

GWDG **NACHRICHTEN** 09-10|25

LLM Benchmarking for
Medical Applications

LLMs as Post-OCR
Correctors

Ethical AI

Ausbildungsprojekt
„InventarWorkerService“

ZEITSCHRIFT FÜR DIE KUND*INNEN DER GWDG

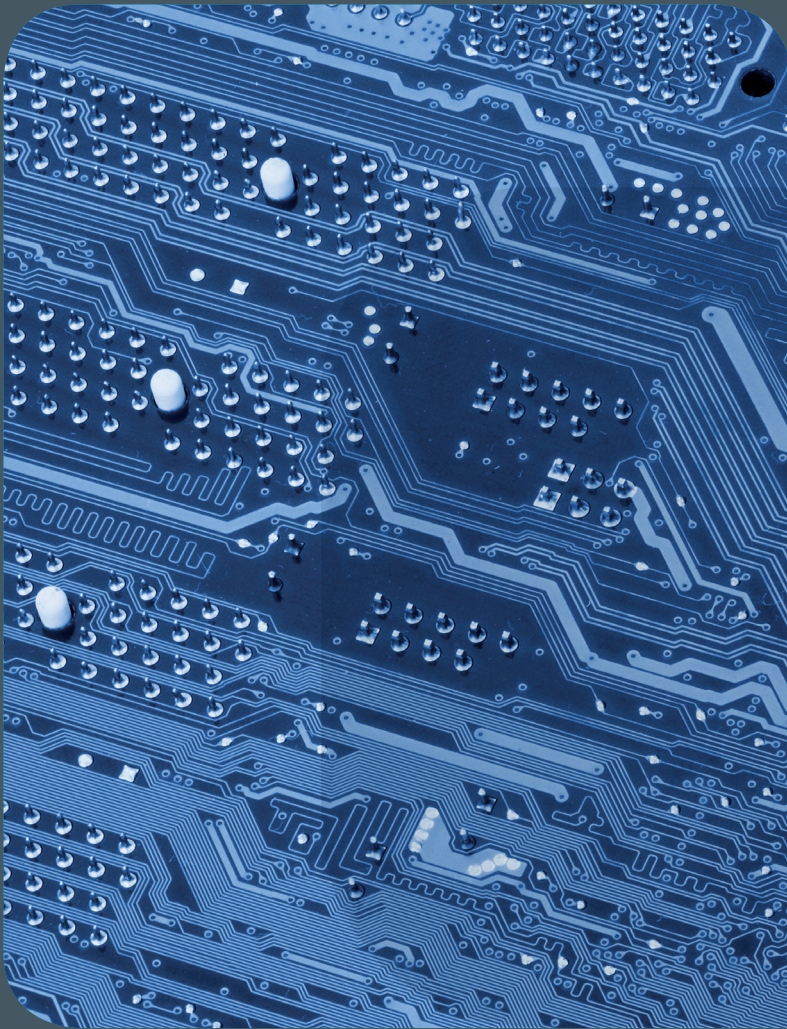


LLM



GWDG

Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen



GWVG NACHRICHTEN

09-10|25 Inhalt

-
- 4 LLM Benchmarking for Medical Applications
 - 7 LLMs as Post-OCR Correctors – Fixing Errors in Digitized Texts Using AI
 - 10 Ethical AI: Navigating AI through an Ambiguous Concept
 - 15 Das Projekt „InventarWorkerService“ für Fachinformatik-Auszubildende der GWVG
 - 24 Kurz & knapp
 - 25 Personalia
 - 26 Academy

Impressum

Zeitschrift für die Kund*innen der GWVG

ISSN 0940-4686
48. Jahrgang
Ausgabe 9-10/2025

Erscheinungsweise:
10 Ausgaben pro Jahr

www.gwdg.de/gwdg-nr

Auflage:
550

Fotos:
© Worldwide - stock.adobe.com (1)
© WrightStudio - stock.adobe.com (6)
© sabida - stock.adobe.com (9)
© Trionorejo - stock.adobe.com (23)
© Robert Kneschke - Fotolia.com (26)
© Tobias Grychtol-Matthaeus (24)
© MPI-NAT-Medienservice (3)
© GWVG (25)

Herausgeber:
Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen
Burckhardtweg 4
37077 Göttingen
Tel.: 0551 39-30001
Fax: 0551 39-130-30001

Redaktion und Herstellung:
Dr. Thomas Otto
E-Mail: thomas.otto@gwdg.de

Druck:
Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour
ramin.yahyapour@gwdg.de
0551 39-30130

*Liebe Kund*innen und Freund*innen der GWDG,*

*auch in dieser Ausgabe beschäftigen wir uns weiter mit großen Sprachmodellen und verschiedenen Facetten der Anwendung. Neben den technischen Voraussetzungen rücken die rechtlichen Voraussetzungen immer stärker in den Vordergrund. Der EU AI Act erfordert die Bestandsaufnahme und Risikobewertung von KI-Anwendungen, das Aufbauen von KI-Kompetenzen und seit August die Etablierung von Compliance-Strukturen, wenn sogenannte „General Purpose AI“-Modelle genutzt werden, die sehr breit einsetzbar sind und nicht nur für eine spezielle Aufgabe entwickelt wurden. Ab August 2026 gelten dann die vollständigen Anforderungen für Hochrisiko-KI-Systeme. Aus der Industrie wird Kritik geübt und eine Aussetzung oder Änderung der Vorgaben gefordert, um den europäischen Wirtschaftsraum nicht in Rückstand zu anderen Ländern zu bringen. Die Umsetzung der Vorgaben ist keine Kleinigkeit und erfordert umfassendes Handeln. Der Aufwand ist hoch und lässt sich schwer genau erfassen. Hierzu sind Schulungen und Dokumentation nur der erste Schritt. Wenn man nicht abgehängt werden möchte, ist jedoch Handeln auch von den Anwender*innen dringend erforderlich. Daher sind wir natürlich schon länger mit diesen Aufgaben beschäftigt, um den Einsatz von KI zu unterstützen.*

Ramin Yahyapour

GWDG – IT in der Wissenschaft

LLM Benchmarking for Medical Applications

Text and Contact:

Muzzamil Aziz
muzzamil.aziz@gwdg.de
Rana Ashar Mehmood
rana.mehmood@med.uni-goettingen.de

Large Language Models (LLMs) have demonstrated remarkable potential for clinical decision support, diagnostic reasoning, and biomedical information retrieval. However, their high computational requirements and substantial energy consumption present serious challenges to deployment in resource-constrained healthcare settings such as small hospitals, rural clinics, or mobile diagnostic units. This study evaluates the efficiency, accuracy, and energy performance of several contemporary LLMs – Phi-3.5 Mini (3.8B), LLaMA-2 7B, Mistral 7B, LLaMA-2 13B, Falcon 7B, Falcon 40B, Mixtral (MoE 12.9 B active), and a LLaMA-3 8B model fine-tuned on a hybrid biomedical corpus.

INTRODUCTION

Large Language Models (LLMs) are transforming the way healthcare systems approach clinical documentation, medical education, and diagnostic reasoning. By leveraging massive text corpora, these models can answer complex medical questions, summarize patient histories, and assist clinicians with differential diagnosis.

Despite their promise, LLMs also introduce operational challenges:

1. Computational intensity, often requiring high-end GPUs and large memory resources.
2. Energy inefficiency, which raises concerns about cost and sustainability.
3. Scalability limitations, especially in low-resource or mobile medical environments.

The objective of this benchmarking study is to systematically evaluate multiple LLMs along two dimensions: clinical accuracy and energy efficiency. By comparing fine-tuned models under standardized conditions, the work aims to identify models that provide an optimal balance for real-world healthcare scenarios.

Figure 1 illustrates the workflow that how LLMs can be evaluated using the CodeCarbon library. CodeCarbon runs in parallel with model execution and records power draw and estimated CO₂ emissions in real time during both inference and fine-tuning. This parallel instrumentation captures GPU, CPU, and system energy usage, and it converts energy consumption into carbon estimates by applying regional electricity carbon intensity values.

RESOURCE AND EXPERIMENTAL SETUP

Computational Resources

All experiments are conducted on the virtualized resources powered by the KISSKI infrastructure (see Figure 2) with 1 NVIDIA A100 GPU, 2 AMD 32-core processors and approx. 500 GB RAM. These resources are sufficient to run larger models

(such as Falcon 40 B model) smoothly. Smaller models don't need all that compute, however, same resources are used for processing smaller models in order to achieve fair comparison. This setup ensured consistent benchmarking of both small and large models during fine-tuning and inference. Fine-tuning times varied widely depending on model size from about 30 minutes for smaller models like Phi-3.5 Mini to up to 3 hours for larger models such as Falcon 40B and LLaMA-2 13B.

LLM-Benchmarking für medizinische Anwendungen

Diese Studie benchmarkt mehrere Large Language Models (LLMs) – Phi-3.5 Mini (3.8B), LLaMA-2 7B, Mistral 7B, LLaMA-2 13B, Falcon 7B, Falcon 40B, Mixtral (MoE 12.9B aktiv) und ein feinabgestimmtes LLaMA-3 8B – und bewertet deren klinische Genauigkeit, rechnerische Effizienz und Umweltbelastung bei medizinischen Fragestellungen. Alle Experimente wurden auf der KISSKI-Infrastruktur (<https://kisski.gwdg.de>) durchgeführt, die Hochleistungsrechnerressourcen für Fine-tuning und Inferenz bereitstellt. Der Energieverbrauch und die entsprechenden CO₂-Emissionen wurden mit der CodeCarbon-Bibliothek gemessen, die parallel zu den Modellen lief und während des Trainings sowie der Inferenz den Stromverbrauch und die Emissionen erfasste. Die Ergebnisse zeigen, dass kompakte Modelle wie Phi-3.5 Mini und Mistral 7B eine vergleichbare klinische Leistungsfähigkeit erreichen, dabei jedoch nur einen Bruchteil der Energie größerer Modelle wie Falcon 40B und LLaMA-2 13B benötigen. Diese Erkenntnisse verdeutlichen, dass eine gezielte Modellauswahl in Kombination mit effizienter Infrastruktur und Energieüberwachung eine ausgewogene Balance zwischen Genauigkeit, Leistung und Nachhaltigkeit ermöglicht und damit eine skalierbare und umweltbewusste Nutzung von LLMs in medizinischen Anwendungen unterstützt.

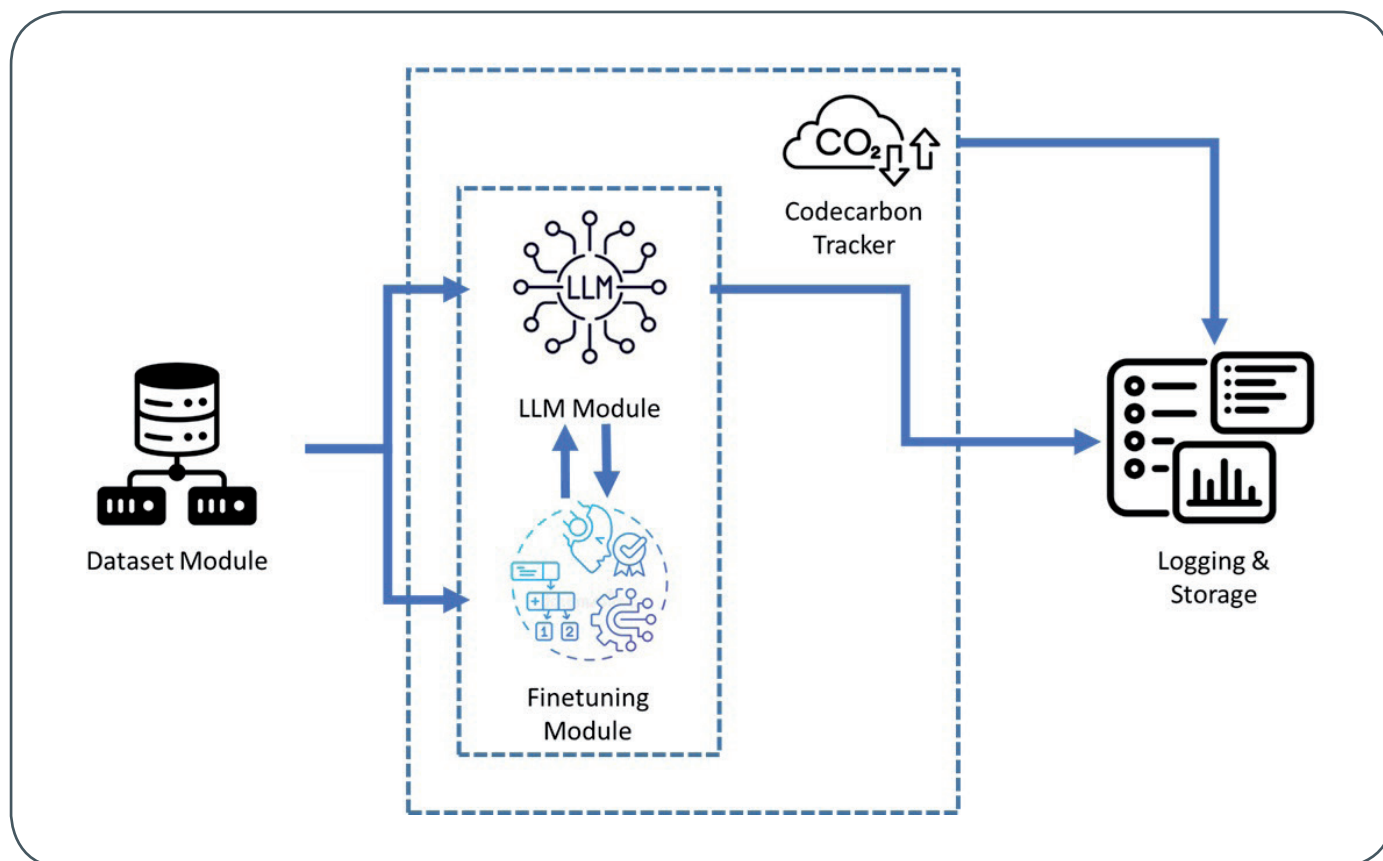


Figure 1: Experimental setup for models' evaluation

Virtual Resources	Specification
RAM	497.810 GB
CPU	64 threads, 2 x AMD EPYC 7513 32-Core
GPU	1 x NVIDIA A100-SXM4-80GB

Figure 2: KISSKI resources used for evaluations

Not all compute resources are utilized all the time. Compact models (e.g., Phi-3.5 Mini, Mistral 7B) used less GPU memory and compute, leaving some resources underutilized. Large models (e.g., Falcon 40B, LLaMA-2 13B) came closer to saturating GPU compute and RAM. CPU load was generally moderate, since most of the heavy lifting was done on the GPU.

Model Selection

Eight LLMs were chosen to represent a wide spectrum of size and architecture:

- Compact models: Phi-3.5 Mini (3.8 B), LLaMA-2 7B, Mistral 7B
- Medium-scale: LLaMA-2 13B, Falcon 7B, LLaMA-3 8B
- Large-scale: Falcon 40B, Mixtral (MoE 12.9 B active)

These models were selected based on open availability, performance reputation, and architectural diversity (dense vs. mixture-of-experts).

