Department of Informatics, University of Zürich

MSc Thesis

Energy Efficiency of Web Applications

Robert Dewor

Matrikelnummer: 08-715-922

January 30, 2014

supervised by Prof. Dr. Lorenz M. Hilty and Dr. Wolfgang Lohmann

University of Zurich[UZH]

Department of Informatics

Abstract

The increase of web applications and web services is a contributing factor to the yearly increase of emissions caused by the ICT Industry.

The goal of this master thesis is to develop an evaluation method for the energy efficiency of web applications by identifying a collection of criteria for their energy efficiency. Those criteria then provide information on what areas the web application could be improved in order to be more energy efficient. Based on those criteria an evaluation model is created. This model will then be applied to an example web application. After that problems and challenges that arose during the development of this thesis are discussed.

Zusammenfassung

Die zunehmende Anzahl von Webapplikationen und Webservices trägt erheblich zur jährlichen Emissionszuname, verursacht durch die IKT-Branche, bei.

Das Ziel dieser Masterarbeit ist die Entwicklung einer Bewertungsmethode für die Energieeffizienz von Webapplikationen durch die Identifikation einer Anzahl von Kriterien für die Energieeffizienz. Diese Kriterien stellen Informationen zur Verfügung, an welchen Stellen die Webapplikation verbessert werden könnte, um energieffizienter zu arbeiten. Basierend auf diesen Kriterien wird ein Bewertungsmodell erstellt. Dieses Modell wird auf eine Beispiel-Webapplikation angewendet. Danach werden die Herausforderungen und Probleme, die während der Entwicklung dieser Masterarbeit aufgetreten sind, diskutiert.

Contents

List of Figures

List of Tables

1 Introduction

The total amount of internet capable devices is constantly increasing. Increasing connectivity has become important to our daily life. The internet offers more and more content and websites and web applications are becoming more advanced. This development comes at a price. The cost of operating data centres, the increased bandwidth and the future need for faster infrastructure and hardware as well as the cost of computation cause increased cost of energy. There are at least three contributing factors for emissions caused by the ICT Industry: data-centres, end-user devices and network infrastructure. Emissions caused by network infrastructure are estimated to increase by 4.7 percent and end-user devices by 2.3 percent. Due to the increasing demand for data storage, emissions caused by data centres are estimated to increase by 7.1 percent by 2020, more than the increase for end-user devices and network infrastructure together. While the switch to more mobile end-user devices as well as more efficient hardware has decreased the emission growth rate for end-user devices, the overall emission growth rate is still 3.8 percent per year until 2020 [10, p.21-22]. The increase of web applications and web services is a contributing factor to the yearly increase of emissions. The energy consumed by data-centres all over the world is significant. In *The Energy Efficiency Potential of Cloud-Based Software: A U.S. Case Study* Masanet et al. explore the energy consumption of data-centres. Currently all data-centres account for 1-2% of the global usage of electricity. The study indicates that there are roughly around 4.7 million servers in data-centres all over the world [34]. By finding ways to optimize the energy efficiency of web applications, the overall energy consumption and emissions caused by web applications and their supporting infrastructure could be lowered, lowering the overall carbon footprint of the ICT industry.

The aim of this master thesis is to propose a collection of criteria in order to determine whether a web application is energy efficient or not. Those criteria then provide information on what areas the web application could be improved in order to be more energy efficient. In order to do that, it must be explored as to how to measure them. Identifying the right criteria as well as measuring them is a difficult task. Not every criteria can be measured directly.

This thesis will discuss the existing approaches and challenges when measuring certain aspects related to the energy consumption of web applications. In addition to that, I will identify relevant criteria that determine the energy efficiency of a web application. By discussing the problems and challenges associated with evaluating each criterion, this thesis provides a first step in creating a method to evaluate the energy efficiency of a web application. Based on this evaluation, one will be able to compare different web applications and their energy efficiency. It displays the efforts and decisions made in order to reduce the energy consumption of a web application. Those efforts determine how green a web application is. A web application will be considered to be green if all criteria are met.

In a first step the current state of the art in related scientific fields will be evaluated. The purpose of this research is to determine how necessary information can be retrieved, analysed and translated into energy consumption. In order to do this, it is necessary to create a model that incorporates all cooperating structural units involved in the process of using a web application. Based on this model, one will be able to determine the energy efficiency of a web application.

In a second step, the evaluation model will be applied to a practical example. Then a web application will be created that enables the user to evaluate the distance between client- and server-side. In order to acquire the necessary information from the web application, various existing API services will be researched.

This thesis is structured as follows:

In Chapter 2, background information based on related work will be given. In Chapter 3, the selected criteria for the evaluation of web applications will be discussed in more depth and the evaluation model will be displayed. Chapter 4 will present the application of the evaluation model to a practical example and the demo web application that has been developed. In Chapter 5, the results and problems of this master thesis will be discussed. Chapter 6 will give a conclusion. In Chapter 7, an outlook on future work will be displayed.

2 Background Information

This chapter will give relevant background information about the topic of this thesis. First, important terminology will be defined. Then necessary background information will be provided by discussing the papers most relevant for the approach presented in this thesis.

The topic at hand has been researched extensively. Nevertheless, most papers focus on single factors in order to evaluate the energy consumption of web applications. Since the aim of this thesis is to provide a full and detailed analysis of every relevant factor, various papers researching different factors will be presented.

2.1 Terminology

In table 2.1, basic terms used throughout this thesis will be defined.

Term	Definition	Source
Web Application	A web application is a software system based on technologies and standards of the World Wide Web Consortium (W3C) that provides Web specific resources such as content and services through a user interface, the Web Browser	[11, p. 2]
LCA	Life cycle assessment (LCA) involves the evaluation of some aspects - often the environmental aspects - of a product system through all stages of its life cycle. Sometimes also called “life cycle analysis”, “life cycle approach”, “cradle to grave analysis” or “Ecobalance”, it represents a rapidly emerging family of tools and techniques designed to help in environmental management and, in the longer term, in sustainable development.	[16, p. 9]
Green IT	Optimal use of information and communication technology (ICT) for managing the environmental sustainability of enterprise operations and the supply chain, as well as that of its products, services and resources, throughout their life cycles.	[14]
CDN	A Content Delivery Network represents a group of geographically dispersed servers deployed to facilitate the distribution of information generated by Web publishers in a timely and efficient manner.	[15, p. 15]
Resource	A Medium, which can be used in a process. A resource may be tangible or intangible.	[12]
Resource Efficiency	Ratio between a certain benefit or result in relation to the necessary input of resources.	[12]

Table 2.1: Terminology

2.2 Scope

The scope of this thesis is researching the energy efficiency of web applications. The term energy efficiency will be explained in more detail later on. Defining factors which influence the energy efficiency of a web application can potentially help improve the energy consumption and allows the comparison of different web applications. Deciding whether a web application is more or less energy efficient than another is not a trivial task. Using a web application consumes energy on various structural units. In this context a structural unit is considered as an organisational unit that is part of the architecture explained in Figure 2.1. The main three structural units in this thesis are: server-side, data-transfer and client-side. In order to determine the energy efficiency of a web application, all structural units have to be evaluated individually. The total energy efficiency of a web application is an aggregation based on the

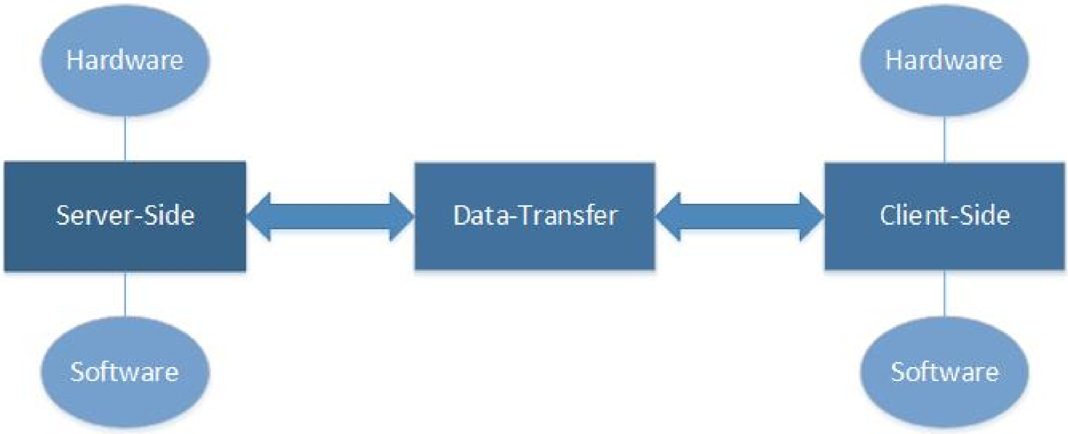

Figure 2.1: Overview

energy efficiency of each structural unit. The following will present some of these approaches as a basis for future discussion.

2.3 Clarification of the Term Energy Efficiency

In the paper *Impact of Office Productivity Cloud Computing on Energy Consumption and Greenhouse gas Emissions*, Williams et al. determine the energy consumption of office applications that are run in a cloud service. In order to measure the total cloud service energy consumption, data was collected at three stages: the data center, the network transfer and the end user device.

At the data center stage, the energy consumption was apportioned to a single Office 365 and Outlook Exchange 2010 session. The result of that stage was a single device energy consumption per session [7].

Since those sessions cannot be clearly identified without access to the operational servers, the energy consumption related to each session was estimated based on the average session value over all Office 365 / Outlook Exchange 2010 sessions [7].

The energy consumed by the network was accounted for by estimating the average number of hops taken from the end-user device to the Office 365 data centre and the total size of data that was transmitted with each activity [7].

The energy consumption of the end user device was estimated for each activity. By segmenting the software product and the energy consumption of the operating system, the energy consumption for the cloud activity was appropriated as a portion of the devices power consumption. The power consumption of the device itself was measured with a Computer Oscilloscope [7].

The results of this paper indicate that cloud computing was more energy efficient for software services with high levels of processing and storage or those with constant uptime but low data transfer sizes [7].

The term *energy efficiency* in the context of this thesis is based on the coefficient between the useful energy output of a system S_{Out} and the energy input of that system E_{In} [9]:

$$E_{ConversionEfficiency} = E_{Out} / E_{In}$$

In the case of web applications the useful output will not be energy but any number of services and functionalities provided by the application to the user (S_{Out}). Those services could be creating a new auction on ebay or performing a search on Google. The energy input is the energy consumed by the execution of such a service.

Therefore, in this context, the ***energy efficiency of a web application*** will be defined as [9]:

$$Webapp_{EnergyEfficiency} = S_{Out} / E_{In}$$

Determining the usefulness of S_{Out} is difficult and depends on the type of service provided. It is therefore hard to evaluate on its own. It is however possible to estimate a comparison between two web applications providing similar services in order to determine which one offers a better energy efficiency. Web applications providing similar services could be Ebay, Amazon and Ricardo. Those web applications all provide a service that allow users to auction items to other users.

2.4 General Approaches for the Evaluation of Web Media

In the paper *Approaches to the Dynamic Energy Footprinting of Online Media*, Schien et al. present approaches to a dynamic energy footprinting for digital media focusing on the delivery of a web application to an end-user. The particular structural units that are discussed are the data-centre, the server for the web application, the connecting network and the end-user device [2].

In this paper web applications are considered to be hosted as multi-tiered applications. In order to calculate the energy cost, power monitoring of individual devices and performance monitoring of process activity on those devices will be combined [2].

In order to determine the energy used for data-transfer in the network, the parts of the network used for a given request have to be identified. The energy will then be allocated proportionally according to what percentage of traffic the specific services request created [2].

Due to the large variations in power consumption for different end-user devices like mobile phones or desktop computers, the end-user is required to provide information about the device. Furthermore it is assumed that a user can only lend his attention to a maximum of two separate applications, one serving audible while the other serves visual content. The power consumption of the end-user device is the power consumption of the web browser over the time of reading the website plus the power consumption of the home network during the time of the visit [2].

In the paper *Systems Modelling of New Media Services*, Yearworth et al. offer a different approach to the energy consumption of web media. They estimate the dynamics of energy use for the delivery of media services based on three factors: the overall energy efficiencies deliverable by the ICT supply chain into the data centre, the overall media richness of content and how that is effected by investment and syndication and growth in the user base. As a result of this theory, the authors claim that if a web application has a exponentially growing user base as well as exponentially growing data volume, the increasing energy consumption cannot be countered by mere efficiency improvements. This paper therefore strongly suggests that the actual published content, especially the data volume of that content that is being transferred to the client, offers various opportunities in order to lower the energy consumption of the web application [3].

