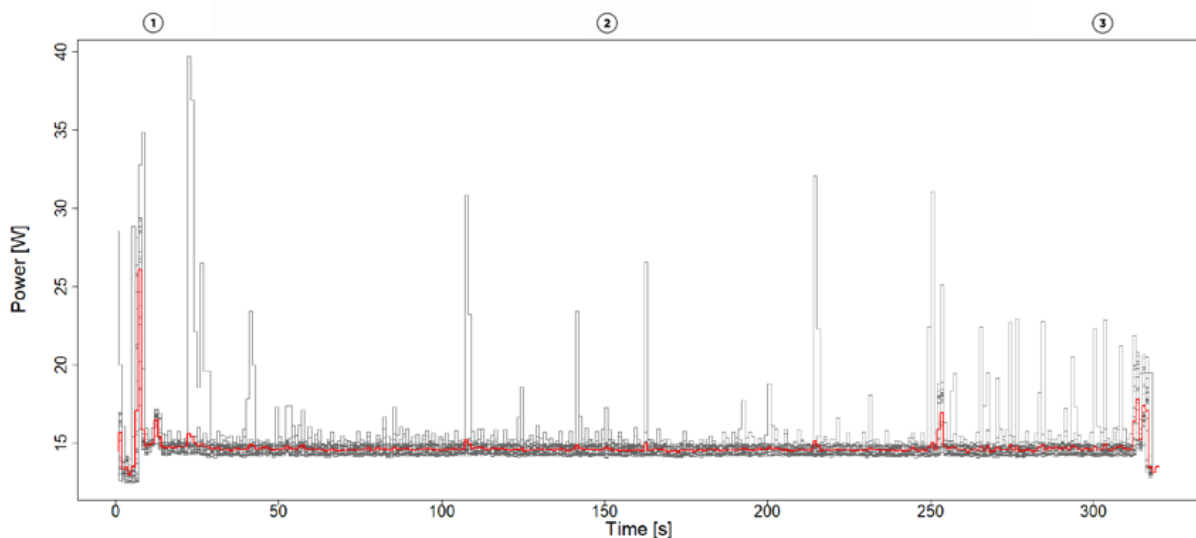


Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

Desktop-Softwareprodukt

Im Folgenden wird die Auswertung der beispielhaften Messung eines Desktop-Softwareprodukts mit insgesamt 30 Messdurchläufen betrachtet. Die Abbildungen Abbildung 9 und Abbildung 10 zeigen dabei die Kurven der Leistungsaufnahme während des Abspielens eines Videos in einer Desktopanwendung. Das Szenario beinhaltet den Start der Anwendung (Aktion 1), das Abspielen des Videos (Aktion 2), das direkt nach Aktion 1 um Sekunde 5 beginnt und kurz vor dem Beenden der Anwendung (Aktion 3) bei Sekunde 320 endet. Einzelne stark abweichende Ausschläge, welche in den grauen Linien erkennbar sind, gehen in dem Mittelwert über 30 Durchläufe unter. Die Ausschläge zu Beginn und am Schluss der Messung, die bei allen Durchläufen vorkommen, können auf das Starten und Beenden des Videos zurückgeführt werden.

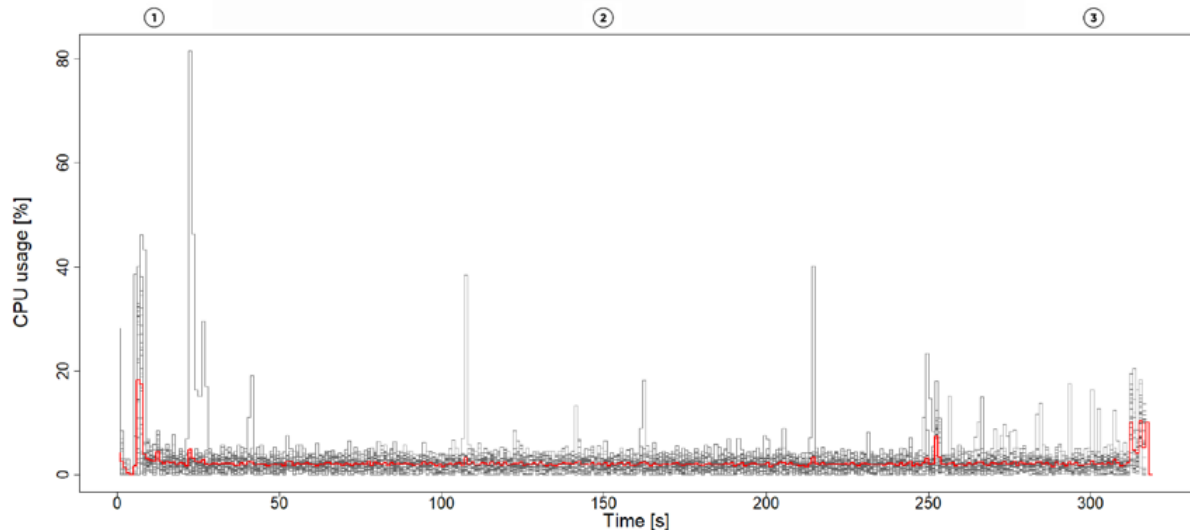
Abbildung 9: Leistungsaufnahme eines Softwareprodukts im 6-minütigen Messvorgang



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

In Abbildung 10 ist die Auslastung der CPU über der Zeit dargestellt. Die Ausschläge ergeben ein ähnliches Muster wie bei der Leistungsaufnahme in Abbildung 9.

Abbildung 10: Prozessorauslastung eines Softwareprodukts im 6-minütigen Messvorgang



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

Wie in Kapitel 4.1.2 dargelegt, soll geprüft werden, welche IP-Adressen von einer Anwendung kontaktiert werden, um Datenverkehr durch Werbung oder Tracking ausschließen zu können. Daher wurden in einem weiteren Szenario-Test die aufgerufenen IP-Adressen und URLs mit einem Analysewerkzeugs aufgezeichnet. Das Ergebnis ist in Tabelle 3 aufgelistet. In der 5-minütigen Messung wurde der durch die Aufrufe erzeugte Datenverkehr aufgeschlüsselt und die daraus resultierende Menge an Paketen manuell analysiert. Diese wurden mit den aufgerufenen IP-Adressen und URLs während der Messung der Baseline abgeglichen, um betriebssystembedingten Datenverkehr zu identifizieren. An dieser Stelle ist es empfehlenswert, die zu untersuchende Software genau zu kennen, um einschätzen zu können, welcher Netzwerkverkehr zum Betriebssystem und welcher zur eigentlichen Software gehört.

Tabelle 3: Aufgerufene IP-Adressen und URLs eines Softwareprodukts im 5-minütigen Messvorgang

Liste aller aufgerufenen IP-Adressen	URL	Angabe zum Eigentümer (eigener oder externer Dienst)	Frequenz: Anzahl Pakete/Minute	Hinweis
176...	http://...	eigener Dienst	5	
194...	http://...	eigener Dienst	1598	
62...	http://...	eigener Dienst	4,4	
52...	http://...	externer Dienst	9,2	Betriebssystembedingt
224...	http://...	externer Dienst	1,4	
224...	http://...	externer Dienst	8,4	Broadcastadresse
255..	http://...	-	10,4	

Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

6 Ableitung der Vergabekriterien für ein Umweltzeichen

Die Kriterien für den Blauen Engel DE-UZ 215, Version 1 sind von Gröger et al. (2018) und von Naumann et al. (2020) bereits hergeleitet und ausführlich begründet worden. Seitdem ist das Umweltzeichen zweimal ohne Veränderungen verlängert worden, sodass die Version 3 die zuletzt gültige Fassung ist. Nachfolgend werden die überarbeiteten Vergabekriterien abgeleitet, die sich passagenweise wortgleich in dem separaten Dokument der Vergabekriterien wiederfinden (Blauer Engel 2024).

Das Softwaredesign und die Entwicklung der Software haben Einfluss auf die Dimensionierung und Lebensdauer der Hardware, ebenso wie auf die Höhe des Energiebedarfs durch die Nutzung der Software. Mit dem Umweltzeichen Blauer Engel für „Ressourcen- und energieeffiziente Softwareprodukte“ (DE-UZ 215) können Produkte gekennzeichnet werden, die bezüglich ihres Energieverbrauchs und ihrer Hardware-Inanspruchnahme besonders transparent sind, die bei Updates auch auf älteren Computern lauffähig bleiben und deren Energieverbrauch bei Updates nicht stark ansteigt. Aufgrund geringerer Leistungsanforderungen wird eine längere Hardware-Nutzungsdauer möglich. Darüber hinaus zeichnet sich die Software durch eine hohe Transparenz aus und ermöglicht Nutzenden besondere Freiheiten im Umgang mit der Software.

Nachfolgend werden vor allem die vorgeschlagenen Überarbeitungen und die sich daraus ergebenden Änderungen am Vergabedokument beschrieben und begründet. Die Änderungen ergeben sich aus einer Vielzahl an Teiluntersuchungen, wie einer Literaturanalyse, eigenen Experimenten, Interviews mit Expert*innen und einem Beteiligungsverfahren, in dessen Rahmen mehrere Workshops mit Software-Entwickler*innen und weiteren interessierten Stakeholdern stattgefunden haben.

6.1 Geltungsbereich

Die Ausweitung des Untersuchungsgegenstandes in Kapitel 2 hat sich durch die weiteren Untersuchungen hindurch als praktikabel erwiesen. Die Analyse der Hemmnisse (siehe Abschnitt 3.2.2) hat ergeben, dass der Geltungsbereich von Anwendungssoftware mit Nutzerschnittstelle, die ihre hauptsächliche Last lokal ausführt, nicht ausreichend ist. Auch die Analyse des Marktes (Abschnitt 3.1) impliziert, dass der Geltungsbereich um Anwendungssoftware für mobile Geräte (Apps) und um verteilte Systeme erweitert werden sollte. Neben Anwendungssoftware für Konsument*innen soll der Geltungsbereich explizit auch Unternehmenssoftware beinhalten. Es gibt genug Verfahren und Werkzeuge (Kapitel 5), um einen erweiterten Geltungsbereich sinnvoll messtechnisch zu untersuchen.

Software ist ohne die darunterliegende Hardware nicht ausführbar. Je nach **Gerätetyp** unterscheiden sich wesentliche Parameter, bspw. wie viele Hardwarekapazitäten eine Software in Anspruch nehmen kann. Für diesen Überarbeitungsvorschlag werden drei Gerätetypen unterscheiden:

- **pc device:** Personal Computer zum überwiegend stationären Einsatz wie beispielsweise Desktop-PC, Workstation, Laptop oder Convertible jeweils mit einem Desktop-Betriebssystem (Windows, MacOS, Linux oder vergleichbar).
- **mobile device:** Batteriebetriebenes tragbares Endgerät wie beispielsweise Smartphone, Tablet, Smartwatch mit einem mobilen Betriebssystem (Android, iOS oder vergleichbar).

- **server device:** Server-Hardware zur zentralen Datenverarbeitung, betrieben in der Regel in einem Rechenzentrum oder Serverraum. Der Server verfügt über eine Netzwerkverbindung und kann mit mehreren Clients gleichzeitig Daten austauschen. Ein Server verfügt in der Regel nicht direkt über Eingabe- oder Ausgabegeräte (Human Interface Devices). Die Kommunikation mit dem Server erfolgt stattdessen softwaregetrieben über die Netzwerkschnittstelle.

Um sicherzustellen, dass die wesentlichen Bestandteile der Software und deren Umweltwirkungen erfasst werden, werden dazu folgende Einschränkungen vorgeschlagen.

