

BLUE ANGEL

The Environmental Label

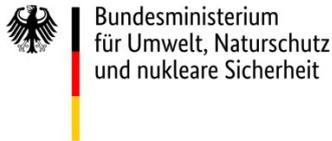


Resource and Energy-Efficient Software Products

DE-UZ 215

**Basic Award
Criteria**
Edition January 2020
Version 1

The environmental label is underpinned by the following institutions:



The Federal Ministry for the Environment, Nature Conservation and Nuclear Safety (Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit) is the owner of the label. It regularly provides information on the decisions taken by the Environmental Label Jury.



The Federal Environmental Agency (Umweltbundesamt) in the specialist department “Ecodesign, Eco-Labeling and Environmentally Friendly Procurement” acts as the office of the Environmental Label Jury and develops the specialist criteria in the form of the Basic Award Criteria for the Blue Angel environmental labels.



The Environmental Label Jury is the independent, decision-making body for the Blue Angel and includes representatives from environmental and consumer associations, trade unions, industry, the trade, crafts, local authorities, academia, the media, churches, young people and the German federal states.



RAL gGmbH is the awarding body for the environmental label. It organises the process for developing the relevant award criteria in independent, expert hearings – which involve all relevant interest groups.

If you require further information please contact:

RAL gGmbH

RAL ENVIRONMENT

Fränkische Straße 7

53229 Bonn

Tel.: +49 (0) 228 / 6 88 95 - 0

E-mail:

umweltzeichen@ral.de

Table of contents

1	Introduction.....	5
1.1	Preface	5
1.2	Background	5
1.3	Objectives of the environmental label	6
1.4	Definitions.....	7
1.5	Overview of possible future requirements.....	9
1.6	Information on submitting an application and compliance verification	9
2	Scope	10
3	Requirements	10
3.1	Requirements at the time of application	10
3.1.1	Resource and energy-efficiency.....	10
3.1.1.1	Minimum system requirements.....	10
3.1.1.2	Hardware utilisation and electrical power consumption in idle mode	11
3.1.1.3	Hardware utilisation and energy demand when running a standard usage scenario.....	12
3.1.1.4	Support for the energy management system.....	12
3.1.2	Potential hardware operating life.....	13
3.1.2.1	Backward compatibility.....	13
3.1.3	User autonomy	13
3.1.3.1	Data formats.....	13
3.1.3.2	Transparency of the software product.....	14
3.1.3.3	Continuity of the software product	15
3.1.3.4	Uninstallability.....	15
3.1.3.5	Offline capability.....	16
3.1.3.6	Modularity	16
3.1.3.7	Freedom from advertising.....	16
3.1.3.8	Documentation of the software product, licence conditions and terms of use	17
3.2	Requirements during the term of the contract	18
3.2.1	Requirements for the further development and update of the product	18
3.3	Requirements at the end of the term of the contract.....	18
3.3.1	Final evaluation.....	18

4	Applicants and parties involved	19
5	Use of the Environmental Label	19
	Appendix A Measurement instructions for criteria 3.1.1.2 and 3.1.1.3	20
	Appendix B Information for the measurement of the criteria.....	24
	Appendix C Data format for passing on the product information about resource and energy efficiency	29
	Appendix D Reference system and standard usage scenario	34

1 Introduction

1.1 Preface

In cooperation with the Federal Ministry for the Environment, Nature Conservation and Nuclear Safety, the Federal Environmental Agency and considering the results of the expert hearings conducted by RAL gGmbH, the Environmental Label Jury has set up these Basic Criteria for the Award of the Environmental Label. RAL gGmbH has been tasked with awarding the Environmental Label.

Upon application to RAL gGmbH and on the basis of a Contract on the Use of the Environmental Label to be concluded with RAL gGmbH, the permission to use the Environmental Label may be granted to all products, provided that they comply with the requirements as specified hereinafter. The product must comply with all the legal requirements in the country in which it is to be marketed. The applicant shall declare that the product meets these conditions.

1.2 Background

The annual energy consumption by the telecommunications network and data centers in Germany stood at around 18 TWh in 2015. Due to the advance of digitalisation, this figure is expected to increase to 25 TWh in 2025, especially because of the increase in Internet-based services¹. The volume of data that is transmitted via the Internet is also constantly increasing, dramatically in some cases. According to the Federal Network Agency, the volume of data transmitted via the fixed-line network increased 4.5 fold between 2011 and 2016 (28 billion GB in 2016). On the mobile phone network, the volume of data increased even more drastically, by more than 800% compared to the figure in 2011 (913 million GB in 2016)². Nevertheless, the subject of digitalisation is rarely mentioned in connection with any environmental impacts³. A fact that is even more rarely appreciated is that the key to increasing energy efficiency and protecting natural resources lies not with the hardware but rather above all with the software.

Processing power has doubled about every two years since 1970. This means that functions are processed twice as fast and thus less energy is required for the same functions. A similar improvement in efficiency cannot be observed in the field of software. In contrast, the efficiency gains in hardware have been overcompensated for by higher software requirements and the need for larger volumes of data. This often results in the so-called "rebound effect", e.g. improvements in resource efficiency increase the demand for a product or service so that less resources are saved or, in some cases, even more resources are required than would have been the case if demand had remained constant⁴. The availability of more and more powerful hardware has resulted in software becoming more and more bloated from version to version so that more resources are required for only minimal or even no enhancement of the functionality.

¹ <https://www.bmwi.de/Redaktion/DE/Downloads/E/entwicklung-des-ikt-bedingten-strombedarfs-in-deutschland-abschlussbericht.html>

² Federal Network Agency Activity Report 2016/2017
https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Allgemeines/Bundesnetzagentur/Publicationen/Berichte/2017/TB_Telekommunikation20162017.pdf

³ https://www.dbu.de/123artikel38000_2442.html

⁴ Hilty, L. et al. (2013): Grüne Software - Ermittlung und Erschließung von Umweltschutzpotenzialen der Informations- und Kommunikationstechnik (Green IT) (Green Software - Establishing and exploiting potentials for environmental protection in information and communication technology (Green IT))

Examples for this “software bloat” can be found, for example, in the area of operating systems. It is not uncommon for the launch of an operating system that places, in some cases, much higher requirements on the hardware to result in the replacement of hardware that was generally still functioning well. In the area of desktop application software, there are, however, also clear differences in the consumption of energy and resources depending on which software product is used to complete the same task: for example, the energy demand from one text processing program was four times as high as another text processing program when completing the same standard usage scenario on the same hardware (3.6 Wh compared to 0.93 Wh)⁵. Therefore, software plays a special role here because the characteristics and functions of the software determine which hardware capacities are required and consequently what energy demands are made by the information and communications technology (ICT).

The environmental label for “Resource and Energy-Efficient Software Products” provides special guidance to purchasers and users of software products and raises awareness about the role played by software in the area of ICT. The focus will firstly be placed on one area of ICT: application software that is primarily run on desktop systems (see Scope). The aim is to expand the scope of the environmental label later on to other architectures, such as client server systems and mobile apps.

1.3 Objectives of the environmental label

The design of the software and its programming have an influence on the hardware specifications and also the energy consumption due to the use of the software.

The Blue Angel environmental label for “Resource and Energy-Efficient Software Products” (DE-UZ-215) may be awarded to products that use hardware resources in a particularly efficient manner and consume a low amount of energy during their use. Due to the lower performance requirements, longer operating lives for the hardware are possible. In addition, these software products stand out due to their high level of transparency and give users greater freedom in their use of the software. The Blue Angel will be awarded to the best products that are currently available on the market in a category and the label will not make any comparative assessment between the products.