In the paper *The GREENSOFT Model: A reference model for green and sustainable software and its engineering*, Naumann et al. present the GREENSOFT model. It was created as a reference model in order to structure concepts, strategies, activities and process of Green IT. It displays the importance of including all relevant components and services involved in the life-cycle of a software product as displayed in Figure 2.1.

In the following, the three main structural units, server-side, data-transfer and client-side will be discussed in more detail.

2.5 Server-Side

On the server-side, the most important elements are the data-centre running the server that hosts the web application as well as the development life-cycle of that application (Figure 2.3). When evaluating the software aspects of a web application, one also has to consider the initial and ongoing development of that application. In the paper *LCA for Green System Design of Digital Media*, Schien et al. discuss the challenges of applying the life cycle assessment as an assessment tool in order to improve the energy efficiency of digital news media products. First of all, the different phases of a product life cycle are not clear when developing software. Also, without the right level of granularity when observing resource flows, it is impossible to determine which process caused which amount of energy consumption. Due to the high complexity of Green IT systems changes applied to a web application can have a huge impact on the energy consumption of the application. Increasing or reducing the size of data published via the web application as well as making minor changes to the routing can increase or decrease the energy consumption substantially.

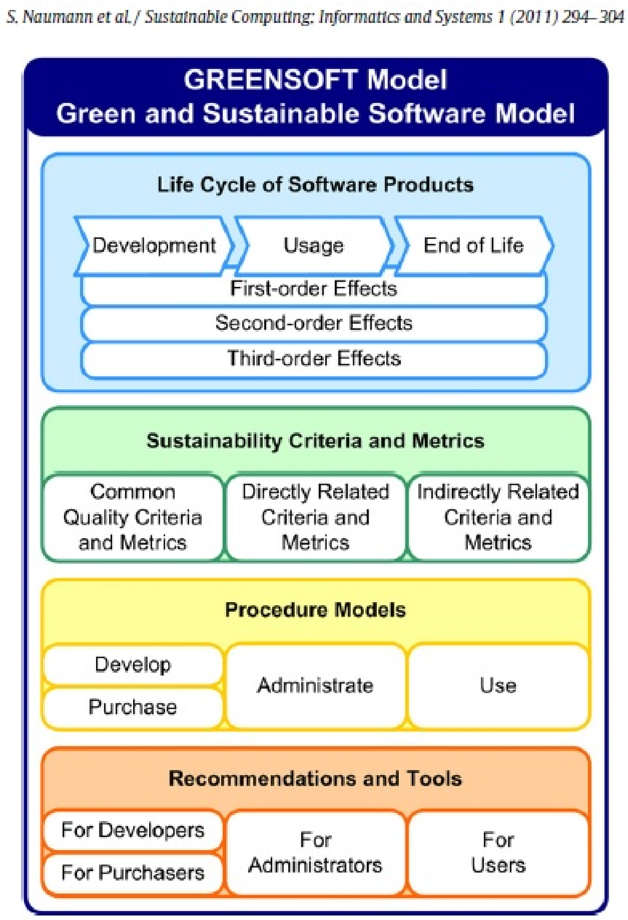

Figure 2.2: [GREENSOFT Model - 4, Figure 1]

This complexity also makes it hard to predict impact results of adjustments to the web application [5].

In the final report *Grüne Software*, Hilty et al. discuss various opportunities for lowering the energy consumption created in data-centres. While lowering the energy consumption of data-centres with the help of more efficient hardware solutions is already applied in practice, improving software solutions in order to increase energy efficiency has been mostly ignored in research. In this report, the author proposes three different ways of improving energy efficiency: Dynamic predictive load management, information and data management and data compression and de-duplication [1, p.29-33].

Due to the service level agreements of most hosting providers that require them to deliver a very high uptime, a lot of data centres are run inefficiently. Because of that, the servers are

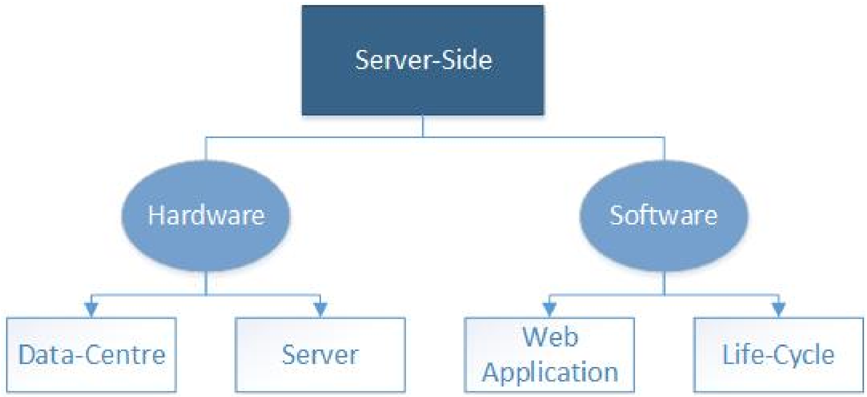

Figure 2.3: Server-Side

almost always up and running, even when there is nothing to compute. Dynamic predictive load management could help control the IT infrastructure in a way that saves energy by shutting down devices that are not needed. A lot of providers for cloud computing, for example, the Microsoft Azure service have started adopting similar mechanics to balance out different physical servers with each other based on the workload [1, p. 29-33].

Due to increasingly cheap data storage units there is a lack of incentive for handling data efficiently. Therefore the volume of data is constantly increasing and because of that the energy consumption for handling data because there is no incentive for handling data efficiently. Removing redundant or duplicate data as well as storing data on the most energy efficient data storage could help improve the overall energy efficiency of a data centre [1, p. 29-33].

Decreasing the size of data due to data compression lowers the amount of data being transferred as well as overall data size in the data storage, therefore reducing energy consumption. There are various methods in order to compress data. De-duplicating data identifies redundant data and eliminates duplicates [1, p. 29-33].

2.6 Data-Transfer

The communication between the web application and the end-user device plays an important role for energy consumption. In the scope of this thesis, the three main factors that will be considered concerning the data transfer between the client and the server are the size of the transmitted data, the number of packages as well as the routing of that data (Figure 2.4). In the paper *Impact of Location on the Energy Footprint of Digital Media*, Schien et al. assess the

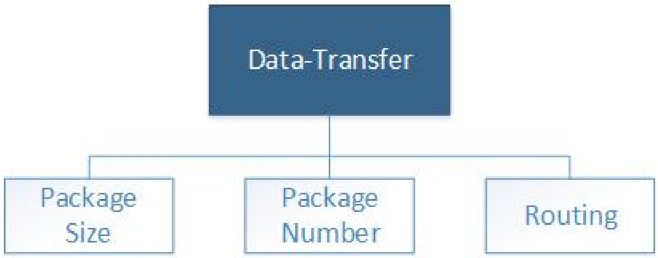

Figure 2.4: Data-Transfer

energy consumption for data transmission between two specific geographic locations. According to Schien et al., the energy consumption for routing data greatly depends on the volume of the data that is being transferred as well as the geographic location of user and server. He estimates the energy consumption of the network to be similar to the energy consumption of the device used to access the data. Depending on whether the user device is a desktop computer or a smart phone, the energy consumption for the data transmission is therefore higher or lower. Due to the introduction of CDN servers, the route length and resulting energy consumption by the network is smaller. In conclusion, the routing and geographic location of the end-user has little effect on the energy consumption when the CDN servers utilized are effectively deployed [6].

Since mobile devices are getting increasingly popular, it is important to distinguish between the internet network and the cellular network. Currently, the energy intensity for transmitting the data of a cellular network is significantly higher than utilizing wireless LAN or normal internet networks. Cisco estimates, that more than half of the global IP-traffic will be based on mobile devices utilizing cellular networks. Due to ever increasing data volume for mobile devices with increasing capabilities, the energy efficiency of mobile devices accessing web applications is an important factor. This is in context to how much data should be sent to the client and which computations should be performed server-side and which should be performed client-side in order to optimize the energy efficiency [1, p. 12-15].

In *The Direct Energy Demand of Internet Data Flows*, Coroama et al. estimate the direct energy demand of transmitting data over the internet to be 0.2 kWh/GB for a wired internet connection. When considering connections that are operating near maximum capacity, the author claims that the energy consumption could potentially reach up to 8.97 kW. This might only arise under unlikely assumptions, nevertheless, it seems very important to consider high traffic when evaluating web applications. Web Applications provide services that depending on the application, might be used a lot during peak times e.g. after work hours are over. Therefore, the load of the network is an important factor to consider when researching the energy consumption of web applications. Based on this result, transmitting an e-book with the size of 1 MB would cost no more than 0.2 Wh of energy on average (the amount of energy needed to light a 60W bulb for twelve seconds). This is considered to be a reasonable value.

This value changes significantly when considering mobile networks. In *Power Trends in Communication Networks* Kilper et al. estimate a projection for the power consumption of mobile networks through 2020. According to the projections in this paper, the power consumption per user for mobile networks will increase. Even though there are ongoing improvements in efficiency, the growth rate for mobile traffic increases the energy consumption more than the efficiency improvements can reduce it [20]. It is therefore believed that finding criteria that help measure and optimize data traffic between client devices and web applications can help in the ongoing efforts to reduce the power consumption per user for mobile networks.

In *Effects of Internet-based multiple-site conferences on greenhouse gas emissions* Coroama et al. show how the use of ICT technology can reduce greenhouse gas emissions. According to their experiment, greenhouse gas emissions could be potentially reduced by up to 50% by utilizing ICT technology. The idea is simple: by replacing work related travel with video conference technology, the greenhouse gas emissions can be reduced based on the situation [18]. Web applications can be utilized in order to increase energy efficiency as well. This depends on the type of web application and it should be considered when evaluating the energy efficiency of a web application.

The overall increased demand for internet usage increases the energy consumption for transferring data. In the master thesis *Measuring and Optimizing Energy Efficiency in Internet Communication*, Gitanjali Sachdeva researches ways and mechanisms to measure the energy consumption of networks in order to optimize them. The author proposes a packet-level energy accounting model collecting energy-related information in IP packet headers. This information can then be related to the energy consumption for each packet for each hop on the route between the client device and the server [17]. Creating a link between the the energy consumed and the part of the architecture that is consuming it is one of the most challenging aspects researched in this thesis. It is therefore believed that the concept presented in this thesis has a lot of potential. Nevertheless, the practical implementation has not been tested for bigger networks topologies, which is what web applications are all about.

2.7 Client-Side

On client-side, there are four elements to consider: the kind of device that is used to access the web application, the life-cycle of the device, the browser and the processes related to running the web application on the device (Figure 2.5). Evaluating the resources allocated to a soft-

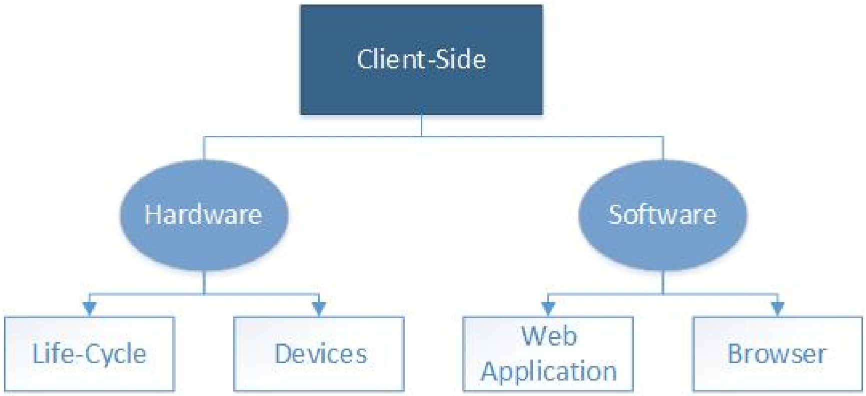

Figure 2.5: Client-Side

ware process is a difficult task. Every piece of software is part of a complex system consisting of multiple components, each consuming energy. Existing key performance indicators usually do not allow isolating the influence of software. In order to evaluate the energy consumption of a web application on the client-side, one has to determine the energy consumption of the browser as well as the computational tasks performed by displaying the web application. Existing tools, for example powertop for Linux, allow the allocation of CPU usage to a software process in order to do so. In addition to that, the share of the idle energy consumption of the device has to be considered [1, p. 37].

In the paper *Profiling energy consumption of smart phone users for environmentally efficient business decisions*, Bernstein et al. link the user behaviour of mobile devices and energy consumption through various types of ICT infrastructure. As a result, four main user groups can be defined based on user behaviour: users that use low amounts of 3G data as they approach their limit, users that use heavy amounts of data while exhibiting limited overage cost sensitivity, users that use low amounts of data even though their subscription would allow for higher data usage and users who use heavy amounts of data closely fitting their plan.