Nicht unter den Geltungsbereich dieses Umweltzeichens fallen Softwareprodukte,

- bei denen mehr als 10 % des IKT-bedingten Energiebedarfs in der Nutzungsphase der Software außerhalb der oben genannten Computer-Plattformen aufgewendet wird (z.B. Software die rechenintensive *externe* Cloud- und Datendienste nutzt, Software zum Einsatz in Routern, Netzwerkkomponenten, eingebetteter Elektronik, Druckern, Haushaltsgeräten, Unterhaltungselektronik),
- der Energiebedarf im Betrieb aufgrund fehlender Datenverfügbarkeit nicht bilanziert werden kann (z.B. Ausführung der Software in Cloud-Rechenzentren, auf die der Antragsteller keinen messtechnischen Zugriff hat, Peer-to-Peer-Architekturen, Edge-Computing, Nutzung rechenintensiver externer Cloud- und Datendienste),
- der IKT-bedingte Energiebedarf zur Entwicklung der Software höher ist als der Energiebedarf in der Nutzungsphase der Software (z.B. Erstellen von Datenbanken, Machine Learning, Anwendungen der Künstlichen Intelligenz), bei einer angenommenen Nutzung über den Zeitraum eines Jahres und einer repräsentativen Anzahl an genutzten Softwareprodukten.

Bei einigen Anforderungen wird dabei unterschieden, auf welcher Computer-Plattform das Softwareprodukt ausgeführt wird. Der Nachweis solcher Anforderungen muss dann nur für die genannten Computer-Plattformen erbracht werden.

6.2 Ressourcen- und Energieeffizienz

Ein Softwareprodukt soll seine Funktionalität mit einem minimalen Ressourcenaufwand und Energiebedarf erbringen. Die Ressourcen- und Energieeffizienz des Softwareprodukts soll optimiert werden. Zur Operationalisierung der Ressourceneffizienz werden als Bezugsgrößen genutzte Hardwarekapazitäten und Energiebedarfe verwendet.

6.2.1 Erforderliche minimale Systemvoraussetzungen

Je weniger „anspruchsvoll“ ein Softwareprodukt ist, desto kleiner sind seine direkten und indirekten Umweltwirkungen. Steigende Anforderungen an die Hardware sind eine relevante Ursache für Software-induzierte Obsoleszenz. Software soll nicht dazu führen, dass neue Hardware beschafft werden muss und funktionierende Hardware nicht weiter genutzt werden kann. Software ist allerdings so vielfältig, dass hier keine direkte Kategorisierung möglich ist, und so wenig vergleichbar, dass keine allgemeinen Mindestanforderungen an die Genügsamkeit von Softwareprodukten formuliert werden können. Deshalb fordert dieses Kriterium

Transparenz. Die Genügsamkeit der Software kann somit Beschaffungs- und Entwicklungsentscheidungen beeinflussen.

Die Anforderungen, welche die Software an die Hardware-Ausstattung und den vorausgesetzten Softwarestack stellt, sollen sowohl für die Nutzer*innen als auch für die Fachöffentlichkeit transparent sein.

6.2.2 Dokumentation des Messaufbaus

Die Bestimmung der Ressourcen- und Energieeffizienz soll wissenschaftlichen Standards genügen, um dem Umweltzeichen Glaubwürdigkeit zu verleihen. Dazu gehören Plausibilität, Nachvollziehbarkeit, Repräsentativität und Reproduzierbarkeit der Messergebnisse (siehe Kapitel 5).

In den letzten Jahren wurden diverse Verfahren und Methoden entwickelt, um die Hardwareinanspruchnahme und den Energiebedarf von Software zu bestimmen (siehe Abschnitt 4.2).

Dieser Entwicklung Rechnung tragend, werden im Blauen Engel Qualitätsanforderungen an den Messaufbau und die Durchführung der Messung formuliert (Umweltbundesamt 2020). Dazu gehört bspw., dass die Messungen unter Last mindestens 10-mal wiederholt werden und die Streuung der Ergebnisse klein sein muss. Die relative Standardabweichung der in den Wiederholungen ermittelten Messwerte sollte nicht größer sein als 5 %.

Die Messergebnisse der Hardwareinanspruchnahme und des Energiebedarfs der Software sind erst zusammen mit der Dokumentation der Hardware und den Spezifika der verwendeten Last, z.B. mit der Dokumentation des Standardnutzungsszenarios, aussagekräftig.

6.2.3 Messung der Grundauslastung und der Last des Softwareprodukts im Leerlaufzustand

Die Energie- und Hardwareinanspruchnahme des Softwareprodukts verursachen eine wesentliche Umweltwirkung (siehe Kapitel 4.1)

Der Hersteller des Softwareproduktes soll die Möglichkeit und einen Anreiz haben, die Energie- und Hardwareinanspruchnahme des Softwareprodukts zu optimieren. Dazu müssen Energie- und Hardwareinanspruchnahme des Softwareprodukts transparent sein.

Die Grundauslastung ist die Energie- und Hardwareinanspruchnahme des unterliegenden Softwarestacks (also des Betriebssystems und aller zum Betrieb des Softwareprodukts notwendigen Software, aber ohne das Softwareprodukt selbst). Diese Software dient der Ausführbarkeit des Softwareproduktes, weshalb die Grundauslastung anteilig dem Softwareprodukt zugeteilt werden muss. Die Berechnungsvorschrift von Gröger et al. (2018) wird weiterhin als geeignet betrachtet.

Viele Softwareprodukte, die eine Nutzer-Interaktion zulassen, sind die meiste Zeit ihres Betriebs im Leerlaufzustand. Die Energie- und Hardwareinanspruchnahme im Leerlauf zu minimieren, hat daher einen relevanten Einfluss auf die Umweltwirkung der Software.

Die belastbare Messung des Energiebedarfs mobiler Apps stellte sich in eigenen Untersuchungen als herausfordernd dar. Zum einen beeinflusst der Akku die Messungen, da dieser einem gewissen Alterungsprozess unterliegt und auch die Spannung abhängig vom Ladestatus variiert. Selbst wenn die Akku-bedingten Einflüsse ausgeschlossen werden (z.B. durch Ausbau und direkter Messung), können weitere Schwankungen nicht vollkommen ausgeglichen werden. Der Einfluss weiterer Faktoren auf die Auslastung und Leistungsaufnahme, beispielsweise der

Sensoren, anderer Apps, der Ausführungsumgebung und des Betriebssystems, kann bei mobilen Geräten nicht vernachlässigt werden und ist oft unvorhersehbar. Es zeigte sich, dass die relative Abweichung bei der mehrfachen Messung des Energiebedarfs von mobilen Apps wesentlich größer ist als bei anderen Hardware-Plattformen und nicht den Ansprüchen an Repräsentativität und Reproduzierbarkeit der Messergebnisse genügt. Daher wird das Erfassen des Energiebedarfs mobiler Apps zwar als sinnvoll bewertet, ist für den Blauen Engel jedoch weniger praktikabel. Bei mobilen Apps ist dagegen die Anbindung an externe Dienste aller Art besonders präsent und Nutzenden gegenüber intransparent. Daher wird empfohlen, die aufgerufenen URLs und die Menge des Datenverkehrs transparent zu machen.

6.2.4 Messung der Energie- und Hardwareinanspruchnahme während der Nutzung

Ebenso relevant sind die Energie- und Hardwareinanspruchnahme des Softwareprodukts während der Nutzung (siehe Kapitel 4.1). Diese sollen ebenso plausibel, nachvollziehbar, repräsentativ und reproduzierbar ermittelt und kommuniziert werden.

Dafür eignen sich die Simulation einer repräsentativen Nutzung mithilfe eines Standard-nutzungsszenarios oder die Messung in einer produktiven Umgebung unter realen Bedingungen (siehe Kapitel 5.1)

Darüber hinaus werden in Anhang A der Vergabekriterien Qualitätsanforderungen an den Messaufbau und die Durchführung von Messungen definiert (Umweltbundesamt 2020)).

6.3 Potenzielle Hardware-Nutzungsdauer

Ein Softwareprodukt soll nicht dazu beitragen, dass vorhandene Hardware vorzeitig durch eine leistungstärkere Hardware ersetzt werden muss, weil sie den Leistungsanforderungen der Software nicht gerecht wird. Software-Updates sollen darüber hinaus nicht dazu führen, dass auch ein Hardware-Update erfolgen muss. Stattdessen muss dem Nutzenden eine unabhängige Entscheidung über Software- und Hardware-Erneuerung ermöglicht werden. Dies reduziert die Umweltwirkungen im Bereich des Materialverbrauches von Software (siehe Abschnitt 4.1.3).

6.3.1 Abwärtskompatibilität

Tools zum Kompilieren der Software, die von Gatekeepern zur Verfügung gestellt werden, unterstützen meist ausschließlich das aktuelle Betriebssystem oder solche Betriebssysteme, die nicht älter sind als 5 Jahre. Dies trifft insbesondere auf Software für einige mobile devices zu. Gleichzeitig ist zu verzeichnen, dass Hersteller von Smartphones vereinzelt längere Garantienzeiten auf ihren Softwaresupport geben (Fairphone BV 2024; Chau und Zack 2023). Die Abwärtskompatibilität unter Beibehaltung der Sicherheit bedeutet u. U. für Softwareentwickler*innen einen Mehraufwand. Dieser steht aus Sicht eines Umweltzeichens in einem angemessenen Verhältnis zur Verlängerung der Lebensdauer der Hardware, welche einen signifikanten Einfluss auf die Umweltwirkung von Software hat.

Es ist für den gesamten Geltungsbereich wenig praktikabel, Referenzsysteme zu definieren, da dies zur Folge hat, dass die Antragsteller*innen diese spezifischen Geräte für den Nachweis der Lauffähigkeit anschaffen müssten. Stattdessen sollte Transparenz geschaffen werden, mit welchem Computer-System aus welchem zurückliegenden Kalenderjahr die Software noch lauffähig ist. Der erforderliche Softwarestack (z.B. Betriebssystem, Framework, Anwendersoftware) und die Konfiguration von Softwareprodukt und Softwarestack müssen bei der Bestimmung dieser Jahreszahl berücksichtigt werden. Es wird empfohlen, dass dies mindestens 5 Jahre sind.

In einigen Fällen ist es praktikabel, das zu zertifizierende Softwareprodukt auf alter Hardware aufzusetzen. In einigen Fällen ist es unter Umständen und trotz Sicherheitsbedenken notwendig, ein altes Betriebssystem zu nutzen. Die Lauffähigkeit der Software auf einem 5 Jahre alten Betriebssystem impliziert in der Regel die Kompatibilität mit einer 5 Jahre alten Hardware. Deshalb kann die Abwärtskompatibilität als optionale Alternative zur Angabe der Hardwarespezifikationen sichergestellt werden, indem die Betriebssystem-Version als Referenz genutzt wird.

6.4 Nutzungsautonomie

Ein Softwareprodukt soll die Entscheidungsfreiheit der Nutzenden im Umgang mit dem Produkt nicht einschränken, und es sollen möglichst wenig Abhängigkeiten geschaffen werden. Dadurch werden indirekte Umweltwirkungen der Software reduziert (siehe Abschnitt 4.1.4).

6.4.1 Datenformate

Die durch das Softwareprodukt erzeugten Daten sollen auch mit anderen Softwareprodukten weiter verarbeitbar sein und es soll nicht zu einer Abhängigkeit zu einzelnen Softwareherstellern kommen („Vendor Lock-in“). Um dies zu erreichen, sollen die Formate der von der Software verarbeiteten und erzeugten Daten oder alternativ exportierbaren Daten so gut dokumentiert sein (Syntax und Semantik), dass eine Verarbeitung durch andere Softwareprodukte ohne Verlust wesentlicher Informationen ermöglicht wird.