The aim of the environmental label is to reduce the total energy consumed by information and communication technology and improve resource efficiency. In particular, the Blue Angel will highlight those products whose manufacturers provide information about their products, especially with respect to their energy and resource efficiency, to recognise this transparency. In addition, the environmental label will also highlight those products whose manufacturers are actively engaged in improving the resource and energy efficiency of their products.

Therefore, the following benefits are stated in the explanatory box:



⁵ https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2018-12-12_texte_105-2018_ressourceneffiziente-software_0_

1.4 Definitions

Application software: Software programs that provide a “benefit” for end users. Examples of application software are: image processing applications, e-mail programs, web browsers, multimedia applications, text processing or spreadsheet applications.

End of sale: The time at which the software product is no longer provided for sale by the applicant. It is still possible that support is provided for the product, e.g. it does not necessarily mean an end of support (see End of support).

Desktop computer: A computer whose main unit is set up in a fixed location, is not designed for mobile use and is operated with an external display as well as external peripheral devices such as a keyboard and mouse.

Escrow: Depositing the source code and documentation for a software product with a third-party company or notary to guarantee that the source code can still be viewed in the event of certain events (e.g. a transfer of power or support, further development). An agreement defines when the source code can be issued to the user.

Energy efficiency: Generally, the amount of “useful work” divided by the amount of energy it requires. In the context of this document, “useful work” is understood to mean the successful execution of standard usage scenarios.

Hardware: All of the material goods required to run programs or to store or transport data.

Hardware capacity: Quantifiable characteristic of a hardware system which represents its performance limit on a given dimension of performance (e.g. working memory capacity, computing power, bandwidth).

Hardware system: Delimitable unit of hardware that performs defined functions.

Indicator: An empirically determinable quantity that provides insight into a matter that cannot be measured directly.

Idle mode: A state in which the software product has been started but no functionalities have been executed.

Usage pattern: Abstracted form of a sequence of interactions with a given software product.

Usage scenario: Description of a usage pattern which is generally machine executable.

Reference system: A hardware system that is defined as generally customary in terms of its most important capacities (e.g. working memory, processor performance) during a defined period of time (e.g. one year).

Resource: A natural resource, in particular a raw material, a form of energy, or also the capacity of an environmental medium to absorb emissions. To differentiate natural resources from technical ones, especially hardware resources, the more precise term “hardware capacities” is used here for the latter. Since the use of hardware capacities always results in the use of natural resources, this distinction (which ultimately requires a difficult to define differentiation between the ecosphere and the technosphere) is not of decisive importance here.

Resource efficiency: Generally, the amount of “useful work” divided by the amount of resources it requires. In the context of this document, “useful work” is understood to mean the successful execution of standard usage scenarios.

Software: Programs and associated data in digital form.

Software product: A licensed copy of software that is designed for carrying out the required services, which are implemented and processed on a hardware system.

Software stack: Software components that build on one another to form the software product or which are required for its execution.

Standard usage scenario: A usage scenario that is used for testing a software product and which should be as representative as possible for the customary use case.

End of support: Time at which the applicant neither makes the software product itself available for sale (end of sale) nor provides support for the product.

System Under Test (SUT): A hardware system on which the energy consumed and the utilised hardware capacities are measured (e.g. during the completion of a standard usage scenario). Alongside the hardware, it also includes all of the software (software stack: e.g. operating system, runtime environment, etc.) required to operate the software product and the configuration of the software product and software stack.

Spreadsheet file for recording the criteria: This file is a spreadsheet in which information about the software product, the reference system used for the measurements, the standard usage scenario and the results of the assessment of the criteria for the software product are recorded.

Update: An updated and, where relevant, improved version of a software product (also including patches). In the case of these Basic Award Criteria, an update is

relevant when a new standard usage scenario is required following the further development of the labelled software product, e.g. it is no longer possible to fully cover the main functions (typical user interactions) of the software with the existing standard usage scenario.

1.5 Overview of possible future requirements

Due to the lack of measurement methods and benchmarks, these Basic Award Criteria for the environmental label do not specify any quantitative minimum requirements. This area will be examined again during the next revision and minimum requirements may be set.

The scope of these Basic Award Criteria covers software products that belong to the group of application software with a user interface. The next revision of the Basic Award Criteria will expand the scope to include, above all, server-client software products. This will require, amongst other things, the definition of a reference system for servers. The question of whether system software should also be included within the scope of the environmental label will also be examined in the medium term.

In addition, the next revision will examine whether testing laboratories can be used to carry out the measurements of the energy consumption and hardware utilisation.

Due to the lack of usage statistics, the standard usage scenarios that are completed to evaluate the resource and energy efficiency of the software product can be freely selected in these Basic Award Criteria. This area will be examined again during the next revision and standardised usage scenarios may be set for each product group.

1.6 Information on submitting an application and compliance verification

The requirements for energy and resource efficient software products formulated in these Basic Award Criteria are valid at both the time of application and also during the use of the environmental label. Compliance with the requirements must be verified at the time of application before the environmental label is awarded (Paragraph 3.1) and, in the event of updates, during the term of the contract (Paragraph 3.2) as well as at the end of the term of contract for the final evaluation (Paragraph 3.3).

When submitting an application, all of the compliance verifications stated in Paragraph 3.1 must be provided. The central verification document at the time of application is the spreadsheet file for recording the criteria, in which information about the software product, the reference system used for the measurements, the standard usage scenario and the results of the assessment of the criteria for the software product are recorded (Annex 2).

The values contained in this file must be examined and confirmed by an auditor. RAL gGmbH, with the support of the Federal Environmental Agency, will approve auditors. The approval of the auditors guarantees that the testing of these values is carried out independently and with a high level of professional qualifications.

2 Scope

Within the scope of these Basic Award Criteria, a software product is defined as follows:

A software product provides a benefit for end users. These Basic Award Criteria for energy and resource efficient software products refer to application software that has a user interface and can be run on one of the reference systems stated in Appendix D. Software products for which the vast majority of the computing work is not carried out on the local computer but is instead outsourced to a remote server do not fall under the scope of these Basic Award Criteria.

In addition, application software that comes under the scope of these Basic Award Criteria must comply with the requirements of the currently valid version of the Interstate Treaty on the Protection of Minors (JMStV)⁶ and must not display any of the characteristics stated in "§ 4 Illegal Content" of the JMStV.

3 Requirements

3.1 Requirements at the time of application

Verification of compliance with the requirements stated in Paragraph 3.1 must be submitted at the time of application. Verification must be provided for the current or explicitly stated version of the software product. The Contract on the Use of the Environmental Label will be issued on this basis. If this version of the software is updated during the term of the contract, verification of compliance with the requirements in Paragraph 3.2 must also be provided.

The requirements are split into three sections:

- 3.1.1 Resource and energy efficiency
- 3.1.2 Potential hardware operating life
- 3.1.3 User autonomy

3.1.1 Resource and energy efficiency

A software product must deliver its functions with a minimal use of resources and a minimal energy demand. The resource and energy efficiency of the software product should be maximised. To operationalise the resource efficiency, the hardware resources and energy consumption will be used as reference parameters.