Based on these four user groups, Bernstein et al. propose different business scenarios where the internet service providers create incentives for users to shift their behaviours towards lowering the over energy consumptions of their mobile device [8]. Knowing the user base of a web application can provide the host of web applications valuable information in order to increase the energy efficiency of the application.

3 Approach: Evaluation of a Web Application

In this chapter, an approach for the evaluation of the energy efficiency of web applications will be presented. First, the structural unit server-side will be explained, then the structural unit data-transfer, followed by the structural unit client-side. After that, the total energy consumption when executing a web application will be shown. The focus of this approach is to identify the elements with the greatest impact on the energy consumption in total that can be influenced by the host of the web application. The energy efficiency of those elements will then be measured.

In this chapter the following notation will be used:

X : the element X refers to a specific term or object, for example the workload of a web application, or the amount of database calls executed by the web application

E_X : the energy efficiency associated with element X
C_X : the energy cost associated with element X
N_X : the number of elements
W_X : the element X belonging to the web application

For this approach three main stakeholders will be considered. The host of the data-centre will be the stakeholder that provides the server a web application is hosted on and the host of the web application will be the stakeholder that develops and runs the web application. The user base will be the stakeholder that accesses the functionality of a web application.

3.1 Example

While some of the criteria presented in this approach can be evaluated directly, others have to be compared to benchmark values in order to create a meaningful statement. While the example web application is a real web application, not all information was available. Some informations for this example have been researched on the internet. In order to visualize the approach, the following example web application will be utilized. First the context of the example application will be described. Then the offered functionality will be displayed. After this, the information relevant for the evaluation of the example web application will be shown and called parameters.

Context: The purpose of the web application is time management. It can be utilized by users in order to record and manage their time at work, their holidays or absence. The web application is based on the MVC 4 framework and is hosted on a data-centre based on cloud computing. In order to measure the energy efficiency of the server hardware the SPECpower_ssj2008 Benchmark was used. The Benchmark was developed by the Standard Performance Evaluation Corporation (SPEC) with the goal of creating a comprehensive standard for energy efficiency of server Hardware [26][27]. While there are other metrics like the Green500 List which takes the theoretical aggregate floating-point operations per second and divides it by the measured peak power consumption, the SPECpower_ssj2008 benchmark was chosen because it considers the fact that servers are not always operating under full load [28]. The SPECpower_ssj2008 benchmark calculates the server side Java operations per watt. This benchmark is considered to be a close enough estimation that enables one to compare the energy efficiency of server hardware.

Functionality: The example application offers a wide range of functionality to the user. It allows a user to create and manage a company. More users can be added to that company as employees. Employees can create, edit and delete time, holiday and absence entries with the help of an interface similar to the outlook calendar. In addition, employees can create pdf reports for a chosen time frame. The company can calculate bills based on the reported time entries of the employees of the company.

Parameters: The important parameters of the example web application are:

- *Development-team*: The current development team consists of five people developing the application for one year

- *User base*: The current user base is rather small and consists of 100 people utilizing the web application each month
- *Server-Location*: The data-centre is located in Ireland
- *Data-centre monthly energy consumption*: 765 kWh per server and user
- *Power Source*: The data-centre is not powered by any renewable energy
- *Amount of Servers*: The data-centre consists of 300 servers
- *Average Workload of the data-centre*: The average workload of the server that the web application is hosted on is 70%
- *Amount of Applications*: The amount of applications run on the server fluctuates heavily, since the data-centre is based on cloud hosting applications are moved between servers dynamically depending on the workload of the server
- *User-Location*: The users are located in Switzerland
- *Amount of Database Calls per Month*: 6,000 Database Calls
- *Amount of Function Calls per Month*: 10,000 Function Calls
- *Communication Structure*: The web application is built asynchronously
- *Caching*: The web application does not utilize more than basic caching in its current iteration
- *Database Structure*: The web application is supported by a relational SQL database as well as a blob storage that stores larger files such as pictures
- *Computation*: The web application heavily utilizes JavaScript, only basic functions are computed server side while most of the relevant computation is done on the client devices
- *Average Packet size*: The average size of the data packets sent between the web application and the client devices is around 50kb
- *Average amount of data packages*: Around 20,000 packages

- *Required software*: A browser capable of displaying the web application (older versions of chrome, fire-fox or internet explorer are not supported), Microsoft Silverlight installed on the client device

- *Server Efficiency*: Around 7000 ssj_ops/watt

- *Average power consumption of the client devices*: 140 kWh

3.2 Definition of the Structural Unit Server-Side

In this section, the different elements relevant for determining the energy consumption of the structural unit server-side will be discussed (Figure 3.1).

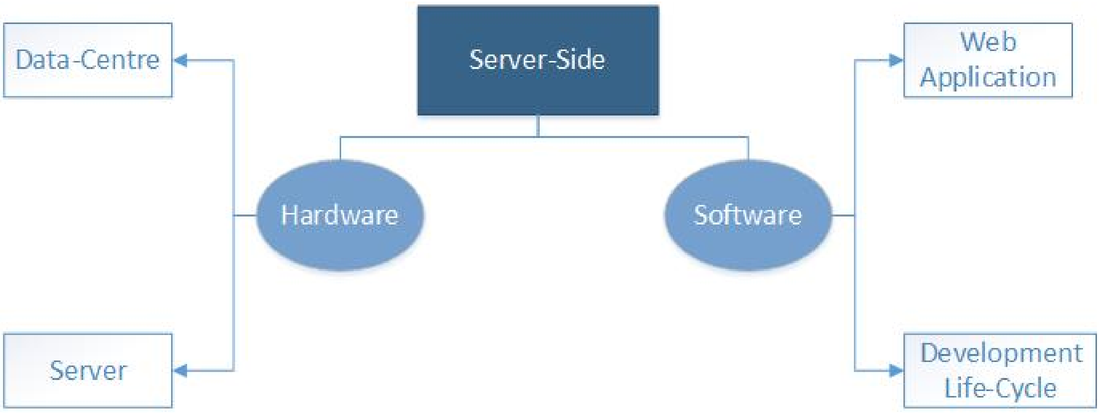

Figure 3.1: Server-Side

3.2.1 Hardware

The Data-Centre

Any web application is usually run on one or more servers. The servers are usually hosted in a data-centre. The energy consumption created by the server hosting the web application can vary greatly depending on various factors. First of all, most data-centres operate under the *service-level agreement of 99,9% availability*. Because of that, the servers are up and running, even when there is nothing to compute. Providers of cloud-based server solutions have managed to increase server workloads with the help of load balancing. During the day there are times of low traffic and peak times, where there are a lot of requests issued on a web application. In order to deal with high peaking requests most hosts of web application will *scale up* either the amount of servers or virtual machines in order to secure stability of the web application. The *location of a data-centre* in relation to its target audience impacts the length of the routing between the client devices and the server. The more jumps in a route between the client devices and the server, the more energy is consumed during data transfer. A data-centre which is run inefficiently, powered solely by renewable energy is not energy efficient, nevertheless, it is a factor that has to be considered when comparing the energy efficiency of data-centres. The different *power sources* that are available influence the energy efficiency of the server.

Those criteria cannot be measured directly from outside. Accurately measuring the exact energy consumption created by a specific web application hosted in a data centre for different data-centres is not something that will be considered in the scope of this thesis. Therefore, in order to estimate the energy consumption of the hardware, the web application is hosted on, the host of the web application has to provide the following information: *the server energy efficiency* ($E_{ServerEfficiency}$), *the workload efficiency* ($E_{Workload}$) , *the cost of energy* (C_{Energy}) , *scaling* ($C_{Scaling}$), *the power source* (P_{Source}), *the distance between the data-centre and the clients* ($C_{Distance}$).

The Server

The server energy efficiency ($E_{ServerEfficiency}$) : The server energy efficiency is the efficiency with which the server consumes energy in order to perform computations e.g. how good the energy efficiency of the server that the web application is hosted on is. The server energy efficiency can be heavily influenced by the software controlling the server. In *Analysing the Energy Efficiency of a Database Server* Tsirogiannis et al. research the role of database software for the energy efficiency of servers in a data-centre. They conclude, that there can be a difference of up to 60% of CPU usage for a server depending on the controlling software. Higher CPU usage performing the same tasks results in a higher energy consumption and therefore less energy efficiency. The authors conclude that the software configuration that provides the best energy efficiency is also the one that provides the best performance values [25].

Example: The energy efficiency of the server that the example web application is hosted on is 7,000 ssj_ops/watt. Based on the latest SPECpower_ssj2008 benchmark for forth quarter of 2013, this puts the energy efficiency of the server into the middle tier of modern servers. The important factor is finding the appropriate evaluation method for this criteria in order to create comparable values that can be a deciding factor when choosing a data centre for a web application.

$E_{Workload} : \frac{C_{AvgWorkload} * N_{WebApp}}{N_{Server}}$

The workload efficiency ($E_{Workload}$) : The workload efficiency is the *average monthly workload of the data-centre* ($Avg_{Workload}$) divided by *the number of servers* related to host the web application (N_{Server}) divided by the *number of applications* run on those servers (N_{WebApp}). The workload efficiency determines the average workload of each server in a data-centre. Modern data-centres utilize load balancing in order to even out the workload on all servers and increase the operating efficiency of the data-centre. There are various types of load balancing. In *Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges* Buyya et al. address the issue of high energy consumption in data-centres based on cloud computing. Depending on the way the balancing between the different servers is handled, the energy consumption for the data-centre may vary. In order to increase the energy efficiency as well as performance of a data-centre, the authors propose dynamic resource provisioning and allocation algorithms that consider the synergy between various data center infrastructures [23]. The important aspect of this approach is to understand that data-centres operate under strict service level agreements that require a certain level of performance. Service level agreements ensure a good performance for users accessing the web application hosted on that server. Sacrificing performance in order to increase the energy efficiency is therefore not an option since the performance has to remain constant. This approach shows that performance and energy efficiency do not have to be exclusive. By choosing an appropriate data-centre, the overall energy consumption for a web application can be decreased significantly.

Example: The example web application is hosted in a data-centre in Ireland and has an average monthly workload of 70%. Cloud-based solutions tend to achieve higher overall workload values when compared to classic data centres where one rents one or more servers for himself which then run with a lower overall workload. While cloud-based solutions are becoming more and more popular due to cheap pricing, there are certain benefits, like protecting sensitive information, for renting a whole physical server for oneself. Nevertheless, newly developed cloud solutions provide efficiency gains that are quite substantial mostly due to the superior load balancing mechanism [22]. An overall higher workload for cloud-based solutions increases the energy efficiency compared to classical data-centres solutions where each user rents their own server. One of the newer approach is the Ananta load balancer. In *Ananta: Cloud Scale Load Balancing*, a load balancing mechanism is presented that has been inspired by the Azure public cloud service. It is designed to meet the scale, reliability, tenant isolation and operational requirements of multi-tenant cloud environments [24]. Creating your own load balancing mechanism provides greater control over data and might be a viable alternative for companies to existing cloud-based data-centres.

$C_{Energy} : \frac{Avg_{PowerConsumption} * N_{WebApp}}{N_{Server}}$

The cost of energy (C_{Energy}): The cost of energy is the *average monthly power consumption* of the data-centre ($Avg_{PowerConsumption}$) divided by *the number of servers* related to the host of the web application (N_{Server}) divided by *the number of applications* run on those servers (N_{WebApp})

Example: The monthly power consumption of the data-centre the example web application is is hosted at is 765 kWh per server and user. In a real world case study about the U.S. General Services Administration (GSA), Google published a report indicating that after migrating to a cloud-based solution the monthly power consumption for one servers was 465 kWh per user. Therefore, the server that the example web application is hosted on has a higher energy consumption than a typical server in cloud-based solution for Google Apps [29].

Scaling ($C_{Scaling}$): Depending on the infrastructure of the web application, there are different ways of handling user requests. A lot of web applications are hosted on virtual machines. When there are too many user requests at the same time, additional virtual machines are started in order to handle all requests. In some cases additional servers have to be utilized to handle heavy workloads on the server. While scaling of the web application increases the energy efficiency of the web application by powering down unnecessary resources, depending on how often the web application has to be scaled up, the scaling process causes additional energy consumption.

Example: Utilizing a cloud-based solution the example web application provides scaling via virtual machines. Depending on the workload and CPU usage, additional virtual machines are started and stopped in order to handle all incoming requests appropriately.

Due to the small user base of the example web application, the energy consumed when starting and stopping additional virtual machines possibly on other physical servers might exceed the benefits of the scaling mechanism.

The power source (P_{Source}): Energy consumption of the hardware in a data-centre is one of the major factors when analysing its efficiency. By utilizing alternative energy sources in order to power the data-centre, the energy efficiency of any web application hosted at the data-centre can be increased.