6.4.2 Transparenz des Softwareproduktes

Zur Erweiterung des Softwareproduktes um zusätzliche Funktionalitäten sollen die Anwendungs-Programmier-Schnittstellen (APIs) klar dokumentiert sein. Falls APIs vorhanden sind, sollen diese offenen Standards entsprechen.

Im Sinne einer langfristigeren bzw. umfangreicheren Nutzung soll idealerweise außerdem der Quellcode offengelegt werden, spätestens wenn das Produkt nicht weiterentwickelt wird, sowie die Erlaubnis dazu erteilt werden, das Produkt zu verändern. Da eine Verpflichtung zu Open Source weit verbreiteten Geschäftsmodellen widerspricht, soll zumindest Erklärung darüber eingefordert werden, was mit dem Softwareprodukt nach Supportende geschieht, d.h. wenn das Produkt bzw. die zertifizierte Version des Produktes nicht mehr erworben werden kann und/oder für die Nutzung des Produktes bzw. für die Version des Produktes kein Support mehr besteht.

6.4.3 Kontinuität des Softwareproduktes

Es soll möglich sein, das Softwareprodukt über einen längeren Zeitraum zu nutzen, ohne dass schwerwiegende Nachteile (insbesondere Probleme der IT-Sicherheit) auftreten. Dazu soll der Softwarehersteller eine Funktionalität anbieten, mit der das Softwareprodukt auf dem aktuellen Stand gehalten werden kann. Sicherheitsupdates sollten kostenlos erfolgen, Updates, die ausschließlich neue Funktionalitäten und keine Sicherheitsupdates enthalten, können dagegen kostenpflichtig sein.

Für pc und server devices sollen kostenlose Sicherheitsupdates für das zu kennzeichnende Produkt für mindestens 5 Jahre ab Bereitstellungsende bereitgestellt werden.

Für mobile devices soll das Produkt mindestens 3 Jahre nach der Kennzeichnung mit dem Umweltzeichen sicher nutzbar bleiben, indem bei Bedarf kostenlose Sicherheitsupdates

angeboten werden und das Produkt auf den jeweiligen Vertriebsplattformen in aktualisierter Version bereitgehalten wird.

6.4.4 Deinstallierbarkeit

Im Sinne der Nutzungsautonomie ist eine Software am Ende seiner Nutzungszeit vom Nutzenden so entfernbar, dass keine Daten zurückbleiben. Davon ausgenommen sind Daten und Dateien, die von Nutzenden mithilfe der Software erstellt oder bearbeitet wurden. Es wird empfohlen, gegenüber der zuletzt gültigen Version 3 der Vergabekriterien keine Änderungen vorzunehmen.

6.4.5 Modularität

Softwareprodukte bieten oft eine Vielzahl von Funktionen an, die nur selten oder nur von wenigen Nutzenden aufgerufen werden. Dadurch können Software-Installationen bzw. Aktivierung von Teilen des Softwareprodukts unnötig Speicherplatz, Rechenkapazitäten und damit Energie benötigen. Um dies zu vermeiden, ist anzustreben, Softwareprodukte modular zu gestalten, d.h. eine Basisfunktionalität anzubieten, die bei Bedarf durch die Installation weiterer Module ergänzt werden kann. Das Prinzip von mobilen Apps, jeden Nutzen in jeweils einer App abzubilden, die unabhängig von anderen installiert und de-installiert werden kann und geeignete Schnittstellen zu anderen Apps anbietet, folgt dem Prinzip der Modularität. Demgegenüber ist die Architektur von einzelnen Apps meist nicht modular aufgebaut. Auch für server und pc devices soll das Softwareprodukt den Nutzenden die Möglichkeit anbieten, die Funktionalität auf den gewünschten Funktionalitätsumfang zu beschränken. Folgende Angaben sollten dazu in den Produktinformationen gemacht werden müssen:

- Angabe, ob der Installationsprozess eine Möglichkeit bietet, einzelne Funktionalitäten, die über die Basisfunktionalität des Softwareproduktes hinaus gehen, bewusst nicht zu installieren. Wenn zutreffend, sollte dieses Vorgehen beschrieben werden.
- Angabe, ob während der Nutzung des Softwareprodukts einzelne Module bzw. Funktionalitäten durch den Nutzenden deaktiviert werden können. Wenn zutreffend, sollte dieses Vorgehen beschrieben werden.

6.4.6 Werbefreiheit und Vermeidung von Tracking

Werbung unterscheidet sich von einem Informationsangebot darin, dass Konsument*innen die Inhalte nicht aktiv aufsuchen und ihnen nicht ausweichen können. In den meisten Fällen ist sie von den Nutzenden der Software unerwünscht oder wird billigend in Kauf genommen.

Tracking als eine Grundlage von personalisierter Werbung generiert ebenso ungewollten Datenverkehr. Es beinhaltet das Auslesen und Sammeln von Nutzendenverhalten, bspw. Verweildauern bei den jeweiligen Inhalten, und andere personenbezogene Daten, z.B. dem Standort. Mithilfe statistischer Verfahren werden die durch Tracking gewonnen Informationen dazu genutzt, personalisierte und situativ passende, z.B. an die wahrscheinliche Gemütslage angepasste, Werbung auszuspielen. In seltenen Fällen ist Tracking jedoch nicht vermeidbar, beispielsweise ist es für Anbieter von zertifizierten Medizinprodukten gesetzlich vorgeschrieben, bestimmte Daten in Echtzeit zu überwachen.

Durch die Einblendung von Werbung kann der Ressourcen- und Energiebedarf durch Softwareprodukte z.T. stark erhöht werden. Insbesondere kann die Einblendung von Werbung, bspw. über einen externen Dienstleister, eine Übertragung zusätzlicher Datenmengen bedeuten. Dieser

Netzwerkverkehr hat für die von Nutzenden gewünschte Funktionalität des Softwareproduktes keine Bedeutung und sollte vermieden werden. Unter "Werbung" sind also insbesondere über das Netzwerk übertragene Medieninhalte (z.B. über Werbeprovider wie Google AdSense) zu verstehen. Werbung z.B. für das eigene Unternehmen und weitergehende Versionen der Software erzeugen weniger Datenverkehr und sind ggf. sogar von Nutzenden gewünscht.

Insgesamt wird im Sinne der Nutzungsautonomie sowie der Ressourcen- und Energieeffizienz empfohlen, Werbung auszuschließen und Trackingfunktionen ausschließlich in Sonderfällen zuzulassen.

6.4.7 Dokumentation des Softwareproduktes

Das Softwareprodukt sollte mit einer Dokumentation versehen sein, die eine langfristige und ressourcenschonende Nutzung des Produktes ermöglicht. Entsprechende Informationen zum Softwareprodukt sollen sowohl öffentlich (beispielsweise auf der eigenen Website) als auch in der Distribution des Softwareproduktes selbst in Form von Dateien (Textdokumente, Grafiken, allgemein lesbare Datenformate, Tabellenkalkulationsdatei) oder Hilfe-Funktionalitäten zur Verfügung gestellt werden.

6.4.8 Offlinefähigkeit

Offlinefähigkeit ergibt für server devices keinen Sinn, für mobile und pc devices in verteilten Systemen nur in speziellen Fällen. Es ist beispielsweise nicht allgemein energieeffizienter, alle benötigten Daten einer mobilen App auf dem Endgerät zu speichern, beispielsweise bei Karten- und Navigationsapps. In anderen Beispielen, wie etwa bei Streaming-Diensten, ist dies gar nicht praktikabel. Ist die Software auch ohne Internetverbindung lauffähig, reduziert dies nicht in jedem Fall die Abhängigkeit von energieintensiven mobilen Datennetzen. Es wird daher empfohlen, die Anforderung in der Überarbeitung des Blauen Engel für Softwareprodukte zu streichen und stattdessen einen größeren Wert auf die übertragenen Datenmengen zu legen (vgl. Abschnitt 6.2.3 und 6.2.4).

6.5 Anforderungen an Weiterentwicklung und Updates des Produktes

Updates des Produkts

In der Praxis werden Software-Updates teilweise mehrfach pro Woche veröffentlicht und der Quellcode wird über verschiedene Kanäle verteilt, die nicht in jedem Fall alle von den Entwickler*innen kontrolliert werden. Daher ist eine Energiemessung bei jedem Update unverhältnismäßig.

Änderungen an der Software z.B. durch neue Funktionen oder Änderung bei den Hardware-Anforderungen können es jedoch nötig machen, Messumgebung und Testszenario weiterzuentwickeln und die Dokumentation fortzuschreiben. Änderungen könnten z.B. folgende sein.

Zusätzliche Funktionen beim Szenario-Test nach einem Update.

Das Standardnutzungsszenario sollte entsprechend angepasst werden, um die zusätzlichen Funktionen zu integrieren. Eine Vergleichbarkeit mit der vorherigen Version des Softwareprodukts könnte durch eine gesonderte Betrachtung des neuen Szenarios ohne die neuen Funktionen erfolgen.

Geänderte Funktionen beim Szenario-Test nach einem Update.

Das Standardnutzungsszenario sollte angepasst werden, um geänderte Funktionen zu umfassen

(ggf. Funktionen austauschen). Eventuell erfordert dies auch die Anpassung der Automatisierung und anderer Elemente des Software- oder Hardwarestacks.

Entfallende Funktionen beim Szenario-Test nach einem Update.

Entfallende Funktionen sollten aus dem Standardnutzungsszenario entfernt werden. Auch hier sollte ggf. die Automatisierung und weiteres angepasst werden. Die Vergleichbarkeit mit der vorherigen Version könnte durch Entfallen der entsprechenden Funktion aus der Bewertung des alten Szenarios erreicht werden.

Bei der erneuten **Durchführung eines Langzeit-Tests** kann es zu größeren Abweichungen gegenüber der vorangehenden Messung kommen, da die Messergebnisse nicht nur von der Software selbst abhängen, sondern auch von externen Parametern wie veränderte Nutzerzahlen, verändertes Nutzerverhalten oder neue Hardware z.B. innerhalb eines Rechenzentrums. Solche Einflussfaktoren sollten während eines Vergleiches analysiert und begründet werden. Prinzipiell wäre es zweckdienlich, wenn die spezifischen Kennzahlen der Software (z.B. pro User oder pro Zeiteinheit) mit der Zeit besser (effizienter) werden, die Computer-Systeme besser ausgelastet sind oder im laufenden Betrieb weiter optimiert werden würden. Geht die Entwicklung dennoch in die Richtung eines steigenden Energie- und Hardware-Bedarfs, ist es ratsam, wenn die Zeichennehmer des Umweltzeichens triftige Gründe dazu nennen können.

Die öffentlichen Messwerte des Energie- und Ressourcenbedarfs sollten deshalb mindestens einmal pro Jahr aktualisiert werden. Dabei wird empfohlen, dass die aktualisierten Messwerte um maximal 10 % von den Messwerten aus dem Vorjahr nach oben abweichen dürfen oder begründet werden müssen, beispielsweise mit einer Funktionserweiterung.