3.1.1.1 Minimum system requirements

The minimum system requirements for the operation of the software product must be stated.

- Minimum processor architecture incl. generation (e.g. Intel i5-3570k 3.4 GHz 4 Core (64-Bit), Intel Atom x7-Z8700 processor)

⁶ Interstate Treaty on the Protection of Human Dignity and the Protection of Minors in Broadcasting and in Telemedia (Interstate Treaty on the Protection of Minors – JMStV) in the version of the 19th Treaty for amending the Interstate Treaties with regard to broadcasting law (19th Interstate Broadcasting Treaty), https://www.kjm-online.de/fileadmin/user_upload/Rechtsgrundlagen/Gesetze_Staatsvertraege/JMStV_Genese/Jugendmedienschutz-Staatsvertrag_JMStV_in_der_Fassung_des_19_RA_StV.pdf

- Minimum local working memory required (MByte)
- Minimum local permanent storage required (MByte)
- Requirements for other software (operating system, middleware and auxiliary applications: software stack) (e.g. Windows 7, .NET Framework and browser version XY)
- The required external services that are not available on the reference system (e.g. cloud services, storage services, API usage, ...)
- The required additional hardware (e.g. graphics card, peripheral devices such as a camera connected via USB). A prerequisite is that the additional hardware can be directly integrated into the reference system (see Appendix D).

Compliance verification

The applicant shall state the minimum system requirements in Annex 1 to the contract and enter this information in the spreadsheet file for recording the criteria (Annex 2 to the contract).

3.1.1.2 Hardware utilisation and electrical power consumption in idle mode

The hardware utilisation and electrical power consumption when the software product is in idle mode must be stated. It is determined as the sum of the additional load on the hardware due to loading the software product and the percentage share of the base load (calculation guidelines: see Appendix B).

The following information when the software product is in idle mode must be stated:

- Average processor utilisation in idle mode (%)
- Average working memory utilization in idle mode (MByte)
- Average permanent storage utilization in idle mode (MByte/s)
- Average bandwidth utilization for network access in idle mode (Mbit/s) (only contains the additional load, not the percentage share of the base load)
- Average electrical power consumption (net) (W)

Compliance verification

The software product shall be installed on a reference system and the hardware utilisation determined using the measurement method described in Appendix A. One of the computer systems stated in Appendix D shall be used as the reference system, whereby the choice of system and its reference year is left up to the applicant. The selected reference system, the software stack used and any configuration (of the software product and software stack) that deviates from the delivered state shall be stated in the spreadsheet file for recording the criteria (Annex 2 to the contract). The applicant shall state the measurement values in Annex 1 to the contract and document them in the spreadsheet file in Annex 2 to the contract. In addition, the applicant shall submit the measurement report (see Appendix A) as Annex 3 to the contract.

3.1.1.3 Hardware utilisation and energy demand when running a standard usage scenario

The hardware utilisation and energy demand when using the software application to run a standard usage scenario must be stated (calculation guidelines: see Appendix B).

The basis for the calculations are log files that are saved by the reference system when running the software product. The hardware utilisation includes both the additional load caused by the use of the software product and also a percentage share of the base load (see Appendix B).

The measurement units for hardware utilisation are units for work performance, such as %*s (processor utilisation), MByte*s (working memory utilisation), MByte/s*s = MByte (permanent storage utilisation, reading and writing) and MBit/s*s = MBit (volume of data transferred in the network).

Alongside the hardware utilisation, the electrical energy demand of the reference system must also be determined. The required electrical energy is determined by integrating the electrical power consumption over the time taken to run the standard usages scenario. Contrary to the specifications for the hardware utilisation, the measurements for the electrical energy only take into account the power values that go beyond the base electrical load for the reference system (net power) (see: Appendix B).

The following information must be provided when executing the standard usage scenario under the standard configuration:

- Processor utilisation (%*s)
- Working memory utilisation (MByte*s)
- Permanent storage utilisation (reading and writing) (MByte/s*s)
- Volume of data transferred for network access (Mbit/s*s)
- Average energy demand (net) (Wh)

Compliance verification

The measurements shall be carried out under the same conditions used for determining the criteria in Paragraph 3.1.1.2. The applicant shall state the measurement values in Annex 1 to the contract and document them in the spreadsheet file in Annex 2 to the contract. In addition, the applicant shall submit the measurement report (see Appendix A) as Annex 3 to the contract.

3.1.1.4 Support for the energy management system

A software product must not require that an already existing energy management system on the computer (e.g. standby mode/idle mode) is deactivated/uninstalled for it to operate perfectly. The energy management system on the computer must not be negatively influenced by the software product. The functionality of the software product must not be influenced by the energy management system (e.g. loss of data, impaired usability). The need to reregister (without any loss of data) with a server-based software or the time required to reactivate the computer system from an energy saving mode are not considered impairments to the usability and functionality of the software.

The following requirement must be fulfilled:

There must be no restriction to the functionality of the software product when an energy management system on the underlying system layers or the connected client system has been activated.

Compliance verification

The applicant shall declare compliance with the requirement in Annex 1 to the contract.

3.1.2 Potential hardware operating life

The software must not contribute to the early replacement of existing hardware with more powerful hardware because the existing hardware no longer meets the performance requirements of the software. In addition, software updates must not result in the need for a hardware update. Instead, users must be able to make mutually independent decisions on the purchase of new software and hardware. Therefore, the requirements in this section address the potential hardware operating life.

3.1.2.1 Backward compatibility

It must be possible to run the software product on a reference system from a calendar year that is at least five years before the time of application. The supplier of the software product must make a binding declaration about which reference system from which calendar year can still be used to run the software. Reference systems from the year 2015 onwards are already defined (see Appendix D). The required software stack (e.g. operating system, framework, application software) and the configuration of the software product and software stack must be taken into account when determining the relevant year.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract by stating the calendar year for the relevant reference system including the required software stack and document this information in the spreadsheet file as Annex 2 to the contract (“Backward compatibility”).

3.1.3 User autonomy

The software product must not restrict user autonomy with respect to the product and must not create any dependencies. The requirements described below contribute to different aspects of user autonomy.

3.1.3.1 Data formats

To enable interoperability of the software product, the data formats (file and data stream formats) used by the software product to store the data generated by users or to exchange data with other programs must be adequately documented. The data formats must be disclosed in order to enable further use of the data with another software product and thus promote the basic principle behind this Blue Angel of labelling those products whose manufacturers work in a particularly transparent way (see Paragraph 1.2).

The data formats must be documented using one of the following methods:

- Submitting the manuals or technical data sheets in which the data formats are documented or
- Providing examples of other software products (from other suppliers) that can process these data formats or
- Stating the data formats and assigning them to an open standard.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract using one of the options described above and submit corresponding verification documentation as Annex 4 to the contract.

3.1.3.2 Transparency of the software product

In order to enhance the software product with additional functions, the application programming interfaces (APIs) must be clearly documented. The APIs should correspond to open standards. In the interests of long-term and comprehensive use of the product, the source code should also ideally be fully or partially published, at the latest when the product is no longer being updated, and permission to change the product granted.