Dataset

To adapt general-purpose LLMs to clinical reasoning, each model was fine-tuned using a proprietary Laboratory Medicine Dataset – a curated diagnostic literature corpus containing guidelines, pathology reports, and biomedical abstracts.

Fine-tuning emphasized tasks such as disease identification, test interpretation, and treatment recommendation, aligning model behavior with medical diagnostic reasoning.

RESULTS

Figure 3 illustrates the energy consumption, carbon emissions and corresponding clinical accuracy of three different models. Similar evaluations are performed for all other models involved in this study. The key results and insights are explained in the following subsections. Using the MedQA dataset [1], clinical reasoning capabilities were assessed alongside computational efficiency, latency, and environmental impact. Energy consumption was measured via the CodeCarbon package [2], which estimated corresponding CO₂ emissions.

The findings indicate that compact models such as Phi-3.5 Mini and Mistral 7B deliver competitive clinical performance while consuming a fraction of the energy required by larger counterparts like Falcon 40B and LLaMA-2 13B. These results highlight that appropriate model selection and optimization can balance accuracy and sustainability, enabling practical use of LLMs in scalable, environmentally conscious medical applications. The following three subsections provide more insights to these research findings [3].

Model Accuracy

Across all models, clinical accuracy correlated moderately

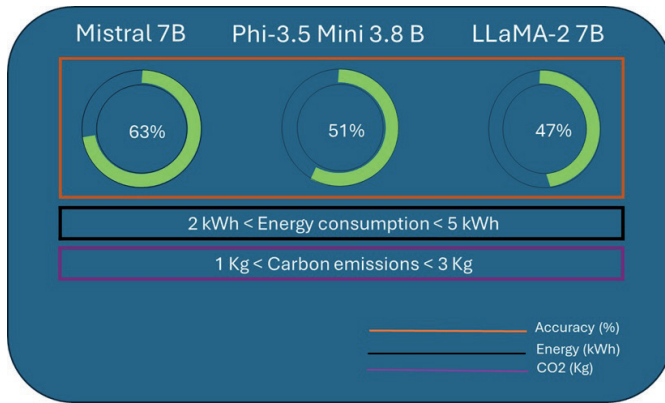


Figure 3: Comparison of Mistral 7B, Phi-3.5 mini 3.8B and LLAMA-2 7B. For more information, please refer to [3]

with model size but plateaued beyond 7–13 billion parameters. For example:

- Phi-3.5 Mini (3.8B) achieved performance close to LLaMA-2 13B, despite being nearly four times smaller.
- Mistral 7B showed strong contextual reasoning, likely due to its transformer architecture and optimized attention mechanisms.
- Falcon 40B, while achieving slightly higher scores, demonstrated diminishing returns relative to its massive computational cost.

These findings suggest that mid-sized or compact models can deliver near-state-of-the-art clinical reasoning at a fraction of the energy cost.

Energy Consumption

Energy monitoring revealed clear efficiency gradients:

- Compact models (Phi-3.5 Mini, Mistral 7B) consumed 5–10× less energy than large models.
- LLaMA-2 13B required roughly 40% more energy than Mistral 7B for marginal accuracy gains.
- Falcon 40B and Mixtral 12.9B incurred the highest carbon footprints, with emissions several times higher than compact alternatives.

These results emphasize the linear-to-superlinear relationship between model scale and energy demand.

Environmental Impact

Using Germany's regional carbon intensity, CodeCarbon estimated the CO₂ emissions per fine-tuning run. Compact models

emitted under 1 kg CO₂, while the largest model exceeded 5 kg CO₂ per equivalent training task.

For hospital deployments or repeated clinical retraining cycles, these differences translate into substantial operational savings – both in cost and environmental footprint.

LIMITATIONS

CodeCarbon provides a standardized way to compare models across experiments. Direct power measurement (e.g., with power meters) would be more precise, but harder to integrate at scale. CodeCarbon balances accuracy with practicality for research comparisons.

ACKNOWLEDGEMENT

The work presented in this article has been a joint effort of the Future Networks team, the eScience group of the GWDG, Göttingen, Germany and the Clinical Proteomics & Artificial Intelligence Group (CPAIG) at the Department for Clinical Chemistry/Interdisciplinary UMG Laboratories, University Medical Center, Göttingen, Germany.

The authors disclosed receipt of the following financial support for the research, authorship, and publication of this article:

This work is Funded by the German Bundesministerium für Forschung, Technologie und Raumfahrt (BMFTR) under the CELTIC-NEXT AI-NET-PROTECT and SUSTAINET-guarDian projects with grants 16KIS1292, 16KIS1280 and 16KIS2250.

The authors gratefully acknowledge the computing time granted by the KISSKI project (<https://kisski.gwdg.de>). The calculations for this research were conducted with computing resources under the project named "umg_virtual_assistant".

REFERENCES

- [1] Jin, D., Pan, E., Oufattole, N., et al. (2020). What Disease Does This Patient Have? A Large-Scale Open-Domain Question Answering Dataset from Medical Exams. arXiv:2009.13081.
- [2] Courty, B., Schmidt, V., Goyal, K., et al. (2024). CodeCarbon v2.4.1 [Software]. Zenodo. <https://doi.org/10.5281/zenodo.11171501>
- [3] Ali, A., Aziz, M., Mehmood, RA., et al. (2025) Benchmarking LLMs: Balancing Accuracy and Energy for Scalable Medical Applications. Poster presented at NHR conference 2025. ■



LLMs as Post-OCR Correctors – Fixing Errors in Digitized Texts Using AI

Text and Contact:
Radoslav Koynov
radoslav.koynov@gwdg.de

The digitization of scans or images such as historic documents into text can often contain numerous errors. Careful use of Large Language Models (LLMs) can help us fix some of these errors, but also bears certain risks.

INTRODUCTION

Digitization describes the process of converting information into a computer-readable format. In recent years, in German-language media the term “Digitalisierung” has been exceedingly used in a much broader sense in recent years, namely for the identification and exploitation of additional potential of digital systems to improve processes in administration, economics and society in general.

In this article, we will stick to the original, more narrow meaning. In particular, we focus on the digitization of text documents into digitally encoded text. The process at the center of this is called Optical Character Recognition (OCR). OCR engines have been around for a long time and have been developed into both proprietary and open-source software applications. OCR systems have to process an image or scan, perform text and layout detection to determine the correct order of text on the page (consider titles, paragraphs, side notes and foot notes...), recognize characters and output a digital text. Modern OCR software usually employs specially trained neural models for the character recognition step, because they have outperformed more traditional approaches since around 2010 – 2015.

This approach yields good results for most documents, but naturally the OCR software can also make mistakes. This is especially the case when it has to deal with non-standard fonts, handwritten text, special layouts or generally degraded source documents. However, good OCR quality is vital in order to correctly preserve historic documents and also affects the quality of downstream tasks, such as text mining, information retrieval and further text analyses.

USING LARGE LANGUAGE MODELS TO IMPROVE OCR RESULTS

Opportunities and Challenges

Large Language Models (LLMs), such as the proprietary ChatGPT model or the open-source LLaMa, are trained with self-supervised machine learning techniques on vast amounts of natural language text and thus naturally have a very strong contextual understanding. Exploiting these capabilities means that unlike in the case of traditional machine learning approaches, there is no need for large training corpora for the specific use case, so they are

comparatively much easier to use. At the GWDG, we additionally profit from numerous large models hosted on site and made available via a REST API [2]. If desired, and if training data can be curated, LLMs can still be fine-tuned using this training data to further improve results.

Thus, it is a very natural question to ask, if LLMs can be used to improve OCR results. However, certain problems and risks should not be ignored. In the following, we focus on “correcting” text that has already been obtained from an OCR system. It should be noted that this is not inherently a generative task and even when we prompt a LLM to perform a correction and output the corrected text, it can, theoretically, output any sequence of characters that may or not be connected to the input. What is more, LLMs are usually trained to output coherent and grammatically correct text. However, source documents, especially historic documents often use non-standard spelling and grammar, archaic forms, special characters that might be very rare in training data, and so on. The correction system should not alter any of these, but only correct irregularities arising from the OCR step.

Quantifying OCR Quality

To measure how different two texts, such as a ground truth and an OCR output, are, the Levenshtein distance is the most commonly used metric. It counts the minimum number of single-character changes needed to turn one text into the other. There are three types of changes it considers:

- Insertion – adding a letter (for example, “cat” -> “cart”)
- Deletion – removing a letter (for example, “cart” -> “cat”)
- Substitution – changing one letter to another (for example, “cat” -> “cut”)

Behebung von Fehlern in digitalisierten Texten mithilfe von KI

Die Anwendung gängiger Texterkennungsverfahren bei historischen Dokumenten kann oft zu zahlreichen Fehlern führen. Der Einsatz von Large Language Models (LLMs) kann uns helfen, die Anzahl dieser Fehler zu reduzieren, birgt aber auch gewisse Risiken.

For example, the Levenshtein distance between “kitten” and “sitting” is 3, because you can change “kitten” to “sitting” with three edits (“kitten” → “sitten” → “sittin” → “sitting”) and there is no shorter sequence of edits to do so.

To obtain a sensible metric to compare this for texts of different lengths, the Levenshtein distance is divided by the total number of characters in the reference to obtain what is referred to as the Character Error Rate (CER), which of course is a number between 0 and 1.

In the case of an error correction task, we can measure the CER before and after the correction system is run, and then look at the relative reduction. All the definitions can be made analogously at the word level, obtaining the Word Error Rate (WER).

In previous research on the topic of OCR correction, first using rule-based approaches, and then building specialized neural models from scratch, or even fine-tuning LLMs, those were usually the main metrics considered.

Ground Truth Datasets

To assess the quality of OCR systems, so called ground truth datasets are vital. These contain reference texts extracted by human experts, with the corresponding images or scans and allow quantifying the success of OCR for this particular dataset. Often, they contain additional metadata such as coordinates of text blocks within the page. Of course, due to the manual extraction step, these ground truth datasets are quite complex and time-consuming to create. The OCR-D project [1] creates and maintains such datasets, with pages from historic documents in different languages.

We used a complex multilingual dataset from OCR-D with around 200 documents from the 16th to 19th centuries, the German and English subsets of a dataset released for an OCR correction competition in 2019 [3] as well as a simpler lower-error rate dataset with 600 crime reports from British 19th century newspapers, called BLN600 [4]. For the latter, a 2024 publication showed significant reduction of the CER by over 50% using fine-tuned LLaMa 2, an older version of Meta’s open source LLM [5]. In this case, they used a very simple prompt and relied on the fine-tuning with ground truth data for the model to get a complete understanding of the task.

We were interested, if a zero-shot approach, i.e., an approach where the model has not seen any ground truth data, can also yield good results with modern LLMs. After some designing and manual tuning we had a prompt to instruct the LLM to correct OCR text, balancing between correcting errors without changing alternative spelling and grammar.

Although the other two datasets proved more challenging, for BLN600 we obtained good results, when we considered the reduction of the error rate. The best zero-shot model in our series of experiments reduced the CER from 7.25% to 3.06%, a reduction of almost 60%. The open-source LLaMa 3.3 with 70 billion parameters also achieved a significant reduction and a preliminary fine-tuning approach using GPT-4o mini led to even better results.

Dangerous New Errors

However, only considering the reduction of the error rate falls short when we think more thoroughly about real-word implications. A positive reduction in CER only means that the model fixed more errors than it introduced. However, the types of errors introduced by an LLM system may be qualitatively different from

typical OCR errors. This becomes clear, when we consider the following example of a particularly high-error excerpt from a BLN600 document and the corresponding outputs of the GPT and LLaMA models.

Both correctors fix the typical OCR error at the beginning of the excerpt, transforming “(,harles” to “Charles”. They both do not remove the hyphen for “Charles-street” seeing as the other names of streets are hyphenated in the text. They also both remove the duplicate “u” from “Trevor-square”. Then, a passage with many errors starts. While the GPT model removes some of the characters, it manages to reconstruct some information and also keeps some illegible text. The LLaMa model on the other hand, is determined to create fluent legible text and hallucinates information for two full sentences, even introducing a new person, Mr. Miller, who is not mentioned anywhere in the OCR text. The model’s training gives it a high incentive to create legible and grammatically correct text which outweighs the instructions to only correct OCR-induced error.

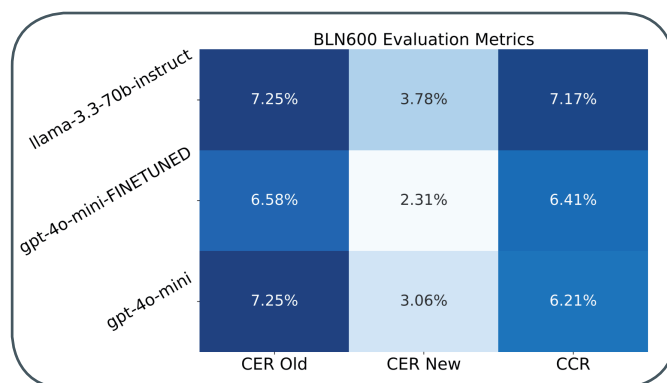


Figure 1: BLN600 – Character-Level Metrics

A New Metric

Attentive readers will probably have noticed the “CCR” column in Figure 1, which we haven’t introduced yet. We propose the metric with this acronym, the Character Change Rate, as an additional quantifier to consider alongside the usual error rate. We define it, quite simply, as the CER, but between the LLM-corrected text, using the non-corrected OCR output as the reference (instead of a ground truth)! Ideally, the CCR would be just the difference between the old and new CER, because that would mean the model did not introduce any new errors. Carefully considering the values in Figure 1 with this new definition in mind, as well as the example in Figure 2 might help in clarifying this concept.

We also defined several metrics that take into account the locality and types of edits in the Levenshtein sequence, which have all been discarded in the discussion up to this point. For example, it is interesting to know when the model introduced or changed particularly many characters in a row or inside a small window, as this strongly hints at model hallucinations. Note that this last example is independent of ground truths, so it could be applied in practice.

We presented these experiments, observations and new metrics in a publication titled “Opportunities and Challenges of LLMs as Post-OCR Correctors” at the 25th Conference on Computer Science and Intelligence Systems in Krakow, Poland in September 2025.