6.6 Empfohlene Veränderungen gegenüber der zuletzt gültigen Fassung des Umweltzeichens DE-UZ 215, Ausgabe Januar 2020 (Version 3)

Zusammenfassend ergeben sich aufbauend auf den genannten Überlegungen für die Überarbeitung der Vergabekriterien folgende wesentliche Empfehlungen zur Anpassung:

- ▶ Erweiterung des Geltungsbereichs für Client-/Server-Systeme sowie Apps für mobile Endgeräte.
- ▶ Es sollte kein einzelnes Messverfahren festgelegt, sondern nur vorgegeben werden, was zu messen ist, aber nicht, wie Messaufbau oder Vorgehensweise konkret umzusetzen sind. Stattdessen sollte auf einschlägige Verfahren verwiesen und die Bewertung der Plausibilität über Auditor*innen geprüft werden.
- ▶ Es sollte kein konkretes Hardware-Referenzsystem festgelegt werden. Individuelle Hardware kann als Basis der Messungen verwendet werden, sollte dann aber unverändert für Folgemessungen wiederverwendet werden.
- ▶ Es sollten bei verteilten Anwendungen, alle Computer-Plattformen, auf denen die Software lauffähig ist gemessen werden, je nach Vorgehensweise parallel oder seriell.
- ▶ Bei mobilen Endgeräten sollten nur übertragene Datenmenge und Quelle/Ziel verschickter/empfangener Datenpakete (z.B. URLs/IP-Adressen), sowie weitere Eckdaten,

aber kein Energieverbrauch und keine Hardwareinanspruchnahme des Endgeräts erfasst werden.

- ▶ Es sollten zusätzlich zu den Szenario-Tests Langzeit-Tests von Produktivsystemen, insbesondere für den Server-Bereich, zugelassen werden, die über einen längeren Zeitraum laufen.
- ▶ Die Abwärtskompatibilität des zu zertifizierenden Produkts sollte nicht zwingend über Hardware-Referenzsysteme nachgewiesen werden müssen, sondern kann optional alternativ durch eine Referenz auf (frühere) Betriebssystem-Versionen nachgewiesen werden.
- ▶ Die Software soll nicht nur Werbefreiheit gewährleisten, sondern auch keine Tracking-funktionen enthalten. Ausgenommen sollten Trackingfunktionen sein, die aufgrund eines Gesetzes enthalten sein müssen, beispielsweise in Medizinprodukten.
- ▶ Die Anforderung an Offlinefähigkeit sollte entfallen.
- ▶ Die Einhaltung der Kriterien sollte während der Laufzeit nicht bei jedem Update, sondern einmal pro Jahr bestätigt werden. Die aktualisierten Messwerte sollten ebenfalls veröffentlicht werden müssen.

6.7 Ausblick

Mit der Ausweitung des Geltungsbereichs des Blauen Engels auf Software für Arbeitsplatz-Computer (pc devices), mobile Endgeräte (mobile devices), Server (server devices) und beliebige Kombinationen können zwei Ziele erreicht werden. Zum einen wird die Anzahl an Softwareprodukten deutlich erhöht, die potenziell mit dem Umweltzeichen gekennzeichnet werden kann. Zum anderen fühlen sich deutlich mehr Software-Entwickler*innen angesprochen, Nachhaltigkeit bei der Software-Entwicklung zu berücksichtigen und die Kriterien des Umweltzeichens Blauer Engel bei ihren Produktentwicklungen anzuwenden. Letzteres wurde durch die rege Teilnahme von Entwickler*innen an den Workshops innerhalb des Forschungsprojektes und der abschließenden Schulung für interessierte Auditor*innen deutlich (zu den empfohlenen Anforderungen an Auditor*innen siehe Anhang B).

Außerhalb des Geltungsbereichs sind jedoch solche Softwareprodukte, die in virtualisierten Hardware-Umgebungen (z.B. „in der Cloud“) ausgeführt werden und bei denen die notwendigen Messwerte nicht hinreichend genau erfasst werden können. Dadurch ist es bei diesen Systemen nicht möglich, die notwendige Transparenz über den Energie- und Ressourcenverbrauch von Software herzustellen, die ein Kernbestandteil der vorliegenden Vorgabekriterien ist. Zukünftig könnte sich diese Situation verbessern. Mit dem zunehmenden gesellschaftlichen und politischen Druck auf Cloud-Provider und Betreiber*innen von Rechenzentren, einen Beitrag zum Klimaschutz und zur Circular Economy zu leisten, könnte es zukünftig selbstverständlich werden, dass auch Software, die in virtuellen Hardware-Umgebungen ausgeführt wird, über ihre jeweilige Umweltperformance berichtet.

Denkbar wäre beispielsweise, dass digitale Dienstleistungen, wozu auch Cloud-Services und gemietete RZ-Infrastruktur zählen, selbstverständlich durch sie beanspruchte Hardware-Kapazitäten, Energieverbrauch und CO₂-Emissionen berichten, beispielsweise als Eintrag im Header. Genauso, wie es jetzt üblich ist, Uhrzeit, Sprache, Datenpaketgröße und Datentyp zu übertragen, könnten auch umweltbezogene Informationen mitgeliefert werden. Indem diese

Umweltinformation bei jeder Datenübertragung bekannt gegeben werden, könnte in Zukunft auch Cloud-Software optimiert und deren Umweltwirkungen sichtbar gemacht werden. Statt lokaler Messgeräte könnte man sich der Messgeräte der jeweiligen Anbieter bedienen. Auf diese Weise kann der Geltungsbereich des Umweltzeichens zukünftig noch auf weitere Softwareprodukte ausgeweitet werden.

Eine weitere Einschränkung des Geltungsbereichs des Umweltzeichens war der Ausschluss von Software mit einem Machine Learning-(ML)-Anteil bzw. mit einem Training Künstlicher Intelligenz (KI), sofern der Energieaufwand des Trainings wesentlich ist. Diese Aufwendungen treten nicht gleichzeitig mit der Nutzung der Software auf, sondern zeitlich versetzt und sie können auf grundsätzlich anderen Hardware-Plattformen stattfinden als die spätere Auslieferung der Software. Um die Aufwendungen für ML und KI der jeweiligen Software zuzuordnen, ist noch ein erheblicher Forschungsbedarf gegeben. Grundsätzlich wäre es wünschenswert, wenn jede Software, egal ob lokal ausgeführt, in der Cloud oder innerhalb einer KI-Architektur, möglichst nachhaltig entwickelt und ihre Umweltwirkungen bestmöglich reduziert würden. Der Blaue Engel DE-UZ 215 zeigt bereits jetzt Wege auf, wie dies geschehen kann, und sollte für weitere technologische Entwicklungen offen sein.

7 Quellenverzeichnis

ADEME - Agence de la transition écologique (2021): Référentiel méthodologique d'évaluation environnementale des services numériques - La librairie ADEME. Agence de la transition écologique (Hg.). Online verfügbar unter <https://librairie.ademe.fr/produire-autrement/6022-referentiel-methodologique-d-evaluation-environnementale-des-services-numeriques.html>, zuletzt aktualisiert am 03.04.2024, zuletzt geprüft am 03.04.2024.

Ahmad, R. W.; Gani, A.; Hamid, S. H. Ab; Xia, F.; Shiraz, M. (2015): A Review on Mobile Application Energy Profiling: Taxonomy, State-of-the-art, and Open Research Issues. In: Journal of Network and Computer Applications 58, S. 42–59. DOI: 10.1016/j.jnca.2015.09.002.

Alarcos (2025): Software and Information for a Digital and Sustainable World. Alarcos (Hg.). Online verfügbar unter <https://alarcos.esi.uclm.es/index.php>, zuletzt geprüft am 08.04.2025.

Anthony, L. F. W.; Kanding, B.; Selvan, R. (2020): Carbontracker, <https://github.com/lfwa/carbontracker>. Online verfügbar unter <https://doi.org/10.48550/arXiv.2007.03051>, zuletzt geprüft am 25.03.2025.

Beschaffungsamt des BMI (2024): Daten und Fakten. Beschaffungsamt des BMI (Hg.). Online verfügbar unter https://www.bescha.bund.de/DE/DasBeschaffungsamt/DatenUndFakten/datenundfakten_inhalt.html, zuletzt geprüft am 27.03.2025.

Bitkom e.V. (2021): Ressourceneffiziente Programmierung, Wie Ressourcenschonung, Langlebigkeit und Nachhaltigkeit in der Softwareentwicklung berücksichtigt werden können. Unter Mitarbeit von Geiger, L. und Termer, F. Berlin, 2021. Online verfügbar unter https://www.bitkom.org/sites/default/files/2021-03/210329_if_ressourceneffiziente-programmierung.pdf, zuletzt geprüft am 25.03.2025.

BITMI - Bundesverband IT Mittelstand e.V. (2023): Zertifizierte Unternehmen. Bundesverband IT Mittelstand e.V. (Hg.). Online verfügbar unter <https://www.software-made-in-germany.org/siegeltraeger/zertifizierte-unternehmen/>, zuletzt geprüft am 08.04.2025.

Blauer Engel (2024): Ressourcen-und energieeffiziente Softwareprodukte DE-UZ 215, Version 4. Umweltbundesamt (Hg.). Online verfügbar unter <https://produktinfo.blauer-engel.de/uploads/criteriafile/de/DE-UZ%20215-202001-de%20Kriterien-V4.pdf>, zuletzt geprüft am 07.04.2025.

Bonamy, C.; Boudinet, C.; Bourgès, L.; Dassas, K.; Lefevre, L.; Ninassi, B.; Vivat, F. (2023): Good practices in digital service ecodesign for software developers, 07.02.2023. Online verfügbar unter <https://hal.science/hal-03977001/document>, zuletzt geprüft am 25.03.2025.

Calero, C.; Piattini, M. (2015): Introduction to Green in Software Engineering. In: Calero, C. und Piattini, M. (Hg.): Green in Software Engineering. Cham: Springer International Publishing, S. 3–27.

Capgemini (2023): Capgemini delivers another record performance in 2022. Capgemini (Hg.). Online verfügbar unter <https://www.capgemini.com/news/press-releases/capgemini-delivers-another-record-performance-in-2022/>, zuletzt geprüft am 25.03.2025.

Chau, S.; Zack, S. (2023): 7 years of software updates for the Pixel 8 series. Google (Hg.). Online verfügbar unter <https://blog.google/products/pixel/software-support-pixel-8-pixel-8-pro/>, zuletzt geprüft am 25.03.2025.

Cruz, L. (2021): Tools to measure software energy consumption from your computer. In: Blog post. Online verfügbar unter <https://luiscruz.github.io/2021/07/20/measuring-energy.html>, zuletzt geprüft am 25.03.2025.

Eco:Digit (2024): ECO:DIGIT: IT-Nachhaltigkeit messbar machen. Eco:Digit (Hg.). Online verfügbar unter <https://ecodigit.de/>, zuletzt geprüft am 25.03.2025.

Europäische Kommission (2019): Verordnung (EU) 2019/424 der Kommission vom 15. März 2019 zur Festlegung von Ökodesign-Anforderungen an Server und Datenspeicherprodukte gemäß der Richtlinie

2009/125/EG des Europäischen Parlaments und des Rates und zur Änderung der Verordnung (EU) Nr. 617/2013 der Kommission. Online verfügbar unter <http://data.europa.eu/eli/reg/2019/424/oj>, zuletzt geprüft am 25.03.2025.

Europäische Kommission (2023): Verordnung (EU) 2023/1670 der Kommission vom 16. Juni 2023 zur Festlegung von Ökodesign-Anforderungen an Smartphones, Mobiltelefone, die keine Smartphones sind, schnurlose Telefone und Slate-Tablets gemäß der Richtlinie 2009/125/EG des Europäischen Parlaments und des Rates und zur Änderung der Verordnung (EU) 2023/826 der Kommission. Online verfügbar unter <http://data.europa.eu/eli/reg/2023/1670/oj>, zuletzt geprüft am 25.03.2025.