Example: The example web application does not utilize any alternative sources of energy. In order to determine the impact that the type of power source has on the energy efficiency of a data-centre there has to be a measurement of the energy efficiency of the data-centre as a whole. A popular method of doing so is the power usage effectiveness (PUE) metric. It is determined by dividing the total power entering a data-centre by the power used to run the IT equipment inside the data-centre. Adding alternative power sources based on renewable energy would then simply reduce the amount of total power entering the data-centre optimizing the PUE ratio [32].

In addition to those two key figures provided by the host of the data-centre, the following information is necessary:

The amount of users per month (U_{Amount}): this information is usually available through existing diagnostic tools most developers utilize to monitor their web application. The amount of end users is a very important key factor. The efficiency of most server-side and data-transfer criteria highly depends on the user base. The more users the energy consumption of a certain criteria is divided into, the more efficient a certain solution becomes. Unfortunately it is hard to estimate the user base for a web application during development. While there are extreme cases e.g. a web application for 3 users or a web application for 3 million users, the end result is hard to predict. In addition to that, a user base can change over time, existing users can stop using the web applications and new users can start using it. Evaluating the energy consumption in relation to the user base therefore only provides a current snapshot.

Example: The example web application has a fairly small user base of only 100 people, but it also only utilizes one server that is being shared with other applications. It is therefore only responsible for a share of the energy consumption that that server requires. On the other hand, there are web applications like the Google search engine with 5,922,000,000 daily search requests on average [31]. In order to handle such a huge amount of users, a lot more servers are required. In order to be able to compare the energy efficiency of both web applications, the user base takes a key role.

Based on those three key figures, the following function determines the average energy efficiency of the server or data-centre that the web application is hosted at for one user:

E_{Server} = f($\frac{E_{Workload} * U_{Amount}}{C_{Energy}}$, $C_{Distance}$, P_{Source}, $E_{ServerEfficiency}$)

- The coefficient $\frac{E_{Workload} * U_{Amount}}{C_{Energy}}$ influences the function in a positive way, the higher the coefficient the better the average workload efficiency and the the higher the amount of end users is.
- $C_{Distance}$ influences E_{Server} in a negative way, the higher the distance between the server and the client is, the higher the energy consumption for data-transfer is.
- Depending on the chosen type of power source, P_{Source} influences the overall energy efficiency in a positive or negative way, the more renewable energy is utilized to run the data-centre the better.
- The better the efficiency of the hardware of the utilized servers $E_{ServerEfficiency}$, the better the overall energy efficiency.

This function is only a first step integration the different criteria for the hardware aspect of the server-side. In future work one could look into a more specific evaluation in order to create an overall benchmark for the energy efficiency of the server-side hardware.

3.2.2 Software

Web Application

When evaluating the energy efficiency of a web application, there are various key figures to consider. While not all of them can be translated into energy consumption directly, they can give an indication as to whether a web application is running more efficiently or less efficiently than another web application.

Operational cost $(C_{Operational})$: the cost of all relevant operations performed on the server side in order to run the web application. Those operations are:

Average amount of function calls per month $(N_{FunctionCalls})$: the average amount of function calls executed by the web application during the time a user utilizes the web application. Based on this information, the energy cost for server-side computation can be estimated. This includes all function calls executed by user interaction.

Example: the example web application on average has around 10,000 function calls per month divided by its current user base. Evaluating such a number is difficult. Each function call and each computation consumes energy. Whether the energy is consumed in an efficient way or not depends on various factors. If the function substitutes a similar activity that requires more energy, it is saving energy. Depending on which is operating more efficiently, the client-device or the server, it might be better to run more functions server-side or more functions client-side.

Average amount of database calls per month $(N_{DatabaseCalls})$: Depending on the task a user wants to perform, requests have to be performed on one or more databases. Performing queries on a database can be an energy intensive task, depending on the complexity of the query that is executed. In addition, the data-transfer from the database towards the web application consumes energy as well.

Example: the example web application performs around 6,000 database calls per month. In general it is considered good practice to minimize the amount of database calls for a web application by saving important data that is used frequently in the cache of the web application, for example in the form of cookies.

Based on those key figures, the operational cost of energy can be determined by the following formula:

$$C_{Operational} : \mathrm{f}(N_{FunctionCalls} * C_{Operations}, N_{DatabaseCalls} * C_{DatabaseCalls})$$

The operational cost of energy is the average number of executions of each operation multiplied by the energy cost associated with performing one operation as well as the amount of database calls multiplied with the energy cost associated with them. Determining the individual energy cost is something that will not be done in the scope of this thesis.

In addition to the operational key figures of the web application, there are structural properties that influence the energy efficiency of a web application:

Structural cost ($C_{Structural}$) : the energy cost of all structural properties of the web application. Those properties are are:

Communication structure of the web application ($W_{CommunicationStructure}$) : Web applications can communicate synchronously or asynchronously. The communication structure determines the way the web application transfers data to the client device. Synchronous web applications reload the whole web page each time a function is called that requires data from the server. That means that overall, more data than necessary has to be sent between the server and the client. Depending on the content of the web application and the size of the unnecessary data, this is inefficient and increases the amount of energy used by the web application. The more unnecessary data that is sent, the more energy is consumed. Web applications that are build based on asynchronous communication only transfer data that is needed for the specific function that is executed and only refresh a part of the web page, not the whole web page.

Example: The example web application is build based on asynchronous communication which is becoming common practice. Asynchronous communication not only increases the energy efficiency of the web application, but it also increases the performance for the end-user. Minimizing data traffic is an important aspect when evaluating web applications, especially with the increase of mobile devices accessing web applications over mobile networks. However, asynchronous web applications shift more computations to the client-device. It therefore also depends on the energy efficiency of the server hardware as well as the client-device hardware as to how much energy can be saved.

Caching of the Web application ($W_{Caching}$) : Website Caching is a method that is gaining more and more popularity. It allows the creation of multiple representations of a website in a cache storage. There are two different types of caching: server-side caching and client-side caching. The cache storage can be local in the browser, or in a search engine like Google. Frequent requests on the same web application can be satisfied by cache storage if certain criteria are met. This technique can greatly reduce traffic as well as user requests issued on a web application. Caching reduces the power consumption because content that is requested frequently can be accessed multiple times without having to compute that content again. The most common occurrence for website caching is browser caching but other forms are being utilized more frequently as well. Streaming hosts offer servers providing similar content in different geographic locations in order to minimize the routing for the data-transfer.

Example: While caching provides several benefits when it comes to energy consumption, it also has practical downsides. In the case of the example web application, it does not support caching due to security concerns. Because multiple users access the application from the same terminal, it is important that one user cannot access data from another user.

Structure of the utilized databases ($W_{DatabaseStructure}$) : Databases can be set up in various ways: Relational and Non-Relational Databases, Blob Storages or In-Memory databases.

Example: Choosing the appropriate database structure for a web application highly depends on the situation. In the case of the example web application, it was decided to utilize two different databases. A blob storage for storing larger files pictures and documents, and one relational SQL database to store information like time entries and user information. This is for a user base of 100 people. Once the user base grows larger, adding additional databases in order to split up the information could become necessary. Further evaluations to determine the optimal database structure for web applications depending on the context might be an interesting topic for future work.

Website Category ($W_{Category}$) : A web application can belong to different categories. It can provide a streaming service or perform various computations for a user. A web application that performs services in order to increase the energy efficiency of something would be in a different category than a web application that provides movie streaming. In addition to that, the user base could be utilized in order to categorize web applications. Standard software is more likely to appeal to a larger user base than specialized software e.g. a niche product for a small group of users. In order to compare standard and specialized web applications, a malus value could be introduced. Determining the correct malus value that would allow a fair comparison between web applications that are from a different category could be an interesting research idea for future work.

Example: The example web application is probably best described as a general online application. It provides certain functionalities to the user by simulating a normal application on the client-device while being run over the internet on a server.

Creating categories allows a differentiation which determines how well different web applications can be compared with each other.

Life-Cycle

Development ($W_{Development}$) : Modern web applications are complex software products. A very popular web application is the Google search engine. It was developed for two years before it was released, after which it has been continuously improved. Most web applications that are run over an extended period of time need to be maintained. Software bugs have to be fixed and additional features have to be developed. New application updates have to be deployed on the server. Depending on the size of the development team, those efforts can consume more or less energy, for example electricity and heat at the work place. The development of a web application can consume a significant amount of energy. The decisive factor whether the energy consumed during the development was used efficiently is the ratio between the amount of energy that has been consumed and the amount of end users utilizing the application. The energy consumed during development per end user is the factor that decides if the application was developed energy efficient or not.

Example: The example web application has been developed by five people over one year. In this case adding another developer would not have significantly decreased the development due to dependencies on external sources, which is why the size of the development appears to be appropriate. Optimizing the energy consumption of the development process by modifying the amount of developers working on the project as well as the project time frame is something that could be interesting to research in.

Based on those key figures, the structural cost of energy can be determined by the following formula:

$$C_{Structural} = \mathrm{f}(W_{Caching}, W_{CommunicationSystem}, W_{Development})$$

The structural cost of energy is based on the way the web application is caching its content, the chosen form of communication, synchronous or asynchronous and the energy consumed during development.

3.2.3 Total Energy Evaluation of the Structural Unit Server-Side

Based on the operational and the structural cost of energy, the following formula determines the average energy efficiency of the structural unit server-side for one user:

E_{Server} = f($C_{Operational}, C_{Structural}, E_{Hardware}$)

The server-side energy efficiency evaluation is based on the operational cost of energy, the structural cost of energy and the hardware efficiency of the server.

3.2.4 Significance Ranking of Server-Side Evaluation Criteria

In the table 3.1 all factors impacting the energy efficiency of the structural unit server-side will be ranked in an descending order according to their influence on the energy efficiency starting with the most important factor. For each element there is a short description explaining its importance. Ranking the server-side criteria is a complex task that depends a lot on the web application that is evaluated and where it is hosted at. Depending on the context, the ranking order might be different. This ranking is only a first step, determining the importance of each factor for a certain web application is an interesting topic for future work.

3.3 Definition of the Structural Unit Data-Transfer

In this section, the different elements relevant to determining the energy consumption of the structural unit data-transfer will be discussed (Figure 3.3). The route between the server and

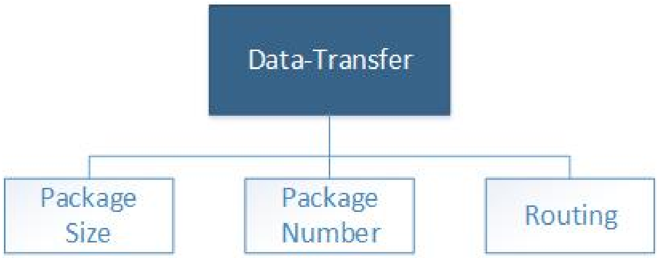

Figure 3.2: Data-Transfer

the client cannot be influenced directly by the host of the web application. Data is transferred by third parties that provide the network infrastructure for the internet. Nevertheless, there are factors which can be influenced by the host of the web application: *the distance between the data-centre and the clients* ($C_{Distance}$), *The average data size of the packets being sent to the clients* ($Avg_{DataSize}$), *The average amount of data packages being sent to the clients* ($Avg_{AmountofDataPackages}$).

The distance between the data-centre and the clients ($C_{Distance}$) : The average routing distance between the physical address of the server the web application is hosted on to the physical address of the targeted client devices. By identifying the target users, the developer of the web application can optimize the location for his server. When data is being transferred over the internet it usually cannot reach its target destination directly. Data is therefore routed through additional computers until it can be transferred to the end-user. Each additional computer required for the data routing is called a jump. Minimizing the distance between the physical locations of the clients and the chosen data-centre reduces the amount of jumps that have to be performed when sending information from the server to the clients.

Example: The servers the example web application is hosted on are in Ireland while the targeted user base is in Switzerland. Every time a user connects to the web application, the data is being routed between Switzerland and Ireland. If the server would be closer to Switzerland, the energy efficiency could be increased due to a shorter distance for the data traffic.