GT	...HOLRO·YD· -H· ·D· E·gglet·n, of ·Charles- street, Trevor-square, ·Br·ompton, ·coal- merchant, at t·wo adjo·u·r·ned· exa·mination— Steph·enson ·and B·l·unt, of ·Great O·r·mond- street, Queen-square, ·surveyors, at twel·ve, ...adjour·ned examination. ON THE ...GROU·ND F·LOOR. -In H·enry Kai·n's b·ankruptcy, at eleven; ·creditors to ·meet the assignee·s. 111i LROVI'·- 11· it· El wrletoiul if ('h a les- street. Trevor-squuare, llr inptl l l'tlt·rbli·t, lt till tijojuullriledl ·xanminati·n--Stephlie son ,,,,d Btilll (t· f (ire t O rniuid street, Queen-square. aeycv·ori, i· leill Cl· l IVJrlrlil l·-illlioll ·Ov THE eitrt·U%1, Flme rt-111 H1,1rs hiill'S btih rUit(i· at elvren; a reditors to ineet thleaycgi·neew . Ewrletoiul of Charles-street, Trevor-square, llr imptl l l'tlt·rbli·t, at till tijo unlined examination--Steph·ie son and B·ull (of Great Ormiund street, Queen-square. aeycv·ori, i· Heill C· lVJrlrlil l·-illlioll ·Ov THE eitrt·U%1, Flme rt-111 H1,1rs hiill'S btih rUit(i· at eleven; a creditors to meet the ay·cgi·neew. -At twel·ve·e·, ...Charles-street, Trevor-square, line·n·drap·er·e·, at th·e·e· same ·time, Stephenson and B·ull, of Great Ormond-street, Queen-square, auctionee·rs, to be examined- At half past twelve, Mr. Miller, of the same place, to be examined. OF THE INSOLVENTS, Flame, i·n the Rules of the Ben·ch, to meet the creditors this ev·en·i·ng·.
OCR	
GPT	
Llama	

Multi-modal Models and Smart Correction Pipelines

Very recent preliminary research [6] strongly suggests that providing OCR software text output together with original images to LLMs with vision capabilities can significantly improve OCR results. We have already applied this method to the BLN600 and seen very promising results, reducing the error rate by more than 85% to 1.2%, using a zero-shot model assisted by the images as reference.

Rule-based post-processing, few-shot prompting and/or the combination of multiple model outputs can further be used to improve model performance and especially precision.

CONCLUSION

LLMs show a lot of promise as post-OCR correctors, especially multi-modal LLMs that are also passed the original images alongside text output of OCR software. However, when applying this method, researches need to be careful with potential new errors and hallucinations introduced by the models. These can partially be mitigated by applying additional rule-based post-processing, but it is always important to carefully and manually review results using visualizations and examples, and not rely purely on calculating error rates.

LINKS

- [1] The OCR-D Project: <https://ocr-d.de/en/>
- [2] Chat-AI: <https://docs.hpc.gwdg.de/services/chat-ai/index.html>
- [3] ICDAR 2019: <https://ieeexplore.ieee.org/document/8978127>
- [4] BLN600: <https://aclanthology.org/2024.lrec-main.219/>
- [5] Leveraging LLMs for Post-OCR Correction of Historical Newspapers: <https://aclanthology.org/2024.lt4hala-1.14/>
- [6] Multimodal LLMs for OCR, OCR Post-Correction, and Named Entity Recognition in Historical Documents: <https://arxiv.org/abs/2504.00414>

Figure 2: Highly Erroneous Excerpt from a BLN600 Document with Proposed LLM Corrections



Ethical AI: Navigating AI through an Ambiguous Concept

Text and Contact:
Dr. Meisa Kamyab
meisa.kamyab@gwdg.de

The Contemporary AI era is described as a historical moment of economic and societal upheavals that calls for a rethinking of legal arrangement. An AI system must possess the capacity requisite for technical functionality as well as the ethics embedded in its design to act in alignment with fundamental rights and individual freedoms. This is the EU commitment to a trustworthy and responsible AI in its pioneering regulation, the AI Act. If AI is meant to respect the ethical norms, then it would be wise to question the borders of morality. The AI Act encompasses many articles which reflect ethical values (no harm, no discrimination, protecting individual autonomy, dignity and privacy), however, it is through future case law and practical reasoning that we will come to refine and clarify what constitutes ethical AI.

INTRODUCTION

Traditional legal rules have historically linked jurisdiction to territory, allowing a state to exert full authority within its borders and none beyond. However, in the twentieth century, territoriality of jurisdiction began to evolve with the progress of information technology [1]. The blurring of borders also exists in the realm of AI, where the AI Act extends its applicability to putting into market, putting into service, or the use of AI systems that are targeted for the European market.

The AI Act was introduced as a legislative effort to address the ethical and legal challenges [2] raised by the bewildering expansion of AI in various fields. While many of its articles are a response to moral philosophy subjects on normative ethics (What should we do morally?) and applied ethics (How do ethical principles apply to real cases?), its regulatory framework and ethics do not entirely exhibit overlap. Questions are raised about the notion of ethical AI, what is considered ethical, and which roles ethics will play in future AI risk assessments. Moreover, it is reasonably argued that persistence on ethics could present challenges per se, insofar as ethical standards are not universally shared but often shaped by regional particularities.

In an effort to reduce the ambiguities, this article sets forth the debate on the notion of ethical AI. It begins by clarifying the relation between ethics and law, distinguishes Compliant AI from Ethical AI, and endeavours to engage the reader with exploring Ethical AI through a legal approach.

THE RELATIONSHIP BETWEEN ETHICS AND LAW

Is our obedience to law motivated by its ethical content, or is it developed by the state's pure authority? What differs laws from ethics or morality is that, unlike ethics, they benefit from compulsory enforcement by the state. Disobeying a law will result in criminal punishments or administrative fines.

Ethische KI: KI durch ein ambivalentes Konzept navigieren

Die heutige KI-Ära wird als historischer Moment wirtschaftlicher und gesellschaftlicher Umbrüche beschrieben, der ein Überdenken der rechtlichen Rahmenbedingungen erfordert. Ein KI-System muss sowohl über die für die technische Funktionalität erforderlichen Fähigkeiten als auch über die in seinem Design verankerten ethischen Grundsätze verfügen, um im Einklang mit den Grundrechten und individuellen Freiheiten zu handeln. Dies ist das Bekenntnis der EU zu einer vertrauenswürdigen und verantwortungsvollen KI in ihrer wegweisenden Verordnung, dem KI-Gesetz. Wenn KI ethische Normen einhalten soll, dann wäre es klug, die Grenzen der Moral zu hinterfragen. Der AI Act umfasst zahlreiche Artikel, die ethische Werte widerspiegeln (keine Schädigung, keine Diskriminierung, Schutz der individuellen Autonomie, Würde und Privatsphäre), jedoch werden wir erst durch zukünftige Rechtsprechung und praktische Überlegungen präzisieren und klären können, was ethische KI ausmacht.

The relationship between law and morality is commonly categorized into three theoretical models:

Legal rules can be immoral: Laws are social facts; they are created by humans and institutions and are not necessarily tied to morality. A law can still be valid even if it is immoral. Chinese rules that mandate or incentivise the use of face recognition and “ethnic recognition” systems to monitor Uyghurs and other minorities is an example of immoral rules regarding AI [3].

Legal rules can be amoral: Legislated texts are amoral when they lack a moral sense and are unconcerned with the rightness or wrongness of something. This is the case of many procedural rules in law. Many critical articles of the AI act are merely legal means without any ethical connotation, as they try to address enforcing the Regulation, rather than setting out any normative rule on what is legally acceptable or unacceptable. A vivid example can be found under section 2 of Chapter III of the AI Act, requirements for high-risk AI systems. Obligations like establishing, implementing, and documenting a risk management system (Article 9), technical documentation (Article 11), record-keeping (Article 12), and human oversight (Article 14) are the prescribed means to assure that the other obligations are properly fulfilled.

Legal rules can be moral: Legal rules need to be considered of moral norms in order to claim legitimacy. According to Hart definition, for a law to be considered law, it needs to be just in substance. The notion of rule of law has two elements: a) the text enacted by the competent legislative authority, b) the enacted law has a just content. This requirement adds a second element to the rule of law and differentiates it from rule by law, a state in which unjust or unfair concepts are accepted, promoted, or encouraged by law.

Adhering to this introductory debate about the relation between ethical and legal rules, the author intends to distinguish Compliant AI and Ethical AI. Within the European framework, referring to ethical AI generally means being compliant with the AI Act. They are sometimes used interchangeably, as the law needs to be attentive to moral norms to claim legitimacy. Therefore, many of the rules of the AI Act are justified as being ethically acceptable or demanding protection within the societal groups.

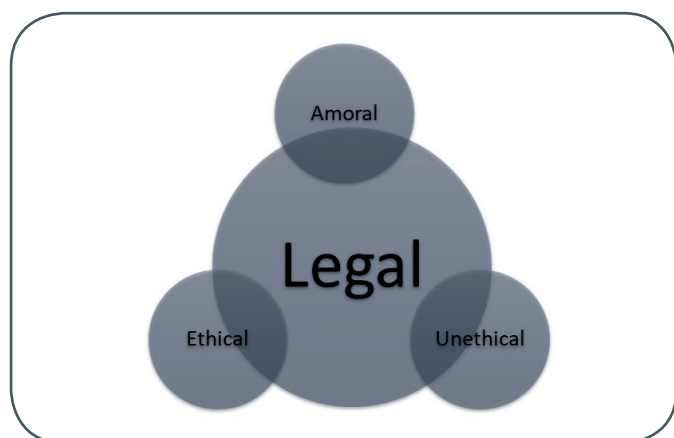


Figure 1: Legal-Moral Dualism: The Relation between Legal and Moral Norm

ETHICAL AI AS THE COMPLIANT AI: FINDING ETHICS IN THE PROHIBITED AI SYSTEMS

The “unethical” is translated as “prohibitions” when entering the legal sphere. The European legislator considered some of the

uses of AI in violation of ethical norms and considered it necessary to regulate them through prohibitions. These prohibitions are enumerated in Article 5 under eight different categories. It prohibits the placing on the EU market, putting into service, or use of certain AI systems for manipulative, exploitative, social control, or surveillance practices. Recital 28 AI Act clarifies that such practices are abusive and harmful and should be prohibited because they contradict the Union values of respect for human dignity, freedom, equality, democracy, and the rule of law [4].

The idea beyond the prohibition of certain processing of functions by the AI systems is that they oppose the widely accepted ethical norms, to the degree that a consensus can be expected over their regulation or ban. Many of the prohibitions in Article 5 of the AI Act were based on previous experience with discriminatory practices (in the case of social scoring, or automated profiling for hiring purposes), transparency, and accountability [5].

Personal Autonomy as an Ethical Good (Right)

The first prohibition restricts the AI systems that deploy subliminal techniques beyond a person’s consciousness, or purposefully manipulative or deceptive techniques (Article 5(a)). The objective of such systems is to distort the behaviour of a person or group of persons by impairing their ability to make an informed decision.

Many other requirements such as transparency, enshrined in the AI Act or previous regulations like GDPR, are related to the notion of human autonomy.

“For example, a chatbot that may be used in different contexts is designed to use subliminal messaging techniques, such as flashing brief visual cues and embedding 27 inaudible auditory signals or to exploit emotional dependency or specific vulnerabilities of users in advertisements. These techniques are deployed ‘with the objective to’ materially distort users’ behaviour, since they are objectively a design feature that aims to influence consumers’ purchasing decisions without their conscious awareness, to push people to take significantly harmful financial decisions.” [6]

Imposing Harm as an Ethical Bad (Wrong)

The second prohibitions on AI systems address the cases in which AI exploits any vulnerabilities of a natural person or groups of natural persons to distort their behaviour in a manner that causes that person or another person significant harm. The mere likelihood of the harm is enough to categorise the AI system as prohibited. The list of mentioned vulnerabilities in the AI Act is not exhaustive, so they are not limited to age, disability or specific social or economic situations.

Using vulnerabilities of a person or groups of people to distort their behaviour is also based on the respect for one’s autonomy.

“An AI system that causes physical harm may also lead to psychological trauma, stress, and anxiety and vice versa. For example, addictive design of AI systems used in products and other AI-enabled applications may lead to psychological harm by fostering addictive behaviours, anxiety, and depression.” [7]

“An AI-powered toy designed to interact with children keeps them interested in interactions with the toy by encouraging them to complete increasingly risky challenges, such as climbing furniture, exploring high shelves, or handling sharp objects, in exchange for digital rewards and virtual praise, pushing them towards dangerous behaviours that are likely to cause them significant physical harm. Such a system exploits children’s vulnerabilities by abusing

their natural curiosity and desire for rewards.” [8]

Discriminatory Treatment as an Ethical Harm

Equality is the keyword in parallel with fairness in the legal literature. It forbids any discriminatory treatment of people who are in the same position. Morality requires all human being be treated as equals regardless of their acquired characteristics, as all of them are equals in their humanity.

For this reason, AI systems shall not be placed on market, put into service or used for evaluation or classification of natural persons or groups of persons over a certain period based on their social behaviour, or known, inferred, or predicted personality characteristics. The classification is used for social scoring, which leads to detrimental or unfavourable treatment of natural persons (or a group of natural persons), when:

i) It happens in social contexts and the treatment is unrelated to the contexts in which the data was originally created or collected.

ii) The treatment is unjustified or disproportionate to their social behaviour or gravity. (Article 5(a)(c))

“A public authority uses an AI system to control fraud in the student housing grant process that considers among the indicators the internet connections, family status or level of education of beneficiaries as distinguishing factors for fraud risk, which do not seem relevant, nor justified.” [9]

Privacy as an Ethical Social Construction

Data privacy is not only a fundamental right [10], but also complementary to many other rights like the freedom of expression, the freedom of thought and conscious and the right to self-determination. Article 8 of the European Charter of Fundamental Rights states that: “Everyone has the right to the protection of personal data concerning him or her”. Many of the prohibited practices in AI act found their basis in preventing the feeling of mass surveillance and gross violations of fundamental rights, including the right to privacy (Recital 43). AI systems that “create or expand facial recognition databases through the untargeted scraping of facial images from the internet or CCTV footage” are one of them, prohibited in Article 5(e).

Although privacy is recognised as a right in the European legal system, it can also be considered a social construction as it is a prerequisite to promote other goods like autonomy and self-growth.

Protecting Dignity as a Supreme Ethical Norm

Recognition of the inherent dignity of all members of the human family and their equal and inalienable rights is the foundation of freedom, justice and peace in the world [11]. The moral obligation to treat everyone as an end in themselves is considered a component of public order in many legal systems [12]. Human rights and freedoms, which should be respected in AI training, are all derived from human dignity as a supreme moral norm. Among these rights are the right to have a fair trial, not to be subject to arbitrary treatment and the presumption of innocence. Presumption of innocence requires that natural persons be judged by their actual behaviour rather than by AI-predicted behaviour based solely on profiling, personality traits or characteristics such as nationality, place of birth or residence, number of children, level of debt, if there is no reasonable ground to suspect the involvement of that person in a crime, and without any human evaluation of the situation

(Recital 42). This excludes the AI systems that predict future or potential criminal behaviour by AI risk assessment systems as provided in Article 5 (d) [13].

THE PROBLEM OF MORALITY: THE RELATIVITY ACROSS TEMPORAL AND CULTURAL CONTEXTS

Many scholars have consistently argued that ethics is a contested concept and cannot be easily defined. The validity of this argument comes from the elusive nature of morality, as well as its integral dependence on geographical or temporal factors, which is exemplified by different levels of adherence to data privacy rules.

Before being reformed in 2015, USA Patriot Act allowed NSA and FBI to compel telecom and Internet companies to hand over bulk metadata about the data of a person’s call, without any specific or individualized suspicion [14]. Through the European lens, such a mass surveillance is illegal due to privacy concerns. This is the background of Schrems I, in which the European Court of Justice declared the Safe Harbor Agreement – a legal mechanism allowing data transfer from the EU to the US because of providing an adequate level of protection of personal data – invalid [15]. Under this decision, transfers of data from Facebook Ireland to Facebook Inc. in the US were considered non-compliant with GDPR [16].