Europäische Kommission, Joint Research Centre; Kamiya, G.; Bertoldi, P. (2024): Energy consumption in data centres and broadband communication networks in the EU. Publications Office of the European Union (Hg.). Luxembourg, 2024. Online verfügbar unter <https://data.europa.eu/doi/10.2760/706491>, zuletzt geprüft am 26.03.2025.

Europäisches Parlament und Rat (2009): Richtlinie 2009/125/EG des Europäischen Parlaments und des Rates vom 21. Oktober 2009 zur Schaffung eines Rahmens für die Festlegung von Anforderungen an die umweltgerechte Gestaltung energieverbrauchsrelevanter Produkte (Neufassung). Online verfügbar unter <http://data.europa.eu/eli/dir/2009/125/oj>, zuletzt geprüft am 25.03.2025.

Europäisches Parlament und Rat (2019): Richtlinie (EU) 2019/770 des Europäischen Parlaments und des Rates vom 20. Mai 2019 über bestimmte vertragsrechtliche Aspekte der Bereitstellung digitaler Inhalte und digitaler Dienstleistungen. Online verfügbar unter <http://data.europa.eu/eli/dir/2019/770/oj>, zuletzt geprüft am 25.03.2025.

Europäisches Parlament und Rat (2024a): Richtlinie (EU) 2024/825 des Europäischen Parlaments und des Rates vom 28. Februar 2024 zur Änderung der Richtlinien 2005/29/EG und 2011/83/EU hinsichtlich der Stärkung der Verbraucher für den ökologischen Wandel durch besseren Schutz gegen unlautere Praktiken und durch bessere Informationen. Online verfügbar unter <http://data.europa.eu/eli/dir/2024/825/oj>, zuletzt geprüft am 25.03.2025.

Europäisches Parlament und Rat (2024b): Verordnung (EU) 2024/1689 des Europäischen Parlaments und des Rates vom 13. Juni 2024 zur Festlegung harmonisierter Vorschriften für künstliche Intelligenz und zur Änderung der Verordnungen (EG) Nr. 300/2008, (EU) Nr. 167/2013, (EU) Nr. 168/2013, (EU) 2018/858, (EU) 2018/1139 und (EU) 2019/2144 sowie der Richtlinien 2014/90/EU, (EU) 2016/797 und (EU) 2020/1828 (Verordnung über künstliche Intelligenz). Online verfügbar unter <http://data.europa.eu/eli/reg/2024/1689/oj>, zuletzt geprüft am 25.03.2025.

Europäisches Parlament und Rat (2024c): Verordnung (EU) 2024/1781 des Europäischen Parlaments und des Rates vom 13. Juni 2024 zur Schaffung eines Rahmens für die Festlegung von Ökodesign-Anforderungen für nachhaltige Produkte, zur Änderung der Richtlinie (EU) 2020/1828 und der Verordnung (EU) 2023/1542 und zur Aufhebung der Richtlinie 2009/125/EG. Online verfügbar unter <http://data.europa.eu/eli/reg/2024/1781/oj>, zuletzt geprüft am 25.03.2025.

Fairphone BV (2024): Fairphone 5. Für dich entwickelt. Fair produziert. Online verfügbar unter <https://shop.fairphone.com/de/fairphone-5>, zuletzt aktualisiert am 31.01.2024, zuletzt geprüft am 31.01.2024.

García-Berna, J.; Carrillo de Gea, J.; Moros, B.; Fernández-Alemán, J.; Nicolás, J.; Toval, A. (2018): Surveying the Environmental and Technical Dimensions of Sustainability in Software Development Companies. In: Applied Sciences 8 (11), S. 2312. DOI: 10.3390/app8112312.

Gesellschaft für Informatik e.V. (2023): Green Coding, Potentials of green coding: Perspectives on implementation. Gesellschaft für Informatik e.V. (Hg.). Online verfügbar unter

<https://gi.de/aktuelles/projekte/green-coding>, zuletzt aktualisiert am 08.04.2024, zuletzt geprüft am 08.04.2024.

Green Coding Solutions GmbH (o.J.): Wir machen Software nachhaltig. Green Coding Solutions GmbH (Hg.). Online verfügbar unter <https://www.green-coding.io/de/>, zuletzt geprüft am 25.03.2025.

Green Soft Engineering (o.J.): Die Arbeitsgruppe Green Software Engineering. Green Soft Engineering (Hg.). Online verfügbar unter <https://www.umwelt-campus.de/forschung/projekte/green-software-engineering/home>, zuletzt geprüft am 25.03.2025.

Green Software Foundation (o.J.): We are building a trusted ecosystem of people, standards, tooling and best practices for Green software. Green Software Foundation (Hg.). Online verfügbar unter <https://greensoftware.foundation/>, zuletzt geprüft am 25.03.2025.

Gröger, J. (2020): Digitaler CO2-Fußabdruck, 2020. Online verfügbar unter <https://www.oeko.de/fileadmin/oekodoc/Digitaler-CO2-Fussabdruck.pdf>, zuletzt geprüft am 25.03.2025.

Gröger, J.; Köhler, A.; Naumann, S.; Filler, A.; Guldner, A.; Kern, E.; Hilty, L. M.; Maksimov, Y. (2018): Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik, Öko-Institut e.V. Umweltbundesamt (Hg.). Online verfügbar unter <https://www.umweltbundesamt.de/publikationen/entwicklung-anwendung-von-bewertungsgrundlagen-fuer>, zuletzt aktualisiert am 18.10.2021, zuletzt geprüft am 18.10.2021.

Gröger, J.; Liu, R.; Stobbe, L.; Druschke, J.; Richter, N. (2021): Green Cloud Computing, Lebenszyklusbasierte Datenerhebung zu Umweltwirkungen des Cloud Computing. Umweltbundesamt (Hg.), März 2021. Online verfügbar unter <https://www.umweltbundesamt.de/publikationen/green-cloud-computing>, zuletzt geprüft am 26.03.2024.

Guldner, A. (2024): Green-Software-Engineering, Software Energy and Resource Efficiency Analysis. GitLab (Hg.). Online verfügbar unter https://gitlab.rlp.net/green-software-engineering/serena/-/tree/main?ref_type=heads, zuletzt aktualisiert am 17.04.2024, zuletzt geprüft am 17.04.2024.

Guldner, A.; Bender, R.; Calero, C.; Fernando, G.; Funke, M.; Gröger, J.; Hilty, L. M.; Hörschemeyer, J.; Hoffmann, G.-D.; Junger, D.; Kennes, T.; Kreten, S.; Lago, P. et al. (2024): Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—Green Software Measurement Model (GSMM). In: Future Generation Computer Systems 155, S. 402–418. DOI: 10.1016/j.future.2024.01.033.

Hilty, L. M.; Naumann, S.; Maksimov, Y.; Kern, E.; Filler, A.; Guldner, A.; Gröger, J. (2017): Kriterienkatalog für nachhaltige Software, UFOPLAN-Projekt „Sustainable Software Design-Entwicklung einer Methodik zur Bewertung der Ressourceneffizienz von Softwareprodukten“. Umweltbundesamtes (Hg.), 31.05.2017. Online verfügbar unter https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/Greensoft/Kriterienkatalog_nachhaltige_Software_v01_2017-05-31.pdf, zuletzt geprüft am 25.03.2025.

Hindle, A. (2016): Green Software Engineering: The Curse of Methodology. In: 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). Osaka, Japan (5), S. 46–55.

Hochschule Trier (2024): OSCAR v0.190404. Hochschule Trier (Hg.). Online verfügbar unter <https://oscar.umwelt-campus.de/>, zuletzt aktualisiert am 17.04.2024, zuletzt geprüft am 17.04.2024.

HTF Market Intelligence Consulting Pvt. Ltd. (04.10.2021): Open Source Software Market to See Huge Growth by 2026 | OpenText, ClearCenter, Astaro. In: openPR, 04.10.2021. Online verfügbar unter

<https://www.openpr.com/news/2416613/open-source-software-market-to-see-huge-growth-by-2026-opentext>, zuletzt geprüft am 07.04.2025.

IT-Beauftragter des Bundes (2021): Nachhaltigkeit in der IT-Beschaffung. Bundesministerium für Informationstechnik (Hg.). Online verfügbar unter <https://www.cio.bund.de/Web/CIO/DE/digitale-loesungen/it-einkauf/nachhaltigkeit-in-der-it-beschaffung/nachhaltigkeit-in-der-it-beschaffung-node.html>, zuletzt geprüft am 27.03.2025.

ITU (2014): L.1410rev: Methodology for environmental life cycle assessments of information and communication technology goods, networks and services. International Telecommunications Union (Hg.), 2014. Online verfügbar unter https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-L.1410-201412-S!!PDF-E&type=items, zuletzt geprüft am 25.03.2025.

Jaeger-Erben, M.; Poppe, E.; Wagner, E.; Schaefer, A.; Druschke, J.; Gröger, J.; Behrens, F. (2023): Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten. Umweltbundesamt (Hg.). Online verfügbar unter <https://www.umweltbundesamt.de/publikationen/analyse-der-softwarebasierten-einflussnahme-auf>, zuletzt geprüft am 19.05.2023.

Jašarević, L. (2022): Die Top 15 der großen deutschen Softwareunternehmen. Online verfügbar unter <https://deutschlandtest.de/rankings/software-hersteller>, zuletzt aktualisiert am 08.12.2022, zuletzt geprüft am 20.05.2023.

Jay, M.; Ostapenco, V.; Lefevre, L.; Trystram, D.; Orgerie, A.-C.; Fichel, B. (2023): An experimental comparison of software-based power meters: focus on CPU and GPU. In: Simmhan, Y. (Hg.): 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing. CCGrid 2023 : 1-4 May 2023, Bangalore, India : proceedings. 2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid). Bangalore, India, 5/1/2023 - 5/4/2023. Institute of Electrical and Electronics Engineers; Association for Computing Machinery. Piscataway, NJ: IEEE, S. 106–118.

KDE e.V. (o.J.): Opt Green - Independent, energy-efficient Free & Open Source Software for long-term hardware use. KDE e.V. (Hg.). Online verfügbar unter <https://eco.kde.org/de/>, zuletzt geprüft am 25.03.2025.

Kern, E.; Hilty, L. M.; Guldner, A.; Maksimov, Y. V.; Filler, A.; Gröger, J.; Naumann, S. (2018): Sustainable software products—Towards assessment criteria for resource and energy efficiency. In: Future Generation Computer Systems 86, S. 199–210. DOI: 10.1016/j.future.2018.02.044.

Listflix (2023): Liste mit 43.950 Softwareunternehmen + Branchenübersicht. Online verfügbar unter <https://listflix.de/dienstleistung/it-dienstleister/softwareunternehmen/>, zuletzt aktualisiert am 18.04.2023, zuletzt geprüft am 20.05.2023.

Mancebo, J.; Calero, C.; Garcia, F.; Moraga, M. A.; Garcia-Rodriguez de Guzman, I. (2021): FEETINGS: Framework for Energy Efficiency Testing to Improve Environmental Goal of the Software. In: Sustainable Computing: Informatics and Systems 30, S. 100558. DOI: 10.1016/j.suscom.2021.100558.

Mercier-Ganady, J. (2024): actiona: Cross-platform automation tool. GitHub (Hg.). Online verfügbar unter <https://github.com/Jmgr/actiona>, zuletzt aktualisiert am 17.04.2024, zuletzt geprüft am 17.04.2024.

mitmproxy (o.J.): mitmproxy is a free and open source interactive HTTPS proxy. mitmproxy (Hg.). Online verfügbar unter <https://mitmproxy.org/>, zuletzt geprüft am 08.04.2025.

