BISE Student

https://bise-student.io

BACHELOR'S THESIS

Trade-Offs Concerning the Design of Artificial-Intelligence-as-a-Service

Publication Date: 2023-11-07

Author
Felix FELIX LINNEMANN
Karlsruher Institut für Technologie
Karlsruhe, Germany
felixlinnemann@outlook.de
0xf22ab176648EDe6095172b0e9889f64B3e84D9A4

Abstract

While Artificial Intelligence (AI) offers many potential benefits in a magnitude of application areas, organizations are still struggling to integrate AI capabilities on a large scale across different business functions. To address this issue, cloud providers offer cloud-based Artificial Intelligence services also known as Artificial-Intelligence-as-a-service (AIaaS). These services aim to support customers in using, developing, deploying, and managing AI capabilities in an easy and flexible way on demand in the cloud. Since AIaaS is a combination of cloud computing and AI, it shares key characteristics of these technologies. Additionally, the combination also yields new AIaaS-specific characteristics leading to increased complexity in service design. There is no one-size-fits-all AIaaS design and consequently, providers face challenges and trade-offs between characteristics that impact the service design and value proposition of AIaaS. To foster an understanding of interdependencies between AIaaS characteristics, trade-offs...

Keywords: AI, Cloud Computing, AIaaS, AI services, MLaaS
Methods: expert interviews, semi-structured interviews, open coding

Submission Date: 2023-10-31
Submission Contract: 0x531a6DC6945b572DdC00688b97a06F9Bb082A227

Trade-Offs Concerning the Design of Artificial-Intelligence-as-a-Service

Bachelor Thesis

by

Felix Linnemann

Degree Course: Industrial Engineering and Management

Matriculation Number: 2230291

Institute for Applied Informatics and Formal Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Prof. Dr. Ali Sunyaev
Second Advisor:	Prof. Dr. Andreas Oberweis
Supervisor:	Dr. Sebastian Lins
Submitted:	7. November 2022

Abstract

While Artificial Intelligence (AI) offers many potential benefits in a magnitude of application areas, organizations are still struggling to integrate AI capabilities on a large scale across different business functions. To address this issue, cloud providers offer cloud-based Artificial Intelligence services also known as Artificial-Intelligence-as-a-service (AIaaS). These services aim to support customers in using, developing, deploying, and managing AI capabilities in an easy and flexible way on demand in the cloud. Since AIaaS is a combination of cloud computing and AI, it shares key characteristics of these technologies. Additionally, the combination also yields new AIaaS-specific characteristics leading to increased complexity in service design. There is no one-size-fits-all AIaaS design and consequently, providers face challenges and trade-offs between characteristics that impact the service design and value proposition of AIaaS. To foster an understanding of interdependencies between AIaaS characteristics, trade-offs as well as to assess the consequences of design decisions, this thesis provides an overview of AIaaS-specific characteristics and prevailing trade-offs from a provider's perspective. Moreover, additional challenges perceived by AIaaS providers are discussed. The results were derived from six conducted semi-structured expert interviews with AIaaS providers. Findings incorporate 10 identified categories of characteristics, each including multiple aspects and a total of 8 identified trade-offs that underline the complexity of AIaaS service design. Furthermore, findings highlight the multitude of design dimensions in AIaaS and suggest that complexity abstraction and performance characteristics are central to most of the identified trade-offs.

Table of Contents

List of Abbreviations ..... **iv**

List of Figures ..... **v**

List of Tables ..... **vi**

1. Introduction **1**

1.1. Problem Definition 1

1.2. Research Objectives 3

1.3. Structure of Thesis 3

2. Background **3**

2.1. Cloud Services 4

2.2. Artificial Intelligence and Machine Learning 4

2.3. AIaaS 6

3. Research Approach **7**

3.1. Interview Partner Acquisition 8

3.2. Interview Conduction 9

3.3. Interview Transcription 9

3.4. Interview Analysis 10

4. Trade-offs between AIaaS Characteristics **11**

4.1. AIaaS in Practice 11

4.2. Characteristics of AIaaS 12

4.3. Trade-offs and Service Design Implications 24

4.4. Additional Challenges in AIaaS Service Design 32

5. Conclusion **34**

5.1. Principal Findings 34

5.2. Implications for Research and Practice 35

5.3. Limitations 36

5.4. Further Research 36

Appendix **38**

A. Appendix A 38

B. Appendix B 39

C. Appendix C 51

References **54**

Declaration about the Thesis **58**

List of Abbreviations

AI	Artificial Intelligence
AIaaS	Artificial Intelligence-as-a-Service
AutoML	Automated Machine Learning
CSP	Cloud Service Provider
GPU	Graphical Processing Unit
GUI	Graphical User Interface
IaaS	Infrastructure-as-a-Service
IP	Intellectual Property
IT	Information Technology
ML	Machine Learning
NLP	Natural Language Processing
PaaS	Platform-as-a-Service
PII	Personal Identifiable Information
POC	Proof of Concept
SaaS	Software-as-a-Service
SMEs	Small and Medium-sized Enterprises
SO	Subobjective
TPU	Tensor Processing Unit
UI	User Interface
XaaS	Everything-as-a-Service

List of Figures

Figure 1: Stages of ML Workflow (Amershi et al., 2019) 5
Figure 2: Overview of Identified Trade-offs between AIaaS Characteristics 24
Figure 3: Trade-offs between Complexity Abstraction, Performance and Genericness and resulting AIaaS Types 25

List of Tables

Table 1: Overview of Interviewees 8
Table 2: Aspects of the Characteristic Complexity Abstraction 13
Table 3: Aspects of the Characteristic Customizability 15
Table 4: Aspects of the Characteristic Genericness 16
Table 5: Aspects of Characteristic Trustworthiness 17
Table 6: Aspects of the Characteristic Security and Privacy 19
Table 7: Aspects of the Characteristic Deployment and Integration 19
Table 8: Aspects of the Characteristic Performance 20
Table 9: Aspects of the Characteristic Centralization 21
Table 10: Aspects of the Characteristic Functionalities 23
Table 11: Aspects of the Characteristic Cloud 23

1. Introduction

1.1. Problem Definition

In recent years Artificial Intelligence (AI) has become one of the most discussed topics among researchers and practitioners in the field of information technology (IT). Driven by its potential for economic benefits and new technological approaches, AI is applied across a variety of industries and has manifold applications areas including but not limited to smart manufacturing (Ding et al., 2020), economics and finance (Gogas & Papadimitriou, 2021, pp. 3–4), e-commerce, healthcare, natural language processing (NLP) or image recognition (Sarker, 2021, pp. 15–16). According to a recent survey by McKinsey & Company, 56% of companies adopted AI in at least one business function and AI became an important success factor for most businesses (Chui et al., 2021, pp. 2–4). However, organizations still struggle to implement and integrate AI capabilities into their business at scale. Accordingly, 46% of organizations only have deployed AI capabilities on a small scale or don't achieve significant outcomes (Ammanath et al., 2021, p. 7). Reasons for this include the scarcity of AI expertise and knowledge or the complexity of data collection and processing (Pandl et al., 2021, p. 1). Furthermore, especially small and medium-sized enterprises (SMEs) are struggling to allocate the required computational infrastructure at an affordable rate that is especially essential for the training of an AI model (Ribeiro et al., 2015, p. 896).

As a consequence, well-known cloud providers such as Microsoft, Google, Amazon, and IBM have started to offer cloud services with AI capabilities to lower entry barriers in the field of AI. Additionally, start-ups and SMEs provide tailored solutions for specific industries or use cases (Lins et al., 2021, p. 441). These AI-related services, are also commonly referred to as Artificial-Intelligence-as-a-Service (AIaaS) and can be defined as "cloud-based systems providing on-demand services to organizations and individuals to deploy, develop, train, and manage AI models" (Lins et al., 2021, p. 442). The primary objective of AIaaS is to enable every business to leverage AI capabilities over the cloud regardless of the businesses' intellectual, computational, and financial resources (Pandl et al., 2021, p. 1769).

However, the service design and provision of AIaaS comes not without challenges. In general, the process of developing AI and machine learning (ML) solutions requires lots of manual work and includes iterative human interaction and tuning (Karmaker et al., 2022, pp. 3–6). One has to adapt the model to a specific use case and its performance depends on used algorithms, parameters, and training datasets (Ribeiro et al., 2015, p. 897). Therefore, the development of AI solutions differs fundamentally from software development (Paleyes et al., 2022, p. 18) and conflicts with the business model of cloud services, as it relies on automated and easy service provision to diverse users, possibly even to users without technical knowledge.

Further, the use cases for AIaaS differ, for example in "accuracy, responsiveness, and monetary budget constraints" (Halpern et al., 2019, p. 34). It is not feasible for the provider to automatically provide custom-tailored AI services for every single use case and hence the provider is forced to make a "static design time decision based on generic needs" (Halpern et al., 2019, p. 34), resulting in an one-size-fits-all design of AIaaS systems (Halpern et al., 2019, p. 34). Javadi et al. (2020) demonstrate this problem with an illustrative example: “For instance, the same object recognition service used by one customer for warehousing might be used by another to support video surveillance” (Javadi et al., 2020, p. 300).

The design of AIaaS is dependent on the specification of characteristics of AIaaS (Lins et al., 2021, pp. 447–448). Characteristics are key properties of an AIaaS service that can be designed and adjusted by the provider, for example, the complexity abstraction or customizability. As some AIaaS characteristics are contrary to each other, providers face challenges and trade-offs in the service design of AIaaS to meet customers’ differing requirements. In this case, trade-offs are situations in which two opposing characteristics of the service cannot be improved simultaneously, and thus the improvement of one characteristic can diminish other characteristics of the corresponding AIaaS service. For example, AIaaS providers possibly enter a trade-off between the performance of an AIaaS service and the complexity of using an AIaaS service (Yao et al., 2017, pp. 388–390). An AIaaS service that is easy to use for every possible user is limited in the provision of customizability options. This can result in a less performant AI model compared to a manually well-tuned model.

To date, extant research lacks knowledge of these provider-sided trade-offs and how they influence service design and configuration (Lins et al., 2021, pp. 452–454). Existing research has mainly focused on platform architecture, specific applications and challenges (Philipp et al., 2020, p. 404), AIaaS adoption (Pandl et al., 2021; Zapadka et al., 2020), or more recently on conceptualizing and creating taxonomies for the growing field of AIaaS (Geske et al., 2021; Lins et al., 2021).

The combination of AI and cloud computing leads to AIaaS-exclusive characteristics, ranging from purely technical characteristics such as performance to socio-technical characteristics, for instance, customizability (Lins et al., 2021, p. 446). These novel characteristics present AIaaS providers with novel challenges and trade-offs. Halpern et al. (2019) investigated accuracy vs. latency trade-offs in AI services, whereas Yao et al. (2017) evaluated the performance of AIaaS systems, uncovering a trade-off between complexity and performance. Additionally, Elshawi and Sakr (2017) are discussing a trade-off between the accuracy and generalizability requiring manual user-defined inputs to the service (Elshawi & Sakr, 2017, p. 6).