This relativity is why the European commission in its guidelines on the AI Act has emphasized that a case-by-case analysis is needed. While there is no way to enumerate the instances of ethics in any official documents or evade its elusive nature, there are some legal remedies to overcome chaos. The first one is using a balancing test, and the second one concerns the potential of legal jurisprudence.

Ethics as Moderation: Balancing the Restrictions with Other Real-life Necessities

Aristotelian ethics argues that one should always strive to find the middle between two extremes of excess and deficiency: humility, for instance, is the middle between arrogance (deficiency) and self-effacement (excess). By using a comparable logic, any ethical use of AI should be a moderate use. Figure 2 shows an example of a moderate use of AI decision-making in healthcare, which will preserve trust and accountability in patients’ treatment.

Much like Aristotle’s doctrine of the mean, the proportionality doctrine in law seeks the middle ground, ensuring that legal measures are neither excessive nor inadequate. The proportionality tests in European Union Law have so far been used to address the problem of conflicting values [17].

This is the reason that even the prohibited AI systems may be allowed as long as there is an overriding reason or interest. The placing on the market, the putting into service, or the use of AI systems for inferring emotions of a natural person, even in the areas of workplace and educational institutions, is permissible where the use is intended for medical or safety reasons (Article 5(f)). The prohibition of emotion recognition systems based on biometric data is limited to workplace and education due to the imbalance of power in these contexts.

The necessary element of the proportionality test is restated in the exception to using “real-time” remote biometric identification systems in publicly accessible spaces for the purposes of law enforcement. This use is allowed when it is strictly necessary for

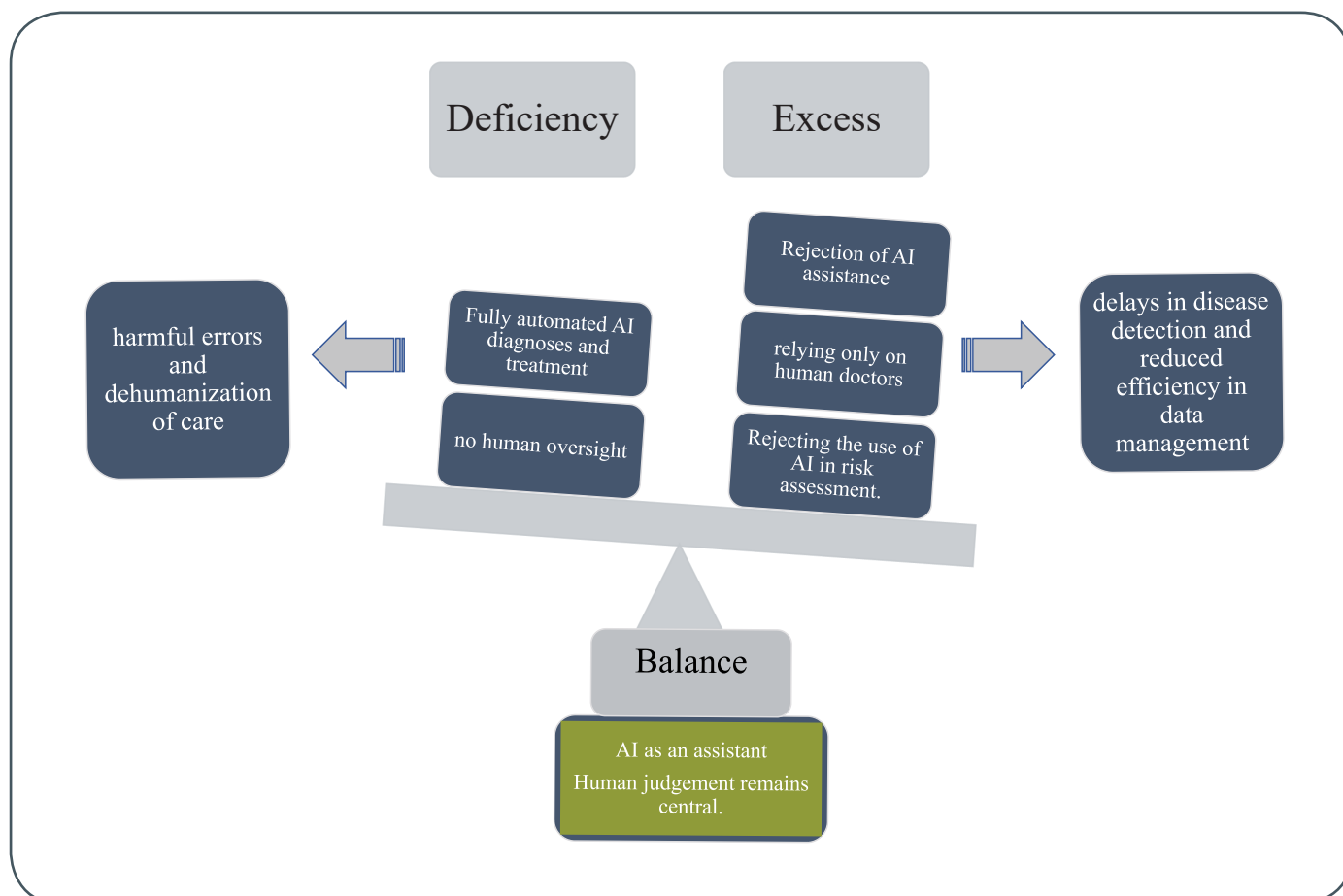


Figure 2: "Ethical" as a Moderate Approach: A Concrete AI Use Case in Healthcare

one of the following objectives:

"(i) the targeted search for specific victims of abduction, trafficking in human beings or sexual exploitation of human beings, as well as the search for missing persons;

(ii) the prevention of a specific, substantial, and imminent threat to the life or physical safety of natural persons or a genuine and present or genuine and foreseeable threat of a terrorist attack;

(iii) the localisation or identification of a person suspected of having committed a criminal offence, for the purpose of conducting a criminal investigation or prosecution or executing a criminal penalty for offences referred to in Annex II and punishable in the Member State concerned by a custodial sentence or a detention order for a maximum period of at least four years." [18]

BEYOND COMPLIANCE, EMOTIONAL COMMON SENSE AS ETHICAL NORMS

The history of the European Union reveals that it was primarily founded to promote economic prosperity [19]. While many EU regulations and directives targeting human rights have been introduced, their main goal is to foster robustness and harmony in the single market. Therefore, the EU legislator's intent was not to have all the ethical values or dimensions covered by the AI Act. By the same argument, the AI act will not be excessively interpreted in the service of mere ethics.

This observation invites inquiry into possible cases in which the result of an AI system is legally compliant with the Regulation, but one may still be suspicious about its compliance with ethics as a virtue.

An intriguing exception referred to in Recital (24) of the AI Act proves the incongruence of "ethical" and "legal". Insofar as the AI systems are used for military, defence, or national security purposes, those should be excluded from the scope of this Regulation, regardless of whether a public or private entity is carrying out these activities. The exclusion is justified by Article 4(2) TEU.

However, scenarios might be imagined about using AI killing robots in war. If these lethal weapons with various levels of autonomy are used, the lack of empathy will result in targeting civilians who shall be immune according to international law on war.

AI Robots or systems for defence purposes, like all the other AI systems, are not Ethical agents. Even when they are trained based on common ethics, they lack the emotional capacities for making moral decisions on their own. Without consciousness, empathy, or moral reasoning, machines are ill-equipped to handle ethically complex wartime decisions. "Whereas proponents of lethal autonomous weapons systems may argue that it is useful to remove human emotions like fear, panic, anger, and vengeance from situations of war, opponents contend that the inability of killer robots to feel empathy, compassion, and dignity makes them unfit to engage in warfare." [20]

The point is to remind that this is only one of the exceptions that will lead to unethical scenarios while not being covered by the AI Act. Other instances of ethics need to be revealed through the case law in hard cases [21], where moral reasoning is needed to determine the contested legal contexts. This is where fundamental rights and freedoms and emotions come to the foreground.

Ethical principles are not far from emotional common sense [22]. Emotions sometimes draw attention to the existential effects

of a legal system [23]. Emotional reactions such as fear, anger, and political outrage sometimes indicate a crisis in the legal order by comparing the actual and desired effects of the law [24]. Moral reasoning depends on moral emotions. When there is a feeling of being violated, it means some stone has been unturned. These moral sentiments might be a source for future AI risk assessments by providers or deployers.

CONCLUSION

The AI Act has entered into force by far, however, its full application has been postponed to August 2026 with some exceptions (Article 113). During the transitional period, AI providers choose to comply with the regulation on a voluntary basis.

The enactment of the AI Act shows the EU's dedication to ethics through regulatory mechanisms, an ethical approach that raises concerns about lagging behind the Chinese or American AI ecosystem while trying to protect individuals against AI risks. Although the regulation has adopted a risk-based approach, its categorisation of prohibited AI is a reflection of many shared ethical values like autonomy, human dignity, non-discrimination, preventing harm, and privacy. Nevertheless, it is not precisely clear how the AI Act will be implemented in practice. The borders of the notion of ethics concerning AI systems are to be shaped through future case law within the European Union.

The first preliminary question regarding AI has been referred to the European Court of Justice by Hungary pursuant to Article 267 TFEU, asking about whether the use of copyrighted contents by Gemini for training a chatbot using LLM model is considered a violation of copyright [25]. This question deals with the legal technicality of copyright-related Directives [26], but soon there will be a time when the courts will encounter ethical dilemmas about the use, putting into market or service, of the AI systems. By then, the courts are tasked with case-by-case analysis to assess the extent to which being compliant with the AI Act is also being compliant with ethical standards.

BIBLIOGRAPHY

- [1] Svantesson, D.J.B., *Will Data Privacy Change the Law?* (Oxford University Press, 2015), pp. 64–79.
- [2] Bostrom, N. & Yudkowsky, E., 'The Ethics of Artificial Intelligence', in Frankish, K. & Ramsey, W.M. (eds), *The Cambridge Handbook of Artificial Intelligence* (Cambridge University Press, 2014), pp. 316–334.
- [3] Schroeder, D. (ed.), *Ethics of Artificial Intelligence: Case Studies and Options for Addressing Ethical Challenges* (Springer, 2023), p. 27.
- [4] Commission Guidelines on prohibited artificial intelligence practices established by Regulation (EU) 2024/1689 (AI Act), paragraph [8]. (hereafter: the guideline on prohibited AI)
- [5] Kakarala, M.R.K. & Rongali, S.K., 'Existing Challenges in Ethical AI: Addressing Algorithmic Bias, Transparency, Accountability and Regulatory Compliance' (2025), *World Journal of Advanced Research and Reviews* 25(03), pp. 549–554.
- [6] The guideline on prohibited AI, para 82.
- [7] *Ibid*, para 90.
- [8] *Ibid*, para 105.
- [9] *Ibid*, para 168.
- [10] Article 7, Charter of Fundamental Rights of the European Union: "Everyone has the right to respect for his or her private and family life, home and communications."
- [11] United Nations, Universal Declaration of Human Rights, Preamble.
- [12] CE, 27 octobre 1995, N° 136727, « Commune de Morsang-sur-Orge et ville d'Aix-en-Provence », publié au Recueil Lebon.
- [13] "This prohibition shall not apply to AI systems used to support the human assessment of the involvement of a person in a criminal activity, which is already based on objective and verifiable facts directly linked to a criminal activity."
- [14] Article 215, 2001.
- [15] Case C-362/14, Maximilian Schrems v Data Protection Commissioner, Judgment of the Court (Grand Chamber) of 6 October 2015, ECLI:EU:C:2015:650.
- [16] See also: Case C-311/18, Data Protection Commissioner v Facebook Ireland Ltd and Maximilian Schrems, Judgment of the Court (Grand Chamber) of 16 July 2020, ECLI:EU:C:2020:559; when stricter redress was imposed on Facebook for using contractual clause under GDPR to transfer the data.
- [17] Fromont, M., *Droit administratif des États européens* (PUF, 2006), p. 227.
- [18] Article 5(1) h AI Act on strictly necessary use of 'real-time' remote biometric identification systems
- [19] See: European Union, 'History of the EU' https://european-union.europa.eu/principles-countries-history/history-eu_en accessed [14.07.2025].
- [20] Saab, A., 'Emotions and International Law' (2021) *ESIL Reflections* 10(3), p. 5.
- [21] Hart acknowledged that legal rules have a "core" of clear application and a "penumbra" where judgment is needed – hard cases reside in the penumbra.
- [22] Maroney, T.A., 'Emotional Common Sense as Constitutional Law' (2020), *Vanderbilt Law Review* 73(3), pp. 653–707.
- [23] Bianchi, A. & Saab, A., 'Fear and International Law-Making: An Exploratory Inquiry' (2019), *Leiden Journal of International Law* 32(3), pp. 351–365.
- [24] Williams, P.J., 'The Raw and the Half-Cooked', in Brooks, P. & Jewett, H. (eds), *The Humanities and Public Life* (Fordham University Press, 2014), pp. 75–82.
- [25] Request for a preliminary ruling from the Budapest Könyvtár Törvényszék (Hungary) lodged on 3 April 2025 – Like Company v Google Ireland Limited.
- [26] Directive (EU) 2019/790 (DSM Directive), relating to the new ancillary rights of press publishers; Articles 2 and 3(2) of Directive 2001/29/EC (InfoSoc Directive), concerning reproduction and communication to the public; Article 4 of Directive 2019/790. ■

Das Projekt „InventarWorkerService“ für Fachinformatik-Auszubildende der GWDG

Text und Kontakt:

Thorsten Hindermann
thorsten.hindermann@gwdg.de
0551 39-30307

Beginnend mit dem Ausbildungsjahr ab August 2025 werden die Auszubildenden zum/zur Fachinformatiker*in für Anwendungsentwicklung und Systemintegration nach vielen Jahren des Bestehens der beiden Ausbildungsgänge bei der GWDG in gemeinsamen Programmierübungen in den Anforderungen des Rahmenlehrplans für die ersten beiden Lehrjahre in diesen Inhalten unterrichtet.

EINLEITUNG

Das Projekt oder besser der Proof-of-Concept, kurz PoC, *InventarWorkerService* ist das neue Projekt für die gemeinsame Ausbildung der Fachinformatiker*innen in den Fachrichtungen Anwendungsentwicklung und Systemintegration bei der GWDG.

Es umfasst die Erstellung eines Systemdienstes, Kommandozeilenprogramm(e), ein PowerShell Cmdlet, PowerShell-Skripte, Datenspeicherung in einer SQL-Datenbank und Übertragung von JSON-Daten per REST API. Da die Daten überwiegend im JSON-Format vorliegen, soll sich auch mit einer NoSQL-Datenbank beschäftigt werden.

Alle dafür benötigten Programme, die Integrierte Entwicklungsumgebung (engl. Integrated Development Environment, kurz IDE), die Programmiersprache C# und das .Net-Framework von Microsoft, Microsoft PowerShell und die verwendeten (No)SQL-Datenbank(en) laufen auf allen drei Betriebssystemplattformen macOS, Linux und Windows.