Naumann, S.; Kern, E.; Guldner, A.; Gröger, J. (2020): Hintergrundbericht Umweltzeichen Blauer Engel für ressourcen- und energieeffiziente Softwareprodukte, Hintergrundbericht zur Entwicklung der Vergabekriterien DE-UZ 215, Ausgabe Januar 2020. Institut für Softwaresysteme am Umwelt-Campus Birkenfeld. Umweltbundesamt (Hg.), März 2020. Online verfügbar unter

<https://www.umweltbundesamt.de/publikationen/umweltzeichen-blauer-engel-fuer-ressourcenund>, zuletzt geprüft am 25.03.2025.

Ninassi; Benjamin (2021): Resources of the Mooc Environmental impacts of digital technologies, The good practices of digital service ecodesign. Online verfügbar unter <https://learninglab.gitlabpages.inria.fr/mooc-impacts-num/mooc-impacts-num-ressources/en/Partie3/FichesConcept/FC3.4.2-bonnespratiques-MoocImpactNum.html>, zuletzt geprüft am 08.04.2024.

OekoJ (2022): Softwarefootprint.py, Python Script zur Abschätzung von Energieverbrauch und CO2-Emissionen von Software. Online verfügbar unter <https://github.com/oekoJ/softwarefootprint>, zuletzt aktualisiert am 18.04.2024.

Raymond, E. S. (2003): The art of UNIX programming (1st edition) (Addison-Wesley professional computing series). Boston: Addison-Wesley Professional.

Schade, T. (2023): GreenCoding.measuring-tools: Tools overview for measuring tools around software energy and CO2 consumption. GitHub (Hg.). Online verfügbar unter <https://github.com/schaDev/GreenCoding-measuring-tools>, zuletzt geprüft am 25.03.2025.

SDIA (2024): Principles for Sustainable Software Design. SDIA (Hg.). Online verfügbar unter <https://knowledge.sdialliance.org/software/principles-for-sustainable-software-design>, zuletzt aktualisiert am 08.04.2024, zuletzt geprüft am 08.04.2024.

SDIA (o.J.): Building Better Digital Infrastructure. SDIA (Hg.). Online verfügbar unter <https://www.sdia.io/>, zuletzt geprüft am 25.03.2025.

Softwareallianz Deutschland (2023): Die richtigen IT-Experten zu jeder Zeit, IT-Mittelstand für Konzerne. Softwareallianz Deutschland (Hg.). Online verfügbar unter <https://softwareallianz.de/>, zuletzt geprüft am 08.04.2025.

Statista (2024a): App - Weltweit | Statista Marktprognose. Online verfügbar unter <https://de.statista.com/outlook/dmo/app/weltweit#umsatz>, zuletzt aktualisiert am 26.03.2024, zuletzt geprüft am 26.03.2024.

Statista (2024b): Deutschland: Umsatz im Segment Office Software 2028 | Statista. Online verfügbar unter <https://de.statista.com/prognosen/970247/prognose-zum-umsatz-mit-office-software-in-deutschland>, zuletzt aktualisiert am 26.03.2024, zuletzt geprüft am 26.03.2024.

Statista (2024c): Mobile Apps - Umsatz in Deutschland 2022 | Statista. Online verfügbar unter <https://de.statista.com/statistik/daten/studie/173810/umfrage/umsatz-mit-mobilen-apps-in-deutschland-seit-2009/>, zuletzt aktualisiert am 26.03.2024, zuletzt geprüft am 26.03.2024.

Statista (2024d): SAP - Umsatz weltweit bis 2023 | Statista. Online verfügbar unter <https://de.statista.com/statistik/daten/studie/28261/umfrage/umsatz-des-unternehmens-sap-seit-dem-jahr-2001/>, zuletzt aktualisiert am 26.03.2024, zuletzt geprüft am 26.03.2024.

Statista (2024e): Software - Deutschland | Statista Marktprognose. Online verfügbar unter <https://de.statista.com/outlook/tmo/software/deutschland>, zuletzt aktualisiert am 29.01.2024, zuletzt geprüft am 29.01.2024.

Statista (2024f): Software - Weltweit | Statista Marktprognose. Online verfügbar unter <https://de.statista.com/outlook/tmo/software/weltweit#key-players>, zuletzt aktualisiert am 29.01.2024, zuletzt geprüft am 29.01.2024.

Stobbe, L.; Nissen, N. F. (2023): Energie- und Ressourcenbedarf der IKT in Deutschland bis 2035. Fraunhofer-Institut für Zuverlässigkeit und Mikrointegration IZM. Veranstalter: MikroSystemTechnik Kongress 2023. Dresden, 25.10.2023.

Sun, Y.; Agostini, N.; Dong, S.; Kaeli, D. (2019): Summarizing CPU and GPU Design Trends with Product Data. Online verfügbar unter https://www.researchgate.net/publication/337545059_Summarizing_CPU_and_GPU_Design_Trends_with_Product_Data, zuletzt geprüft am 25.03.2025.

Tarara, A. (2024): Green Metrics Tool. Green Coding Solutions GmbH (Hg.). Online verfügbar unter <https://www.green-coding.io/de/projects/green-metrics-tool/>, zuletzt aktualisiert am 17.04.2024, zuletzt geprüft am 17.04.2024.

te Brake, C. (2020): Untersuchung der Perspektive von Akteur*innen der Softwarebranche auf die Umweltwirkungen von Softwareprodukten. Bachelorarbeit im Studiengang Umweltwissenschaften, betreut von Herr Prof. Dr. Eckhard Bollow, Leuphana. Lüneburg, 2020.

Umweltbundesamt (2020): Blauer Engel - Ressourcen-und energieeffiziente Softwareprodukte (DE-UZ 215), Produktseite. Umweltbundesamt (Hg.). Online verfügbar unter <https://www.blauer-engel.de/uz215>, zuletzt geprüft am 25.03.2025.

Umweltbundesamt (2024): Digitalisierung. Umweltbundesamt (Hg.). Online verfügbar unter <https://www.umweltbundesamt.de/themen/digitalisierung>, zuletzt geprüft am 25.03.2025.

uptotown (o.J.): ProxyDroid, 2.75. uptotown (Hg.). Online verfügbar unter <https://proxydroid.de.uptodown.com/android>, zuletzt geprüft am 08.04.2025.

Wireshark (o.J.): The world's most popular network protocol analyzer, Get started with Wireshark today and see why it is the standard across many commercial and non-profit enterprises. Wireshark (Hg.). Online verfügbar unter <https://www.wireshark.org/>, zuletzt geprüft am 08.04.2025.

A Beispielhafte Methoden zur Bestimmung der Energie- und Hardwareinanspruchnahme

Durch den erweiterten Geltungsbereich des Umweltzeichens nehmen die Menge und Bandbreite der zertifizierbaren Softwareprodukte zu. Ebenso hat die Anzahl an verfügbaren Messmethoden, die an verschiedenen Stellen entwickelt und teilweise bereitgestellt werden, zugenommen. Eine Reihe privater Unternehmen und auch Open Source-Initiativen entwickeln Messmethoden und entsprechende Softwaretools zum Messen der Energieverbräuche von Software. Im Folgenden soll ein Einblick in die Vielzahl der bestehenden Messmethoden und -Tools gegeben werden.

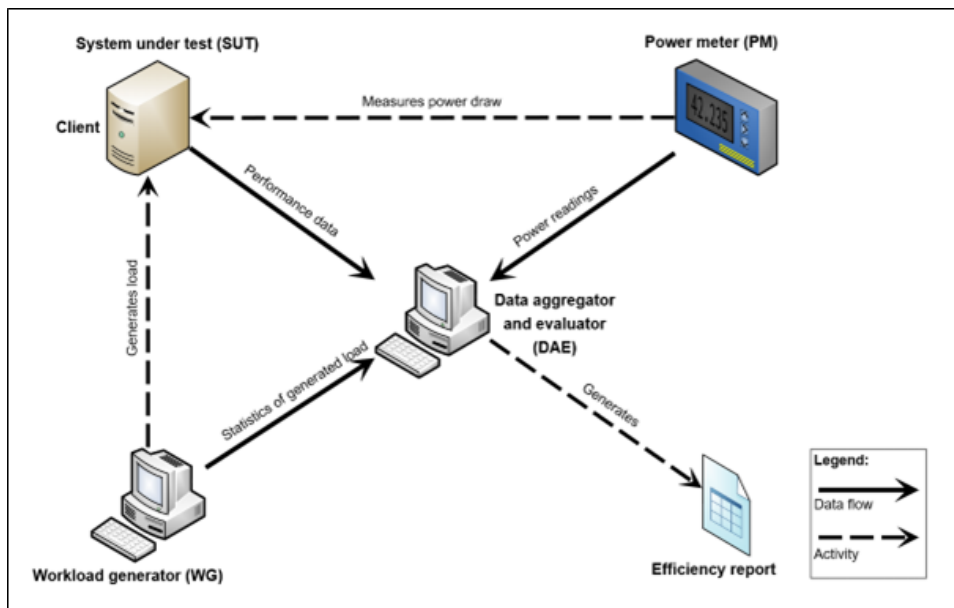
A.1 Messmethode SERENA

Die Messmethode SERENA (Software Energy and Resource EfficieNcy Analysis) dient der Bewertung des Energie- und Ressourcenbedarfs von Software und wurde an der Hochschule Trier, Umwelt-Campus Birkenfeld, entwickelt (Guldner 2024). Auf ihr basiert die Referenzmethode für die bisherigen Vergabekriterien des Blauen Engel für Softwareprodukte DE-UZ 215, Ausgabe Januar 2020 (Version 3) eingesetzt.

A.1.1 Software Energy and Resource EfficieNcy Analysis (SERENA)

Die Messung des Energiebedarfs eines **pc devices** mittels SERENA erfolgt standardmäßig als ein Szenario-Test mit einem Standardnutzungsszenario, welches z.B. mit einer Automatisierungssoftware ausgeführt wird. Die Messung selbst beinhaltet die Grundauslastung, auch Baseline genannt, welche den Energie- und Ressourcenbedarf ohne die zu betrachtende Software erfasst. Weiterhin ist eine Leerlaufmessung Bestandteil, also die Messung des Zustandes nach dem Starten der zu untersuchenden Software, ohne dass die Software aktiv genutzt wird. Schließlich wird die Auslastung des Systems während der Ausführung des Szenarios erhoben. Bei allen Messungen werden die mittlere Auslastung der CPU, des Arbeitsspeichers und des Permanent-speichers sowie die über das Netzwerk übertragene Datenmenge und der Gesamtenergiebedarf des Systems erfasst. Alle Messungen werden mehrmals wiederholt, um statistisch aussagekräftige Daten zu erhalten.

Abbildung 11: Schematischer Messaufbau Client-Messung bei SERENA



Quelle: Gröger et al. (2018)

Abbildung 11 zeigt den Messaufbau von SERENA. Durch den Workload Generator (WG) wird Last auf dem Client (pc device) erzeugt, der gemessen wird. Dabei muss es sich nicht zwingend um einen anderen PC handeln, der Aktionen ausführt. Es kann auch ein Programm direkt auf dem Client laufen, welches Last erzeugt. Klassischerweise wird ein Standardnutzungsszenario ausgeführt. Es kann jedoch auch der Live-Betrieb in Sinne einer Langzeitmessung erfasst werden. Die Art und Weise, wie die Last erzeugt wird, sollte stets dokumentiert werden. Mittels eines externen Power Meter (PM) wird die elektrische Leistungsaufnahme gemessen. Auch diese Messung kann alternativ mit einem geeigneten Software-Tool auf dem Client selbst erfolgen. Die restlichen Daten der Hardwarekomponenten werden bei SERENA auf dem SUT erfasst. Anschließend werden die Daten mit einem Data Aggregator and Evaluator (DAE) ausgewertet und in einem Bericht dargestellt. Dieser sammelt zusätzlich auch die Messwerte des Power Meter und die Informationen zur Lasterzeugung. Mit den gesammelten Informationen wird z.B. durch das Auswertungssystem OSCAR ein Report erzeugt, welcher für eine Antragstellung des Blauen Engels Software genutzt werden kann.