So far, research only examined trade-offs separate from each other. To the best of this author’s knowledge, there is no complete overview of interdependencies and trade-offs between AIaaS characteristics from a provider’s viewpoint and their influence on service design. Moreover, researchers call for further investigation of existent trade-offs and how they influence service provision. Lins et al. (2021) propose that future research should “analyze whether the simplification offered by AIaaS concerning the implementation of AI is in relation to the performance losses associated with a potential non-optimal configuration“ (Lins et al., 2021, p. 452) and the need to create a balance in the trade-off between accuracy and fairness vs. generalizability (Lins et al., 2021, p. 452). To close this research gap, the thesis answers following the research question:

RQ: How do trade-offs between AIaaS characteristics influence AIaaS-service design?

1.2. Research Objectives

To answer the research question, the thesis can be split into the following sub-objectives (SO):

1. What are the core characteristics of AIaaS?
2. What are the trade-offs between these characteristics?
3. What are the consequences for AIaaS service design?

Answering these questions will foster the understanding of AIaaS-specific characteristics and their interdependencies. In particular, trade-offs and challenges in service design are revealed, addressing the overarching challenge to offer AI capabilities as turnkey, easy-to-use, on-demand cloud services. Furthermore, the implications for AIaaS design are discussed. To base the research as closely as possible on the reality and perceived trade-offs of providers, expert interviews are conducted. The investigation will make it possible to derive differences, limitations, and risks between different AIaaS service designs. This thesis can help future researchers to address these trade-offs and challenges and develop more balanced design solutions by integrating new approaches of AI and cloud research in AIaaS. Moreover, it will support practitioners to unlock the full potential of AIaaS and increase the value proposition for the customer. The thesis sets a focus on socio-technological trade-offs which are especially present in AI software services, as these AI services are high-level services and thus require more design time decisions to abstract away lower-level complexities (Lins et al., 2021). AI software services include software that provides a query interface to pre-trained models or that enables the creation of custom models.

1.3. Structure of Thesis

The remainder of the thesis is structured as follows. First, the background chapter describes the cloud computing paradigm, the field of AI as well as the main topic AIaaS that is a combination of the two former mentioned fields. Second, the applied research method and the research process are outlined in chapter 3. After describing the research process, the findings of the semi-structured expert interviews are reported in chapter 4 as follows. First, different AIaaS services in practice are outlined to provide a better understanding of characteristics and trade-offs. Section 4.2 addresses *SO1* and outlines characteristics of AIaaS, whereas Section 4.3 presents provider-sided trade-offs between these characteristics in AIaaS service design, thereby addressing *SO2*. Where existent, practical implications and consequences of trade-offs are also mentioned in Section 4.3, thereby answering *SO3*. Further, Section 4.4 deals with additional provider-sided challenges. Finally, the thesis is concluded in chapter 5.

2. Background

2.1. Cloud Services

Cloud computing is a computing paradigm that can be defined as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of computing resources […] that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell & Grance, 2011, p. 3). Cloud offerings are provided as cloud services that underpin a multitude of applications (Cobbe & Singh, 2021, p. 5). These services consist of five essential characteristics (Hogan et al., 2011, p. 14). On-demand self-service enables the easy and automated provision of computing capabilities without the provider’s interaction. In addition, resources are pooled to dynamically adapt resources to consumers’ needs and accessed over a network, typically the internet. Cloud services allow for rapid elasticity and near-unlimited scalability from a consumer’s viewpoint. Further, the services are measured and thus provide cost transparency (Hogan et al., 2011, p. 14). These characteristics are the foundation for the promising benefits perceived by enterprises. Perceived benefits include, but are not limited to, high availability, scalability, security, computing platforms and cost reduction (Phaphoom et al., 2012, p. 51).

In general, three service models are differentiated. On the lowest abstraction level service providers provide “fundamental computing resources where the customer is able to deploy and run arbitrary software, which can include operating systems and applications” (Hogan et al., 2011, p. 15). This service model is called Infrastructure-as-a-service (IaaS). Within Platform-as-a-service (PaaS), service providers manage the underlying infrastructure, the operating system and provide programming languages and tools. Hence, the consumer can create his own customized applications on top of the platform. On the highest abstraction level providers provide turn-key applications running on a cloud infrastructure with limited customizability options on the consumer’s side, also known as Software-as-a-service (SaaS) (Hogan et al., 2011, p. 15). However, in the last years, more and more diverse computing services were offered as cloud services, leading to the offering of so-called everything as a Service (XaaS) (Duan et al., 2015, p. 626). XaaS includes for example data and information management as a service or security as a service (Duan et al., 2015, p. 621).

Cloud services can be deployed in different ways, including the deployment models private cloud, public cloud and hybrid cloud (Savu, 2011, p. 2). A private cloud is operated on dedicated infrastructure, that can resist on-premises or off-premises and is managed by the organization itself or a third party. Whereas public clouds are made available to the general public by the respective service provider and can be accessed by diverse organizations or individuals. Public clouds are based on large-scale infrastructure that is managed by the cloud service provider (CSP). A hybrid cloud is the combination of different cloud deployment models that remain separate entities, but a hybrid model can combine the advantages of the two former mentioned deployment models.

2.2. Artificial Intelligence and Machine Learning

AI can be defined along different dimensions (Russell & Norvig, 2009, pp. 1–16), but one commonly used definition describes AI as „a branch of computer science devoted to developing data processing systems that performs functions normally associated with human intelligence, such as reasoning, learning, and self-improvement“ (National Institute of Standards and Technology, 2019, p. 25).

Closely related to AI is the term ML that is a subfield of AI and can most basically be described as statistical-computational methods that enable computer systems to automatically improve through experience (Jordan & Mitchell, 2015, p. 255). As such, ML works to build representative models of training data to make inferences from formerly unknown data without explicit programming rules (Bishop, 2006, pp. 1–2). In practice, ML is widely used for manifold applications including computer vision (CV), speech recognition, NLP and others (Jordan & Mitchell, 2015). Developing and managing an ML application in practice is a multi-stage workflow (Amershi et al., 2019, 292.293). As an example, the stages of an ML workflow can be divided into the following steps.

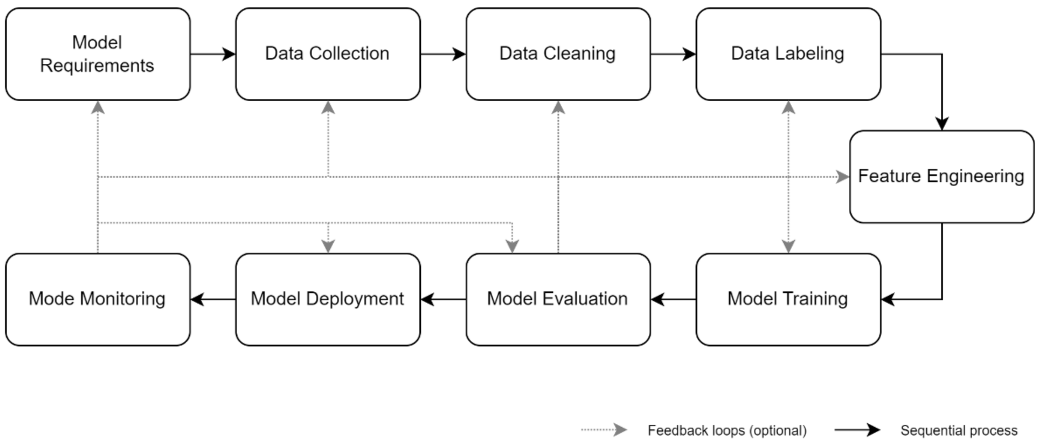


Figure 1: Stages of ML Workflow (Amershi et al., 2019)

1. **Model Requirements:** Deciding where to implement ML for the solution of a problem. Depending on the problem, suitable requirements are defined.
2. **Data collection:** One can use either already available internal data or collect new data. Already available data can include internal datasets or, if appropriate, generic open-source datasets.
3. **Data cleaning:** Removing noisy and inaccurate records, for example, removing outliers or replacing missing values.
4. **Data Labeling:** Supervised learning techniques need labeled data to learn a generalized representation of this respective data (Arqane et al., 2021, p. 3). However, this step might not be required when other learning techniques are used.
5. **Feature engineering:** Input data for ML models are referred to as features. Raw data must be transformed into suitable features for the respective ML model.
6. **Model training:** ML models are trained and tuned on the engineered features.

7. **Model evaluation:** The model is tested on test datasets using pre-defined metrics. In addition, this might involve human evaluation in critical domains.
8. **Model deployment:** Deployment of the ML model on devices.
9. **Model monitoring:** Continuous monitoring to detect errors during inference and operations.

Figure 1 shows that machine learning workflows are not linear, much more they are highly iterative and include several feedback loops (Amershi et al., 2019, p. 293).

Important to note is that ML models are probabilistic and therefore the development of ML models is different from traditional software development, where functions and outcomes are explicitly programmed and outcomes are deterministic (Cobbe & Singh, 2021, p. 7). ML is based on the provided data, its processing, model configuration and parameters, and consequently the performance depends on these properties and their adjustments. Additionally, there are also trade-offs concerning ML. For example, investigated trade-offs in the field of AI include a trade-off between privacy and performance in differentially private ML (Cristofaro, 2021, p. 27), a trade-off between training and inference energy consumption and accuracy ((Brownlee et al., 2021, p. 17) or trade-offs between quality and efficiency in ML-based text processing (Baeza-Yates & Liaghat, 2017, pp. 903–904). These technical trade-offs are inherited by AIaaS as well.

2.3. AIaaS

Overall, AIaaS can be described as a combination of cloud computing and AI. In practice, most of AIaaS systems are based on ML (Javadi et al., 2020, p. 301). However, in this thesis, the term AIaaS is used to adapt to the terminology observed in practice (Javadi et al., 2020, p. 301). In combination with the cloud computing model, AIaaS enables an easy, flexible, and on-demand integration of AI capabilities in customers' applications. In general, AIaaS abstracts away the complexity of developing an AI model and customers can focus on the data and the business problem instead of dealing with the implementation and the underlying computer infrastructure (Ribeiro et al., 2015, p. 896).

AIaaS services can be categorized in three layers depending on the abstraction level of the service (Lins et al., 2021, pp. 442–444). This is equivalent to the three service models in cloud computing. In detail that are (i) AI software services, (ii) AI developer services, (iii) and AI infrastructure services. AI developer services and AI infrastructure services don't abstract away details of the implementation of AI capabilities and rather are assisting developers in creating AI capabilities purely from scratch. AI developer services provide tools for assisting the coding of AI capabilities such as Jupyter Notebooks, and additionally including AI frameworks such as TensorFlow. Whereas AI infrastructure services provide raw computing resources specifically suited for AI applications. However, the majority of AIaaS services today can be categorized as AI software services that hide certain details from the customer such as infrastructure details and the computing environment to make the development process easier (Geske et al., 2021, p. 10).