The following information must be provided:

- If application programming interfaces are available: Template for the interface documentation.
- Information on the extent to which program components with documented source code have been published or whether the entire software product has been published as open source code. If the source code for the software product has been published, the applicant must provide information on how the source code can be accessed (e.g. via standard source code management platforms such as GitHub, GitLab, Bitbucket, Sourceforge, etc.)
- An explanation of what will happen to the software product after the end of support, e.g. when the product or the certified version of the product can no longer be purchased and/or support is no longer available for the use of the product or the version of the product (e.g. publication and free licensing of the supplier's own source code, further support for users via third parties, escrow contract, ...)
- Information on the licence under which the software product will be sold and the rights that the licence provides including information on whether third parties are permitted to develop the software product.
- Information on the licences for all software components, insofar as they differ from the licence under which the software product is sold.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract and submit the corresponding verification documents (interface documentation, permalink to the software source code on a source code management platform, software licences or similar) as Annex 5 to the contract.

3.1.3.3 Continuity of the software product

It must be possible to use the software product over a long period of time without any serious disadvantages arising (especially problems related to IT security). In addition, the software manufacturer must offer a function that can be used to keep the software product up-to-date. Security updates must be provided free of charge, updates offering additional functionalities are excluded from this rule. The applicant undertakes to supply security updates for the labelled product for at least 5 years after the end of sale.

When software updates are made available by the supplier, the user must be given the option of whether to install only security updates or also other (e.g. functional) updates. In general, the requirements described in Paragraph 3.2 for the handling of updates during the term of the contract must be taken into account when making software updates available.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract, mark the corresponding sections of the product information in which software updates are indicated and submit the relevant pages of the product information as Annex 6 to the contract.

3.1.3.4 Uninstallability

It must be possible to completely remove the software product from the computer system after the end of its operating life without leaving any unnecessary traces of data. Data that has been created and processed by the user with the software product is excluded from this rule and must not be automatically deleted when uninstalling the software product (exception: the user confirms the deletion of this data). For this purpose, it must be possible for the user (e.g. a system administrator) to easily uninstall the software product including any additional components or libraries that may be installed by the software within a short period of time.

The product information must include a description of the process for uninstalling the product.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract, mark the corresponding sections of the product information in which the process for uninstalling the software is indicated and submit the relevant pages of the product information as Annex 6 to the contract.

3.1.3.5 Offline capability

The functionality and availability of the software must not be negatively influenced by external factors, such as the availability of a licence server.

In addition, the software product must also remain largely functional even without a network connection (offline) insofar as this is not explicitly required for fulfilling its intended function. If a network connection is required, this must be indicated in the product information

and the need for the network connection must be justified. Offline use of the product must be possible in all cases (limited where relevant).

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract, mark the corresponding sections of the product information in which the offline capability is indicated and submit the relevant pages of the product information as Annex 6 to the contract.

3.1.3.6 Modularity

Due to the modular installation of the software product, which includes the main function and other non-essential functions, it should be possible to reduce the memory and energy demands. Users of the software product should be given the option of limiting the functions to the desired range of functions.

The following information must be provided:

- Information on how individual modules of the software product can be deactivated during the installation process.
- Information on the extent to which individual modules of the software product (especially those that do not belong to the functions of the software product such as tracking, etc.) can be deactivated during the use of the software product.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract, mark the corresponding sections of the product information in which the modularity is indicated and submit the relevant pages of the product information as Annex 6 to the contract.

3.1.3.7 Freedom from advertising

Displaying advertising can increase the resource and energy demands of the software product. In particular, displaying advertising that is provided e.g. by an external service provider will result in the transmission of additional volumes of data from the client to the server and vice versa. This network traffic is not important for the direct functionality of the software product and should be avoided. Therefore, software products that are awarded this environmental label must be free of advertising. Advertising for the manufacturer's own company, other versions of the software or other software from the same manufacturer are exempt from this rule.

Compliance verification

The applicant shall declare compliance with the requirement in Annex 1 to the contract.

3.1.3.8 Documentation of the software product, licence conditions and terms of use

The software product must be provided with documentation that enables long-term and resource-conserving use of the product. The applicant must provide corresponding information about the software product both publicly and also in combination with the product itself.

The following usage information must be included in the documentation and also in the relevant product information (manual, website, etc.):

- a) Description of the processes for installing and uninstalling the software
- b) Description of the data import and export processes
- c) Information on reducing the use of resources (e.g. activating energy saving mode, deactivating unused modules, influencing the size of swap files, cleaning up residual data)
- d) Information on the licensing terms and terms of use to enable, where relevant, the legally compliant further development of the software product
- e) Information on software support
- f) Information on the handling of data, in the sense of existing data protection laws This also applies if the applicant's services or parts thereof are operated or provided on servers located outside of the German or European area of justice.
- g) Information on data security, data collection and data transmission:
 - Information on how individual modules of the software can be deactivated during the installation process.
 - Information on the extent to which individual modules of the software (especially those that do not belong to the functions of the software such as tracking, advertising, etc.) can be deactivated during the use of the software.
 - Declaration that data will only be collected and stored to the extent required for the direct provision of the functions of the software.
 - Confirmation that the type, scope, duration of storage and purpose of use of the collected data will be explained to the user and a corresponding declaration of consent will be obtained.
 - Declaration that data will only be transferred to third parties with the explicit agreement of the user and only to the extent required for the function of the software.
 - Confirmation that the collected data will not be used for advertising purposes.
 - Confirmation that any data that is no longer required will be immediately and permanently deleted after the end of use of the software.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract, mark the corresponding sections of the product information and submit the relevant pages of the product information as Annex 6 to the contract. The applicant shall also declare that he/she is willing to publish the contents of the spreadsheet file for recording the criteria (Annex 2) on the website for the relevant product and will supply the relevant data in a defined data format (see Appendix C) as Annex 7.

3.2 Requirements during the term of the contract

3.2.1 Requirements for the further development and update of the product

If the product is changed (e.g. due to an update), the applicant must ensure that the software product still complies with all of the criteria. The changes must not result in a need for the hardware to be replaced in order to continue using the software. In addition, it must still be possible after an update to run the same standard usage scenario on the same reference system as used for the application and to run the software on the reference system that was valid five years before the application. If this is not the case, a new application for the Blue Angel must be submitted.

An exemption is possible for the energy demand value: The energy demand must not increase by more than 10% compared to the value at the time of application, otherwise the reasons for the change must be stated and reported to RAL gGmbH. Following an update, the energy demand value must be measured on the reference system that was also used for the application for the Blue Angel.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract. The applicant shall update the spreadsheet file for recording the criteria (Annex 2) and the information provided on the product website, as well as Annex 7, accordingly with the new values.

3.3 Requirements at the end of the term of the contract

3.3.1 Final evaluation

The applicant must submit a Resource Efficiency Report to RAL gGmbH at the end of the relevant term of contract. This report must include the values measured during the term of the contract (Paragraph "3.1 Resource and energy efficiency", for example in the event of an update). The measures that were taken to increase the resource and energy efficiency of the software during the term of the contract must be described.

If the holder of the environmental label does not fulfil his/her obligation to submit a Resource Efficiency Report or the report documents that the requirements placed on the software product by the Blue Angel have not been observed, this represents an infringement of the Contract on the Use of the Environmental Label (see Paragraph 5) and can be sanctioned by RAL gGmbH with the removal of permission to use the environmental label. A new application or extension of the Blue Angel for the software product or a software product with the same name is then also not possible in the event of other versions/updates.

Compliance verification

The applicant shall declare compliance with the requirements in Annex 1 to the contract. In addition, the applicant shall submit a Resource Efficiency Report at the end of the term of the contract. For this report, the applicant can provide the spreadsheet file for recording the criteria supplemented with a description of the measures taken for improving resource efficiency. The values for the requirements from Paragraph 3.1.1 must be included in the Resource Efficiency Report.