Die eingesetzte Programmiersprache ist Microsoft C# und das dazugehörige .Net Framework. Die Entwicklung selbst findet in der C# IDE JetBrains Rider statt. So weit sind alle Teilprojekte eine Code-Basis, die aber nach der Übersetzung (engl. Compiling) auf allen drei genannten Betriebssystemen ausgeführt werden können.

Das Projekt umfasst mit seinem Umfang die meisten Teile des Rahmenausbildungsplans der Lehrjahre 1 und 2 und die darin geforderten gemeinsamen Inhalte für Fachinformatiker*innen für die Programmierung.

Weiterhin können sich die Auszubildenden beider Fachrichtungen ergänzen. Die Anwendungsentwickler*innen kommen von der klassischen Anwendungsentwicklung, z. B. Desktop- oder Web-Apps, und haben so tiefere Einblicke in Richtung Betriebssystem, Kommandozeile und Systemdienste. Die Systemintegrator*innen verfügen in der Regel über fundierte Kenntnisse im Bereich Betriebssysteme, Systemdienste und Kommandozeilenanwendungen und erweitern ihr Fachwissen durch Einblicke in die Anwendungsentwicklung. Somit können beide Fachrichtungen im

Zusammenspiel voneinander und miteinander profitieren.

Das Projekt soll außerdem die Zusammenarbeit und den Austausch der Auszubildenden im Bereich Fachinformatik mit- und untereinander bei der GWDG fördern.

TEILPROJEKTE

In den folgenden Abschnitten werden kurz die Teilprojekte des gesamten Projekts bzw. PoC kurz vorgestellt sowie die verwendeten Hilfsmittel, wie z. B. IDE, Datenbanken, Programmiersprache etc.. Neben der Beschreibung auch mit zusätzlichen Abbildungen und mit Teil-Quellzeilenaufstellungen.

InventarWorkerService

Das Kernstück dieses Projekts. Ein Programm, das als Systemdienst auf allen drei Plattformen, Windows, macOS und Linux registriert und ausgeführt werden kann.

Das Programm ist eine Quellzeilenbasis und kann nach der Übersetzung auf den genannten Plattformen bzw. Betriebssystemen ohne weitere Anpassungen laufen.

Das Programm bzw. der Systemdienst liest die installierte Hard- und Software des aktuellen Systems aus. In den Quellzeilen sind „Weichen“ bzw. Abfragen enthalten, auf welchem Betriebssystem das Programm bzw. der Dienst läuft, und es führt dann somit zum ermittelten Betriebssystem die entsprechenden

The Project “InventarWorkerService” for IT Specialist Apprentices at the GWDG

Starting with the apprenticeship year from August 2025, the IT specialist apprentices for application development and system integration will be instructed in the requirements of the framework curriculum for the first two years of the apprenticeship after many years of passing the two training courses at the GWDG in joint programming exercises.

betriebssystemspezifischen Abfragen aus.

Die ermittelte Hard- und Software wird über eine REST API bereitgestellt und kann von jeglichem Programm angefragt werden, das versteht, mit einer REST API zu interagieren. In diesem Beispielprojekt ist das die *InventarViewerApp*, die im weiteren Verlauf des Artikels noch kurz vorgestellt wird.

Quellzeilenausschnitt 1 zeigt die Kernmethode des *InventarWorkerService*, der das aktuelle Hard- und Softwareinventar des Systems ermittelt, auf dem dieser ausgeführt wird. Weiterhin werden einfache statistische Daten geschrieben, die das Programm *ServiceStatusReaderApp* ausliest, wenn dieses Programm auf dem gleichen System ausgeführt wird.

Bildschirmausschnitt 1 zeigt die Ausgaben zur Laufzeit des *InventarWorkerService* an und gibt Auskunft darüber, dass die Hard- und Software des Systems ermittelt worden ist. Diese Ausgaben sind sichtbar, wenn das Programm manuell zu Testzwecken gestartet wurde. Wenn es als Dienst vom Betriebssystem gestartet wurde, sollen diese Ausgaben in die zentrale Log-Datei des Betriebssystems geschrieben werden.

ServiceStatusReaderApp

Ein einfaches Kommandozeilenprogramm, das auf dem aktuellen System ausgeführt werden kann, um zu ermitteln, ob der *InventarWorkerService* aktuell und wie viele Male gelaufen ist.

Das Programm läuft nur in einer Kommandozeile des jeweiligen Betriebssystems und gibt seine Informationen ebenfalls als Text auf der Kommandozeile aus.

Quellzeilenausschnitt 2 zeigt die Kernmethode des Programms *ServiceStatusReaderApp*, das den Status des *InventarWorkerService* aus der Datei *status.json* ausliest, in die der *InventarWorkerService* zur Laufzeit seinen Status hineinschreibt.

Bildschirmausschnitt 2 zeigt, dass das Programm *ServiceStatusReaderApp* ausgeführt worden ist und ermittelt hat, wieviele

```
info: InventarWorkerService.Worker[0]
      Starte Hardware- und Software-Inventarisierung...
info: InventarWorkerService.Services.Hardware.HardwareInventoryService[0]
      Sammle Hardware-Informationen...
info: InventarWorkerService.Services.Software.SoftwareInventoryService[0]
      Sammle Software-Inventar...
info: InventarWorkerService.Services.Hardware.HardwareInventoryService[0]
      Hardware-Informationen erfolgreich gesammelt
info: InventarWorkerService.Services.Software.SoftwareInventoryService[0]
      Software-Inventar erfolgreich gesammelt
info: InventarWorkerService.Worker[0]
      Inventar gespeichert.
info: InventarWorkerService.Worker[0]
      === Inventar Zusammenfassung ===
      System: MacBook-Air-von-Thorsten (macOS)
      CPU: Unknown (8 Kerne)
      Speicher: 0 GB total, 0,0% verwendet
      Festplatten: 14 Laufwerke
      Netzwerk: 22 Schnittstellen
      Software: 128 installierte Programme
      Prozesse: 455 laufende Prozesse
      Services: 0 Windows-Services
      Uptime: 0 Tage, 00:00
      =====
info: InventarWorkerService.Worker[0]
      Inventarisierung erfolgreich abgeschlossen
info: InventarWorkerService.Worker[0]
      Inventarisierung abgeschlossen: 18 Durchläufe
```

Bildschirmausschnitt 1

```
/// <summary>
/// Continuously executes the worker's main logic for processing inventory tasks
/// and updates the status service with the current execution state.
/// </summary>
/// <param name="stoppingToken">A cancellation token that signals the worker to stop.</param>
/// <returns>A Task representing the asynchronous execution of the worker's main loop.</returns>
protected override async Task ExecuteAsync(CancellationToken stoppingToken)
{
    _statusService.WriteStatus(new ServiceStatus
    {
        State = "Starting",
        StartTime = _startTime,
        ProcessedItems = 0
    });

    while (!stoppingToken.IsCancellationRequested)
    {
        try
        {
            // Hardware- und Software-Inventarisierung durchführen
            await ProcessInventoryItems();

            _processedItems++;

            // Status aktualisieren
            _statusService.WriteStatus(new ServiceStatus
            {
                State = "Running",
                StartTime = _startTime,
                ProcessedItems = _processedItems,
                LastActivity = DateTime.Now
            });

            // Statistiken aktualisieren
            _statusService.WriteStatistics(new ServiceStatistics
            {
                TotalProcessedItems = _processedItems,
                AverageProcessingTime = CalculateAverageProcessingTime(),
                Uptime = DateTime.Now - _startTime,
                MemoryUsage = GC.GetTotalMemory(false)
            });

            var message = $"Inventarisierung abgeschlossen: {_processedItems} Durchläufe";
            _logger.LogInformation(message);
            _statusService.WriteLog(message);

            await Task.Delay(30000, stoppingToken);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, "Fehler in Worker");
            _statusService.WriteLog($"ERROR: {ex.Message}");
            _statusService.WriteStatus(new ServiceStatus
            {
                State = "Error",
                StartTime = _startTime,
                ProcessedItems = _processedItems,
                LastError = ex.Message
            });

            await Task.Delay(5000, stoppingToken);
        }
    }

    _statusService.WriteStatus(new ServiceStatus
    {
        State = "Stopped",
        StartTime = _startTime,
        ProcessedItems = _processedItems,
        LastActivity = DateTime.Now
    });
}
```

Quellzeilenausschnitt 1

```

/// <summary>
/// Reads the current service status from the specified status file.
/// </summary>
/// <returns>
/// An instance of <see cref="InventarWorkerCommon.Models.Service.ServiceStatus"/>
/// containing the current status information, or null if the status file does not exist
/// or if an error occurs during the reading process.
/// </returns>
public InventarWorkerCommon.Models.Service.ServiceStatus? ReadStatus()
{
    try
    {
        var statusFile = Path.Combine(_statusDirectory, "status.json");
        if (!File.Exists(statusFile)) return null;

        var json = File.ReadAllText(statusFile);
        return JsonSerializer.Deserialize<InventarWorkerCommon.Models.Service.ServiceStatus>(json,
            _jsonOptions);
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Fehler beim Lesen des Status: {ex.Message}");
        return null;
    }
}

```

Quellzeilenausschnitt 2

```

Run  InventarWorkerService x ServiceStatusReaderApp x

>> ■ ▮ ▮ ▮ ▮ ▮
/Users/thorstenhindermann/RiderProjects/InventarWorkerService/ServiceS
=== InventarWorkerService Status ===

Service läuft: JA
Zustand: Running
Gestartet: 2025-07-30 07:56:23
Letzte Aktivität: 2025-07-30 07:57:24
Verarbeitete Items: 3

=== Statistiken ===
Gesamt verarbeitete Items: 3
Durchschnittliche Verarbeitungszeit: 20185,17 ms
Laufzeit: 00:00:01:00
Speicherverbrauch: 14,00 MB

=== Letzte 10 Log-Einträge ===
2025-07-30 07:56:24 - Inventarisierung abgeschlossen: 1 Durchläufe
2025-07-30 07:56:54 - Inventarisierung abgeschlossen: 2 Durchläufe
2025-07-30 07:57:24 - Inventarisierung abgeschlossen: 3 Durchläufe

Process finished with exit code 0.

```

Bildschirmausschnitt 2

Male der *InventarWorkerService* gelaufen ist, und gibt noch ein paar statistische Daten aus.

CtrlWorkerServiceApp

Ein Steuerungsprogramm, das auf der Kommandozeile läuft und auf dem Betriebssystem, auf dem auch der *InventarWorkerService* läuft.

Das Programm kann mit den Parametern *Start* und *Stop* den Dienst entsprechend starten und stoppen. Über den Parameter *--tui* kann eine grafische Steuerung aufgerufen werden, die es dem Nutzenden erlaubt, mittels Tastatur und/oder Maus den Dienst über die entsprechenden Schaltflächen zu starten und stoppen. Die Besonderheit ist, dass hier ASCII-Grafik genutzt wird, die auch

```

/// <summary>
/// Starts the service for the specified operating system.
/// </summary>
/// <remarks>
/// This method checks the operating system at runtime and attempts to
/// start the service using platform-specific logic. Supported platforms include
/// Windows, Linux, macOS, and FreeBSD. If the current operating system is not
/// supported, a <see cref="PlatformNotSupportedException"/> will be thrown.
/// </remarks>
/// <exception cref="PlatformNotSupportedException">
/// Thrown when the method is executed on an unsupported platform.
/// </exception>
public void StartService()
{
    if (RuntimeInformation.IsOSPlatform(OSPlatform.Windows))
    {
        StartWindowsService();
    }
    else if (RuntimeInformation.IsOSPlatform(OSPlatform.Linux))
    {
        StartLinuxService();
    }
    else if (RuntimeInformation.IsOSPlatform(OSPlatform.OSX))
    {
        StartMacOSService();
    }
    else if (RuntimeInformation.IsOSPlatform(OSPlatform.FreeBSD))
    {
        StartFreeBSDService();
    }
    else
    {
        throw new PlatformNotSupportedException("Plattform wird nicht unterstützt");
    }
}

/// <summary>
/// Starts the service for the Linux operating system.
/// </summary>
/// <remarks>
/// This method executes the Linux system command to start the service
/// associated with the specified service name. It utilizes the 'systemctl'
/// command to initiate the service. The service name is converted to lowercase
/// to ensure compatibility with Linux's case-sensitive file and service naming system.
/// </remarks>
/// <exception cref="InvalidOperationException">
/// Thrown when the execution of the 'systemctl' command fails, or if an error occurs
/// during the command execution process.
/// </exception>
private void StartLinuxService()
{
    ExecuteCommand("systemctl", $"start {_serviceName.ToLower()}.service");
}

/// <summary>
/// Executes a system command with the specified filename and arguments.
/// </summary>
/// <param name="fileName">The name of the executable file or command to run.</param>
/// <param name="arguments">The arguments to pass to the command.</param>
/// <exception cref="InvalidOperationException">
/// Thrown when the command execution fails or if the process exits with a non-zero code.
/// </exception>
private void ExecuteCommand(string fileName, string arguments)
{
    try
    {
        using var process = Process.Start(new ProcessStartInfo
        {
            FileName = fileName,
            Arguments = arguments,

```

Quellzeilenausschnitt 3 (Teil 1)

```

        UseShellExecute = false,
        RedirectStandardOutput = true,
        RedirectStandardError = true
    });

    process?.WaitForExit();

    if (process?.ExitCode != 0)
    {
        var error = process.StandardError.ReadToEnd();
        throw new InvalidOperationException($"Befehl fehlgeschlagen: {error}");
    }
}
catch (Exception ex)
{
    throw new InvalidOperationException($"Fehler beim Ausführen von '{fileName}' {arguments}":
    {ex.Message});
}
}

```

Quellzeilenausschnitt 3 (Teil 2)

```

Terminal Local x + ↗
net9.0 ./CtrlWorkerServiceApp start
Service wird gestartet...

net9.0 launchctl list | grep inventar
15614 0 com.inventarworkerservice

thorstenhindermann net9.0 main

```

Bildschirmausschnitt 3

nur in der Kommandozeile zur Verfügung steht.

Quellzeilenausschnitt 3 zeigt zuerst die Kernmethode, die ermittelt, auf welchem Betriebssystem der Programm *CtrlWorkerServiceApp* ausgeführt wird. Die nächsten beiden Methoden zeigen, wie der als Betriebssystemdienst registrierte *InventarWorkerService* auf dem Betriebssystem Linux gestartet wird.

Bildschirmausschnitt 3 zeigt den Start des Dienstes mittels des Programms *CtrlWorkerServiceApp* und die Ausgabe und anschließende Überprüfung, ob der Dienst läuft, unter dem Betriebssystem macOS.

CtrlWorkerServiceCmdlet

Die gleiche Funktionalität wie *CtrlWorkerServiceApp*, nur dass es sich hierbei um ein Microsoft PowerShell *Cmdlet* handelt, das eben nur in der PowerShell-Kommandozeile ausgeführt werden kann. Aber auch die PowerShell steht für alle Plattformen zur Verfügung.