These software services can be further divided into services that provide software to assist users along the process of building an ML model, often referred to as MLaaS (Ribeiro et al., 2015, p. 896) and those that offer pre-trained models in the cloud where customers can send input data to receive back predictions by the AI model (Javadi et al., 2021, p. 598). These services are also known as "Prediction-as-a-service" (Javadi et al., 2021, p. 598; Lins et al., 2021, pp. 443–444). MLaaS can be described as a platform for building, training and deploying ML models (Philipp et al., 2020, p. 399). As such MLaaS provides a generic architecture for the development and deployment of one or more ML algorithms (Ribeiro et al., 2015, p. 897). Nevertheless, this categorization is not mutually exclusive and since providers offer a wide range of AIaaS services, some services are a combination of prediction services and MLaaS (Cobbe & Singh, 2021, p. 7). For instance, there are AIaaS services offering pre-trained models that can still be customized by the customer.

Depending on the particular AI software service, the processing chain and the actors involved differ. According to Cobbe and Singh (2021), the AIaaS processing chain involves at least two entities, namely providers and customers (Cobbe & Singh, 2021, p. 13). Customers of AIaaS directly interact with the AI software services either for adding functionality to their applications that are in turn used by third-party end-users or for processing activities performed by end-users. In this thesis, however, a broader view is taken as customers and end-users can be the same entity in the case of AIaaS. AIaaS targets users of different roles, including traditional ML engineers and data scientists or non-expert users such as salespeople, citizen data scientists or business stakeholders (Wang et al., 2021, p. 7). In the case of non-experts, customers are often end-users of the AI service. This is especially true for AI inference services, where the user directly uses the pre-build AI service to generate ad-hoc predictions instead of developing AI services (Geske et al., 2021, p. 8). For example, Microsoft offers easy-to-use services, called AI Builder, that are targeted at end-users to process documents with the help of ML.[1]

3. Research Approach

Since there is not much previously published research in the field of AIaaS, especially on the topic of trade-offs within AIaaS service design, a qualitative research approach is chosen to explore characteristics and possible trade-offs between these characteristics in depth (Myers, 2013, p. 28). In comparison to a quantitative research approach, qualitative research "places more emphasis on the study of phenomena from the perspective of insiders" (Lapan et al., 2012, p. 3). This research is based on inductive reasoning, which is commonly used in qualitative research (Myers, 2013, p. 45). The inductive approach can even be used "if no theoretical concepts are immediately available to help you grasp the phenomenon being studied (Linneberg & Korsgaard, 2019, p. 263)". Whereas deductive reasoning is not a suitable approach for this research as there does not exist a theory to confirm. First, data is collected and then analyzed to explore and find emerging patterns (Myers, 2013, p. 45). The chosen data collection method

[1] https://powerapps.microsoft.com/en-us/ai-builder/

in this thesis is expert interviews as these allow valuable insights into practice from an AIaaS provider's viewpoint. In the following, the data collection and research method are described in detail.

3.1. Interview Partner Acquisition

To derive insights from practice and gain a deeper understanding of AIaaS characteristics, related trade-offs and challenges, employees of AIaaS providers were selected to be interviewed. There were no particular requirements for a specific position or job role to facilitate an extensive insight and to include technical, socio-technological and economical aspects into the discussion. However, the selected participants had to have several years of experience in AI in technical or business roles. For acquisition, a flyer with information about the interview process and the topic of research was created (see Appendix A). To get a suitable sample group and high-quality insights, the professional network LinkedIn[2] was used. The search function was leveraged to identify professionals that work at CSPs that offer AI capabilities. The results were filtered to German-speaking countries. Third-party consultants, users of AIaaS, and software providers that provide products where AI capabilities do not represent the key product feature were excluded to get a focused understanding of the providers' viewpoint. In addition, the website of the German AI Association[3] was used to identify promising start-ups within the field on AI platforms or AI services in contrast to the huge, well-known cloud providers. This led to 73 identified, possible interviewees who were contacted via LinkedIn messages or E-mail. After sending a short description and the flyer, six professionals agree to be interviewed. An overview of the interviewees and detailed information can be found in Table 1.

Table 1: Overview of Interviewees

Pseudonym	Date	Duration (h:m)	Professional experience with AI / Cloud / IT in years	Job Title	Size of Company
i01	19.07.2022	01:13	11	Cloud Solutions Architect	Large
i02	27.07.2022	0:42	2,5	Head of Product	Start-up
i03	29.07.2022	0:53	14	Cloud Solutions Architect	Large
i04	09.08.2022	0:49	15	Chief Technical Officer	Start-up
i05	29.08.2022	0:54	9	Product Manager	Large

[2] https://www.linkedin.com/
[3] https://ki-verband.de/

i06	23.09.2022	0:46	7,5	Product Manager AI/ML	Large

3.2. Interview Conduction

The one-to-one interviews were conducted in the period between July 2022 and September 2022. All interviews were carried out online via Microsoft Teams. Therefore, the setup did not need any special requirements and was consistent across all interviews. At the beginning of each interview, the interview process and structure were presented to the participants, and they were asked if they agreed to be recorded. In every conducted interview the participants agreed verbally to the voice recording of the interview to allow an accurate transcription afterwards. To answer the research question, semi-structured interviews were selected as the most appropriate method, following Myers (2013) approach. Semi-structured interviews allow some pre-formulated questions, and they are still consistent across interviews to a certain degree. However, the interviewee is able to talk freely and new questions can emerge during the interview (Myers, 2013, p. 187). An interview guide (see Appendix B) was created, structured as follows: First, the participants were asked to provide general information about their job role, their company, experience, and background. This was done by asking structured, pre-formulated questions. Second, the interviewees were asked about their understanding of the term AIaaS to get a shared agreement for the rest of the interview and address the different terms and concepts encountered in practice (Lins et al., 2021, p. 442). The following part included questions regarding typical characteristics of AIaaS and the description of different AIaaS offerings. For the final section questions about the experienced trade-off between these characteristics and further challenges were included. Each of the three parts contained a brainstorming question at the beginning so that the interviewees were able to freely express their thoughts without being influenced by the interviewer's questions. This brainstorming part was continued with follow-up questions which varied across the interviews as these were based on the previous answers and mentioned characteristics of the respective interviewee. In the end, participants were asked again if they had experienced additional trade-offs or challenges after discussing pre-formulated trade-offs and characteristics. The interview ended if there were no additional trade-offs or challenges to be mentioned. The interview guide was adapted during the course of interviewing to integrate new knowledge learned and ask questions about newly emerging concepts. This process is also known as theoretical sampling (Corbin & Strauss, 2015, p. 136). On average interviews took 52 minutes to complete.

3.3. Interview Transcription

To simplify the analysis, the interviews were transcribed after conduction. The transcription was done by using the software *f4transkript*. Each interview was transcribed, adapting and slightly adjusting the transcription rules from Dresing and Pehl (2018). In detail, the following rules were applied:

- Interviews are transcribed literally; informal contractions and dialects are translated to standard language if possible.
- The interviewing person is marked by an “I:”, the interviewee by a “B:”. Every contribution is indicated by a new paragraph with a blank line in between.
- Incomprehensible words are marked by (unv.). If a particular word can be assumed it is noted, followed by a question mark
- Discontinuations are marked by /
- Longer breaks are marked by (…)
- Punctuation and stuttering are smoothed in favor of readability
- Reception signals that do not interrupt the other participant’s flow of speech are not transcribed

The interview transcripts were pseudonymized so that participating interviewees and their companies remain anonymous. Therefore, the names of the interviewees, names of mentioned products, the interviewees’ employer, or other personal references were replaced by a more general, but descriptive term, placed in square brackets. For example, the fictional sentence “We at Google have introduced …” would have been replaced with “We at [large, international cloud provider] have introduced …”.

3.4. Interview Analysis

After transcription, the interviews were analyzed using techniques proposed by Corbin and Strauss (2015). For the analysis, the software *f4analyse* was used. The aim of this thesis is the description of AIaaS characteristics and trade-offs and additionally conceptual ordering (Corbin & Strauss, 2015, p. 59). That is, “organization of data into discrete categories […] according to their properties and dimensions” (Corbin & Strauss, 2015, p. 61). Thus, no complete construction of grounded theory was pursued and consequently not every proposed step of grounded theory was followed (Corbin & Strauss, 2015, pp. 215–309). However, the steps which were taken to derive results will be explained in more detail in the following.

First, the open coding technique was used. Each interview was scanned line-by-line and to each relevant text passage an open code was assigned if there was no matching code already present. This technique can be described as an inductive approach (Linneberg & Korsgaard, 2019, p. 263). By first using basic-level concepts as a foundation for further analysis one assures to stay close to the data and to make sure to preserve details as well as variations of the empirical data (Corbin & Strauss, 2015, pp. 75–77). The initial aim is to fracture the raw data and explore relevant concepts. If different text passages described the same concepts, they were assigned the same open code. If there was no existing code that described the concept a new code was created with a short description of its meaning (Corbin & Strauss, 2015, pp. 220–238). This process was conducted iteratively. In several rounds of open coding, code descriptions and the meaning of the codes were expanded as well as edited, and similar open codes were merged. For example, the phrase "the idea with the services is that they can be used by someone who also has no data science knowledge" (i05, paragraph 26), was coded as “Complexity Abstraction” and the phrase “the highlight would be if you can customize the model to your use case […] a transfer learning from a generic, pre-trained model to its specific use case" (i01, paragraph 12) was assigned to the code “Customizability”. Identified concepts were divided into five top-level categories, namely conceptual understanding, foundations, characteristics, trade-offs, and challenges. After several rounds of open coding, 187 text passages were assigned to 10 different open codes regarding characteristics. Similarly, a total of 89 text passages regarding trade-offs were coded, yielding to 9 differing open codes in the category of trade-offs. Lastly, 48 coded text passages made up 9 identified concepts concerning general challenges of AIaaS providers.

After the open coding step, axial coding was applied to delineate concepts gathered in the first step along different dimensions (Corbin & Strauss, 2015, pp. 156–160). This supported a broader understanding of characteristics, challenges, and trade-offs. The axial dimensions included conditions, action-interactions and consequences as proposed by Corbin and Strauss (2015). For example, during open coding the sentence “if you use these higher-level AI services the development takes much less time” (i03, Paragraph 28) and the sentence “the main aspect there lies on automation so that you do not have to search a model and set up trainings” (i02, Paragraph 22) were both coded as “Complexity Abstraction” in a first step. During the axial coding step, the interdependencies between different aspects of “Complexity Abstraction” were analyzed in depth, leading to an action-type subcode “Automation” for the latter sentence and to a consequence-type subcode titled “Rapid Development” for the former sentence. However, axial coding only provides a broad framework of which relations should be analyzed and only supports but should not completely pre-define the analysis process (Corbin & Strauss, 2015, pp. 156–160). Thus, new codes were formed when appropriate, irrespective of the before mentioned dimensions.