4 Applicants and parties involved

Manufacturers or distributors of software products shall be eligible for application. In the case of a software product that was produced collaboratively (e.g. free software), an individual company or a legal person is eligible for application. In the event of updates to the certified products, it is necessary to check whether a new application is required (see Definitions and Paragraph 3.2).

Parties involved in the award process are:

- RAL gGmbH to award the Blue Angel environmental label,
- the federal state being home to the applicant's production site,
- Umweltbundesamt, (Federal Environmental Agency) which after the signing of the contract receives all data and documents submitted in application for the Blue Angel in order to be able to further develop the Basic Award Criteria.

5 Use of the Environmental Label

The use of the environmental label by the applicant is governed by a contract on the use of the environmental label concluded with RAL gGmbH.

Within the scope of such contract, the applicant undertakes to comply with the requirements under Paragraph 3 while using the environmental label.

Contracts on the Use of the Environmental Label are concluded to fix the terms for the certification of products under Paragraph 2. Such contracts shall run until 31/12/2023.

They shall be extended by periods of one year each, unless terminated in writing by 31/03/2023 or 31 March of the respective year of extension.

After the expiry of the contract, the Environmental Label may neither be used for labelling nor for advertising purposes. This regulation shall not affect products being still in the market.

The applicant shall be entitled to apply to RAL gGmbH for an extension of the right to use the ecolabel on the product entitled to the label if it is to be marketed under another brand/trade name and/or other marketing organizations.

The Contract on the Use of the Environmental Label shall specify:

- Applicant (manufacturer/distributor)
- Brand/trade name, product description
- Distributor (Label User), i.e. the marketing organization.

Appendix A Measurement instructions for criteria 3.1.1.2 and 3.1.1.3

A possible method for carrying out the software measurements to record energy consumption data and hardware utilisation is described below.

These instructions are an example process for measuring the consumption data. The steps were tested with the operating systems Windows 7, Windows 10, Ubuntu 16.04 and Ubuntu 18.04 and can be transferred to other operating systems.

1 Measurement process

In order to determine the additional energy consumption and additional hardware utilisation due to the software product, it is necessary to complete 3 measurements:

- a) Measurement of the base load
- b) Measurement of the idle load
- c) Measurement of the load when executing the standard usage scenario

The basic measurement process is as follows:

- To measure the base load, the operating system and the required software stack are installed on the reference system. Software components that are supplied together with the software product (e.g. Java runtime, database or similar that are supplied with the software product) are excluded from this process. If required, an automation software (see below) is also installed. The measurement is then carried out on this system in accordance with the measurement instructions. Except for the operating system and, if necessary, any required automation software (see below), no other software is started.
- To measure the load in idle mode, the software product to be certified with the Blue Angel is installed and started, in addition to the components that were installed for the measurement of the base load. After starting the software, the measurement is carried out on this system as described below.
- To measure the load when executing the standard usage scenario, the installed software is identical to the system used for measuring the load in idle mode. For the measurement, the software product and, if required, any automation software are started and the standard usage scenario is run automatically. The load is measured as described below during this process.
- Once the measurement has been completed, the generated data is evaluated.

2 Equipment required for the measurement

The following equipment is required to carry out a measurement:

- SUT - System Under Test
- Power measurement device
- Computer for reading the power measurement device and calculating the average hardware utilisation and the energy consumption (see below).

3 Statistical significance of the measurements

To enable valid statistical evaluations of the measurements, it is important to complete the same measurement multiple times in succession. This can be achieved automatically using software that records macro commands or using a script or program developed for this purpose that repeatedly runs a scenario on the reference system.

These measurements can be used to determine whether the electricity consumption values for the various measurements differ significantly from one another (e.g. after updates). The measurements can also provide useful information for software developers on how different procedures or architectures impact the consumption of energy and resources. To minimise the risk of random outliers in the measurements, the measurements must be repeated at least 10 times, although we recommend that they are repeated 30 times.

4 Preparatory work

Before starting the measurements of the software, the following preparatory measures should be completed:

- Time synchronisation: To carry out a valid measurement, the times on the power measurement device, reference system and, if relevant, other hardware must be the same.
- For measuring the load in idle mode and during the standard usage scenario, it is also necessary to install the software product to be certified with the Blue Angel.
- Recording the standard usage scenario.

5 Recording the standard usage scenario

To automatically record manual user inputs on the computer, macro recorders (e.g. WinAutomation⁷, Pullover's Macro Creator⁸) are used that are able to record a sequence of actions that a user would typically carry out using mouse clicks and keyboard entries on the software to be certified with the Blue Angel. These actions can be recorded either by directly starting a recording mode or by manually entering and editing them. Whether only the option of using the described recording mode is available or whether both options can be used is dependent on the relevant software.

Alternatively to this type of recording software, the automated standard usage scenario can also be created using scripts or programs developed in-house. It is important that the same software is used for every measurement (base load, idle mode and standard usage scenario) so that any measurement errors caused by the automated software are minimised.

The result of the recording process is e.g. an executable file or a script that collects together the user inputs so that the standard usage scenario can be run automatically. It is recommended that a log file is also created by the automation software, in which a time stamp for the start and end of the individual measurements is logged.

⁷ <https://www.winautomation.com/>

⁸ <https://www.macrocreator.com/>

6 Tools for determining the performance indicators

Alongside the energy values that are measured using an ammeter, it is also necessary to measure other indicators (processor utilisation, RAM utilisation, hard disk activity and network traffic). For this purpose, another software that can record performance data and the current system status is required (e.g. Windows Performance Monitor⁹, Collectl¹⁰). It is also recommended here that the applicant creates a log file using the software that contains the measured data and creates a time stamp. In addition, it is recommended that the measurements are taken at least every second.

7 Recording the electrical power consumption

To measure the electrical power consumption, a measurement device (e.g. Janitza UMG 604, Gude Expert Power Control 1202) that can record the electrical power consumption during the measurements is required. It is recommended that the applicant selects a measurement device that aggregates the measured power consumption over one second and saves it together with a time stamp in a log file.

8 Process for measuring the performance indicators and power consumption during the usage scenario

After the preparatory steps have been completed (see Sections 4 and 5 of Appendix A), the measurements for the power consumption and performance indicators are started.

- Start the tool for monitoring the performance indicators (see Section 6, Appendix A).
- Start the measurement of the power consumption.
- Now start the file generated by the macro reading software or the automation software or the automation script.
- No interventions are permitted until the measurement has been completed so as not to falsify the measurement results.
- Stop the measurement of the performance data after the standard usage scenario has been completed.

9 Reading the data

The measurement has now been completed. At this point in time, it is recommended that the raw data from the measurements is saved together in one storage location. The recorded data will be evaluated in the next step (see Appendix B).