Ranking	Element	Impact
1	$W_{Development}$	Depending on the size of the development team, each device used for initial and ongoing development requires energy. The larger and more complex the web application is, the higher the energy consumption created during the development process.
2	U_{Amount}	The size of the user base has great impact on the energy efficiency of the web application. The more users a web application has, the better the invested energy for running the web application is utilized.
3	$E_{ServerEfficiency}$	The efficiency with which the server utilizes the consumed energy is an important indicator for the overall effectiveness of the data-centre that the web application is hosted at.
4	C_{Energy}	The energy consumption created by the server is one of the main contributors towards the total energy consumption of a web application. The lower it is the better the energy efficiency of the web application is.
5	P_{Source}	The more renewable energy that is used running the web application, the better the energy efficiency is.
6	$W_{Category}$	The purpose of the web application can influence the overall energy efficiency in a positive way, if the purpose is to increase the energy efficiency of an external item.
7	$C_{Scaling}$	Scaling web applications only increase the provided server power when needed. The better the application handles the scaling process, the better the energy efficiency is.
8	$E_{Workload}$	The higher the workload is, the better the resources used to host the web application are used and therefore the energy efficiency is.
9	$W_{CommunicationSystem}$	The type of communication determines the efficiency of the data-traffic. By optimizing the communication between the server and the client-devices, the energy consumed for transmitting data can be reduced substantially.
10	$W_{Caching}$	If a web application offers caching, it can reduce the energy consumption on the client-side and therefore increase the energy efficiency of the web application
11	$C_{DatabaseCalls}$	A well optimized web application minimizes the amount of database calls by caching relevant data and optimizing their access queries. This can increase the energy efficiency of the web application.
12	$N_{FunctionCalls}$	The more complex the web application is, the more services it provides and operations it performs, the higher the energy consumption of the web application is.

Table 3.1: Significance Ranking Server-Side

The average data size of the packets being sent to the clients ($Avg_{DataSize}$) : The average data size of the packets that are being used to communicate with the end-user devices depend on how much data is being sent for each operation. With the help of caching and by optimizing function calls, the amount of data that has to be sent can be minimized.

Example: The average data size transferred between the example web application and the client-devices is around 50kb. Evaluating this number is difficult. Deciding which data size is appropriate for the service offered by a web application is something that could be interesting to research in future work. It is expected that the amount of data sent by more complex web applications will be growing. Due to increasing possibilities the size of web pages is increasing as well. In the year 2010 the average size of a web page was around 700 kb and it is estimated to exceed 2,300 kb at then end of 2014 [41].

The average amount of data packages being sent to the clients ($Avg_{AmountofDataPackages}$) : Modern web applications provide a lot of information that is based on computational results which are then send to the client. By utilizing client-side computation based on methods like Java-script, the amount of packets that has to be sent can be reduced. Instead of sending the result of each computation every time it is required, the basic data required for all computations is send once and then used to create all required results resulting in less communication between the server and the end-user device.

Example: The average amount of data packages sent between the example web application and the user-devices is around 20000 packages. Just like evaluating the data size, this depends on the related functionality offered by the web application. Creating benchmarks that allow the comparison of data traffic properties between different web applications is something that could be researched in future work.

By optimizing these two factors the developer of web applications can greatly influence the energy consumption created by the transfer of data, especially when facing increasing data traffic through cellular networks which is very costly in terms of energy consumption.

3.3.1 Significance Ranking of Data-Transfer Evaluation Criteria

In the table 3.2 all factors impacting the energy efficiency of the structural unit data-transfer will be ranked in a descending order according to their influence on the energy efficiency starting with the most important one. For each element there is a short description explaining its importance. Ranking the data-transfer criteria is a complex task that depends a lot on the category of the web application as well as the data-traffic. Depending on the data-traffic, the ranking order might be different. A web application for movie streams might prioritize other criteria than web application for an accounting system. This ranking is only a first step, determining the importance of each factor for a certain web application is an interesting topic for future work.

Ranking	Element	Impact
1	$C_{Distance}$	Depending on the size of the data that is being sent, a lower distance between the web application and the client devices increases the energy efficiency of the web server.
2	$Avg_{AmountofDataPackages}$	Packages that are sent usually contain overhead information required for the transfer. By reducing the overall amount of packages, energy consumption can be optimized. However there is a trade-off between the average amount of packages sent to the client and the average data size of each packet. It depends on the web application which way is more efficient.
3	$Avg_{DataSize}$	By reducing the amount of data that is being sent with each package the overall efficiency of the data transfer can be improved. However, there is a trade-off between the average amount of packages sent to the client and the average data size of each packet. It depends on the web application which way is more efficient.

Table 3.2: Significance Ranking Data-Transfer

3.3.2 Total Energy Evaluation of the Structural Unit Data-Transfer

In order to determine how much energy is consumed by the web application the total related power consumption of the data transferred between the web application and the end-user device has to be determined. Influencing factors that the host of the web application can optimize are:

$$E_{Data-Transfer} = \text{f}(C_{Distance}, Avg_{AmountofDataPackages} * Avg_{DataSize})$$

The Data-Transfer energy efficiency evaluation is based on the distance between the physical server and the client-device, and the average amount of data packages as well as the average size of the data traffic between the server and the client-side.

3.4 Definition of the Structural Unit Client-Side

Calling a web application from any client device consumes energy on the client device. The following diagram provides an overview over the structural unit client-side (Figure 3.2).

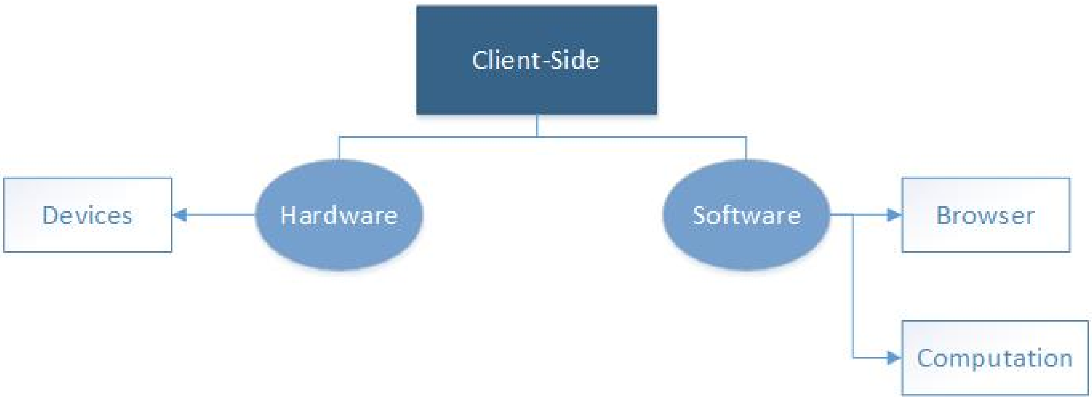

Figure 3.3: Client-Side

3.4.1 Hardware

Devices

The client-hardware efficiency: The *hardware efficiency of a client device* ($E_{ClientEfficiency}$) is the efficiency with which the client-device consumes energy in order to perform computations. The energy efficiency of mobile devices is especially important since they are battery powered. In *Energy efficiency of mobile clients in cloud computing* Miettinen et al. analyse the energy consumption of mobile devices in cloud computing. By shifting the trade-off between energy consumed for computation and energy consumed for data-transfer, one can reduce the energy consumption of mobile devices [37]. This trade-off not only applies to mobile devices but other types as well. By optimizing accordingly, the overall energy efficiency can be improved.

Example: The average energy consumption of the devices used to access the example web application is 140 kWh. Evaluating that value depends on the type of device that is used by the user base. Desktop computers consume more energy than laptops or smartphones. By optimizing the example web application for different type of devices, the energy consumed by those devices can be reduced. A web application run on a smartphone does not have to offer the same functionality in the same way as the same web application run on a desktop

PC. Usually customers do not expect applications for smartphones to provide a similar experience as desktop PC applications do. When developing a web application, one can chose to adapt the user interface for the client device. By identifying the size of the screen of the device, the appropriate interface can be chosen and displayed. By adjusting the web application for the specifications of the client device, the energy efficiency of that device can be increased.

Component Power Consumption ($C_{ComponentPowerConsumption}$) : Any client device has a basic power consumption that is needed in order to stay activated. That power consumption is usually related to the various hardware components that are used like the computer screen, the input devices or the CPU. While there is a huge variety of client devices, for example smart phones, tablets or desktop PCs, the following approach will be general in nature and applicable to all forms of client devices. In *A Method for Characterizing Energy Consumption in Android Smartphones* Corral et al. present an approach for measuring the energy consumption of a mobile device based on the time the battery lasts. This is done by linking the different components of a mobile device and the amount of time they take from the time the battery lasts until it has to be recharged. This is a very interesting approach. It allows for differentiated measurement for various components of mobile devices, just like the general approach of this thesis that focuses on creating a more differentiated evaluation method for web applications.

3.4.2 Software

The Browser

Browser ($C_{BrowserSoftware}$) : Simply determining the energy consumption of the browser application is not enough. One has to determine the energy consumption associated with the web application that has to be evaluated. Operating any client device requires some sort of operating system necessary to access the various hardware components of that device. The energy consumption of the browser software has to be separated from the basic energy consumption of the device. In addition, users tend to browse multiple websites and web applications at the same time. Only the energy consumption related to the specific web application that is examined has to be determined, ignoring other activities in the browser.

Example: The example web application is written in a way so that it works the same for all different browser types. Evaluating the energy consumption of a web application run in a certain browser is different for each browser. Google offers the chrome debugger API in order to track the behaviour of websites in the chrome browser. Microsoft offers the console debugging API for debugging in the internet explorer. In order to evaluate different web applications run in different browsers there has to be a browser-interdependent evaluation method that is standardized and comparable.

Required Software ($C_{RequiredSoftware}$) : Some web applications require software in addition to the browser in order to be run on a client device. For example, streaming web applications may require adobe flash player in order to display any video content available on the web site. Some web applications may require additional software like Microsoft Silverlight in order to display at all. Microsoft Silverlight allows a user to view rich internet applications by downloading the web application into the client browser.

Example: In order to access functionalities like statistics of the data entries, a user has made over time, the Microsoft Silverlight plugin is necessary. It enhances the user experience by increasing the performance for media heavy content. However, as a downside, it downloads the whole content to the client device. Additional software required to access certain content on a web application might increase or decrease the energy efficiency of a web application. In order to evaluate the difference, one has to determine all the required software as well as their impact on the energy consumption.

Computation

Java Script ($C_{JavaScript}$) : Web applications can perform computations server-side or client-side. Outsourcing computations to the client-side is usually done with the help of JavaScript. This is especially advantageous when performing multiple computations based on the same values. It reduces the load on the server and minimizes the amount of data that has to be sent to the client device.

Example: The example web application performs a lot of calculations based on time frames, for example, displaying all time entries for the day, the week, the month or the year. Based on those entries, statistics are made. In order to minimize the data traffic between the server and the client device, the raw data is sent and all calculations are performed client-side with the help of JavaScript. However, there is a trade-off between the increase in energy efficiency due to reduced traffic and the possible decrease in energy efficiency due to performing all computation client-side. This solely depends on which hardware operates more efficiently, the server or the client hardware.

3.4.3 Significance Ranking of Client-Side Evaluation Criteria

In table 3.2 all factors impacting the energy efficiency of the structural unit client-side will be ranked in an descending order according to their influence on the energy efficiency starting with the most important one. For each element there is a short description explaining its importance. Ranking the client-side criteria is a complex task that depends a lot on the type of device that is evaluated. Desktop computers, tablets and smartphones offer different performance, energy efficiency and even browser software depending on the main purpose of the device. Therefore, depending on the device, the ranking order might be different. This ranking is only a first step, determining the importance of each factor for a certain web application is an interesting topic for future work.

3.4.4 Total Energy Evaluation of the Structural Unit Client-Side

In order to determine how much energy is consumed by the web application, the total related power consumption of the client device over the duration of visiting the web application has to be determined. That means the power consumption of all software executed by the client device that is necessary in order to execute the web application as well as the associated energy consumed by any hardware components. In addition to that, the energy consumption of any computations performed on the client-side has to be considered:

E_{Client} = f($C_{ComponentPowerConsumption}$, $C_{BrowserSoftware}$,$C_{JavaScript}$, $E_{ClientEfficiency}$)

The client-side energy efficiency evaluation is based on the component power consumption, the energy cost of the utilized browser, the energy consumption for computations performed via JavaScript and the energy efficiency of the client-device.

3.5 Overall Energy Evaluation of a Web Application

In conclusion, the overall energy efficiency of a web application can be determined by measuring the energy efficiency of the three structural units server-side, client-side and data-transfer:

E_{Total} = f(E_{Server}, E_{Client}, $E_{Data-Transfer}$)

Ranking	Element	Impact
1	$C_{ComponentPowerConsumption}$	In order to access any online content, the device has to be activated and connected to the internet. For most devices like mobile phones, tablets or desktop computers the power consumption of running the device will be the most significant part of the total power consumption.
2	$E_{ClientEfficiency}$	The efficiency with which the client utilizes the consumed energy is an important indicator for the overall effectiveness of the client-device used to access the web application. While the host of the web application cannot influence the type of client-devices used, it has certain control over how much performance is required in order to access the web application which might possibly increase the energy consumption as a whole.
3	$C_{BrowserSoftware}$	The process running the browser software is a portion of the component power consumption. Nevertheless, it is the element that can distinguish different web applications based on their client-side power consumption and therefore very important in order to evaluate the energy consumption of different web applications.
4	$C_{JavaScript}$	Depending on the web application, the computation can be done on the server-side or on the client-side with the help of Java Script computation. The more computational methods that are outsourced to the client device, the higher the energy consumption of the process on the device.
5	$C_{RequiredSoftware}$	Additional software required to use the web application requires additional resources from the client device. Depending on how much additional software is required, the impact on the energy efficiency could be significant.