Additionally, supporting techniques were used to ease the analysis process and to make it more structured, namely constant comparison and memoing. Constant comparison is used to constantly compare codes and concepts leading to better-defined concepts with clear differences and connections (Myers, 2013, p. 166). Memoing is about writing down important notes concerning findings, meaning and interpretations that can be crucial for describing the results at a later stage (Corbin & Strauss, 2015, p. 136). In addition, the visualization tool *Mural*[4] was used to identify and visualize interdependencies between different characteristics.

After this final iteration, the 10 open codes related to characteristics were delineated into 48 additional axial codes. Regarding the 9 conceptual trade-offs, 18 axial codes were found. Lastly, challenges included 21 additional axial codes.

4. Trade-offs between AIaaS Characteristics

4.1. AIaaS in Practice

[4] https://mural.co/

In the interviews, it became apparent that there exists a wide range of AIaaS offerings with diverse characteristics. As outlined in Section 2.3, AIaaS services differ in the level of abstraction provided. The following chapters sometimes consider very different services, which is why this section will briefly clarify what these fundamental differences are in terms of users and value proposition. Since most large cloud providers offer a multitude of services, there is no clear separation of services. Services range from pure inference services (i02) and managed high-level services (i01) to customizable pre-trained services (i03, i05) and MLaaS platforms (i02, i06). Large cloud providers offer extensive platforms where a variety of AIaaS services are offered at the same time (i03).

These services target different customers that have unique and diverse requirements (i02, i03). Business users, domain experts or business analysts might use AI services directly for ad-hoc data analysis (i02) or for simplifying business processes by leveraging AI capabilities (i05). Therefore, they represent end-users in this specific case, who are also called "Citizen Data Scientists" (i02). The value proposition tends to be the provision of pre-trained and ready-to-use services to ease business processes (i03, i05). Contrastingly, other users include developers who aim to integrate AI capabilities into their applications that are in turn used by third-party end-users. These developers have at least some coding skills and do not necessarily need no-code platforms (i02, i03). Lastly, MLaaS services are targeted at more data-savvy customers, e.g., data scientists, data engineers, or ML engineers (i02). MLaaS supports the development and deployment of ML models which might be used by external or internal third parties. The value proposition of these platforms is to help experts work more effectively and bring ML to production. (i03, i06).

4.2. Characteristics of AIaaS

Since service offerings in the field of AIaaS are diverse, the following list is certainly not exhaustive. Rather, it is intended to provide an overview of which characteristics have been addressed repeatedly during the interviews. The findings were grouped into 10 main characteristics, namely centralization, cloud characteristics, complexity abstraction, customizability, deployment and integration, functionality, genericness, performance, security and privacy and trustworthiness. An overview of the core characteristics and respective aspects can be found in Appendix C. Some characteristics might be only applicable to one certain type of AIaaS software services, namely AI inference services or MLaaS, whereas other characteristics are typical for both types of services. Characteristics are these parts of an AIaaS offering that are significantly influenced by design decisions of the providers and determine the value proposition of the respective AIaaS offering.

C1: Complexity Abstraction. The characteristic complexity abstraction relates to the simplification of the usage, development, deployment, and operation of AI capabilities. This simplification is especially achieved by the **automation** of the ML workflow and the implementation of **user-friendly interfaces**. Further, key aspects of complexity abstraction incorporate **provider-managed infrastructure and**

platform, **pre-configured AI components** as well as **managed AI services** offered by the provider, see Table 2 for an overview.

Table 2: Aspects of the Characteristic Complexity Abstraction

Complexity Abstraction	Simplification of the usage, development, deployment, and operation of AI capabilities.
Managed AI services	Ready to use services for inference, no required customization upfront
Pre-configured AI components	Task-specific development kits and pipelines developed and provisioned by the AIaaS provider.
User-friendly Interface	Services provide easy-to-use and intuitive web interfaces, sometimes even with a no-code or low-code interface.
Automation	Automation of the whole or parts of the ML workflow.
Managed Infrastructure and platform	The provider configures the necessary underlying infrastructure and environment.

On the highest abstraction level, there are managed AI services. These services are ready-to-use, turnkey solutions for generic problems and don't need any customization or configuration and can be directly used to get basic predictions (i01, i03, i05). As such, they abstract away almost every complexity within the ML workflow (i01, i05), even the challenging steps of data collection and data cleaning are taken over by the AIaaS provider (i01). In addition, the underlying models get constantly updated by the AIaaS provider to provide state-of-the-art technologies and integrate new knowledge into the service (i03). Examples of such managed AI services include translation services, text-to-speech services, or emotion detection in images, among others (i01).

Another way to abstract complexity lies in providing pre-configured AI components. These components can be freely adapted to individual needs but provide a basic framework for the development of AI models (i05). This can additionally include pre-configured, task-specific AI workflows (i05, i06). For example, a pre-configured pipeline for a visual inspection service where the user can upload his custom data (i06). The model selection, training and tuning are pre-configured and thus static and automated.

Another aspect of complexity abstraction is user-friendly interfaces which are especially targeted at people with no prior coding experience: "what we have as a principle is that the things can also be operated by people who do not have a lot of coding experience" (i03, paragraph 20). It is possible to develop and adjust ML models without coding and instead one can use the platform's web interface (i03). The ease of use is further increased by integrating complementary functionalities directly into the platform's web interface, this includes data labeling and data collection (i01, i03). An example is the creation of an AI-question-answering system based on a knowledge base that is maintainable through a web interface (i03).

Furthermore, the automation of ML can contribute to complexity abstraction, and is also known as Automated machine learning (AutoML). AutoML “can be understood to involve the automated construction of an ML pipeline on the limited computational budget“ (He et al., 2021, p. 1). This functionality is consequently found on ML platforms where users can develop custom models (i01). Theoretically, AutoML can automate multiple steps in the machine learning pipeline, starting at the data preparation step, and continuing to feature engineering, model generation, as well as model evaluation (He et al., 2021, pp. 1-2). Within AutoML data is usually provided by the customer (i05). Depending on the system, some providers implement an automated test of uploaded data as a data preparation step (i02). The user specifies the optimization objective (i02, i01) and can provide further information about the data to support the AutoML system. Depending on the respective system, it automatically performs feature engineering and model generation steps by trying different training algorithms, models, as well as hyperparameters (i01, i02) and proposes optimal models after completion. This automation process uses default values (i01), but the process can be tuned by the user to increase the efficiency of the optimization process or adapt the training and optimization to the users’ restrictions regarding time or computing consumption (i03).

Based on the answers of several experts, MLaaS represents the lowest abstraction level (i03, i05, i06). AIaaS providers provide a managed environment and platform to ease the development, training, and deployment of ML models (i06). Since the provider manages the underlying infrastructure and environment this complexity is transferred to the provider, assuring high availability, security, and scalability (i03, i06). Consequently, "the individual product team no longer has to worry about this as much. Instead, they focus more on the real product, on the ML core" (i06, paragraph 16).

These different aspects of complexity abstraction can be combined, for example, AutoML and an easy-to-use graphical user interface (GUI) (i03, i06). Because of complexity abstraction, AIaaS can be used by everyone, also referred to as the democratization of AI (i03). Business users can leverage AI services and can use AI without in-depth knowledge. This is an advantage as they do not rely on data scientists or ML experts and have better business understanding. But even experienced data scientists can benefit from complexity abstraction: "[...] what it offers is, first of all, a basic security" (i01, paragraph 38). It can support the user, uncover mistakes in the ML pipeline, and especially increases the productivity of a data scientist, for example, AutoML can be used to quickly evaluate the suitability of a dataset for a specific task or model (i03).

C2: Customizability. The characteristic customizability refers to the ability of AIaaS to be adaptable by customers according to their individual needs, their expertise, and the availability of high-quality training data. The importance of customizability for the usefulness of the AI service is emphasized by several experts (i01, i03, i06). As such, one possible customizable property of AIaaS is the underlying pre-trained model itself by adapting the model to customer-specific data, commonly referred to as **domain adoption** (i01, i03). Another customizability option is the **customization of pre-configured AI**

services or the **AutoML** process. Furthermore, some AIaaS services allow manual development of **custom models** for user-specific use cases (i02, i06). Table 3 provides an overview of the three different aspects of customizability.

Table 3: Aspects of the Characteristic Customizability

Customizability	Ability of a service to be customizable to specific needs of a user.
Custom Models	Users can train and develop their own custom models from scratch on MLaaS platforms. This includes custom modeling, custom algorithm selection, custom model tuning and custom deployment.
Custom AutoML	Users can customize AutoML processes and functionalities of the platform to adapt the AutoML process to specific requirements or increase efficiency and performance.
Domain Adaption	Ability to adapt the underlying model of a service to a different domain by leveraging transfer learning techniques.

As outlined in the paragraph about complexity abstraction, MLaaS platforms abstract away only configurational, operational, and infrastructural tasks, leaving the development of purely custom models to the user. Therefore, the users can implement their own ML models by using their preferred libraries and programming languages that are provided on the platform (i03, i06). Thus, the users can fully customize the models and consequently the resulting service.

Additionally, providers can offer customization options regarding the AutoML process. Even though AutoML aims to automate an end-to-end ML workflow, users can benefit from the customization of this process (i02, i03). Customization can accelerate the search for an optimal model (i02) or enhance the performance of the final model (i01, i03). For example, if the feature selection step is performed manually by the user (i06). Additional possible adjustments include the training, tuning and hyperparameter optimization as well as deployment of the final model (i01, i06) so that the user has more customizability options as opposed to pre-build services.

The usage of context-specific training data to adapt underlying pre-trained models of an AIaaS service to a customer-specific domain is another customizability option. This technique is called transfer learning or fine-tuning (i01, i03). Transfer learning is an approach to improve the performance of the model predictions in the target domain by transferring knowledge that is contained in related source domains (Zhuang et al., 2021, p. 43). To fine-tune a model only a few samples are needed which poses an advantage to models trained from scratch (i03). An illustrative example of a use case is adapting an AI translation service to a customer-specific jargon that includes technical terms or proper nouns by providing additional samples as training data (i01, i03).

The different techniques to provide customizability mainly differ in the need for customer-specific training data and the use of pre-trained models. Custom models and custom AutoML models purely rely on data provided by the customer (i03, i04), whereas offering domain adoption relies on pre-trained ML models and usually only a small sample set of data provided by the customer (i01, i03).

C3: Genericness. Genericness refers to what extent the service can be used for different ML tasks or within different domains and use cases (i06). Thus, it is directly correlated with the size of the target group of a respective AIaaS offering (i01, i06). In order to align with the cloud service business model, i.e. automatic self-service without manual provider configuration (Mell & Grance, 2011, p. 3), service providers generally try to target a wide customer base and thus offer generic services (i01). As shown in Table 4, genericness incorporates two dimensions, namely the dimension **task-applicability** and the dimension **domain-applicability**.