⁹ <https://techcommunity.microsoft.com/t5/Ask-The-Performance-Team/Windows-Performance-Monitor-Overview/ba-p/375481>

¹⁰ <http://collectl.sourceforge.net/>

Appendix B Information for the measurement of the criteria

1 Basic definitions

The following definitions are valid for measuring the indicators in Paragraphs 3.1.1.2 and 3.1.1.3:

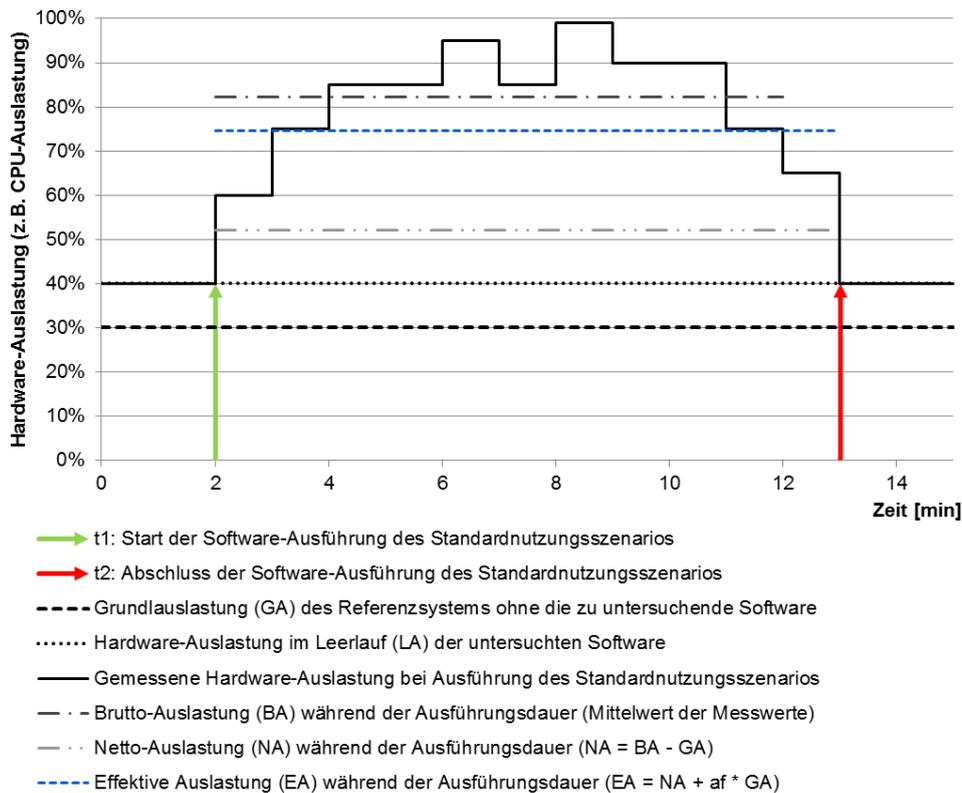
Table 1: Basic definitions for the measurement of the software product (3.1.1.2 and 3.1.1.3)

Identifier	Name	Definition	Comment
FLi	Full load	Upper limit of the capacity i for the reference system	For processing power, the full load is 100%, for working memory the sum of the installed RAM capacities, for network bandwidth the maximum transmission speed, etc.
BLi	Base load	Average load of the capacity i for the reference system when the software product is not installed.	This value represents a constant value for the reference system.
ILi	Idle load	Average load of the capacity i for the reference system when the software product is installed but in idle mode.	It includes both the base load (BLi) and also the additional hardware load when the software is in idle mode.
NILi	Net idle load	$NILi = ILi - BLi$	It describes the idle load of the software without the proportion of the base load (BLi).
t	Time	The time needed by the software product to execute the standard usage scenario on the reference system.	It begins when the standard usage scenario starts and ends when all required actions have been executed, including follow-up processes (such as releasing memory, deleting temporary files).
GLi	Gross load	Average load of the reference system when running a standard usage scenario over the time taken to execute it	It is calculated as a time-weighted average of the measured value over t.
NLi	Net load	$NLi = GLi - BLi$	It describes the average hardware load of the software product after deducting the base load for the reference system.
AFi	Allocation factor	$AFi = NLi / (FLi - BLi)$ (allocation factor for the execution of the standard usage scenario)	Ratio between the net load of the software product and the maximum load available. It is important to note here that the maximum load available is the difference between the full load (FLi) and the base load (BLi). This factor is required for calculating the effective load (EL).

AFLi	Allocation factor idle	$AFLi = NILi / (FLi - BLi)$ (allocation factor for the software in idle mode)	Ratio of the net idle load (NILi) to the total available load (FLi – BLi). This factor is required for calculating the effective idle load (EILi).
ELi	Effective load	$ELi = NLi + AFLi * BLi$	The effective load is the sum of the net load (NLi) and a proportion of the base load (BLi), which is determined using the allocation factor (AFLi).
EILi	Effective load idle	$EILi = NILi + AFLi * BLi$	The effective load idle is the sum of the net load idle (NILi) and a proportion of the base load (BLi), which is determined using the allocation factor idle (AFLi). See Paragraph 3.1.1.2.
HDi	Hardware utilisation	$HDi = ELi * t$	The hardware utilisation is the product of the effective load (ELi) and the time (t). Its units are, for example, percent per second (%s), megabyte per second (MByte*s), megabytes (MByte/s*s) and megabits (MBit/s*s). See Paragraph 3.1.1.3.

The following diagram shows an example measurement cycle and illustrates the different loads and their descriptions.

Fig. 1: Example measurement cycle to determine the hardware utilisation



Source: Final report for the project “Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik” (Development and application of assessment criteria for resource efficient software taking into account existing methodology) (UFOPLAN-SSD-2015)

2 Information on measuring the base load

The hardware utilisation and the electrical power consumption during operation of the system without the software product installed (base load) is required in order to determine the indicators in Paragraphs 3.1.1.2 and 3.1.1.3. To measure the base load, the software (operating system, software stack) required to operate the software product is installed on the reference system but not the actual software product itself or any software components that are supplied with the software product. Except for the operating system and, if necessary, any required automation software, no other applications are started. The hardware utilisation and the electrical power consumption are then determined in accordance with the measurement method described in Appendix A. One of the systems stated in Appendix D must be used as the reference system.

3 Information on measuring the hardware utilisation in idle mode (criterion 3.1.1.2)

3.1 Measurement concept: Hardware utilisation in idle mode

The hardware utilisation when the software product is in idle mode must be stated. For this purpose, the software is installed on a reference system and the hardware utilisation in idle mode is determined using the measurement method described in Appendix A. One of the systems stated in Appendix D must be used as the reference system.

The system used must be stated in the spreadsheet file for recording the criteria (Annex 2). The hardware utilisation in idle mode includes the additional load caused by activating the software product, as a percentage share of the base load. The average electrical power consumption (in watts) is calculated by taking the average of all measurements of the power consumption in idle mode.

3.2 Process for measuring the hardware utilisation in idle mode

The software product being measured is installed on a reference system. The selected reference system, the software stack used and any configuration (of the software product and software stack) that deviates from the delivered state must be documented (see compliance verification for Paragraph 3.1.1.2).

The software product is then started and the measurement of the hardware load of the SUT carried out. The measurements must be repeated at least 10 times, 30 times is recommended (see Appendix A, Section 3), the standard usage scenario must take at least 5 minutes to run (see Appendix D, Section 2), 1 measurement value is recorded per second, it is recommended that you add a waiting time of 1 minute to the process until the system is fully loaded.

The software product is active but there is no user interaction.

Once all of the values have been recorded, the average (idle load) is determined using the formula:

$$LA_i = \frac{1}{n} \sum_{k=1}^n \bar{x}_k$$

X_k is the arithmetic mean for measurement cycle k in % (CPU utilisation), MByte (RAM utilisation and hard disk load) or MBit/s (network).