Table 3.3: Significance Ranking Client-Side

The overall energy evaluation of a web application is based on the energy efficiency of the server-side, the energy efficiency of the client-side and the energy efficiency of the data-transport between both sides.

3.6 Evaluation Model

Based on the identified criteria, an *evaluation model* has been created (Figure 3.4). This model can be used in order to compare different web applications. This is done by introducing a *scoring system* for each web application. First, one determines how well a web application meets a certain criteria. If the criteria is well met, one point is added to the score of the web application. If the criteria is not met sufficiently, no points are added to the score of the web application. In addition to that, the different criteria will be weight based on the ranking order.

This scoring system is only a first step. In future work it would be interesting to determine a more detailed scoring system. The criteria used for this model are complex and often the evaluation of those criteria depends on the web application and the context of the web application. A more differentiated scoring system might allow for a better and more accurate evaluation.

Just like the scoring system, the *weighting system* is an important factor for the evaluation. However the ranking order of the different criteria might change depending on the web application and the context of the web application. As an example, when evaluating a web application that performs a lot of computations, the energy efficiency of the hardware of the server and the client-device as well as how much computation has been moved to the client with the help of JavaScript will be more important than when evaluating a web application for streaming movies. Determining how to weigh different criteria might be an interesting topic for future work.

In addition to that, one has to choose an appropriate time frame for the evaluation process. Evaluating different web applications might lead to different results depending on the time frame. The importance of different criteria for the energy efficiency could depend on on how long the evaluation process is. An evaluation process that is too short might deliver wrong results for criteria like energy consumed for computation or the amount of database and function calls performed. Determining a time frame that allows for an accurate evaluation of one or more web applications is also something that could be researched in future work.

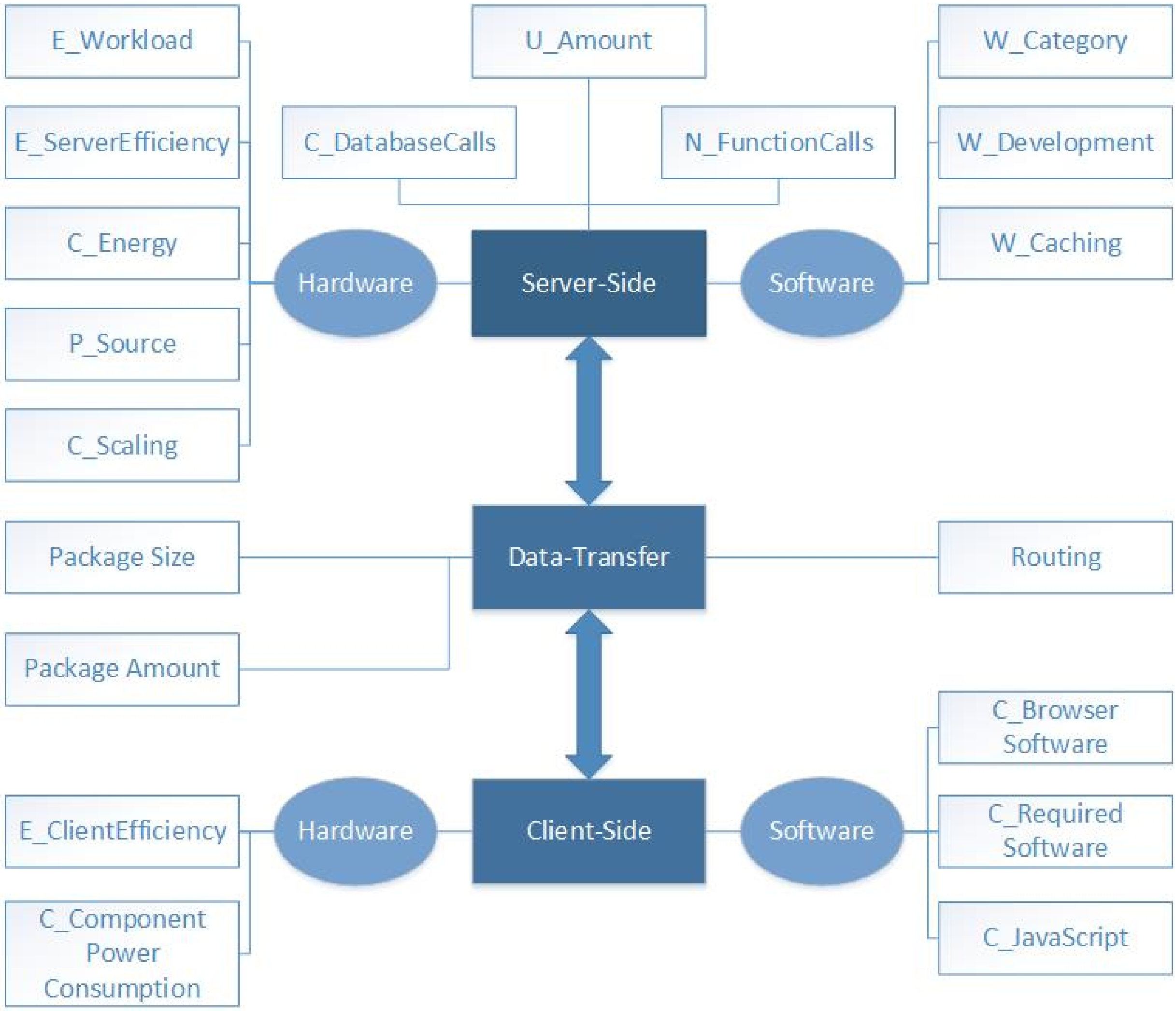

Figure 3.4: Evaluation Model

3.7 Thought-Experiment: What if the web application is run locally

With increasing mobility, a permanent internet connection has become a standard for a lot of end user devices. Most web applications are built around the concept that in order to use them, the user has to be connected to the internet. Being permanently connected might not always be possible and increases the energy consumption of any client device. In addition to that, depending on the energy efficiency of the server and the client-device, it might be more energy efficient to perform more computation client-side.

With the help of technologies like website caching and HTML5, newer web applications can be used without a permanent internet connection. Instead of transmitting data all the time, data is synchronised when possible. The question is where the data is stored locally while the device is off-line.

In *Offline web application and quiz synchronization for e-learning activity for mobile browser* Ijtihadie et al. shows a prototype web application for e-learning activities for students. They created an offline interface that allows students to pull tasks from the university network while they are on campus. They can then solve those tasks at home and submit them once they are connected to the university network again. The data is stored locally in the browsers with the help of the HTML5 web storage functionality [38] [39]. While this example has proven to be a succesful implementation of a web application with offline functionality, it might not work for all types of web applications. It might be hard to provide offline functionality for web application that require parallel editing of data by multiple people.

In *A Web-based, Offline-able, and Personalized Runtime Environment for executing applications on mobile devices*, Kao et al. present an approach to display runtime environment that combines web applications with an offline component in order to run them locally on mobile phones without being connected to the internet. They do so by providing a platform that provides several services like offline service, content adaption service and synchronization service [40]. Web applications are designed to be run while being connected to the internet.

By adding other components that allow certain aspects of the web application to be accessed offline, the energy efficiency of that web application could be increased. However there are several challenges that have to be overcome. First of all, it is hard to evaluate the energy efficiency of devices that are offline because it cannot be measured directly. In addition to that, it is hard to estimate the impact it has on the overall energy efficiency. Overall the improvement of the energy efficiency would depend on the changes in data-traffic, whether computation is more efficient on client or server-side and the energy efficiency of the server and client hardware. Researching the effect of offlining parts of a web application on the energy efficiency of a web application might be an interesting topic for future work.

The evaluation model developed in this thesis could be applied to evaluate a web application providing offline services. Changes to the structural units would include an increased importance of $W_{JavaScript}$ due to more computation performed on the client-device as well as a reduction of $Routing$, $PackageSize$ and $PackageAmount$ to basically zero, depending on the storage solution. More changes would be applied to the structural unit server-side. All hardware criteria as well as $C_{DatabaseCalls}$ and $N_{FunctionCalls}$ would be reduced in amount and importance. It is important to note that the web application will still be a normal web application on the internet, only parts of its functionality will be on the client-device. The software criteria of the web application will remain the same with the exception of $W_{Caching}$ which will be utilized as part of the offline storage in the browser and therefore increase in importance.

4 Practical Application

In this chapter, the evaluation model will be applied to the example web application. The different criteria will be evaluated and rated with either zero or one point. In addition to that, the following weighting system will be applied:

Because the web application provides a basic set of functionalities that could be considered standard content management, all criteria will be weighted the same and multiplied by a factor of 1. However, due to the high amount of computation moved to the client-device, $C_{JavaScript}$ will receive a factor of 1.25.

4.1 Practical Example

Server-Side :

- $C_{DatabaseCalls}$: 6,000 database calls per month is a sign for a web application with a small user base. In this case, due efforts of the developer minimizing the amount of database calls by utilizing cookies to store information, the criteria is well met.
- $N_{FunctionCalls}$: 10,000 Function Calls per month is also a sign for a small user base. The less functions are executed, the less energy is consumed, therefore this criteria is met.
- U_{Amount} : 100 users is a rather small user base for a web application with a year of development time and five developers. That is why this criteria is not met.

  Hardware :

- $E_{Workload}$: The average workload of the data-centre is 70%. Compared to other cloud-based hosting solutions, this is a decent value and therefore the criteria is met.

- $E_{ServerEfficiency}$: The server efficiency is around 7,000 ssj_ops/watt. Compared to the SPECpower_ssj2008 Benchmark ranking, this is an average value and therefore the criteria is met.

- C_{Energy} : The average amount of web applications hosted on the server the example web application is hosted at is changing dynamically and was not available. C_{Energy} can therefore not be calculated directly. However, the monthly energy consumption of the data-centre of 765 kWh per server and the amount of servers in the data-centre (61) indicate that the cost of energy will be a value that is decent in comparison to other data-centres and therefore the criteria is met.

- $C_{Scaling}$: scaling is done by starting up virtual machines in order to handle heavy workloads. This is standard practice for cloud-based solutions and therefore this criteria is met.

Software :

- $W_{Category}$: The web application could be best described as a general online application. This criteria does not award any scoring points as it is ment to create a differentiation between different types of web applications in order to determine if they can be compared with each other or not.

- $W_{Development}$: The example web application has been developed by five people for one year. In order to evaluate this criteria, one has to judge the energy consumed for development and the benefits for the user base. It is found that 100 users is a rather small user base for such a developer team and therefore this criteria is not met.

- $W_{Caching}$: Because the web application does not utilize more than basic caching, this criteria is not met.

Server-Side Score: The overall server-side score is therefore five points multiplied with a weighting factor of one. The example web application therefore scores **5 out of 9** possible points for the **server-side score**.

Data-Transfer :

- $PackageSize$: The average package size is around 50kb. When optimizing the web application by using JavaScript for client-side computation the data size was optimized to be as small as possible. Evaluating what a good size would be is highly dependant on the context and hard to compare. Nevertheless, the smaller the size is, the less energy is consumed for each transmission between the server and the client. In this case the criteria is met since only necessary data is sent to the client.

- $PackageAmount$: The average amount of packages is 20,000. By optimizing the data transmissions, the average amount of packages was also reduced. In this case the criteria was met.
- $Routing$: The data-centre used is located in Ireland. The user base, however, is located mostly in Switzerland. If a data-centre in Switzerland had been chosen, the distance between the server and the client-devices would have been a lot shorter. Therefore this criteria is not met.

Data-Transfer Score: The overall data-transfer score is therefore two points multiplied with a weighting factor of one. The example web application therefore scores **2 out of 3** possible points for the **data-transfer score**.

Client-Side :

Hardware :

- $E_{ClientEfficiency}$: The average power consumption of the client devices is 140 kWh. The web application is designed to be used by desktop PCs, not mobiles devices. The average power consumption for desktop PCs is estimated around 160 kWh based on [29]. Therefore this criteria is well met.
- $E_{ComponentPowerConsumption}$: The component power consumption is ment to link the energy consumption of client-devices with specific hardware parts of the client-device in order to allow for a more differentiated comparison. In case of the example web application, the information about the different components was not available.