Table 4: Aspects of the Characteristic Genericness

Genericness	The extent of how generic a service is in terms of applicability to different use cases and solving diverse ML tasks.
Task-applicability	A service can be task-specific and focus on one specific ML task or task-agnostic to support the development of a variety of AI capabilities.
Domain-applicability	Whether the service includes business-specific knowledge (domain-specific) or is not targeted at one industry (domain-agnostic).

First, a service offering can be domain-specific or domain-agnostic (i02). In the former case, the service is adapted to a specific industry and thus the service is specialized in terms of industry-specific data, data structures, or specific requirements (i02). Whereas domain-agnostic services are not specialized to a specific domain and therefore are generically applicable in several business cases (i06). These services normally address generic problems, for instance, translation services, text-to-speech, and face detection services (i01) as these problems are “so broad that it impacts a huge mass” (i01, paragraph 12) of potential customers.

Contrastingly, there are managed AI services that include domain-specific knowledge (i05). These offerings include pre-trained models (i05). For instance, Google’s pre-trained Vision AI[5] offers a CV service to classify images according to pre-defined labels by the provider or read printed text in images. Since the service only can assign labels, it was trained on, this knowledge must be integrated by the provider at training time.

Second, services can either be task-agnostic or task-specific. A task-specific service supports users in developing models for one specific ML task, such as visual inspection (i06), whereas task-agnostic services support the user across different ML tasks and thus are generically applicable for a multitude of tasks, for example, AutoML can be used for different ML tasks (i06).

[5] https://cloud.google.com/vision

It is important to note that customizable services are more generic as these services allow adaption to different industries (i05). Therefore, the characteristics of genericness and customizability are interconnected to each other. Another observation is that these dimensions are not discrete as there is a multitude of different services, rather some services are more generic whereas other services are more specialized.

C4: Trustworthiness. Trustworthiness refers to all aspects implemented by AIaaS providers to establish trust in AIaaS and foster the responsible usage of AI. Trustworthiness is a key research stream within AI research (Kaur et al., 2023, p. 1) and therefore inevitability is present in AIaaS as well. I integrate several aspects into trustworthiness based on the experts' responses, such as **fairness, ethics, explainability and interpretability** as well as **transparency,** see Table 5. Trustworthiness is particularly important as AIaaS can "exhibit errors, biases, inequalities, and other problems, which, through AIaaS, could be reproduced at scale" (Cobbe & Singh, 2021, p. 10). A term closely related to trustworthy AI that emerged during the interviews is responsible AI (i03).

Table 5: Aspects of Characteristic Trustworthiness

Trustworthiness	Functionalities and properties of AIaaS that aim to increase the customer's and end-user's trust in AIaaS.
Fairness	Functionalities to monitor the fairness of an AI service.
Ethical AI	To what extent the services are ethical and methods to ensure the ethical use of AI services.
Explainability and Interpretability	Functionalities that make inferences as well as the development process of services more interpretable and explainable.
Degree of Transparency	To what extent the services are transparent to the user.

Fairness is mainly based on the underlying data and the resulting model as a representation of the training data (i01). In ML research, fairness is about mitigating inherent algorithmic biases that are caused by historically biased datasets that possibly include proxy attributes, missing data or algorithmic objectives (Pessach & Shmueli, 2023, pp. 2–3). To support users in producing fair models, providers offer frameworks that can be applied to check for user-defined critical attributes (i01). For example, a user can control attributes so that "your salary forecast is definitely gender independent" (i01, paragraph 30). Explainability and interpretability are mentioned in several interviews and, according to the experts, explainability and interpretability features are getting more and more important (i03, i01). However, the importance is also dependent on the specific use case and the user (i01, i05). Depending on the use case, explanations may be irrelevant for end-users of AIaaS services (i05) whereas at the same time they are highly relevant for the service customer, for example for solutions developers or data scientists (i03). Explainability and interpretability methods should therefore be adapted to the user's experience level and needs (i05). To increase explainability and interpretability several service providers implement dashboards and interpretable ML frameworks (i01). The dashboards show different metrics regarding interpretability and additional performance measures, e.g., confidence scores, or feature importance (i01, i02, i03, i05). These metrics also contribute to the system's transparency. Further, providers mentioned that testing functionalities of the service can increase users' trust in the service (i01, i04).

When service providers take over steps of the ML workflow related to data or modeling in the development of AI services, they are at least partially responsible for establishing ethical AI guidelines (i05). However, this is not necessarily true in the case of MLaaS where custom models are trained. Another exception is the post-processing of the predictions as this is out of the providers' control (i05).

Closely related to the transparency are privacy and security of an AI service offering (i01), but the fact that it is offered as a cloud service adds an additional dimension to these concepts, as privacy and security are simultaneously associated with both cloud and AI. To address this complexity, it was coded as an additional characteristic.

C5: Security and Privacy. Security and Privacy are well-known issues in the field of cloud computing (Xiao & Xiao, 2013, p. 844). These two concepts are closely related to each other and regarding cloud services they can be subdivided into the attributes confidentiality, integrity, availability, accountability, and privacy-preservability (Xiao & Xiao, 2013, p. 845). Since AIaaS is typically offered in the cloud, AIaaS providers are responsible for ensuring that these attributes are satisfied. For example, AIaaS providers are taking the following measures.

Security is assured by AIaaS providers by data governance concepts:

> "If you offer such an AI-as-a-service platform, you also need a clear data governance concept, which means you need different classification levels of data confidentiality and also who is allowed to access which data and why" (i06, paragraph 58).

These concepts contribute significantly to integrity, confidentiality as well as accountability and help to establish standards (i05, i06). According to several experts, the security in the cloud is higher compared to private deployment, as AIaaS providers employ dedicated experts only to tackle security issues (i03, i06). Thus, security measures are, in most cases, far superior compared to the ones which are implemented by the customers themselves (i03, i06). Consequently, AIaaS offerings can offer advantages in security as the customers don't have to implement their own security measures and this is taken care of by the provider.

Privacy is highly use case dependent, and ensuring privacy is especially important if personal identifiable information (PII) is transferred for training or inference (i01, i04, i05). To enhance privacy protection, cloud providers generally encrypt data (i06). Other means to assure privacy include product standards that incorporate user protection (i05).

Nevertheless, privacy and security within AIaaS are particularly challenging compared to data in the cloud. AIaaS relies on an interaction between, at least, two parties that exchange data via a network for inference and models incorporate training data. Thus, AIaaS possibly enables new attacks on sensitive data, such as membership inference attacks or model extraction attacks (Tanuwidjaja et al., 2020, pp. 167442–167443).

Table 6: Aspects of the Characteristic Security and Privacy

Security and Privacy	Implemented measures to assure data security and privacy.
Security	AIaaS providers are responsible for security measures that are often more secure than the customers' own security measures.
Privacy	Methods and standards to increase privacy within AIaaS. The importance of privacy depends on the application area.

The concepts of privacy and security are closely related to the characteristic trustworthiness. As standards and transparent regulations about data security and privacy increase the overall transparency of the service and consequently foster trust in the AI service (i01). Table 6 briefly summarizes mentioned aspects of Security and Privacy.

C6: Deployment and Integration. The characteristic of deployment and integration refers to how the AIaaS service is provisioned to the customers and how they can integrate the model into their applications and systems. Thus, these characteristics are mainly related to the inference functionality of a deployed AI model. In the interviews, several aspects were mentioned, namely **inference interface**, **deployment model**, **processing type** and **latency**. These aspects are presented in Table 7.

Table 7: Aspects of the Characteristic Deployment and Integration

Deployment and Integration	How the service is made available to the customer and end-user and how an AI service can be integrated into existing applications.
Inference interface	Provision of the service is possible via different interfaces including REST APIs or web interfaces.
Deployment model	Different deployment types of AI services are possible, ranging from public clouds to on-premises deployment.
Type of Processing	The way inference requests are processed by the system, distinguishing between batch inference and stream inference.
Latency	Amount of time which is needed to process an inference request by the AIaaS service.

The inference interface of an AI service is about how the inference result of a query is provided to the customer. For example, AI services can be embedded in other applications to make them more intelligent and reduce manual workload, so-called embedded AI services (i05). In this case, the result is provided via a REST-API and there is no need for a dedicated user inference interface (i05). In the case that end-users interact with the inference service directly providers can implement a GUI adapted to the users' technical expertise (i01). For instance, a drag-and-drop translation service where users can upload a certain text document and get the result, all integrated in the same graphical interface (i01).

Another key differentiator is different deployment models concerning AI services ranging from public cloud to hybrid cloud to on-premises. Important to note is that there is a difference between the deployment of the platform for model development and the deployment of the model for inference. The development and training might happen on the providers' infrastructure, but eventually, the model is deployed offline for inference, so-called edge deployment (Leroux et al., 2022, pp. 1003–1004). This deployment type is well-suited for use cases that are latency-critical (i01).

Latency is another aspect of deployment and integration. Depending on the point of integration latency is critical (i01). Latency relies on the underlying infrastructure since more computing power can reduce latency and thus latency is closely connected with scalability (i01).

Lastly, there are two main processing types, stream processing and batch processing (i01, i04). This design decision influences the providers' service inference architecture (i01). Batch processing refers to the repetitive execution of training and inference but is not continuous (i01, i04). In contrast, stream processing requires a continuous running endpoint that is dynamically scalable depending on the actual request load. Hence, it is more complex and expensive (i01).

C7: Performance. Performance incorporates multiple dimensions, depending on the regarded technical aspects of an AI service. Thus, central characteristics that were mentioned during the interviews and were grouped to the performance characteristic during analysis include **training performance, model performance** and **model complexity,** refer to Table 8**.**

Table 8: Aspects of the Characteristic Performance

Performance	Concerning the quality and speed of inference requests to the deployed AI model and the training process.
Training performance	Duration and computational costs of training.
Model performance	The quality of the model in terms of context-specific metrics.
Model complexity	Size of the model regarding adjustable parameters and required computing resources for training and inference, e.g., storage and computing.

First**,** training performance describes the efficiency of the training process in terms of computational costs and time (i01). The training performance is for example determined by the model complexity (i01), or defined parameters during the AutoML process (i02, i03).

Model performance is about how effectively a specific task on previously unseen data is performed by the ML model. The performance can be measured by multiple performance metrics (i03), such as F1 score, accuracy, or precision. These metrics are task-dependent, as different ML tasks are evaluated differently (i01, i02). The performance of an ML model depends on several factors. One decisive factor is the training dataset and its representativeness for the specific use case domain (i01). Additionally, the complexity of the model (i01), feature engineering (i06), as well as tuning of the models' hyperparameters influence the model performance in the end (i02). Depending on the use case, customers have differing performance requirements (i03). The complexity of the ML model is another important aspect of performance (i01). Generally, model complexity refers to the size of an ML model, regarding its learnable parameters (i01). Complex models do need more resources for training and inference (i01).