The net idle load, idle allocation factor and effective load idle is then calculated. The supplied spreadsheet for recording the criteria can be used for this purpose.

4 Information on measuring the hardware utilisation and energy demand when running a standard usage scenario (criterion 3.1.1.3)

4.1 Measurement concept: Hardware utilisation and energy demand when running a standard usage scenario

All of the software products that are required to operate the software product are installed (e.g. runtime environment, databases, etc.). The actual software product is then installed.

The standard usage scenario (see Appendix D) is then run automatically and the hardware load of the SUT is measured during the scenario. The measurements must be repeated at least 10 times, 30 times is recommended (see Appendix A, Section 3), the standard usage scenario must take at least 5 minutes to run (see Appendix D, Section 2), 1 measurement value is recorded per second, it is recommended that you add a waiting time of 1 minute to the process until the system is fully loaded.

Once all of the values have been recorded, the average (gross load) is determined using the formula:
 \bar{X}_k is the arithmetic mean for measurement cycle k in % (CPU utilisation), MByte (RAM

$$BA_i = \frac{1}{n} \sum_{k=1}^n \bar{x}_k$$

utilisation and hard disk load) or MBit/s (network).

The net load, allocation factor, effective load and hardware utilisation as an integral of the hardware load are then calculated over the time taken for the scenario. The measurement units for the hardware utilisation are units for work performance (e.g. %*s, MB*s*s=MB, MBit/s*s=MBit). The supplied spreadsheet for recording the criteria can be used to calculate them.

The verifications must be accompanied by information on the system used for the measurements, information on the standard usage scenario (which function of the software were executed?) and information on the configuration of the software product and the software stack – if these differ from the delivered state of the software. One of the reference systems stated in Appendix D must be used.

4.2 Process for measuring the energy demand

The process for installing the software product and measuring the energy demand is the same as the verification process for the hardware utilisation (see Section 2.1 of Appendix B).

The measurement of the energy consumption of the SUT is then started, while the standard usage scenario is run automatically.

After the measurement has been completed, the average energy consumption is calculated using the formula

$$E = \frac{1}{3600 \cdot n} \sum_{k=1}^n \sum_{i=1}^m P_i$$

and recorded as the sum per second of the measured power consumption P_i of the SUT. The electrical work in Wh is derived by dividing this figure by 3600 s/h. If the measurement is not aggregated every second, it is necessary to derive the integral over the measurement period t and calculate the electrical work in Wh using suitable means.