Software :

- $W_{BrowserSoftware}$: The web application is optimized for all basic browsers (chrome, fire-fox and internet-explorer). Due to being optimized for all browsers, the energy efficiency should be similar and therefore browser independent. This criteria is met.
- $W_{RequiredSoftware}$: The only required software besides the browser is Microsoft Silverlight. Additional software may add functionalities to the web application. However,

it also increases the energy consumption on the client device. Therefore this requirement is not met.

- $W_{JavaScript}$: In comparison to average values found on the internet, both the server-side as well as the client-side hardware energy efficiency is decent. Therefore, due to the decrease in data-traffic due to optimizations achieved by performing a lot of computations client-side, this criteria is well met.

Client-Side Score: the overall client-side score is therefore : two points multiplied with a weighting factor of one and 1 point multiplied with a weighting factor of 1.25 ($W_{JavaScript}$). The example web application therefore scores **3.25 out of 4.25** possible points for the **client-side score**.

Overall score: the example web application scores **10.25 out of 16.25** possible points. Evaluating that number on its own is not easy. However, it shows that there is still room for improvement for the energy efficiency of the example web application. Nevertheless, it also shows that a lot of efforts have already been made to optimize the energy consumption of the example web application.

In order to evaluate criteria like $PackageSize$ & $PackageAmount$ or $W_{JavaScript}$, information about the development is necessary. In order to evaluate certain client-side criteria, detailed information about the client-device is necessary. In this example, the development information was accessible but the information about the client-devices was not and had to be researched on the internet. This evaluation model is only a first step. I believe the approach could be refined into a standard that can be applied more easily by reducing the complexity of each criterion. This could be an interesting topic for future work.

4.2 Demo Application

During the research for a practical application, a small demo was developed that calculates the distance between a client-device and the URL of a website or web application. The demo is displayed in Figure 4.1. The demo application was created based on the MVC 4 network. It is a web application with one functional method. A user can enter the URL of a website. The demo application will then calculate the number of jumps on the route between the IP address of the server the website is hosted at and the IP address of the client. The calculation is done based on functionality that is provided by the traceroute application for Windows. The

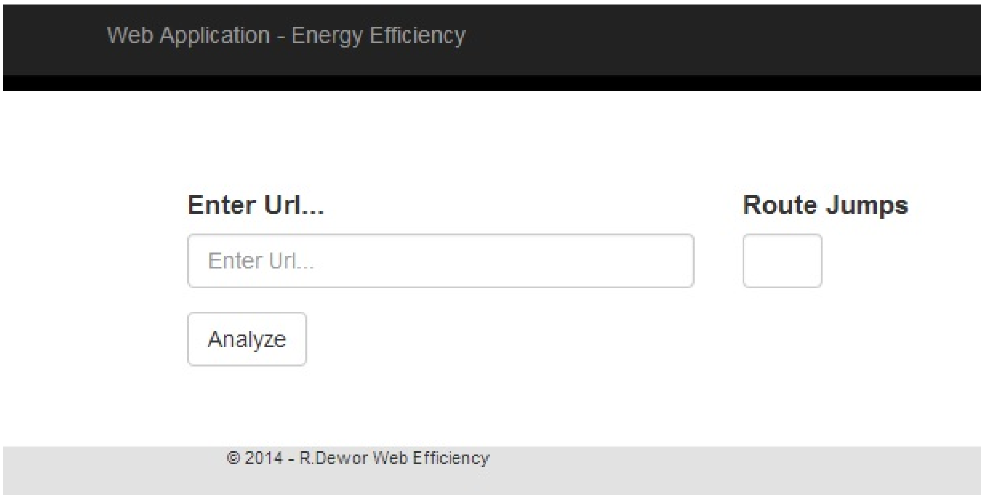

Figure 4.1: Demo

demo was created in order to research the existing possibilities for measuring the necessary information for the evaluation model. Due to time constraints, only the measurement of one criterion ($Routing$) was implemented.

The demo illustrates one of the main challenges for practical evaluation of web applications. While there are a lot of existing tools and API‘s that can evaluate parts of the criteria identified in this thesis, there is no existing solution that covers everything. Google offers the Chrome Platform APIs which offer the functionality to gather information about the data-traffic between the web application in the chrome browser and the client-device. The data size, the amount of requests and function calls as well as the option to debug JavaScript directly covers quite a few of the criteria identified as important factors for the client-side energy consumption [35]. Microsoft offers the Console Debugging API providing similar functionality for the internet explorer [36]. Other options are the Web Inspector for Chrome and Safari, F12 Dev Tools for Internet Explorer or Dragonfly for Opera. Data-centres provide diagnostic tools for the developers. Microsoft provides the azure software development kit and IBM provides the SmartCloud Application Performance Management and Diagnostics portfolio. There are different APIs for different browsers and different software tools for different operating systems, yet a way has to be found to analyse web applications in the same way.

There are already a lot of options for evaluating the energy efficiency of single aspects of web applications. The main task for the future will be finding a way of integrating the different approaches into one piece of software that can do it all and create comparable results independent of the host of the data-centre, the operating system and the browser.

Another challenge is the time frame for the analysis. All the information gathered from a web application in the browser is information recorded over a certain period of time. The results may vary depending on if only the first minute or the first 10 minutes of a user browsing the web applications are evaluated. Choosing the appropriate time frame depending on the type of web application in order to get the best evaluation results is not an easy task. It would require a permanent recording of information from the web application over a certain period of time.

The demo was implemented as a separate web application. However, the existing developer APIs that offer the evaluation of browser information record and evaluate everything that happens in the browser. In order to evaluate a web application it would have to be separated from the web applications doing the evaluation. Another option would be integrating the evaluation functionality into the web applications directly or in form of a browser plugin if equality can be assured over the different browsers.

5 Discussion

In this chapter, challenges and problems that arose during the development of this thesis will be discussed.

One of the main issues when trying to evaluate web applications is acquiring the necessary data that allows one to do so. The server-side hardware information is only available to the provider of the physical server, not the developer of the web application. The server-side software information is partially available to the developer and partially available to the provider. The required client-side information is only available to the end-user. No party possesses all necessary information to fully evaluate the web application. In order to assess the efficiency of the web application, the developer needs all information.

For this reason, evaluating criteria were chosen that are readily available to at least one party and that can be made available with reasonable effort. Most of the server-side hardware information should be available in common metric tools provided by most hosting providers. The server-side software information is available to the responsible party - the developer - and the information about the data-transfer can be acquired based on trace route and browser information available via the Google API. Client-side information is more difficult to acquire because access to the information required to evaluate the energy consumption of client-devices is not something that can be achieved that easily. Also, many users might choose not to provide such sensitive information about their devices.

5.1 Incentives

There are three stakeholders involved in the evaluation process of the energy efficiency of web applications: the host of the data-centre, the developers of the web application and the users. Finding incentives that encourage them to assess the efficiency of web applications is not an easy task. Evaluation approaches in the area of data-centre efficiency presented in this thesis are already going in the right direction. They are incorporating the performance aspect into the approach. The service level agreements are very important. A paying user wants his server to work around the clock with good performance. Approaches to increase the energy efficiency of servers and data-centres have to consider that.

It is found that the hosts of data-centres have a monetary incentive for reducing the energy consumption for the data-centres. Electricity is getting more and more expensive and the increasing popularity of cloud-based data-centres displays that increased operating efficiency is desirable for the host of data-centres.

Developing green software, or in this case, green web applications is not something that can be translated directly into a monetary benefit. During the planing of this thesis it was mentioned that the government might have the interest to create a green stamp for web applications that fulfil certain energy efficiency standards. This could send a positive signal to users who wish to save energy, attracting new customers for the web application.

While the power consumption of software is only a minor factor in our households, it might increase in the foreseeable future. With the internet of things and smart devices entering our households, software will be incorporated in many aspects of our lives. With increasing power prices, having a better energy efficiency in the household is a desirable goal for everyone. However, users are becoming more aware about privacy issues in relation to online activities. Creating other incentives so users will provide the necessary information along with the privacy issues that go with it might be an interesting topic for future research.

5.2 Scope

When evaluating the energy efficiency of a certain criterion, the scope has to be defined. If one is to be thorough with the evaluation of a physical server, one has not only to consider the energy efficiency of the server hardware or the software controlling it, but also the design and development process of the server as well as the creation process. Only then can one get a full and accurate evaluation of the energy efficiency of the server. Gathering so much information about a single part of the infrastructure for a web application might appear excessive. Determining the scope that is sufficient for the different criteria identified in this thesis is an interesting topic for future work.

The weighting system developed in this thesis might help prioritize important criteria that should be evaluated in more depth than others depending on the web application that is about to be evaluated.

5.3 Influence

When discussing the energy efficiency of a web application, the responsible party is the developer of the web application. The developer decides the specifications and properties of the web application and its infrastructure. When evaluating the energy efficiency one is essentially evaluating the decision made by the developer. Therefore, only evaluating criteria have been chosen that can be influenced by the developer of the application directly or indirectly.

Determining the reach of that influence is not easy. While developers can influence the hardware used by users by developing for a certain hardware performance level, the choice which hardware users are utilizing is still their own. It is open for discussion as to how much of the client-side energy consumption should be added to the overall energy consumption of a web application. Determining the influence a developer can have on certain criterion in order to weight its importance might be an interesting topic for the future.

5.4 User base

The size of the user base is a very important aspect for the energy efficiency of a web application. The more users a web application has, the more effectively the energy consumed is used. User groups not only differ in size but also in behaviour. It would be interesting to research if the behaviour of different user groups influences the energy efficiency of a web application as well. The energy efficiency of a network also depends on the workload it experiences. If a web application is used in a way that all users utilize it at the same time, that could potentially increase the energy consumption in comparison to those users using it at different times over the course of a day. It would be interesting to see in future work if, by creating incentives and controlling the user behaviour, the host of a web application could influence the overall energy efficiency of the web application.

5.5 Measurement and Comparability

The aim of this thesis was not to estimate the energy consumption directly but to evaluate if certain criteria are met or not. Then, based on a scoring system, one can effectively compare different web applications.

Direct measurement of the energy consumption of any part of the web application is very difficult. While the energy consumption can be measured, it is challenging to link it to a component. That is why, in research as in this thesis, the evaluating criteria will be measured in tendencies and average estimations. This provides a base for comparison as it can be measured how a web application performs according to certain evaluating criteria compared to others. Many values used for the evaluation criteria are average values. This is due to the fact that some values simply cannot be measured directly. Determining when average values are appropriate and when direct measurement is necessary in order to get decent evaluation results is a challenging task for the future.

Some criteria have to be measured directly. The energy efficiency of hardware or the power consumption of a data-centre are simple numbers. However, numbers only mean something when they are put into context. While there are benchmarks, for example, for the energy efficiency of a server (SPECpower_ssj2008 Benchmark) there are not benchmarks for all criteria identified in this thesis. Developing standard benchmarks for the evaluation model would be something that could possibly enhance the overall evaluation process and would be an interesting topic for future work.

The time frame for measured values is something that depends on which information is available. Basic diagnostic developer programs usually provide monthly and daily information, e.g. the amount of users that visited the web application in that month, how many function calls have been performed that day. This could be problematic if there are different time frames for different criteria making it hard to evaluate them. In addition to that, development of a web application does not stop once it is released. There will be ongoing developments and updates in the future. How that energy consumption should be incorporated today would be an interesting topic for future work.

In *Measurement and Rating of Software Induced Energy Consumption of Desktop PCs and Servers* Kern et al. describe an approach to measure the energy consumption of software for servers as well as desktop computers. This is done by applying a power meter to the test computers as well as with a data evaluation system [42]. Developing ideas in order to gain accurate feedback about the energy consumption of different software applications is an important area of research. However, adding power measurement tools to client-devices in order to evaluate the energy efficiency of web applications is not practical. This thesis finds that in order to evaluate the energy efficiency of client-devices when running a web application, it has to be measured by software potentially integrated in the web application or the browser.

5.6 Complexity

The criteria identified in this thesis are ment to provide an overview of important areas of a web application. When analysing the different aspects of a web application relevant for the evaluation of the energy efficiency, every criteria could be evaluated in great detail. The more details are incorporated in the evaluation process, the more complex it becomes. Determining the level of detail that is required to achieve a certain result is a challenging task. Almost every criteria could be broken down to raw energy consumption of some hardware components. Linking that energy consumption to specific software elements is also quite difficult. A compromise has to be found between the required level of detail for the evaluation process in order to achieve a specific result and the increasing complexity. Finding that balance is an interesting task for future work.

6 Conclusion

In this thesis, the energy efficiency of web applications has been researched. In order to evaluate a web application and all of its supporting infrastructure as a whole, criteria have been identified that allow one to do so.