Das an der Hochschule entwickelte Softwareprodukt **OSCAR** (Open source Software Consumption Analysis and Reporting) wertet vorliegende Messdaten zum verursachten Energiebedarf einer Software und die entsprechende Hardware-Auslastung aus (Hochschule Trier 2024). Die Werte müssen dazu als CSV-Dateien vorliegen. Eine Vorverarbeitung der Rohdaten kann erforderlich sein, um sie an das verarbeitbare Format anzupassen. Abschließend wird der Bericht erzeugt, der jeweils für Leerlaufmessung und die Messung des Standardnutzungsszenarios verfügbar ist.

Ein Vorteil der Messmethode liegt in der Abgrenzung der Komponenten. Im Gegensatz zur softwarebasierten/Chip-basierten Messung wird der Energiebedarf von allen Komponenten des Gerätes erfasst. Allerdings werden so auch Hintergrund- und ggf. andere Aktivitäten gemessen, die nicht auf das Softwareprodukt zurückzuführen sind, und bspw. mit der Baseline herausgerechnet werden können. Nachteilig kann der ggf. kostspielige Aufbau der Messumgebung sein. Zudem kann die Auswertung mit OSCAR durch die Vorverarbeitung der Rohdaten arbeits- und zeitaufwändig sein. OSCAR liefert jedoch einen umfangreichen Überblick über die Messwerte

und weitere statistische Daten, die Aufschluss über die Messung und das Verhalten der Komponenten geben.

A.1.2 SERENA für Client-Server Systeme

Das SERENA-Messverfahren kann auch für **Client-Server-Systemen** angewendet werden. Hier wird mindestens ein Server integriert, der mit mindestens einem Client kommuniziert. Es können somit beliebig viele Clients und Server miteinander interagieren, wobei Clients auf eigener Hardware laufen können oder sich auch simulieren lassen. Die Messungen von Client- und Serverseite erfolgen dabei getrennt voneinander. Das generelle Vorgehen von SERENA bleibt erhalten. Die Last auf den Servern wird üblicherweise durch die Clients erzeugt, sodass der Workload Generator ebenso ausschließlich auf den/die Clients wirkt, bzw. die Clients selbst die Last generieren. Es sind aber auch Lasterzeugungen direkt auf dem Server möglich.

Bei allen Messungen werden ebenfalls die mittlere Auslastung der CPU, des Arbeitsspeichers und des Permanentspeichers sowie die über das Netzwerk übertragene Datenmenge und der Gesamtenergiebedarf des Systems erfasst. Auch hier werden alle Messungen mehrmals wiederholt, um statistisch relevante Daten zu erhalten.

Die Anwendbarkeit dieser Methode wurde mit einem Server und zwei miteinander kommunizierenden Clients getestet. Dazu wurde ein Standardnutzungsszenario entwickelt und dieses mittels der Automatisierungssoftware „Actiona“ automatisiert (Mercier-Ganady 2024).

Herausforderungen zeigten sich in der Synchronisation der Automatisierung der Clients.

Die Messmethode erfüllt die wesentlichen Bedingungen, um Client-Server Systeme messen zu können. Der Vorteil der Methode ist die klare Abgrenzung der Hardware-Komponenten. Die Kenngrößen werden reproduzierbar erfasst und sind nachvollziehbar. Es kann bspw. beobachtet werden, welche Daten zwischen Client und Server versendet werden. Allerdings ist der Aufwand des Aufbaus durch die zusätzlichen Server bzw. Clients entsprechend höher als bei dem pc device Aufbau. Das Erstellen und Ausführen des Szenarios kann dabei zeitintensiv sein. Bei dynamischen Testobjekten kann es zudem zu Schwierigkeiten der Automatisierung kommen. Es wird also eine sorgfältige Planung und Umsetzung benötigt. Die Auswertung mit OSCAR ist ebenfalls umfangreicher und liefert mehrere Berichte (zu jeder Computerplattform bei Szenario-Tests zwei Berichte und einen Bericht pro Plattform bei Langzeit-Tests). Insgesamt ist die SERENA-Client-Server-Methode eine geeignete und nutzbare Messmethode für den erweiterten Geltungsbereich.

A.2 Weitere Verfahren zur Messdatenerfassung

(Guldner et al. 2024; Cruz 2021; Schade 2023) und (Jay et al. 2023) bieten einen Überblick zu weiteren Messmethoden und -instrumenten, die sich zur Messung der Ressourcenbedarfe von Softwareprodukten eignen. Darüber hinaus haben die Autor*innen eigene Erfahrungen mit Messmethoden für mobile Apps und deren Datenverkehr gesammelt. Im Folgenden wird ein grober Einblick über einige der Methoden gegeben.

A.2.1 AppChecker (mobilsicher)

Der „AppChecker“ von „mobilsicher“¹⁴ stellt die Grundlage zur Ressourcenbedarfsanalyse dar. Dieses Kooperationsprojekt des iRights e. V. und des Instituts für Technik und Journalismus e. V. (ITUJ) analysiert Android Apps auf ihr Datensendeverhalten und identifiziert den Advertisement and Tracking Services-(ATS)-Anteil, welcher transparent gemacht wird. Zudem ist es mit den Werkzeugen möglich, die Menge der empfangenen und gesendeten Daten aufzuzeichnen. Weiterhin kann ein „record and replay“-Tool verwendet werden, welches die Testabläufe automatisiert. Dieses ist jedoch nur bei werbefreien Apps und Apps mit statischer Werbung anwendbar. Es wird direkt auf einem mobilen Endgerät, einem Smartphone, gemessen und die Daten auf einem Rechner mithilfe „mitmproxy“ gesammelt¹⁵. Diese Software erzeugt einen Proxy (Vermittler zwischen Client und Server), über den die HTTP-Kommunikation zwischen Endgerät und Internet gesendet werden. Dafür muss die mobile App „ProxyDroid“¹⁶ auf dem Testgerät installiert sein, welche den Netzwerkverkehr entsprechend umleitet.

A.2.2 Green Software Measurement Process - GSMP

Die „Alarcos Reasearch Group“¹⁷ der Universidad Castilla-La Mancha in Ciudad Real, Spanien, hat den Green Software Measurement Process (GSMP) zum Messen lokaler Softwareprodukte, Webanwendungen, Client- oder Serveranwendungen entwickelt.

Der Messaufbau besteht aus einem Hardware-Messgerät (EET – Energy Efficiency Tester) und einem Tool (ELLIOT) zur Analyse der von EET gewonnenen Daten. Zusätzlich zum Gesamtenergiebedarf des DUT (Device Under Test) unterstützt EET die Messung verschiedener Hardwarekomponenten: CPU, Festplatte, GPU und Monitor. Die Abtastfrequenz von EET liegt bei 100 Hz. Bei diesem Messaufbau kann ausschließlich lokale Software, die auf dem DUT läuft, gemessen werden (Mancebo et al. 2021).

A.2.3 Erfassung des Datenverkehrs mobiler Geräte

Für die Betrachtung des Datenverkehrs mobiler Geräte wurde ein handelsüblicher Router als Data collector genutzt. Um den gesamten Netzwerk-Traffic zu protokollieren, wurde der Paketmitschnitt verwendet, den der Router anbietet. Der Daten-Traffic wurde während der Durchführung einer Baseline-, Leerlauf- und Szenario-Messung aufgezeichnet. Für die Messung wurde ein Standardnutzungsszenario erstellt, welches manuell ausgeführt wurde. Anschließend wurden die gespeicherten Datenprotokolle mit Wireshark¹⁸ geöffnet. Wireshark ist ein open-source Analysetool, welches Netzwerkprotokolle überwacht und in diesem Messaufbau als Auswertungstool verwendet wurde. Dazu wurde der gesamte Netzwerk-Traffic nach der IP-Adresse des mobilen Endgerätes gefiltert. Die dort aufgelisteten IP-Adressen wurden manuell dokumentiert und analysiert, um eine Liste aller aufgerufenen Internetadressen zu erstellen, sowie die gesendeten und empfangenen Datenmengen zu erhalten.

¹⁴ <https://appcheck.mobilsicher.de/> (Dieses Projekt wurde am 31.12.2024 beendet)

¹⁵ <https://mitmproxy.org/> mitmproxy (o.J.).

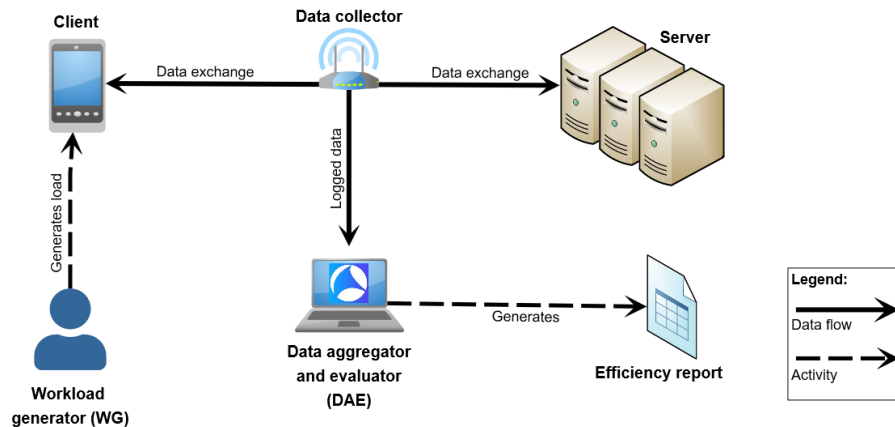
¹⁶ <https://proxydroid.de.uptodown.com/android> uptotown (o.J.).

¹⁷ <https://alarcos.esi.uclm.es/indexENG.php> Alarcos (2025).

¹⁸ <https://www.wireshark.org/> Wireshark (o.J.).

In Abbildung 12 ist der Messaufbau dargestellt. Der Workload Generator (WG) generiert Last auf dem mobilen Client. Der Client tauscht Daten mit einem oder mehreren Servern aus. Diese Daten werden von einem Logging tool, der sich zwischen Server und Client befindet, erfasst und an den Data Aggregator and Evaluator (DAE) weitergegeben. Der DAE generiert aus den erhaltenen Daten einen Efficiency Report.

Abbildung 12: Schematischer Messaufbau zu Messung des Datenverkehrs



Quelle: eigene Darstellung, Hochschule Trier, Umwelt-Campus Birkenfeld, Institut für Softwaresysteme

A.2.4 Green Metrics Tool

Bei dem **Green Metrics Tool (GMT)** werden Anwendungen durch Container isoliert. Dies ermöglicht eine präzise Messung verschiedener Faktoren wie Strom-, Netzwerk-, Festplatten- und Speicherbedarf. Durch die Nutzung bestehender Infrastrukturdateien, die derzeit auf Docker Compose-Dateien beschränkt sind, lässt es sich nahtlos als Drop-in-Messlösung integrieren. Die Container und Netzwerke werden entsprechend der bereitgestellten Infrastrukturdatei orchestriert und es wird ein Nutzungsszenario ausgeführt. Da jede Komponente der Software in einem Container gekapselt ist, können die Messungen komponentenspezifisch erfolgen und alle Aspekte der Anwendung präzise gemessen werden (Tarara 2024).