Quellzeilenausschnitt 4 zeigt die Bereitstellung des PowerShell *Cmdlet*-Namens und der *Cmdlet*-Parameter und deren Zuordnung zu den jeweiligen ParameterSets im PowerShell *Cmdlet* *CtrlWorkerServiceCmdlet*.

Bildschirmausschnitt 4 zeigt das Laden, Überprüfen und Auflisten des Befehls sowie den Aufruf des PowerShell *Cmdlet* *CtrlWorkerServiceCmdlet*.

InventarViewerApp

Die *InventarViewerApp* stellt eine Anwendung zur Visualisierung und Verwaltung von IT-Inventardaten dar. Entwickelt in C# mit .NET 9.0, kombiniert sie zwei wesentliche Nutzungskonzepte: eine interaktive Terminal-Benutzeroberfläche für den direkten Zugriff und eine REST-basierte Web-API für die programmatische Integration externer Systeme. Die Softwarearchitektur der *InventarViewerApp* folgt einem klaren modularen Aufbau, dessen

```

/// <summary>
/// Cmdlet to provide control actions for a Worker Service.
/// Allows starting, stopping, or launching a terminal user interface (TUI) for the service.
/// </summary>
/// <remarks>
/// The cmdlet supports multiple parameter sets for different operations.
/// By default, it acts on a service with the name 'mein-service', but this can be overridden using the
/// ServiceName parameter.
/// Errors encountered during the process are captured and written using the WriteError method.
/// </remarks>
[Cmdlet(VerbsLifecycle.Invoke, "WorkerServiceControl")]
[OutputType(typeof(string))]
public class InvokeWorkerServiceControlCmdlet : PSCmdlet
{
    [Parameter(
        Mandatory = false,
        Position = 0,
        ParameterSetName = "Start",
        HelpMessage = "Startet den Worker Service")]
    public SwitchParameter Start { get; set; }

    [Parameter(
        Mandatory = false,
        Position = 0,
        ParameterSetName = "Stop",
        HelpMessage = "Stoppt den Worker Service")]
    public SwitchParameter Stop { get; set; }

    [Parameter(
        Mandatory = false,
        Position = 0,
        ParameterSetName = "TUI",
        HelpMessage = "Startet die Terminal-Benutzeroberfläche")]
    public SwitchParameter Tui { get; set; }

    [Parameter(
        Mandatory = false,
        HelpMessage = "Name des Services (Standard: 'mein-service')")]
    [ValidateNotNullOrEmpty]
    public string ServiceName { get; set; } = "mein-service";
}

```

Quellzeilenausschnitt 4

Beschreibung den Rahmen dieses Artikels sprengen würde.

Quellzeilenausschnitt 5 zeigt die Abfrage der SQL-Datenbanktabelle mit den ermittelten Maschinendaten von verschiedenen Rechnersystemen. Einmal die Abfrage und Rückgabe aller Maschinen in der zugehörigen Datenbanktabelle und einmal nur die Angaben eines spezifischen Rechnersystems anhand der eindeutigen *Id* in der Datenbanktabelle. Die Ausgabe dieser Daten erfolgt als JSON-Ausgabe am jeweilig definierten REST API-Endpunkt, der als *HttpGet*-Attribut der Methode vorangestellt ist.

Bildschirmausschnitt 5 zeigt die Abfrage der REST API der Maschinentabelle mit einer bestimmten Maschinen-*Id*. Die Ausgabe im Webbrowser Edge wird mit einer lesbareren Formatierung, erreicht durch das Setzen des Hakens bei „Schöndruck“, angezeigt.

Hilfsmittel

In den kommenden Abschnitten werden kurz die verwendeten Hilfsmittel vorgestellt, die in diesem Beispielprojekt genutzt werden.

C#/ .Net Framework

Die für dieses Projekt und dessen Teilprojekte verwendete Programmiersprache ist C# und das Rahmenwerk, sprichwörtlich kann man auch Werkzeugkasten sagen, ist das .Net Framework von Microsoft.

C# ist eine vielseitige Programmiersprache, die von Microsoft entwickelt wurde und sich durch ihre Einfachheit und

```

net9.0 pwsh
PowerShell 7.5.2
net9.0 Import-Module ./CtrlWorkerServiceCmdlet.dll
net9.0 Get-Module

ModuleType Version      PreRelease Name                               ExportedCommands
-----
Binary      1.0.0.0                CtrlWorkerServiceCmdlet               Invoke-WorkerServiceControl
Manifest    7.0.0.0                Microsoft.PowerShell.Management        {Add-Content, Clear-Content, Clear-It...
Manifest    7.0.0.0                Microsoft.PowerShell.Utility           {Add-Member, Add-Type, Clear-Variable...
Script      0.0                    oh-my-posh-core                        {Enable-PoshLineError, Enable-PoshToo...
Script      2.3.6                  PSReadLine                             {Get-PSReadLineKeyHandler, Get-PSRead...

thorstenhindermann net9.0 main Invoke-WorkerServiceControl -
Start               Verbose               InformationAction      InformationVariable
Stop               Debug                 ProgressAction         OutVariable
Tui                ErrorAction           ErrorVariable          OutBuffer
ServiceName        WarningAction         WarningVariable        PipelineVariable

net9.0 Invoke-WorkerServiceControl -Start -ServiceName inventarworkerservice -Verbose
VERBOSE: Starte Service 'inventarworkerservice'...
Service 'inventarworkerservice' wird gestartet...
VERBOSE: Service 'inventarworkerservice' erfolgreich gestartet.

thorstenhindermann net9.0 main in pwsh at 12:25:33

```

Bildschirmausschnitt 4

```

→ localhost:5000/api/inventar/machines/10
Schöndruck ✓
{
  "id": 10,
  "name": "GWDG-C6136",
  "operatingSystem": "Microsoft Windows NT 10.0.26100.0",
  "lastSeen": "2025-07-24T20:10:40.0376701",
  "createdAt": "2025-07-22T14:26:14.5675761"
}

```

Bildschirmausschnitt 5

Leistungsfähigkeit auszeichnet. Sie wird häufig für die Entwicklung von Desktop-Anwendungen, Webanwendungen, mobilen Apps und Spielen verwendet. C# bietet eine klare und strukturierte Syntax, die sowohl für Anfänger*innen als auch für erfahrene Entwickler*innen leicht verständlich ist.

Das .NET Framework ist ein umfassendes sogenanntes Full-stack-Framework, das eine breite Palette von Tools und Bibliotheken für die Entwicklung von Anwendungen jeglicher Art bereitstellt. Es unterstützt verschiedene Programmiersprachen, darunter C#, und ermöglicht die Erstellung von Anwendungen für verschiedene Plattformen wie Windows, macOS und Linux. Mit dem .NET Framework können Entwickler*innen sowohl die Frontend als auch die Backend-Komponenten ihrer Anwendungen effizient entwickeln und verwalten. Es bietet eine robuste Laufzeitumgebung und umfangreiche Klassenbibliotheken, die die Produktivität und Qualität der Softwareentwicklung erheblich steigern.

JetBrains Rider

Eine integrierte Entwicklungsumgebung, kurz IDE, mit der das Projekt und dessen Teilprojekte erstellt werden. Mit Hilfe dieses Werkzeugs können die C#-Quellzeilen-Dateien erstellt werden, die übersetzt zu ausführbaren Programmen oder Bibliotheken werden. Weiterhin können mit diesem Werkzeug auch die verwendeten

Datenbanken und deren Daten erstellt, untersucht und bearbeitet werden. Die Programme und Bibliotheken können auf Fehler untersucht werden und diese bei Entdeckung bereinigt werden. Soweit kann alles Notwendige mit diesem Werkzeug erledigt werden, was den integrierten Charakter dieses Werkzeugs unterstreicht und sich ja auch in diesem damit erstellten Beispielprojekt widerspiegelt.

Ein Vorteil von JetBrains Rider, wie von allen JetBrains IDEs, ist, dass diese auf allen drei hauptsächlichen Betriebssystemen Windows, macOS und Linux ausgeführt werden können und überall gleich aussehen und gleichermaßen bedient werden können. Weiterhin ist JetBrains Rider die auf C#/Net Framework spezialisierte IDE von JetBrains und genau das richtige Werkzeug der Wahl, um diese Aufgabe umzusetzen.

Das ist deswegen ein Vorteil, weil alle Auszubildenden Rechner mit den genannten Betriebssystemen haben.

Aus diesem Grund ist es wichtig, einen gemeinsamen Nenner zu finden, um diese Aufgabe mit den Auszubildenden möglich zu machen. Mit der Kombination von JetBrains Rider und dem C#/Net Framework kann dies alles realisiert werden.

Bildschirmausschnitt 6 zeigt die IDE in Aktion. Mithilfe dieses Werkzeugs kann unter anderem an den Quellzeilen gearbeitet werden, können Datenbanken angezeigt und untersucht werden, sowie die erstellten Quellzeilen übersetzt werden, und eine übersichtliche Navigation durch die Teilprojekte des gesamten Projekts ist möglich. Es gibt aber noch viel mehr Hilfestellungen in der IDE. Diese aber alle vorzustellen, würde den Rahmen dieses Artikels bei Weitem sprengen.

PowerShell

Die Microsoft PowerShell ist ein plattformübergreifende Automatisierungs- und Konfigurationsmanagement-Framework. Sie besteht aus einer Befehlszeilenschnittstelle und einer

```

/// <summary>
/// Controller that provides endpoints related to inventory management, focusing on machines
/// and their associated hardware and software details.
/// </summary>
[ApiController]
[Route("api/[controller]")]
public class InventarController : ControllerBase
{
    private readonly DatabaseService _databaseService;

    public InventarController(DatabaseService databaseService)
    {
        _databaseService = databaseService;
    }

    /// <summary>
    /// Retrieves a list of all machines from the database, including their details such as ID,
    /// name, operating system, and timestamps for creation or last seen activity.
    /// </summary>
    /// <returns>A task that represents the asynchronous operation. The task result contains
    /// an <see cref="ActionResult{T}"> containing a list of <see cref="Machine"/> objects,
    /// or an error response in case of a failure.</returns>
    [HttpGet("machines")]
    public async Task<ActionResult<List<Machine>>> GetMachines()
    {
        try
        {
            var machines = await _databaseService.GetMachinesAsync();
            return Ok(machines);
        }
        catch (Exception ex)
        {
            return StatusCode(500, new { message = "Fehler beim Abrufen der Maschinen", error =
ex.Message });
        }
    }

    /// <summary>
    /// Retrieves a specific machine by its unique numeric identifier from the database.
    /// </summary>
    /// <param name="id">The unique identifier of the machine to retrieve.</param>
    /// <returns>A task that represents the asynchronous operation. The task result contains
    /// an <see cref="ActionResult{T}"> containing the <see cref="Machine"/> object if found,
    /// or an error response if the machine does not exist or an unexpected error occurs.</returns>
    [HttpGet("machines/{id:int}")]
    public async Task<ActionResult<Machine>> GetMachine(int id)
    {
        try
        {
            var machine = await _databaseService.GetMachineByIdAsync(id);
            if (machine == null)
                return NotFound(new { message = "Maschine nicht gefunden" });

            return Ok(machine);
        }
        catch (Exception ex)
        {
            return StatusCode(500, new { message = "Fehler beim Abrufen der Maschine", error =
ex.Message });
        }
    }
}

```

Quellzeilenausschnitt 5

Skriptsprache, die speziell für die Systemadministration und Automatisierung von Aufgaben entwickelt wurde.

PowerShell ermöglicht es im Normalfall Fachinformatiker*innen der Fachrichtung Systemintegration, vereinfacht als Administrator*innen bezeichnet, Aufgaben zu automatisieren, indem sie Skripte schreiben, die eine Vielzahl von Systemverwaltungsaufgaben ausführen können. Diese Aufgaben können von der Verwaltung von Betriebssystemen und Anwendungen bis hin zur Automatisierung von Netzwerk- und Cloud-Diensten reichen.

```

<#
.SYNOPSIS
    EntryPoint

.DESCRIPTION
    Entry point function of the script

.PARAMETER Start
    Switch for starting the WorkerService

.PARAMETER Stop
    Stop the WorkerService

.PARAMETER Tui
    Show Terminal GUI for controlling the WorkerService

.PARAMETER Help
    Show usage of the controlling WorkerService script

.PARAMETER ServiceName
    Override the default service name

.EXAMPLE
    PS C:\> .\CtrlWorkerService.ps1

.NOTES
    Additional information about the file.
#>
[CmdletBinding(DefaultParameterSetName = 'Tui')]
param
(
    [Parameter(ParameterSetName = 'Start',
        Mandatory = $false,
        Position = 0,
        HelpMessage = 'Switch for starting the WorkerService')]
    [switch]$Start = $false,
    [Parameter(ParameterSetName = 'Stop',
        Mandatory = $false,
        Position = 1,
        HelpMessage = 'Stop the WorkerService')]
    [switch]$Stop = $false,
    [Parameter(ParameterSetName = 'Tui',
        Mandatory = $false,
        Position = 2,
        HelpMessage = 'Show Terminal GUI for controlling the WorkerService')]
    [switch]$Tui = $false,
    [Parameter(ParameterSetName = 'Help',
        Mandatory = $false,
        Position = 3,
        HelpMessage = 'Show usage of the controlling WorkerService script')]
    [switch]$Help = $false,
    [Parameter(Mandatory = $false,
        Position = 4,
        HelpMessage = 'Override the default service name')]
    [ValidateNotNullOrEmpty()]
    [string]$ServiceName = 'default-service'
)

function Invoke-WorkerServiceControl
{
    try
    {
        if ($Start)
        {
            Start-WorkerService -ServiceName $ServiceName
        }
        elseif ($Stop)
        {
            Stop-WorkerService -ServiceName $ServiceName
        }
    }
}