In the preceding elaborations, existing approaches for evaluating the energy efficiency of server-side, data-transfer or client-side related subjects have been discussed. The focus of this thesis has been identifying criteria that determine whether a web application is energy efficient or not. Based on the identified criteria, an evaluation model has been created (Chapter 3.6). With the help of this model and a simple scoring system, an example evaluation has been performed (Chapter 4.1). As a result of this evaluation, it was shown that even a web application which was already optimized for performance and energy efficiency still has a lot of room for improvement. In addition, due to the importance of the increasing amount of mobile devices, web applications with offline functionality have been discussed (Chapter 3.7). After that, important problems and challenges identified during this thesis have been discussed.

There is a lot of potential in the area of the energy efficiency of web applications. The technology required to utilize this potential already exists, it simply has to be integrated into one evaluation model for the whole web application as well as its supporting infrastructure. With the help of a such an evaluation model, it could be shown in which areas the energy efficiency of a web application could be improved the most. By focusing on those areas a developer can maximize the results for his efforts to increase the energy efficiency of the web application. The results of this thesis provide a starting point for a lot of possible research. The most interesting tasks and challenges will be displayed under future work.

7 Future Work

In this this thesis, potential tasks and challenges that could be interesting for future work have been identified. The following will list those tasks and challenges: :

- Creating Benchmarks for the server-side, client-side and data-transfer criteria that allow a standardized comparison between different web applications for each criteria.
- Determining the correct malus value that would allow a fair comparison between web applications that are from a different category.
- Creating a more complex but standardized ranking system that identifies the importance of each criteria on the energy efficiency of a web application.
- Advancing the evaluation model in a standard model that can be applied more easily.
- Finding a way to integrate existing evaluation and measurement software into one piece of software that can measure all criteria and create comparable results independent of the operating system and the browser.
- Researching the effect of offlining parts of a web application on the energy efficiency of a web application.
- Determining when average values are appropriate and when direct measurement is necessary in order to get decent evaluation results.
- Optimizing the energy consumption of the development process by modifying the amount of developers working on the project as well as the project time frame.
- Deciding which data size is appropriate for the service offered by a web application.
- Determining how future energy consumption of ongoing development of a web application could be incorporated in the energy efficiency evaluation now.
- Creating a more detailed and more complex score and weighting system for the evaluation model.

- Creating incentives for all stakeholders that encourage them to increase the energy efficiency of structural units they can influence.
- Determining the scope and complexity required for a good evaluation result for each criterion.
- Determining the influence a developer can have on certain criterion in order to weight its importance.
- Research the impact of user groups with different behaviour and the influence the host of a web application has on that behaviour and its impact on the overall energy efficiency.

Evaluating the energy efficiency of software is a very complex and interesting research field. Ongoing efforts in research display a lot of interesting concepts for the future. This thesis has provided an overview of the important aspects for the energy efficiency of a web application. This thesis finds that the most promising areas of research will be in reducing the overall complexity by creating standardized models and concepts that can be applied to a wide range of web applications as well as incorporating existing measurement software into one standardized measuring tool.

Bibliography

[1] Lorenz Hilty, Wolfgang Lohmann, Siegfried Behrendt, Michaela Evers Wölk, Klaus Fichter, Ralph Hintemann 2013. "Grüne Software, Schlussbericht zum Vorhaben:Ermittlung und Erschliesung von Umweltschutzpotenzialen der Informations- und Kommunikationstechnik (Green IT), TV 3: Potenzialanalyse zur Ressourcenschonung optimierter Softwareentwicklung und -einsatz."

[2] Daniel Schien, Chris Preist, Paul Shabajee, Mike Yearworth 2011. "Approaches to the Dynamic Energy Footprinting of Online Media". In *Innovations in Sharing - Environmental Observations and Information Ispra.*, pp. 592-600.

[3] Yearworth, M, Schien, D, Preist, CW & Shabajee, P 2011. "Systems modelling of new media services". In *The 29th International Conference of the System Dynamics Society*. Washington, DC, USA. International System Dynamics Society, 29th International Conference of The System Dynamics Society, Washington DC, United States, 1 July.

[4] Stefan Naumann, Markus Dick, Eva Kern, Timo Johann. "The GREENSOFT Model: A reference model for green and sustainable software and its engineering." In *Sustain. Comput. Inform. Syst., vol. 1, no. 4*, pp. 294–304, 2011.

[5] Schien, D, Shabajee, P, Wood, S, Yearworth, M & Preist, CW 2012, "LCA for Green System Design of Digital Media". In *Klaus-Dieter Lang, Nils F Nissen, Andreas Middendorf, Perrine Chancerel (eds) Proceedings of Electronics Goes Green 2012+ (EGG)*. Fraunhofer Verlag.

[6] Schien, D, Preist, C, Yearworth, M & Shabajee, P 2012, "Impact of Location on the Energy Footprint of Digital Media". In *2012 International Symposium on Sustainable Systems and Technology (ISSST)*. IEEE Computing, NEW YORK, pp. -.

[7] Williams, D. R., Tang, Y. 2013b: "Impact of Office Productivity Cloud Computing on Energy Consumption and Greenhouse Gas Emissions, Environ". Sci. Technol. 2013, 47, 4333–4340, dx.doi.org/10.1021/es3041362.

[8] William Z. Bernstein, Devarajan Ramanujan, Fu Zhao, Karthik Ramani 2013. "Profiling Energy Consumption of Smartphone Users for Environmentally Efficient Business Decisions." *ASME IDETC/CIE* 2013.

[9] Lorenz Hilty 2013. "Einführung in Green It", Lecture on 13-09-2013, Lucerne University of Applied Sciences and Arts.

[10] GeSI 2012,"SMARTer 2020".

[11] Gerti Kappel, Birgit Pröll, Siegried Reich, Werner Retschitzegger John Wiley & Sons 2006. "Web Engineering: The Discipline of Systematic Development of Web Applications".

[12] UBA (2012). Glossar zum Ressourcenschutz. Umweltbundesamt. http://www.umweltdaten.de/publikationen/fpdfl/4242.pdf visited on 31.05.2013.

[13] V. Coroama, L. Hilty, E. Heiri, and F. Horn 2013. "The Direct Energy Demand of Internet Data Flows," In *Journal of Industrial Ecology*.

[14] Mingay, S. (2007). "Green IT: The New Industry Shock Wave", Gartner.

[15] Gilbert Held 2010. "A Practical Guide to Content Delivery Networks (2nd ed.)". CRC Press, Inc., Boca Raton, FL, USA.

[16] Allan Astrup Jensen, Leif Hoffman, Birgitte T. Møller, Anders Schmidt 2006. "Life Cycle Assessment. A guide to approaches, experiences and information sources". In*Environmental Issues Series 6*.

[17] Gitanjali Sachdeva. "Measuring and Optimizing Energy Efficiency in Internet Communication", Master Thesis, NTNU Trondheim , 2013.

[18] Vlad C. Coroama, Lorenz M. Hilty, Martin Birtel 2012. "Effects of Internet-based multiple-site conferences on greenhouse gas emissions, Telematics and Informatics". In *Volume 29, Issue 4*, November 2012, Pages 362-374, ISSN 0736-5853, http://dx.doi.org/10.1016/j.tele.2011.11.006.

[19] Luis Corral, Anton B. Georgiev, Alberto Sillitti, Giancarlo Succi 2013. "A Method for Characterizing Energy Consumption in Android Smartphones". In *Green and Sustainable Software (GREENS), 2013 2nd International Workshop*, on 20-20 May 2013, San Francisco, CA 38 - 45. http://dx.doi.org/10.1109/GREENS.2013.6606420

[20] Daniel C. Kilper, Gary Atkinson, Steven K. Korotky, Suresh Goyal, Peter Vetter, Dusan Suvakovic, Oliver Blume 2011. "Power Trends in Communication Networks," In *IEEE J. Sel. Top. Quantum Electron., vol. 17, no. 2*, p. 275. DOI: 10.1109/JSTQE.2010.2074187

[21] Albert Greenberg, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta 2008. "Towards a next generation data center architecture: scalability and commoditization". In *Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow (PRESTO '08)*. ACM, New York, NY, USA, 57-62. http://doi.acm.org/10.1145/1397718.1397732

[22] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb 2009. "A Taxonomy and Survey of Cloud Computing Systems". In *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC (NCM '09)*. IEEE Computer Society, Washington, DC, USA, 44-51. http://dx.doi.org/10.1109/NCM.2009.218

[23] R. Buyya, A. Beloglazov, and J.H. Abawajy 2010. "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges". In *Proceedings of PDPTA* 6-20.

[24] Parveen Patel, Deepak Bansal, Lihua Yuan, Ashwin Murthy, Albert Greenberg, David A. Maltz, Randy Kern, Hemant Kumar, Marios Zikos, Hongyu Wu, Changhoon Kim, and Naveen Karri 2013. "Ananta: cloud scale load balancing". In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13)*. ACM, New York, NY, USA, 207-218. http://doi.acm.org/10.1145/2486001.2486026

[25] Dimitris Tsirogiannis, Stavros Harizopoulos, and Mehul A. Shah 2010. "Analyzing the energy efficiency of a database server". In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10)*. ACM, New York, NY, USA, 231-242. http://doi.acm.org/10.1145/1807167.1807194

[26] Server Efficiency Rating Tool (SERT)TM. http://www.spec.org/sert/ visited on 02.01.2014.

[27] Klaus-Dieter Lange, Mike G. Tricker, Jeremy A. Arnold, Hansfried Block, and Christian Koopmann 2012. "The implementation of the server efficiency rating tool". In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (ICPE '12)*. ACM, New York, NY, USA, 133-144. http://doi.acm.org/10.1145/2188286.2188307

[28] The Green500 List. http://www.green500.org/news/green500-list-november-2013 visited on 02.01.2014.

[29] Google Apps: Energy Efficiency in the Cloud. http://static.googleusercontent.com/media/www.google.com/de//green/pdf/google-apps.pdf visited on 02.01.2014.

[30] Google Analytics. http://www.google.com/analytics/ visited on 02.01.2014.

[31] Statistic Brain. http://www.statisticbrain.com/google-searches/ visited on 02.01.2014.

[32] GREEN GRID DATA CENTER POWER EFFICIENCY METRICS: PUE AND DCIE. http://www.thegreengrid.org/ visited on 02.01.2014.

[33] M.K. Patterson, S.W. Poole, C. Hsu, D. Maxwell, W. Tschudi, H. Coles, D.J. Martinez, and N. Bates 2013. "TUE, a New Energy-Efficiency Metric Applied at ORNL's Jaguar". In *Proceedings of ISC*. 372-382.

[34] Masanet, E., Shehabi, A., Ramakrishnan, L., Liang, J., Ma, X., Walker, B., Hendrix, V., and P. Mantha 2013. "The Energy Efficiency Potential of Cloud-Based Software: A U.S. Case Study". Lawrence Berkeley National Laboratory, Berkeley, California.

[35] Chrome Platform APIs. http://developer.chrome.com/extensions/api_index.html visited on 010.01.2014.

[36] Console Debugging API. http://msdn.microsoft.com/en-us/library/ie/hh772173(v=vs.85).aspx visited on 10.01.2014.

[37] Antti P. Miettinen and Jukka K. Nurminen 2010. "Energy efficiency of mobile clients in cloud computing". In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10)*. USENIX Association, Berkeley, CA, USA, 4-4.

[38] Ijtihadie, R.M., Chisaki, Y., Usagawa, T., Cahyo, H.B., Affandi, A 2010. "Offline web application and quiz synchronization for e-learning activity for mobile browser". In *IEEE Region 10 Conference TENCON, 2010*, pp.2402–2405

[39] Moodle. https://moodle.org/ visited on 20.01.2014.

[40] Yung-Wei Kao, ChiaFeng Lin, Kuei-An Yang, and Shyan-Ming Yuan. 2012. "A Web-based, Offline-able, and Personalized Runtime Environment for executing applications on mobile devices". In *Comput. Stand. Interfaces 34*, 1 (January 2012), 212-224. http://dx.doi.org/10.1016/j.csi.2011.08.006

[41] Web Performance Today. http://www.webperformancetoday.com/ visited on 20.01.2014.

[42] Dick, Markus; Kern, Eva; Drangmeister, Jakob; Naumann, Stefan; Johann, Timo: "Measurement and Rating of Software-induced Energy Consumption of Desktop PCs and Servers". In *Pillmann, Werner; Schade, Sven; Smits, Paul (Eds.): Innovations in Sharing Environmental Observation and Information. Proceedings of the 25th EnviroInfo Conference "Environmental Informatics"* October 5-7, 2011, Ispra, (VA) Italy. Part 1 and 2, 2011, pp. 290 - 299.