C8: Centralization. A common characteristic of some of today's AI software services is that they are centralized to a certain degree and consequently under the control of the service providers (i01, i03, i05). Centralization is concerning data, models, and computing power. Moreover, providers typically offer other cloud services on their platform to complement the respective AIaaS offering (i06). Thus, the characteristic centralization was divided **into intellectual property (IP), control and observability** as well as **interoperability**, see Table 9**.**

Table 9: Aspects of the Characteristic Centralization

Centralization	AIaaS is centralized and thus under the control of the AIaaS provider, who benefits from the centralization.
Intellectual Property	IP includes large and complex models that require high investments in time, computing power and expertise. Additionally, training data is another part of IP since it is required for pre-trained models.
Control and observability	Control and track service usage for billing or improvement of services.
Interoperability	Interoperability with other products and services of the same service provider or third parties.

To offer pre-trained and turnkey solutions AIaaS providers must have access to high-quality, large training datasets (i01, i03, i04), for example in the field of NLP or CV (i03). These datasets are often not public and part of the provider's **intellectual property (IP)** and consequently represent a competitive advantage for the AIaaS provider (i01, i03). Further, there are pre-trained models that are likewise part of the providers' IP. Especially, recent large-scale language models such as Google's LamDA (Thoppilan et al., 2022) or OpenAI's GPT-3 (Brown et al., 2020) need particularly powerful computing resources and the computational cost of training such models is high (i01, i03). Thus, training large models is not feasible for smaller companies due to the lack of data, skills and time as well as limited financial and computing resources (i03). Recently, these models became also known as foundation models which are models "trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks" (Bommasani et al., 2021, p. 1). In the case of pre-trained and managed models this two aspects are the key value contribution: "Data or compute one of the two has to be a USP that you have an AI model that you can then offer to others" (i01, paragraph 12).

Pre-trained and centralized ML models have additional advantages, according to AIaaS providers (i01, i03). First, they incorporate the providers' knowledge across different use cases and customers (i03) and in turn, increase the quality of the model. Second, these centralized models can get adapted to changing environments without manual interaction of the customer as the service provider takes over the maintenance of such models (i03).

Another aspect of centralization is the ability of control and observability (i02). Providers can for example potentially track customer behavior on the platform (i02) or monitor user queries (i05). This data can then be used for maintenance of the models (i03), to assure more ethical AI applications (Javadi et al., 2020, p. 303) and for improving service offerings. Additionally, the control over the inference endpoints enables providers to easily charge users on a pay-per-use basis that is typical for cloud services (i01, i03, i06).

Centralization further enables the combination of AIaaS offerings with other services and products by the same cloud provider, leading to interoperability of AIaaS services. This is possible since most providers offer other complementary SaaS, PaaS, and IaaS services, for example, data storage. This can in turn increase the value contribution of AIaaS services (i06).

C9: Functionalities. MLaaS integrates a multitude of functionalities that support different steps of the ML workflow outlined in Section 2.2. These functionalities include **data-related capabilities, MLOps** and **integration of third-party providers**. Table 10 provides a short overview of aspects of functionalities.

Based on the conducted interviews data-related capabilities can be delineated into several aspects such as support with data cleaning and data preparation (i02) or human-in-the-loop approaches to increase training data quality (i02, i03).

Furthermore, a central functionality of MLaaS platforms is to facilitate model management, maintenance, and collaboration between actors (i02). For example, one expert mentioned during the interview: “iterative training and also model maintenance. […] That's just, that's just coming more and more" (i03, paragraph 28). MLaaS platforms integrate continuous training capabilities and thereby facilitate the integration of new knowledge as well as continuous improvement of the service by adapting to changing environments (i03, i06). Further, MLaaS platforms enable customers to update an ML model without the need for data scientists (i03). Observing a changing environment and constantly updating the model to account for that change is especially important if ML models are deployed in production in real-world use cases (i06). One expert refers to this as “CI for machine learning models” (i02, paragraph 32). These functionalities to manage and maintain ML models at scale in production align well with the recently emerging concept of Machine Learning Operations (MLOps). According to Kreuzberger et al. (2022) MLOps is a paradigm, concerning the end-to-end deployment of ML models in production and integrates multiple steps, such as conceptualization, implementation, monitoring, deployment and scalability. As such “ML Ops is aimed at productionizing machine learning systems by bridging the gap between development (Dev) and operations (Ops)” (Kreuzberger et al., 2022, p. 8). According to interview partner i06, MLOps is central to AIaaS with the objective of productionizing ML models faster and more effectively (i06). Additionally, MLOps fosters collaboration between different actors in the ML workflow (i02, i06).

Another functionality is the integration of third-party providers to build an ecosystem around the AIaaS platform (i01). This, for example, includes partners that integrate AIaaS in existing applications or partners that support customers in collecting training data (i01, i05).

Table 10: Aspects of the Characteristic Functionalities

Functionalities	Central features and functionalities that are provided within the MLaaS platforms to increase value proposition and extend the functionality of AI service.
MLOPs	Functionalities for ML lifecycle management at scale in production.
Data Capabilities	Functionalities to collect and annotate training data or increase its quality.
Integration of third-party providers	Build up an ecosystem around the platform to support customers in complementary tasks.

C10: Cloud. As most of AIaaS is hosted by CSPs on cloud infrastructure, AIaaS inherits typical cloud characteristics (i06), among others these are **scalability, pay-per-use,** and **on-demand-provision**, see Table 11.

Table 11: Aspects of the Characteristic Cloud

Cloud	AIaaS inherits typical cloud characteristics.
Scalability	Availability of lots of computing resources to scale services according to the requirements and usage.
On-demand	Off-the-shelf AI components are provisioned automatically.
Pay-per-use	Billing is based on the actual usage of the endpoint or platform.

Scalability is a central aspect of AIaaS (i06). Training of ML models is resource-intensive and outsourcing the training to a scalable cloud infrastructure can be beneficial for the customer (i01). Training can be accelerated or even actually made possible by available special computing resources such as graphic processing units (GPUs) or tensor processing units (TPUs) for the training of complex models (Lins et al., 2021, p. 446).

The on-demand aspect refers to the possibility that AIaaS is available without manual provider interaction and can be used off-the-shelf by the customer as it is provisioned automatically (i05, i06). Pay-per-use is the typical billing model for cloud services (Hogan et al., 2011, p. 14), and thus also used by AIaaS providers as a pricing model (i03).

4.3. Trade-offs and Service Design Implications

The identified characteristics in Section 4.2 can be used to differentiate AIaaS offerings and AIaaS providers try to optimize these characteristics during the design time of an AIaaS offering (i01, i05, i06). However, providers are not able to optimize all characteristics at the same time because they interfere with each other. This results in trade-offs and design implications that are described in the following in more detail. Figure 2 illustrates the identified trade-offs at the level of characteristics and in the case of two trade-offs at a lower level regarding only one aspect of a specific characteristic.

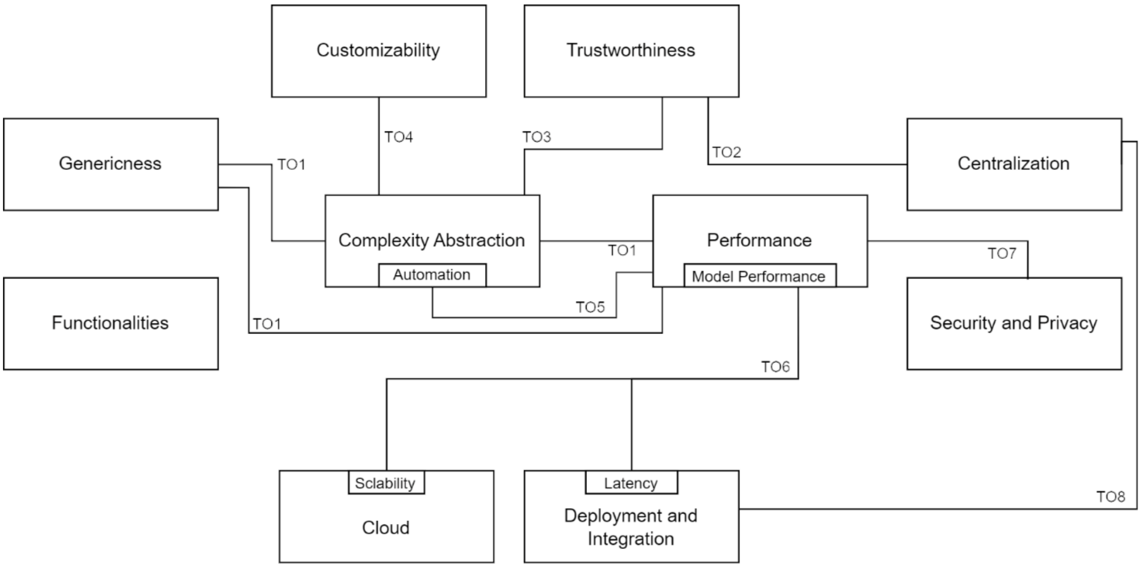


Figure 2: Overview of Identified Trade-offs between AIaaS Characteristics

TO1: Complexity Abstraction vs. Performance vs. Genericness. As an overarching trade-off, an AIaaS provider must decide on the degree of specialization. That is if the service offering should be generic or specialized in terms of the ML task and domain (i01, i02, i05). Specialization is closely related to performance and complexity abstraction. Services that are specialized for a particular domain generally result in higher performance on that respective domain without the need for manual customization (i02). An important question is which actor should adapt the service to specific use cases, as this could be either the customer or the provider. Ultimately, this results in different levels of abstraction.

Since these multiple interdependencies between the three characteristics complexity abstraction, performance and genericness are not clearly dividable, this paragraph describes a three-fold trade-off that is illustrated in Figure 3. For an AIaaS provider the three characteristics complexity abstraction, performance and genericness represent all desirable characteristics, but they cannot be optimized at once without sacrificing at least one characteristic.

First, an AI service that is generic and at the same time abstracts away all the complexities of the ML workflow might imply a loss in performance in special domains (Field A, Figure 3). Generic, pre-trained and fully managed services target wide and undefined use cases and are not purpose build (i01). These services provide basic turn-key AI capabilities such as document information extraction (i05), emotion detection within the field of CV and translation services in NLP (i01, i04). However, these services do not include business logic and are trained on generic datasets. Hence, they are only suitable to generate generic outputs (i01). For instance, in the case of a document information extraction service this might integrate fields like date and name. Another illustrative example would be a generic out-of-the-box translation service that is not adapted to use case-specific vocabulary. In that case, the quality of the model suffers as context-specific vocabulary might not be translated correctly (i01). In most cases, these generic outputs are insufficient for the customers' use case and therefore the performance of the model in terms of its applicability to a special use case is compromised (i05). Consequently, fully managed AIaaS services that prioritize complexity abstraction and genericness in terms of being domain-agnostic are sacrificing performance.