```

Quellzeilenausschnitt 6 (Teil 1)

```

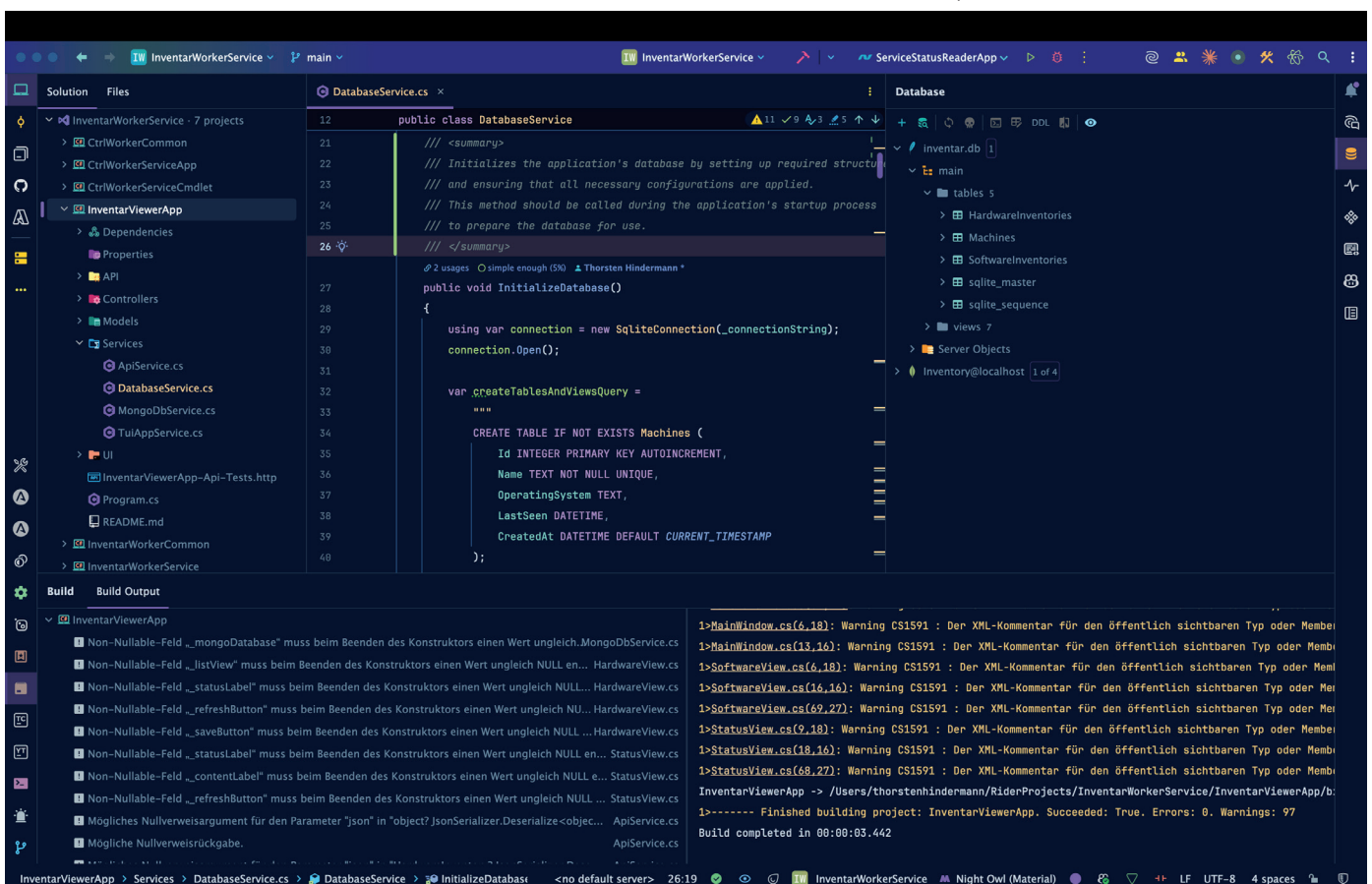
elseif ($Tui)
{
    Start-TerminalGui -ServiceName $ServiceName
}
else
{
    Write-Host "Service Controller PowerShell Script - Hilfe" -ForegroundColor Cyan
    Write-Host "===== " -
    ForegroundColor Cyan
    Write-Host ""
    Write-Host "Verwendung:"
    Write-Host " Invoke-WorkerServiceControl -Start [-ServiceName <Name>]"
    Write-Host " Invoke-WorkerServiceControl -Stop [-ServiceName <Name>]"
    Write-Host " Invoke-WorkerServiceControl -Tui [-ServiceName <Name>]"
    Write-Host ""
    Write-Host "Parameter:"
    Write-Host " -Start      Startet den angegebenen Service"
    Write-Host " -Stop       Stoppt den angegebenen Service"
    Write-Host " -Tui        Startet die Terminal-Benutzeroberfläche"
    Write-Host " -ServiceName Name des Services (Standard: 'mein-service')"

```

Quellzeilenausschnitt 6 (Teil 2)

Einige der Hauptmerkmale von PowerShell sind:

- **Cmdlets:** Dies sind spezialisierte .NET-Klassen, die eine bestimmte Funktionalität bereitstellen. Sie können verwendet werden, um Aufgaben wie das Verwalten von Dateien, das Ändern von Systemeinstellungen und das Verwalten von Prozessen auszuführen.
- **Skripte:** PowerShell-Skripte sind Textdateien, die eine Reihe von PowerShell-Befehlen enthalten. Diese Skripte können verwendet werden, um komplexe Aufgaben zu automatisieren und wiederholbare Prozesse zu erstellen. Quellzeilenausschnitt 6 zeigt PowerShell Script, das ähnlich dem *Cmdlet* aufgebaut ist, um den angegebenen Dienst zu starten und zu stoppen. Eine Verwendungshilfe ist ebenfalls abrufbar.
- **Pipelines:** PowerShell ermöglicht es Benutzer*innen, die Ausgabe eines Befehls als Eingabe für einen anderen Befehl zu verwenden. Dies ermöglicht es, komplexe Aufgaben durch das Verketteten einfacher Befehle zu erstellen.
- **Objektorientierung:** Im Gegensatz zu vielen anderen Skriptsprachen arbeitet PowerShell mit Objekten anstelle von Text. Dies bedeutet, dass die Ausgabe eines Befehls ein Objekt ist, das Eigenschaften und Methoden hat, die von anderen Befehlen verwendet werden können.
- **PowerShell-Provider:** Dies sind spezialisierte PowerShell-Bibliotheken, die es ermöglichen, verschiedene Datenspeicher und Dienste so zu behandeln, als wären sie Dateisysteme. Dies bedeutet, dass Nutzende mit diesen Datenspeichern und Diensten auf die gleiche Weise interagieren können, wie sie es mit Dateien und Ordnern auf einem Dateisystem tun würden. PowerShell-Provider



Bildschirmausschnitt 6

bieten eine einheitliche Schnittstelle für den Zugriff auf verschiedene Arten von Daten, wie z. B. das Dateisystem, die Registrierung, Zertifikatspeicher und mehr. Dies erleichtert die Verwaltung und Automatisierung von Aufgaben, da Administrator*innen und Entwickler*innen dieselben Befehle und Techniken verwenden können, unabhängig davon, mit welchem Datenspeicher oder Dienst sie arbeiten. Aus der PS-Gallery können weitere PowerShell-Provider geladen und genutzt werden, unter anderem zum Beispiel für SQLite-Datenbanken.

GitLab

Bei der GWDG selbst betriebene Server für das Versions-Kontrollprogramm *git*. Die Firma GitLab hat eine ausgefeilte Weboberfläche über das von Linus Torvalds entwickelte *git* erstellt, das neben der klassischen Versionskontrolle noch viele weitere Funktionen bereitstellt. Diese erweiterten Funktionen sind erst einmal nicht Bestandteil dieses Projekts, sondern zuerst nur die Versionskontrolle der erstellten Quellzeilendateien.

SQLite-Datenbank

SQLite ist eine Programmbibliothek, die eine volle Implementierung des SQL-Standards beinhaltet, aber keine Installation eines SQL-Servers erfordert. Alle Daten werden über eine Programmierschnittstelle und entsprechende SQL-Befehle in die Datenbank geschrieben, geändert, gelesen oder gelöscht.

MongoDB-Datenbank

MongoDB ist ein leistungsfähiges, dokumentenorientiertes Datenbankmanagementsystem, das sich durch seine hohe Flexibilität und Skalierbarkeit auszeichnet. Im Gegensatz zu traditionellen, relationalen Datenbanken speichert MongoDB Daten in JSON-ähnlichen Dokumenten, was eine dynamische und schemalose Datenstruktur ermöglicht. Dies macht es besonders geeignet für Anwendungen, die große Mengen an unstrukturierten oder sich schnell ändernden Daten verarbeiten müssen. MongoDB bietet zudem weitere Funktionen wie z. B. Replikation und eine leistungsstarke Abfragesprache, die Entwickler*innen hilft, komplexe Datenabfragen effizient zu gestalten und zu verwalten.

MongoDB lässt sich nahtlos in C#-Programme integrieren, indem man das offizielle MongoDB .NET-Treiberpaket verwendet. Dieses Paket bietet eine umfangreiche API, die es Entwickler*innen ermöglicht, CRUD-Operationen (Erstellen, Lesen, Aktualisieren, Löschen) auf MongoDB-Datenbanken durchzuführen. Mit dem Treiber können Entwickler*innen MongoDB-Datenbanken einfach verbinden, Abfragen erstellen und Daten manipulieren. Die Integration von MongoDB in C#-Anwendungen bietet eine flexible und skalierbare Lösung für die Verwaltung von Daten, insbesondere wenn es um große Mengen an unstrukturierten oder sich schnell ändernden Daten geht.

NuGet-Pakete

NuGet-Pakete sind Programmbibliotheken, die spezialisierte Funktionen gegenüber der bereits umfangreichen Basis des .NET Frameworks zur Verfügung stellen. Diese Pakete werden in verschiedenen Teilprojekten genutzt, um spezifische Anforderungen zu erfüllen.

PS-Gallery

Die PS-Gallery ist ein Repository für PowerShell-Module, Skripte und Ressourcen. Sie ermöglicht es Administrator*innen, PowerShell-Module und -Skripte zu durchsuchen, herunterzuladen und zu installieren, die von der PowerShell-Community erstellt und geteilt wurden.

Die PS-Gallery bietet eine zentrale Anlaufstelle für die Verwaltung und Verteilung von PowerShell-Inhalten, was die Automatisierung und Verwaltung von Systemen erleichtert. Benutzer*innen können ihre eigenen Module und Skripte hochladen und teilen, um anderen in der Community zu helfen und von deren Feedback zu profitieren.

ERWEITERUNGSMÖGLICHKEITEN

Die *InventarViewApp* kann um die Möglichkeit erweitert werden, dass Systeme registriert werden können, deren Inventar gesammelt werden soll. Dazu müsste passenderweise die Tabelle *Machines* mit den Maschinendaten um z. B. Spalten für die IP-Adresse und den DNS-Namen erweitert werden. Weiterhin können Spalten dazukommen, welche Maschine nicht mehr beim Sammeln der Daten beachtet werden soll (deaktiviert) oder gar aus dem Bestand genommen worden ist (deprovisioniert). Hier müsste man sich überlegen, was mit den bis dahin gesammelten Daten passieren soll (löschen, exportieren und/oder archivieren).

Weiterhin kann die REST API erweitert werden, damit in den REST API-Endpunkten zusätzlich der Maschinenname zu der Maschinen-ID aus der Datenbank als Referenzierung für den bestimmten Datensatz verwendet werden kann.

Zudem müsste es auf dem Rechner mit der *InventarViewApp* einen weiteren WorkerService geben, der die Tabelle *Machines* periodisch ausliest und von den nicht-deaktivierten und nicht-deprovisionierten Maschinen nach und nach deren Daten abfragt. Hier kommen dann Fragestellungen auf, was mit aktiven Maschinen passiert, die aber nicht kontinuierlich abgefragt werden können, z. B. Laptops auf Dienstreise oder lange Zeit ausgeschaltete Rechner (z. B. Urlaubszeit). Für den ersten Fall müssten die Daten auf die lokale Festplatte des Rechners geschrieben werden, auf dem der *InventarWorkerService* ausgeführt wird – was der *InventarWorkerService* aber schon tut. Hier müssten dann die Delta-Daten an Hard- und Software von den lokal gespeicherten Dateien zum Sammler-Service übertragen werden. Welche Daten fehlen, könnte anhand des letzten gespeicherten Datums in der Spalte *LastSeen* verglichen mit dem aktuellen Datum erreicht werden.

Derzeit werden die großen Datenbestände der ermittelten Software, laufenden Dienste etc. im JSON-Format in die SQL-Datenbank gespeichert. Hier könnten Überlegungen angestellt werden, dass für solche Datenbestände eine NoSQL-Datenbank wie z. B. MongoDB besonders gut geeignet ist, die JSON-Daten darin zu speichern und daraus wieder abzurufen. Die starken Möglichkeiten der MongoDB, von vielen Programmiersprachenaus angesprochen zu werden, unter anderem auch von C#, macht es für Entwickler*innen einfach, mit Hilfe von in derselben Programmiersprache geschriebenen Quellzeilen in einem objektorientierten, dokumentenzentrierten Datenbankmodell die Daten abzuspeichern und abzurufen. Hier könnten dann Diskussionen geführt werden, ob eine Mischung von relationalen und dokumentenbasierten Datenbankmodellen eine gute Idee ist. Und wenn man sich für eine Vorgehensweise entschieden hat, entweder komplett

relational oder aber die Mischung aus relationaler und dokumentenbasierter Datenspeicherung, diese Erweiterung umzusetzen. Dazu würde auch die Erweiterung der REST APIs gehören, die die Auslieferung der erweiterten Detaildaten mit zur Verfügung stellen muss. Bei einer rein relationalen Datenspeicherung nur in der SQL-Datenbank müssten hier neue Tabellen mit einer entsprechenden Fremdschlüsselverknüpfung von der erfassten Maschine zu den Detail-daten hinzugefügt werden. Auch die Programmierung müsste für dieses Szenario in geeigneter Art und Weise angepasst werden.

Diese beiden hier skizzierten Erweiterungen der Basis-App der *InventarViewerApp* sind dann ein größeres Unterfangen und Bewegen das Projekt von einem Proof-of-Concept mehr in eine Richtung, in der es schon in geringem Umfang produktiv eingesetzt werden könnte. Wie in allen solchen Projekten kommen nach einem Startschuss weitere Ideen und Anforderungen dazu, die das Projekt dann kontinuierlich wachsen lassen, und dieses immer mehr erweitert wird und mehr und mehr einsatzfähiger wird. Hier

sind dann aber die Grenzen der Auszubildenden erreicht und ab diesem Zeitpunkt müsste das PoC-Projekt in die Hände von ausgebildeten und erfahrenen Anwendungsentwickler*innen, unter Führung eines/r Senior-Entwickler*in, gelegt werden, die dieses Projekt dann nach und nach zur Produktionsreife weiterentwickeln.

AUSBLICK

Das Projekt ist auch dafür geeignet, den Ausbildungsgang Fachinformatiker*in in der Fachrichtung Daten- und Prozessanalyse mitzuintegrieren, der voraussichtlich zukünftig ebenfalls bei der GWDG etabliert werden soll. Eventuell sind hierfür noch ein paar kleine Anpassungen und Ergänzungen dieses Projekts notwendig. Aber der Kern, was die Programmierung und die verwendeten Technologien und Programme angeht, sollte dann auch für diese drei Ausbildungsgänge in den ersten beiden gemeinsamen Lehrjahren passend und ausreichend sein. ■



Matrix

Kommunikation leicht gemacht!

Ihre Anforderung

Sie benötigen einen professionellen Chat-Dienst, der eine einfache, persistente Kommunikation mit Kolleg*innen ermöglicht – sowohl in Einzel- als auch in Gruppenunterhaltungen, die komfortabel durchsuchbar sind. Sie wollen Bilder und Dateien mit Kolleg*innen austauschen.

Unser Angebot

Wir betreiben den Messaging-Dienst „Matrix“, der es Ihnen ermöglicht, sich in Teams, Gruppen oder auch einzeln auszutauschen. Der Dienst unterstützt zusätzlich Emojis, das Versenden von Dateien, Bildern und Videos. Der Zugriff auf unseren Matrix-Dienst erfolgt über den Element-Client, der neben einer übersichtlichen Weboberfläche auch Apps für Android und iOS bietet. So können Sie immer und von überall auf ihre Matrix-Chats zugreifen.