Green Coding Solutions hat zudem "ein maschinelles Lernmodell entwickelt, das den Energiebedarf auf der Grundlage leicht verfügbarer Eingabeparameter schätzen kann." Dieses Modell akzeptiert eine Vielzahl von Eingabeparametern, wobei viele dieser Parameter optional sind. "Dies ermöglicht es dem Modell, in eingeschränkten Umgebungen zu funktionieren, wie z. B. in Cloud-basierten Messeinrichtungen", in denen manche Informationen nicht ohne weiteres zugänglich sind. Je mehr Parameter angegeben werden, desto höher ist die Präzision des Modells. "Die Ausgabe des Modells kann entweder die Leistungsaufnahme oder die akkumulierte Energie seit der letzten Ausgabe sein." (ebd.)

A.2.5 Logging der CPU Process ID Rechenzeit

Auf einem unserer Workshops haben wir mit einem selbstgeschriebenen Python-Skript (OekoJ 2022) demonstriert, dass es leicht möglich ist, über CPU-Daten den Energiebedarf einer Software abzuschätzen und einer laufenden Software direkt zuzuordnen. Ein Skript zum Logging

läuft parallel zu der gemessenen Software und erfasst die CPU-Last. Es kann jede lokal laufende Software gemessen werden, die durch ihren Prozessnamen und ihre ID identifiziert wird. Weitere Child-Prozesse, die von der zu testenden Software erzeugt werden, werden ebenfalls erfasst. Das Tool protokolliert die aktiv genutzten CPU-Zeiten und den Anteil der genutzten Prozessorkerne über die Zeit. Damit ist die Messung unabhängig von Hintergrundprozessen, die auf dem Rechner laufen.

B Anforderungen an Auditor*innen

In den vorangehenden Kapiteln des Hintergrundberichts wurde dargestellt, dass mit der Ausweitung des Geltungsbereichs auch eine Ausweitung möglicher Messumgebungen (ehemals: Referenzsysteme) und Messverfahren notwendig wurde. Die dadurch steigende Komplexität der Antragstellung macht es erforderlich, weitere qualifizierte Personen zur Antragsprüfung hinzuzuziehen. Zusätzlich zu den Vergabekriterien wurde daher ein Konzept entwickelt, wie die Antragsunterlagen durch externe Auditor*innen vorab geprüft werden können und welche Anforderungen an diese Auditor*innen gestellt werden.

Es ist beabsichtigt, dass Antragsdokumente für die Kennzeichnung eines Softwareproduktes mit dem Blauen Engel bevor sie bei der RAL gGmbH, der Vergabestelle des Blauen Engels, eingereicht werden können, von einer unabhängigen Auditor*in auf Vollständigkeit, Plausibilität und Korrektheit geprüft und bestätigt werden müssen.

In den nachfolgenden Abschnitten wird daher ein Konzept vorgestellt, wie sich interessierte Personen zur Auditor*in qualifizieren können, welche Aufgaben und Pflichten damit verbunden sind und welche weiteren formalen Anforderungen an die Rolle der Auditor*in geknüpft sind.

Es wird im Rahmen des Forschungsprojektes vorgeschlagen, dass diese Anforderungen vom Umweltbundesamt in Form einer Handreichung oder einer Schulung an Interessent*innen bereitgestellt werden. Außerdem sollte es geeignete Schulungen durch das Umweltbundesamt geben, die Interessent*innen qualifizieren und bereits ernannte Auditor*innen weiterbilden.

B.1 Qualifizierung zur Auditor*in

Um Auditor*in zu werden, sollen nachfolgende Voraussetzungen erfüllt werden:

- ▶ Einschlägiges Hochschulstudium oder mindestens 3-jährige einschlägige Praxiserfahrung.
- ▶ Teilnahme an einer speziellen Schulung des Umweltbundesamtes.
- ▶ Formloser Antrag beim Umweltbundesamt, Referat Z 2.3 Digitale Transformation und Beratungsstelle Green IT auf Zulassung zur Prüfung als Software-Auditor*in zusammen mit der Übersendung eines Qualifikationsnachweises.
- ▶ Interessierte Anwärter*innen bekommen Prüfungsunterlagen zugeschickt, die innerhalb einer genannten Frist zurückgesendet werden müssen.
- ▶ Erfolgreiches Ablegen der Prüfung zur Auditor*in.
- ▶ Benennung zur Software-Auditor*in durch das Umweltbundesamt nach bestandener Prüfung.
- ▶ Auditor*innen sind natürliche Personen und keine Unternehmen oder Institutionen.

Benannte Auditor*innen werden auf der Webseite des Blauen Engels neben den Antragsunterlagen des Blauen Engels DE-UZ 215 aufgelistet: <https://blauer-engel.de/uz215>

Die Benennung zur Auditor*in ist zeitlich befristet und kann vom Umweltbundesamt wieder entzogen werden.

B.2 Aufgaben

Um das Audit durchzuführen und dazu beizutragen, unzureichende Antragsunterlagen zu korrigieren, kann die Auditor*in

- ▶ der Antragsteller*in Rückmeldungen zu den Kriterien geben, deren Überprüfung nicht erfolgreich war,
- ▶ die Antragsteller*in dazu auffordern, Angaben zu korrigieren, Messungen zu wiederholen oder weitere Nachweisdokumente vorzulegen,
- ▶ sich den Messaufbau oder die Lauffähigkeit der Software bei den genannten Systemvoraussetzungen durch die Antragsteller*in vorführen lassen.

Durch ihr Audit bestätigen Auditor*innen

- ▶ die Vollständigkeit und Plausibilität der Antragsunterlagen und der eingereichten Nachweisdokumente,
- ▶ die Angemessenheit des Messaufbaus und des Nutzungsszenarios,
- ▶ die Einhaltung von Fehlertoleranzen der Messergebnisse
- ▶ sowie die Nachvollziehbarkeit aller weiteren Angaben zur Antragstellung.

Die Bestätigung erfolgt im Antragsdokument in Anlage 1 durch eine Unterschrift der Auditor*in.

B.3 Pflichten

- ▶ Die Auditor*in muss die Auditierung mit großer Sorgfalt durchführen.
- ▶ Die Auditor*in muss wirtschaftlich unabhängig von der Antragsteller*in sein. Ausgenommen davon ist die Entgegennahme eines Honorars für die Durchführung des Audits selbst, dessen Höhe unabhängig vom Ergebnis des Audits sein muss.
- ▶ Die wirtschaftliche Unabhängigkeit beinhaltet, dass die Auditor*in nicht maßgeblich bei der Erstellung des Antrags zur Kennzeichnung eines Softwareproduktes mit dem Blauen Engel oder bei der Durchführung der Messungen mitgewirkt haben darf. Anderenfalls würde die Auditor*in ihre eigenen Arbeitsergebnisse überprüfen.
- ▶ Die Auditor*in muss institutionell unabhängig von der Antragsteller*in sein. Das heißt, sie darf nicht in der gleichen Firma oder Institution beschäftigt sein und in keinem institutionellen Abhängigkeitsverhältnis zur Antragsteller*in stehen.
- ▶ Die Auditor*in verpflichtet sich dazu, mit der jeweils neuesten Ausgabe der Vergabekriterien des Umweltzeichens für Ressourcen- und energieeffiziente Softwareprodukte vertraut zu sein. Dazu nimmt die Auditor*in nach Möglichkeit an entsprechenden Fachgesprächen des Umweltbundesamtes zur Weiterentwicklung der Kriterien und an den Expertenanhörungen des Blauen Engels für Software teil.

B.4 Arbeit als Auditor*in

- ▶ Antragsteller*innen des Blauen Engels für Software müssen ihre Anträge erfolgreich durch eine Auditor*in überprüfen lassen. Hierzu beauftragen die Antragsteller*innen eine Auditor*in, die auf der Auditor*innen-Liste des Blauen Engels benannt ist (<https://blauer-engel.de/uz215>). Die Wahl der jeweiligen Auditor*in bleibt der Antragsteller*in überlassen.
- ▶ Die Dienstleistung der Auditierung wird von der Auditor*in in der Regel gegen ein Honorar angeboten. Die Höhe dieses Honorars kann zwischen Auditor*in und Antragsteller*in frei ausgehandelt werden. Die Höhe darf allerdings nicht vom Erfolg der Prüfung abhängig gemacht werden, sondern sollte entweder pauschal oder entsprechend dem Bearbeitungsaufwand vereinbart werden.
- ▶ Eine Auditor*in darf für ihre Dienste auch außerhalb der Auditor*innen-Liste werben und sich als „Auditor*in für den Blauen Engel für Software“ (oder ähnlich) bezeichnen und den Bezug zum Umweltzeichen DE-UZ 215 herstellen. Die Nutzung des Blauen-Engel-Logos ist jedoch den Zeichennehmern (d.h. den erfolgreichen Antragsteller*innen) vorbehalten.
- ▶ Stellt eine Auditor*in Mängel bei der Antragsunterlagen fest, die von der Antragsteller*in nicht behoben werden können, so kann die Auditor*in ihre Unterschrift unter Anlage 1 des Antrags auf Kennzeichnung des Produktes verweigern und damit das Audit für nicht erfolgreich erklären.
- ▶ Auditor*innen werden dazu ermutigt, ihr Wissen weiterzugeben. Bei der Durchführung von Audits können sie mit dem Einverständnis der Antragsteller*innen anderen interessierten Personen Einblicke in die Antragsprüfung gewähren.

B.5 Kontrolle

Die Auditor*innen übernehmen einen Teil der Antragsprüfung für den Blauen Engel für Software. Um eine hohe Qualität der Auditierung zu gewährleisten, gibt es daher folgende Kontrollmechanismen:

- ▶ Die Vergabestelle des Blauen Engels, die RAL gGmbH prüft die von den Auditor*innen bestätigten Antragsunterlagen ihrerseits auf Plausibilität.
- ▶ Sofern eine Auditor*in das Audit für gescheitert erklärt, hat die Antragsteller*in die Möglichkeit, das Umweltbundesamt zur Schlichtung anzurufen. Das Umweltbundesamt prüft dann seinerseits die Antragsunterlagen und vermittelt ggf. zwischen Antragsteller*in und Auditor*in.
- ▶ Durch die Veröffentlichung von wesentlichen Informationen zum Softwareprodukt auf der Webseite des Blauen Engels, wie beispielsweise die Dokumentation des Messsystems, Messergebnisse und Angaben zur Abwärtskompatibilität (Vergabekriterium 3.1.3.7) können diese Angaben von der interessierten Öffentlichkeit eingesehen und überprüft werden.

Das Umweltbundesamt, Referat Z 2.3 Digitale Transformation und Beratungsstelle Green IT benennt einerseits Personen zur Auditor*in für den Blauen Engel, kann diese Benennung aber ebenso wieder entziehen.

Die Benennung zur Auditor*in kann vom Umweltbundesamt insbesondere entzogen werden,

- ▶ wenn die Auditierung fahrlässig oder wiederholt mit fachlichen Mängeln erfolgt,
- ▶ wenn die Auditor*in gegen die oben genannten Pflichten verstößt,
- ▶ wenn das Umweltzeichen seine Gültigkeit verliert oder
- ▶ wenn sich die Vergabekriterien des Umweltzeichens oder die festgelegten Messmethoden so weit ändern, dass ein erneuter Qualifikationsnachweis notwendig wird.