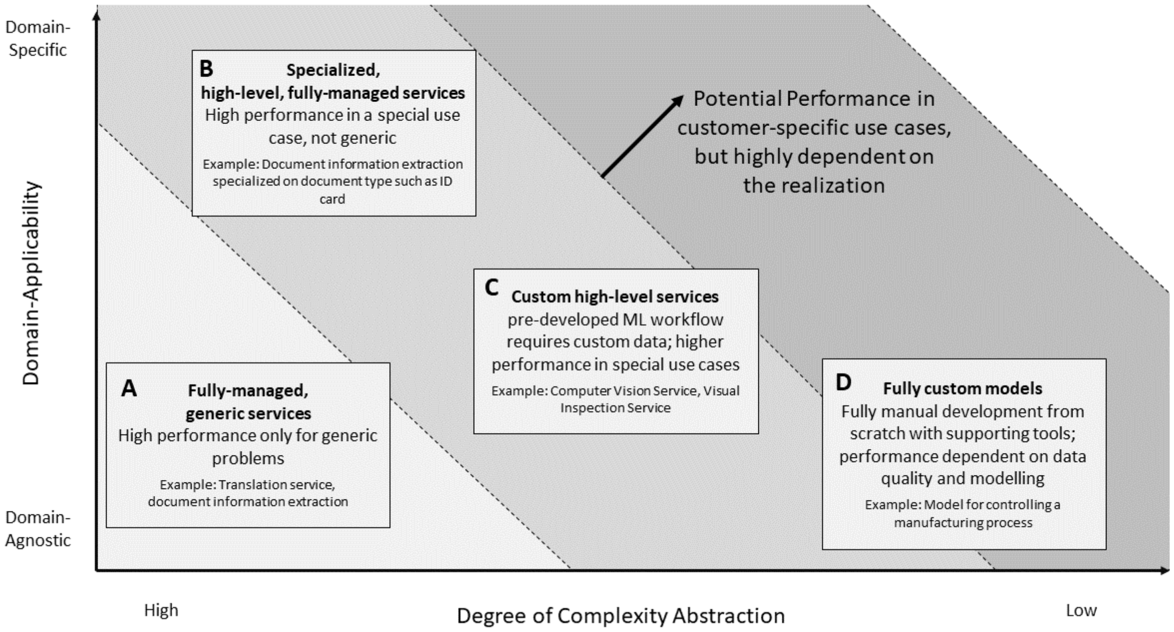

Domain-Specific
Domain-Agnostic
Domain-Applicability
High
Degree of Complexity Abstraction
Low
Potential Performance in customer-specific use cases, but highly dependent on the realization
A Fully-managed, generic services
High performance only for generic problems
Example: Translation service, document information extraction
B Specialized, high-level, fully-managed services
High performance in a special use case, not generic
Example: Document information extraction specialized on document type such as ID card
C Custom high-level services
pre-developed ML workflow requires custom data; higher performance in special use cases
Example: Computer Vision Service, Visual Inspection Service
D Fully custom models
Fully manual development from scratch with supporting tools; performance dependent on data quality and modelling
Example: Model for controlling a manufacturing process

Figure 3: Trade-offs between Complexity Abstraction, Performance and Genericness and resulting AIaaS Types

To solve this issue, AIaaS providers can offer more specialized AI software services that are more specialized and domain- specific (Field B, Figure 3). At the same time, specialized services are easy-to-use because they are managed by the AIaaS provider (i05). This is done by already integrating business logic into the service (i05), inevitably sacrificing the genericness of the service and therefore only targeting a smaller customer base (i02, i06). Specialized, provider-managed services are often based on pre-trained and purpose-built models (i05). One example is a document information extraction service specifically adapted to business invoices. This service is able to recognize typical elements in business invoices such as line items and headers (i05). Therefore, the service might perform well on business documents but cannot be used in other domains and is limited in its application (i06). Additionally, these services are turn-key solutions for specific business problems, as they do not require any configuration.

Thus, a provider trades in genericness for complexity abstraction and performance in one specific use case. A precondition for specialized services is the prior identification of relevant use cases and the integration of business logic into the service, which requires domain expertise and is a challenging task for providers (i03, i05).

Yet, some domains and ML tasks are too customer-specific and therefore business knowledge cannot be integrated in pre-trained models by the AIaaS provider in advance, leaving the adoption to the customer (i05) (Field C, Figure 3). For example, this challenge arises in the case of classifying customer support tickets since possible classes highly vary across different customers (i05). Accordingly, AIaaS providers must sacrifice complexity abstraction to a certain degree and cannot offer fully managed services. Therefore, AIaaS providers offer pre-developed ML workflows where the customers must contribute their own individual training data, thereby making the service customizable. This increases the complexity of service usage (i05) because the customers must perform data collection, data preprocessing, and data labeling steps (i03, i05). Two approaches can be differentiated. Customers can either adapt pre-trained models by leveraging transfer learning workflows or train a fully customized model based entirely on their customer-specific data (i05). Yet, both approaches still abstract away complexity as they are based on pre-configured pipelines (i03, i05, i06). The difference between these two approaches lies in the amount of training data required (i01). The first approach, domain adaption, might offer a promising balance between complexity abstraction and customizability as customers do not need to train models from scratch and thus need less training data (i01). In some cases, they need to upload only a few data samples to adapt the model to their customer-specific data (i01).

These services have the advantage that the genericness in terms of domain applicability can be preserved more easily (i01, i05). The reason for this is that the customer-specific training data matches the respective domain in which the services are used for after the model deployment (i01). However, these high-level AI services are still task-specific as AIaaS providers can only abstract away the complexities of certain ML workflows (i06). The providers are only able to abstract complexities in the model development if these are independent of the respective data provided (i06). For instance, ML tasks that can be offered as a higher-level, pre-developed service by abstracting away development complexities include CV models, object recognition models or chatbots (i04, i06). Take a visual inspection service as a compelling example. This service can be provided as a higher-level service because details of the training process can be abstracted away independently of the customers' varying training data (i06). That enables providers to offer generic services across different domains that at the same time have good performance in customer-specific use cases (i05, i06):

> "We provide […] something generic, you can use that out of the box, but if that's not enough for you, there's the possibility to extend or improve that […] with certain tools that we also provide" (i05, paragraph 38).

In some cases, however, the meaning of the data in the various use cases is so different from one to another such that modeling, and feature engineering are highly dependent on the respective use case. Thus, it is demanding for providers to offer a higher-level service:

> "But if it is really about different company data, i.e., really about business-specific process data, for example, some manufacturing process that is simply specific to this customer, then I don't see any abstraction potential at all, and then the data points also have a completely different meaning in terms of the domain" (i06, paragraph 32).

In these cases, purely custom models must be developed, requiring data science expertise and domain knowledge simultaneously during the modeling process (i06), refer to Field D in Figure 3. Performance in a domain-specific use case that is unique to the respective customer is prioritized and hence complexity abstraction is inevitably reduced in this type of AIaaS. Nevertheless, these offerings are generic, and providers can still abstract away details regarding operational tasks, and deployment tasks (i06). Further, they can offer the possibility to reuse certain ML pipelines and consequently support customers in developing purely custom models (i06). These services are thus categorized as MLaaS.

Therefore, an AIaaS provider must find the right balance between genericness, performance and complexity abstraction in AIaaS design. It is important to note that most of the services are specialized on a specific task, e.g., CV or text-to-speech, but not on a specific domain, leading to a horizontal, domain-agnostic service offering (i06). One promising implication for service design might be the integration of foundation models (i01). As these models can generalize well across different tasks and can be applied in different use cases (i01). For instance, Google's transformer-based neural language model can offer genericness and performance in special tasks at the same time (Thoppilan et al., 2022):

> "It is not somehow special and generic, it can do both, but it is extremely complex to train such a model. That is, you need a lot of computing power and very large amounts of data or perhaps also special challenges on which the model is trained" (i01, paragraph 18).

T02: Centralization vs. Trustworthiness. The centralization characteristic of AIaaS interferes with ensuring trustworthiness in multiple ways. First, to protect their IP in terms of datasets and models providers might conceal certain details (i01, i03), leading to a trade-off between protecting their IP rights and ensuring transparency. For example, the data on which a model was trained might be proprietary and is not publicly available. Large, high-quality training datasets are most likely not published to maintain competitive advantage (i01, i03, i06). This makes it more difficult to assess the suitability of the training dataset to a respective customer's use case. Still, it is important that the training dataset matches the domain where the services are intended to be used (i01).

Additionally, centralization enables the development and training of large complex models, also known as foundation models, due to centralized datasets and the availability of computing power that is not easily available for other organizations (i01, i02). These models are hidden behind an API and controlled only by the provider (i01, i03). Thus, the provider is also responsible for emerging risks and challenges. Among other challenges concerning the trustworthiness of such models, one central challenge is that biases and weaknesses of centralized foundation models will be amplified as foundation models are used in a lot of downstream tasks across industries (Bommasani et al., 2021, p. 152).

TO3: Trustworthiness vs. Complexity Abstraction. The more complexity is abstracted away from the user and the more steps are automated, the lower the user's control over the service. This could even lead to the final AI service being perceived as even more of a black box compared to self-developed ML models resulting in decreased trust in the service.

One central attribute of complexity abstraction is the offering of ready-to-use, provider-managed services. Thus, AIaaS providers are also responsible for the trustworthiness of these services (i01). As the models are typically hidden behind an API (i01, i03), the service provider must ensure sufficient transparency to allow users to test and verify the functionality of the service (i01, i04). However, within fully managed services transparency is not always given: "With the current solution, you can't see what models are behind it, it's completely hidden, opaque" (i04, paragraph 34). On the contrary, MLaaS services allow more transparency if base models are open-source models and not proprietary (i02). But even when the service provides full transparency, ML models remain a black box, a well-known problem within ML (i04). Therefore, it is challenging for providers to explain the outputs of fully managed AI services (i04).

The offering of ready-to-use AI inference services inevitability transfers the responsibility of detecting bias or unfairness within ML models and training data to the AIaaS provider. Since the model prediction quality mainly depends on the training data (i03, i06), AIaaS providers must clearly state training details and the purpose of the model. This can for example be achieved by offering data cards or fact sheets. The former term refers to "structured summaries of essential facts about various aspects of ML datasets needed by stakeholders across a dataset's lifecycle for responsible AI development" (Pushkarna et al., 2022, p. 1778), whereas the latter term describes documents that contain important information about an AI service, for examples its intended use, performance, safety, and security to increase transparency and trust among customers (Arnold et al., 2019, p. 6:1). To foster transparency, AI services should reveal information about the entire lifecycle of an AI model (Li et al., 2022, p. 8), this includes "design purposes, data sources, hardware requirements, configurations, working conditions, expected usage, and system performance" (Li et al., 2022, p. 8).