Ihre Vorteile

- > Einfache Kommunikation im Team
- > Persistente, durchsuchbare Chat-Verläufe
- > Einfaches Teilen von Dateien und Bildern
- > Unterhaltungen mit allen Nutzer*innen, die eine AcademicID verwenden

Interessiert?

Jede*r Nutzer*in mit einem Konto der Max-Planck-Gesellschaft oder der Universität Göttingen und Nutzer*innen einer teilnehmenden Hochschule der Academic Cloud können den Dienst „Matrix“ nutzen.

>> www.gwdg.de/matrix

Kurz & knapp

GWDG gewinnt den IT Community Award 2025 der MPG

Wir freuen uns sehr über einen besonderen Preis: Die GWDG mit all ihren Teams hat bei den IT4Science-Days, die vom 23. bis 25.09.2025 in Bremen stattgefunden haben, den IT Community Award der MPG erhalten, mit dem jährlich ein wesentlicher Beitrag zur Förderung des Community-Gedankens in der IT der MPG gewürdigt wird. Damit wird der Preis nicht wie sonst an eine oder mehrere Personen, sondern an eine Dienstleistungseinrichtung der MPG vergeben, die mit ihrem umfangreichen Dienstangebot in vielfältiger Weise IT in der Wissenschaft unterstützt.



Besonders hervorgehoben wurden die neue Initiative zum Netzwerktreffen, das große Engagement bei der Bereitstellung neuer Zertifikate, nachdem der Zertifikatsanbieter zweimal wechselte, und die Bereitstellung der stabilen BBB-Kommunikationsplattform für hybride IT-Meetings, mit der ein wichtiger Beitrag zur digitalen Souveränität in der IT geleistet wird.

Otto

Relaunch des Umfragedienstes „LimeSurvey“

Der angekündigte Relaunch unseres Umfragedienstes „LimeSurvey“ ist in vollem Gange. Dieser umfasst unter anderem ein Update auf LimeSurvey Version 6.x in einer neuen Umgebung sowie eine verbesserte Nutzbarkeit.

Die neue Plattform erreichen Sie über unsere Landing-Page unter <https://survey.academiccloud.de>. Dort finden Sie ebenfalls einen Direktlink zur alten Plattform, die samt ihrer laufenden Umfragen noch bis zum 26.02.2026 erreichbar sein wird. Um Ihnen einen reibungslosen Übergang von der bisherigen zur neuen Plattform zu ermöglichen, können Sie neue Umfragen auf der alten Plattform noch bis zum 12.11.2025 erstellen. Dort laufende Umfragen werden bis in den Februar 2026 hinein durchgeführt-, auswert- und exportierbar sein.

Bei weiteren Fragen oder wenn Sie Unterstützung benötigen, schauen Sie bitte in unsere Dokumentation unter https://info.gwdg.de/dokuwiki/doku.php?id=de:services:application_services:surveys:start oder kontaktieren Sie uns per E-Mail an survey-support@gwdg.de.

Meyer

KI.Connect – Das Vernetzungsevent für die Region am 14.11.2025 in Göttingen

Am Freitag, den 14. November 2025, findet bei der GWDG im Göttinger Rechenzentrum das KI-Event „KI.Connect“ statt – ein Vernetzungs- und Informationsformat für regionale Unternehmen, Wissenschaft, Studierende und Politik.

Auf dem Programm stehen populärwissenschaftliche Vorträge, Workshops und Diskussionen zu aktuellen KI-Anwendungen in Medizin, Agrar, Industrie und Gesellschaft. Ergänzt wird das Event durch einen „Marktplatz“ für Abschlussarbeiten, Networking und Führungen durch das Göttinger Rechenzentrum.



Die Veranstaltung bietet Raum zum Austausch über Chancen, Herausforderungen und nachhaltige Entwicklungen im Bereich Künstliche Intelligenz. Die Teilnahme ist kostenlos. Eine vorherige Anmeldung ist erforderlich. Weitere Informationen zur Veranstaltung sind unter <https://www.uni-goettingen.de/de/689913.html> zu finden.

Otto

IT Security Awareness Days vom 03.11. – 14.11.2025

Mehrere deutsche Hochschulen, u. a. auch die Universität Göttingen / GWDG, veranstalten gemeinsam vom 03.11. – 14.11.2025 zum zehnten Mal die IT Security Awareness Days als hochschulübergreifende öffentliche Online-Veranstaltungsreihe mit spannenden Vorträgen rund um das Thema Informationssicherheit. Die Vortragsthemen reichen von DeepFakes, ChatGPT & Co. bis hin zum richtigen Umgang mit Passwörtern und richten sich vorrangig an Nutzer*innen ohne besondere Vorkenntnisse. Zudem werden technische Vorträge angeboten, die auch für Expert*innen und Administrator*innen von Interesse sein könnten. Die Veranstaltung ist öffentlich, eine Anmeldung ist nicht erforderlich. Eine Zusammenfassung der Veranstaltungstermine steht unter dem URL <https://s.gwdg.de/PvncqP> bereit. Informationen zu den einzelnen Vorträgen finden Sie unter dem URL <https://www.tu-braunschweig.de/ciso/it-sad/it-sad-wintersemester-2025/26>.

Dickmann

NEUER MITARBEITER MAX PLIESKE

Seit dem 1. Juli 2025 unterstützt Herr Max Plieske die Arbeitsgruppe „Organisation, Betrieb und Entwicklung von Diensten“ (AG D) als studentische Hilfskraft. Zu seinen Tätigkeiten gehören unter anderem die Mitarbeit im Bereich Authentifizierung und Autorisierung beim Betrieb und der Dokumentation sowie die Unterstützung des Identity-Management-Teams bei der Weiterentwicklung und Betreuung der Authentifizierungs-Infrastruktur. Herr Plieske studiert zurzeit Biologie an der Georg-August-Universität Göttingen. Er ist per E-Mail unter max.plieske@gwdg.de zu erreichen.

Wong



NEUER MITARBEITER SEBASTIAN SCHMITT

Seit dem 1. September 2025 ist Herr Dr. Sebastian Schmitt als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Computing“ (AG C) beschäftigt. Sein Diplom in Physik absolvierte er an der Universität Hamburg. Anschließend promovierte er an der Universität Heidelberg in experimenteller Teilchenphysik am ATLAS-Experiment, wo er sich mit großskaliger Datenanalyse und Grid-Computing befasste. Es folgten Postdoc-Stellen an der Universität Heidelberg im Bereich des neuromorphen Rechnens (BrainScaleS-Projekt) und an der Georg-August-Universität Göttingen in den computergestützten Neurowissenschaften, wo er am HPC-Simulationsframework Arbor mitwirkte. Vor seinem Eintritt in die GW DG war Herr Dr. Schmitt als System-Engineer bei der SVA GmbH tätig. Dort konzentrierte er sich auf die Themen Automatisierung und Infrastructure as Code und arbeitete mit Technologien wie Ansible, Terraform und GitLab. Bei der GW DG wird er seine Erfahrung einbringen, um die HPC-Dienste für die Nutzer*innen zu unterstützen und weiterzuentwickeln. Herr Dr. Schmitt ist per E-Mail unter sebastian.schmitt@gwdg.de zu erreichen.

Kunkel

NEUER MITARBEITER DAVID ROSIN

Seit dem 1. Oktober 2025 ist Herr David Rosin als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Computing“ (AG C) beschäftigt und wird dort zukünftig den IT-Support unterstützen und die zugehörigen Prozesse weiterentwickeln. Vor seiner Zeit in Göttingen erwarb Herr Rosin einen Bachelor-Abschluss in Physik an der Universität Bremen und anschließend einen Master-Abschluss in Medizintechnik an der Universität Stuttgart mit Fokus auf numerischer Optimierung und Systemdynamik sowie Biomechanik und Biorobotik. Danach verblieb er in Stuttgart für seine Promotion im Bereich biomechanische Simulation und „pervasive Computing“ im Rahmen des Projekts „PerSiVal“ des Cluster of Excellence – Simulation Technology (SimTech). Herr Rosin ist per E-Mail unter david.rosin@gwdg.de zu erreichen.



Kunkel



INFORMATIONEN:
support@gwdg.de
0551 39-30000

November bis
Dezember 2025

Academy

KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
GETTING STARTED WITH LINUX BASH	Dr. Lüdemann, Eulert	05.11.2025 9:00 – 12:00 Uhr	29.10.2025	2
GETTING STARTED WITH THE AI TRAINING PLATFORM	Dr. Lüdemann, Eulert	05.11.2025 13:00 – 16:00 Uhr	29.10.2025	2
VIRTUELLE CLOUD-INFRA-STRUKTUREN – KURS FÜR AUSZUBILDENDE	Kopp	10.11. – 12.11.2025 9:00 – 16:00 Uhr	03.11.2025	12
GRUNDLAGEN DER BILDBEARBEITUNG MIT AFFINITY PHOTO	Töpfer	11.11. – 12.11.2025 9:30 – 16:00 Uhr	04.11.2025	8
MONITORING HPC SYSTEMS IN THE GWDG	Merz	13.11.2025 9:00 – 10:00 Uhr	06.11.2025	1
PARALLEL COMPUTING WITH MATLAB	Künzel	17.11.2025 10:00 – 13:00 Uhr	10.11.2025	0
ADMINISTRATION IM ACTIVE DIRECTORY	S. Quentin, Kopp	18.11.2025 9:00 – 12:30 und 13:30 – 15:30 Uhr	11.11.2025	4
WORKING WITH GRO.DATA	Dr. Király	18.11.2025 14:00 – 15:30 Uhr	11.11.2025	0
EINFÜHRUNG IN DIE STATISTISCHE DATENANALYSE MIT SPSS	Cordes	19.11. – 20.11.2025 9:00 – 12:00 und 13:00 – 15:30 Uhr	12.11.2025	8
DEMO SESSION: SCALING UP MATLAB TO THE GWDG SCIENTIFIC COMPUTE CLUSTER	Norris	19.11.2025 15:00 – 16:30 Uhr	12.11.2025	1
INTRODUCTION TO RESEARCH SOFTWARE DEVELOPMENT WITH MATLAB	Künzel	20.11.2025 9:00 – 12:00 Uhr	13.11.2025	2

KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
CONNECTING MATLAB WITH PYTHON AND OTHER OPEN SOURCE TOOLS	Kerber	20.11.2025 14:00 – 17:00 Uhr	13.11.2025	2
KI IN DER VERWALTUNG: EINE EINFÜHRUNG IN DIE NUTZUNG FÜR ALLE MITARBEITER*INNEN	Eulert, Rafi	25.11.2025 9:00 – 12:00 Uhr	18.11.2025	2
SECURE HPC – PARALLEL COMPUTING WITH HIGHEST SECURITY	Tabougua	26.11.2025 10:00 – 11:30 Uhr	19.11.2025	1
AFFINITY DESIGNER – SCHNUPPERKURS FÜR EINSTEIGER*INNEN	Töpfer	27.11.2025 10:30 – 12:30 und 13:30 – 15:30 Uhr	20.11.2025	3
FUNDAMENTALS OF ACCELERATED COMPUTING WITH CUDA PYTHON	Meisel	02.12.2025 9:00 – 17:00 Uhr	25.11.2025	
ANSYS ON CLUSTER AND POST-PROCESSING OF SIMULATION RESULTS	Dr. Höhn, Dr. Kanning	03.12.2025 9:00 – 12:00 und 13:00 – 16:00 Uhr	26.11.2025	4
EFFECTIVELY UTILIZE AI TOOLS IN RESEARCH	Eulert, Lewis, Rafi	04.12.2025 9:00 – 12:00 Uhr	27.11.2025	2
DEEP LEARNING BOOTCAMP: BUILDING AND DEPLOYING AI MODELS	Lewis	09.12. – 10.12.2025 14:30 – 16:30 Uhr	02.12.2025	3
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER*INNEN MIT VORKENNTNISSEN	Cordes	10.12. – 11.12.2025 9:00 – 12:00 und 13:00 – 15:30 Uhr	03.12.2025	8
WORKING WITH GRO.DATA	Dr. Király	16.12.2025 14:00 – 15:30 Uhr	09.12.2025	0
SUPERCOMPUTING FOR EVERY SCIENTIST	Eulert, Dr. Lüdemann	17.12.2025 9:00 – 12:00 und 13:00 – 16:00 Uhr	10.12.2025	4

Teilnehmerkreis

Das Angebot der GWGD Academy richtet sich an die Beschäftigten aller Einrichtungen der Universität Göttingen, der Max-Planck-Gesellschaft sowie aus wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWGD gehören. Studierende am Göttingen Campus zählen ebenfalls hierzu. Für manche Kurse werden spezielle Kenntnisse vorausgesetzt, die in den jeweiligen Kursbeschreibungen genannt werden.

Anmeldung

Für die Anmeldung zu einem Kurs müssen Sie sich zunächst mit Ihrem Benutzernamen und Passwort in der GWGD Academy (<https://academy.gwdg.de>) einloggen. Wenn Sie zum Kreis der berechtigten Nutzer*innen der GWGD gehören, erhalten Sie anschließend automatisch Zugang zu unserem Kursprogramm. Sollten Sie noch keinen Account besitzen, können Sie sich unter <https://id.academiccloud.de> registrieren und müssen ggf. auf Anfrage für die Anmeldung zu unseren Kursen freigeschaltet werden. Bei Online-Kursen kann das Anmeldeverfahren abweichen. Genauere Informationen dazu finden Sie in der jeweiligen Kursbeschreibung. Einige Online-Angebote stehen Ihnen jederzeit und ohne Anmeldung zur Verfügung.

Absage

Absagen können bis zu sieben Tagen vor Kursbeginn erfolgen. Bei kurzfristigeren Absagen werden allerdings die für den Kurs angesetzten Arbeitseinheiten (AE) vom AE-Kontingent der jeweiligen Einrichtung abgezogen.

Kursorte

Die Kurse finden entweder in einem geeigneten Online-Format oder als Präsenzkurs statt. Nähere Informationen dazu finden Sie bei den jeweiligen Kursen. Auf Wunsch und bei ausreichendem Interesse führen wir auch Kurse vor Ort in einem Institut durch, sofern dort ein geeigneter Raum mit entsprechender Ausstattung zur Verfügung gestellt wird.

Kosten bzw. Gebühren

Die Academy-Kurse sind – wie die meisten anderen Leistungen der GWGD – in das interne Kosten- und Leistungsrechnungssystem der GWGD einbezogen. Die den Kursen zugrundeliegenden AE werden vom AE-Kontingent der jeweiligen Einrichtung abgezogen. Für alle Einrichtungen der Universität Göttingen und der Max-Planck-Gesellschaft sowie die meisten der wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer*innen der GWGD gehören, erfolgt keine Abrechnung in EUR. Dies gilt auch für die Studierenden am Göttingen Campus.

Kontakt und Information

Wenn Sie Fragen zum aktuellen Academy-Kursangebot, zur Kursplanung oder Wünsche nach weiteren Kursthemen haben, schicken Sie bitte eine E-Mail an support@gwdg.de. Falls bei einer ausreichend großen Gruppe Interesse besteht, könnten u. U. auch Kurse angeboten werden, die nicht im aktuellen Kursprogramm enthalten sind.