Appendix C Data format for passing on the product information about resource and energy efficiency

```
<xs:schema      attributeFormDefault="unqualified"
                elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Blue-Angel-Software-Criteria">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Product">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="ProductName"/>
              <xs:element type="xs:string" name="ProductVersion"/>
              <xs:element type="xs:string" name="ProductManufacturer"/>
              <xs:element type="xs:string" name="SoftwareArchitecture"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="MeasurementPersonnelData">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="MeasurementDate"/>
              <xs:element name="MeasurementSender">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="MeasurementSenderName"/>
                    <xs:element type="xs:string" name="MeasurementSenderMail"/>
                    <xs:element type="xs:string" name="MeasurementSenderOther"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="MeasurementSenderInstitute">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="InstituteName"/>
                    <xs:element type="xs:string" name="InstituteAddress"/>
                    <xs:element type="xs:string" name="InstituteWebsite"/>
                    <xs:element type="xs:string" name="InstituteOther"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="MeasurementNote"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="MeasurementTechnicalData">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element type="xs:string" name="MeasurementDevice"/>
    <xs:element type="xs:string" name="SamplingFrequency"/>
    <xs:element type="xs:string" name="ScenarioLength"/>
    <xs:element type="xs:string" name="SamplingSize"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="MeasurementSUTDetails">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string"
name="MeasurementReferenceSystemYear"/>
      <xs:element type="xs:string" name="MeasurementSUTSystem"/>
      <xs:element type="xs:string" name="MeasurementSUTModel"/>
      <xs:element type="xs:string" name="MeasurementSUTProcessor"/>
      <xs:element type="xs:string" name="MeasurementSUTCores"/>
      <xs:element type="xs:string" name="MeasurementSUTClockSpeed"/>
      <xs:element type="xs:string" name="MeasurementSUTRAM"/>
      <xs:element type="xs:string" name="MeasurementSUTHardDisk"/>
      <xs:element type="xs:string" name="MeasurementSUTGraphicsCard"/>
      <xs:element type="xs:string" name="MeasurementSUTNetwork"/>
      <xs:element type="xs:string" name="MeasurementSUTCache"/>
      <xs:element type="xs:string" name="MeasurementSUTMainboard"/>
      <xs:element type="xs:string" name="MeasurementSUTOperating
System"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
  <xs:element name="BackwardCompatibilitySUTDetails">
    <xs:complexType>
      <xs:sequence>
<xs:element type="xs:string" name="BackwardReferenceSystemYear"/>
        <xs:element type="xs:string" name="BackwardSUTSystem"/>
        <xs:element type="xs:string" name="BackwardSUTModel"/>
        <xs:element type="xs:string" name="BackwardSUTProcessor"/>
        <xs:element type="xs:string" name="BackwardSUTCores"/>
        <xs:element type="xs:string" name="BackwardSUTClock Speed"/>
        <xs:element type="xs:string" name="BackwardSUTRAM"/>
        <xs:element type="xs:string" name="BackwardSUThard Disk"/>
        <xs:element type="xs:string" name="BackwardSUTGraphicsCard"/>
        <xs:element type="xs:string" name="BackwardSUTNetwork"/>
        <xs:element type="xs:string" name="BackwardSUTCache"/>
        <xs:element type="xs:string" name="BackwardSUTMainboard"/>
        <xs:element type="xs:string" name="BackwardSUTOperatingSystem"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

</xs:element>

    <xs:element name="SoftwareStack">
    <xs:complexType>
    <xs:sequence>
    <xs:element type="xs:string" name="SoftwareStackData"/>
    </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ResourceEfficiency">
<xs:complexType>
<xs:sequence>
<xs:element name="HardwareEfficiency">
<xs:complexType>
<xs:sequence>
<xs:element name="MinimumSystemRequirements">
<xs:complexType>
<xs:sequence>
<xs:element name="Indicator" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="IndicatorID"/>
<xs:element type="xs:string" name="IndicatorName"/>
<xs:element type="xs:string" name="IndicatorNote"/>
<xs:element type="xs:string" name="IndicatorResult"/>
<xs:element type="xs:string" name="IndicatorUnit"/>
<xs:element type="xs:string" name="IndicatorNote"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="HWLoadIdle">
<xs:complexType>
<xs:sequence>
<xs:element name="Indicator" maxOccurs="unbounded" minOccurs="0">
<xs:complexType>
<xs:sequence>
<xs:element type="xs:string" name="IndicatorID"/>
<xs:element type="xs:string" name="IndicatorName"/>
<xs:element type="xs:string" name="IndicatorNote"/>
<xs:element type="xs:string" name="IndicatorResult"/>
<xs:element type="xs:string" name="IndicatorUnit"/>
<xs:element type="xs:string" name="IndicatorNote"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="HWUtilisationEnergyDemand">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Indicator" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="IndicatorID"/>
              <xs:element type="xs:string" name="IndicatorName"/>
              <xs:element type="xs:string" name="IndicatorNote"/>
              <xs:element type="xs:string" name="IndicatorResult"/>
              <xs:element type="xs:string" name="IndicatorUnit"/>
              <xs:element type="xs:string" name="IndicatorNote"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="HWOperatingLife">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BackwardCompatibility">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="BackwardCompatibilityYear">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Indicator" maxOccurs="unbounded" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="IndicatorID"/>
                          <xs:element type="xs:string" name="IndicatorName"/>
                          <xs:element type="xs:string" name="IndicatorNote"/>
                          <xs:element type="xs:string" name="IndicatorResult"/>
                          <xs:element type="xs:string" name="IndicatorUnit"/>
                          <xs:element type="xs:string" name="IndicatorNote"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Appendix D Reference system and standard usage scenario

[Extract from the: Final report for the project "Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik" (Development and application of assessment criteria for resource efficient software taking into account existing methodology) (UFOPLAN-SSD-2015)]

1 Reference system

To verify the resource and energy efficiency (criteria 3.1.1.2 and 3.1.1.3), the applicant must use one of the reference systems stipulated below. The relevant measurements (see Appendix A) at the time of application and during the term of the contract (e.g. if the product has been changed, see criterion 3.2.1) must be carried out on the selected reference system. The reference system used must be documented in the spreadsheet file for recording the criteria as the "Measurement system".

In addition, the table of reference systems should be used to determine the backward compatibility of the software (criterion 3.1.2.1), i.e. by stating the calendar year for a reference system on which the software product can still be run. If you want to use older computer systems (2010 - 2014) as the reference system, refer to the computer systems in Tables 16 and 17 of the final report for the project "Entwicklung und Anwendung von Bewertungsgrundlagen für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik" (Development and application of assessment criteria for resource efficient software taking into account existing methodology) (UFOPLAN-SSD-2015). The reference system used for verifying the backward compatibility of the software must be documented in the spreadsheet file for recording the criteria as the "Backward compatibility system". The measurement system and backward compatibility system can be identical.

1.1 Description of the reference system

The computers for the reference system "workplace computer for office software" are professional desktop computers primarily in mini tower casings. As the devices are selected based on their energy efficiency, the efficiency of the network components used in these computers is very high and the energy consumption of the computers is comparatively low.

Description of the reference system for Windows/Linux as a time series 2015 - 2019

Technical parameter	2015	2016	2017	2018	2019
Manufacturer	Fujitsu	Fujitsu	Fujitsu	Fujitsu	Fujitsu
Model	Esprimo P920 ¹	Esprimo P956 ²	Esprimo P957 ³	Esprimo P957 ⁴	Esprimo P958 ⁵
Processor	Intel i5-4590	Intel i5-6500	Intel i5-6500	Intel i5-7500	Intel i5-8500
Cores	4	4	4	4	6
Clock speed	3.3 GHz	3.2 GHz	3.2 GHz	3.4 GHz	3.0 GHz

¹ Data sheet: <https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-ESPRIMO-P920-0Watt-de.pdf>

² Data sheet: <https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-ESPRIMO-P956-E90-de.pdf>

³ Data sheet: <https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-ESPRIMO-P957-E90-de.pdf>

⁴ Data sheet: <https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-ESPRIMO-P957-E85-de.pdf>

⁵ Data sheet: <https://sp.ts.fujitsu.com/dmsp/Publications/public/ds-ESPRIMO-P958.pdf>

RAM	4 GB (DDR3, 1600 MHz)	4 GB (DDR4, 2133 MHz)	8 GB (DDR4, 2133 MHz)	16 GB (DDR4, 2400 MHz)	16 GB (DDR4, 2666 MHz)
Hard disk (HDD / SSD)	HDD SATA III 500 GB (6 Gbit/s)	HDD SATA III 500 GB (6 Gbit/s)	SSD SATA III 128 GB (6 Gbit/s)	SSD M2 SATA 256 GB (6 Gbit/s)	SSD M2 SATA 512 GB (6 Gbit/s)
Graphics card	Intel HD Graphics 4600	Intel HD Graphics 530	Intel HD Graphics 530	Intel HD Graphics 630	Intel UHD Graphics 630
Network	LAN GigaBit	LAN GigaBit	LAN GigaBit	LAN GigaBit	LAN GigaBit

1.1.1 Description of the reference system for Mac OS 2015 and 2019

Technical parameter	2015	2019
Manufacturer	Apple	Apple
Model	Mac mini "Core i5" 2.6 (Late 2014) ⁶	Mac mini "Core i5" 3.0 (Late 2018) ⁷
Processor	Intel i5-4278U	Intel i5-8500B
Cores	2	6
Clock speed	2.6 GHz	3.0 GHz
RAM	8 GB (LPDDR3 SDRAM, 1600 MHz)	8 GB (PC4-21300 DDR4, 2666 MHz)
Hard disk (HDD / SSD)	1 TB HDD	256 GB SSD
Graphics card	Intel Iris 5100	Intel UHD Graphics 630
Network	LAN GigaBit	LAN GigaBit

⁶ https://everymac.com/systems/apple/mac_mini/specs/mac-mini-core-i5-2.6-late-2014-specs.html

⁷ https://everymac.com/systems/apple/mac_mini/specs/mac-mini-core-i5-3.0-late-2018-specs.html

2 Examples for standard usage scenarios

The main basis for measuring some indicators is a standard usage scenario, which must firstly be defined for every software product group being tested. It simulates the realistic use of the software. The standard usage scenario includes the completion of tasks for which the software was developed and, if relevant, the interaction with the user of the software.

2.1 Procedure for developing a standard usage scenario

1. Develop standard usage scenarios that reflect typical functions of the software product. The frequency at which these functions are used should be taken into account.
2. Describe the standard usage scenario in the spreadsheet for recording the criteria (Annex 2).

The duration of the standard usage scenario can be freely selected. However, the scenario must take at least 5 minutes to run.

2.2 Overview of the standard usage scenarios for selected software products

The following overview gives examples of standard usage scenarios in a simplified form:

Word processing	Media player
Edit entire text	Start video
Add & update table of contents	Fast forward and rewind the video
Adjust view	Pause video
Add & edit content	Change playback speed
Create PDF	Resume video
Save	Close program

Other examples for standard usage scenarios are enclosed with the application documentation. The standard usage scenario "word processing" is described in detail below. This is consistent with the description in the spreadsheet file for recording the criteria:

Software: Product name
Product group: XX
Version: Date
Designation: XX
Authors: Name(s)
Reference system: Year
Load generator: Designation

General specifications:

Wait for a period of 60 seconds before starting the measurement.

Requirements:

The files created by the automated process do not exist

Zoom the document to 125% in the program
 Rulers are switched on
 Control characters are switched on
 Copy the text to be added to the clipboard
 Font "XX" is installed

Process:

No.	Time	Description of action	Name of action
1	0:00	Open a link to a document on the desktop (double click)	start
2	00:02	Wait until the document has opened	
3	00:08	Change the entire text	ChangeText
3.1	00:08	Click in the text	
3.2	00:09	Select all: Ctrl+a	
3.3	00:11	Change font to "XX"	
3.4	00:17	Change the format to justified text	
3.5	00:19	Set the font size to 9	
3.6	00:26	Click in the text to cancel the selection	
4	00:28	Save as a new document	Save
5	00:35	Add a table of contents	AddContents
...

After end of the measurement:

Delete both of the generated files to ensure the consistency of the process.