Further, the complexity abstraction characteristic of AIaaS enables the democratization of AI. Thus, AIaaS can be used by non-expert users to leverage AI capabilities. The lack of understanding of underlying concepts of AI might interfere with the value contribution of AIaaS (i06) and might result in decreased trust in the service. Thus, interpretable, and explainable AI measures need to be adapted to the customer's expertise (i05). This includes for example the design decision and how explanations are provided to the end-user and customers (i05).

Another challenge, arising with the increased democratization of AI through AIaaS, is that AI services can easily be integrated and underpin a lot of applications (i01). These applications might misuse predictions of the respective AI service (i05). However, providers cannot monitor and detect all possible methods of misuse (i05). For example, providers cannot control the post-processing of inference requests (i05). This leaves the responsibility to deploy ethical AI services in the end to both, the customer and the service provider, for example, to prevent reputational damage (Cobbe & Singh, 2021, p. 58).

Automation, as another building block of complexity abstraction, is adding complexity to the provision of a trustworthy service. Especially for pre-developed pipelines and AutoML, explanations should not only be targeted to end-users at the inference step (i05), but rather providers should also offer explanations for data scientists and application developers during the development to increase the transparency of automated ML workflow steps. By integrating explanations and transparency into AutoML, providers can increase trust in the AutoML functionalities of the platform (i01). For example, if the feature engineering and feature selection are automated, the platform should display what features were selected to train the model and reasons for the selection (i01).

TO4: Complexity Abstraction vs. Customizability. Providers enter a trade-off between customizability and complexity abstraction in AIaaS design. The more complexity is abstracted away from the users' side, the easier the use of the service, but on the contrary, the customizability of the service in terms of user control is sacrificed. If the provider offers lots of customizability options within an MLaaS platform, the user interface (UI) inevitability is more complex and the risk of false configuration increases (i01).

One reason for this trade-off is the diversity of user groups. Depending on the knowledge and background of the user the service might be too complex or too trivial (i02), resulting in the providers having to find a balance between complexity abstraction and customizability. Business users want easy-to-use services, developers might benefit from AutoML and pre-configured AI components, and data science users need more sophisticated platforms to develop custom models (i02). Consequently, providers are diversifying their services and offering them on different abstraction levels (i03).

Some decisions are highly customer-specific and therefore cannot be abstracted away from the user without making implicit assumptions regarding adjustable parameters: "But we never really want to make decisions for the user, where it really depends on the user's decision. For example, what metric to use when optimizing the models" (i02, paragraph 44). As these decisions are critical and highly influence the final model performance in a specific use case, the user itself must make this decision, consequently the complexity increases (i02). Other examples include decisions based on user-specific requirements regarding accuracy, latency, or costs (i01, i02, i03) that cannot be abstracted away by the provider without sacrificing customizability by making generic assumptions. However, operational, and infrastructure-related tasks can be abstracted without compromising customizability: "With all the infrastructure and operational topics, the more you don't have to do yourself, the better. That's my general opinion" (i06, paragraph 28). During the interviews experts mentioned, for example, that they provide default settings for AutoML to tackle the problem concerning diverse user groups (i01, i03). Thus, customers with less expertise can use default settings but more experienced users are still able to adapt, for instance, used algorithms or hyperparameters (i03).

T05: Automation vs. Performance. AutoML as one aspect of complexity abstraction might come at a price of performance losses of the ML model (i01). The more automated an AI service is, the higher the potential for false implicit assumptions made by the service which ultimately lowers the performance of the model. This includes, for example, how missing values are handled or how the scale of measure is determined (i01). Additionally, if there are no functions offered to do manual tuning, refinements, and adjustments during the AutoML process, one cannot adapt the model to reach the best performance (i04). As such, the development of ML models remains a task that requires a lot of trial-and-error and experimentation to achieve optimal results (i02, i06).

This is consistent with the findings of Yao et al. (2017), who concluded that MLaaS platforms that offer more manual tuning options and therefore are less automated deliver models with better performance if tuned correctly. But at the same time customizability increases the risk of poorly configurated models, hence the performance can be worse than in the automated case (Yao et al., 2017, p. 386). An important factor affecting performance is data quality: "The models will, of course, still only be as good as the data with which the algorithms are fed. That is also important" (i03, Paragraph 47).

Quality can, for example, be increased by pre-processing data and performing manual feature engineering, thereby integrating more domain knowledge, but these steps are difficult to automate (i06). Domain knowledge is central for high performance:

> "And then the data points have a completely different meaning in terms of the domain. And this meaning has to be included in the modeling. And the model is only as good as the meaning that is given to the data" (i06, paragraph 32).

In this context, it already becomes apparent that automation can only support the customer in the ML workflow. If the goal is a powerful model in a specific use case, manual work cannot yet be completely replaced (i02, i03, i06). For optimal results and a high-performant model, customers are required to manually perform data-preprocessing, feature-engineering, and modeling steps (i03, i06).

On the other hand, highly automated ML services enable domain experts to set up their own models through an easy-to-use interface (i03). Typically, these business users have a better understanding of the business domain than pure data scientists and hence can determine relevant data better or interpret the service's output (i03). For example, by using domain knowledge they can evaluate the importance of specific features used for training during the feature engineering step, potentially resulting in higher-quality models (i03). In that case, AutoML helps to create performant models and can automate domain-independent aspects of the ML workflow, such as hyperparameter selection, tuning, and selection of the optimal architecture (i02, i06). The experts mentioned during the interviews that the baseline performance of today's AutoML models is acceptable (i01, i03): "Often the AutoML models are not so far off from what you then achieve […] by fine tuning" (i01, paragraph 38). In addition, an automated AI service can support even experienced users by providing a baseline performance and possibly uncovering errors in the configuration (i01).

TO6: Model Performance vs. Latency and Scalability. The provider must make certain design choices regarding his server infrastructure and model size (i01). One of these choices concerns the server inference architecture (i01, i04). For instance, stream processing allows for the on-demand processing of inference requests. On the contrary stream processing poses certain challenges for AIaaS providers in terms of scalability, latency or resource consumption and allocation (i01, i04).

In particular, complex ML models need a lot of computing resources for training and inference. These large models may offer higher performance (i01), however, the computational costs to run these models are high. Accordingly, scalability and latency are cloud characteristics that interfere with the provision of complex and large ML models, especially in the cases of vision or sound-related services (i01). For example, in the case of a fully managed translation service which is based on a complex ML model, the AIaaS provider might face challenges regarding latency:

> "If it takes 15 seconds to translate a text, nobody has the patience. So, it's a bit like browser latencies […]. And according to that you always have the question how big, how complex is your model, how long does the inference take" (i01, paragraph 42).

If the service is fully managed by AIaaS providers, the challenge of finding the right balance between model performance and computational costs is transferred to them (i04). In other cases, this can also be seen as a user-sided trade-off, as accuracy and latency requirements are use-case dependent and are subjective to the customers. The AIaaS provider can account for flexibility in deployment and model choices to meet customers' differing requirements (i01).

TO7: Security and Privacy vs. Performance. Providers enter a service design trade-off between security and performance. One effective measure to ensure data security for sensitive training data is to delete respective data after the model has been trained (i05). However, this leads to a drop in performance in several aspects. First, the training performance is negatively influenced. For model training, the whole data must be uploaded again and therefore it takes longer to process (i05). Second, even the model performance itself can be compromised. If the training of the model requires manual work as a cause of security measures (i05) and is done in a batch process, there is no continuous improvement of the model, even if new data is already available. Thus, the service cannot automatically improve over time, and the performance of the model might be lower compared to continuously improved models with a continuous learning approach (i04, i05).

Training data in the cloud is normally encrypted to ensure privacy (i06). However, there are additional challenges for privacy within the AIaaS paradigm because data is not only stored but the ML models are trained on that data and integrate the information from the training dataset. Thus, data owners are worried about the safety and privacy of their data and providers are concerned that adversaries could compromise data or models (Tanuwidjaja et al., 2020, p. 167425). Nevertheless, existing privacy-preserving machine learning techniques are problematic for AIaaS because they add computational overhead resulting in performance and scalability issues (Tanuwidjaja et al., 2020, pp. 167441–167442).

TO8: Deployment models vs. Centralization. Typically, AI services that are deployed in the cloud are leading to desirable characteristics from a provider's viewpoint, such as control over the service (i02), better protection of IP (i01), and the possibility to track user behavior (i02). Yet sometimes users have special requirements that do not allow AI services to be deployed in the cloud, including highly regulated industries (i02, i05) and low latency use cases (i01). In these cases, AIaaS providers can offer the possibility to deploy services and models in private clouds on-premises or on edge, possibly giving up provider-sided advantages of centralization. Pre-trained ML models represent a competitive advantage as they integrate training data or investments in computational costs during the model training that customers are not able to allocate (i01). However, if the model or the service is made available locally and not via a web-service, the intellectual property of the provider is at a higher risk to be compromised (i05), thus it is challenging to provide AIaaS on-premises. Nevertheless, researchers already proposed potential solutions. For example, Mo et al. (2021) proposed the usage of trusted execution environments and Zhang et al. (2018) suggest watermarking for deep neural networks.

Further, on-premises deployment might interfere with the typical pay-per-use business model that is facilitated by centralized services and platforms:

> "[…] I give a little bit of control out of my hand and then I also have a completely different business model. Then, I sell my model. And who prevents me now that you take the model and also pass it on again to someone else?" (i01, paragraph 24).

In addition, providers are making trade-offs between the observability that is possible in the case of a centrally provisioned AI service and alternative deployment options (i02). In the case of a centralized online platform, the provider can track user behavior and collect valuable data (i02). This data can be used to optimize and improve the service provider's product offering, and pre-trained models as well.

Lastly, lock-in effects might be reduced if the AIaaS is offered in combination with a private cloud deployment model, because the service can be deployed on different cloud infrastructures, making the service cloud-agnostic (i02). This might interfere with the AIaaS provider's revenue as customers are typically billed for computing hours incurred in the cloud of the respective provider (i01, i03).

4.4. Additional Challenges in AIaaS Service Design

During the conduction of the interviews providers often mentioned challenges in addition to explicit trade-offs. Since these challenges might influence trade-offs and impact service design decisions simultaneously a short overview of the most discussed challenges is provided alongside trade-offs. As mentioned in chapter 3.4, the coding process yielded 9 challenges. In the following, however, only four challenges are mentioned. Challenges were excluded because of one of the following reasons. Either the remaining challenges referred to internal challenges that were not directly connected to service design decisions, challenges were already mentioned within Section 4.3 or challenges only concerned very specific aspects that were not generalizable to several AIaaS services.

CH1: Developing a suitable inference interface. One important design decision is concerning the inference interface of a respective AIaaS model:

> “Ultimately, the question is always: How do you provision your model? […] it's just a question of do you provide an interface or maybe even a translation mask or something where you just drag and drop your Word document or your text in and get the result" (i01, paragraph 62).

The reason it is challenging also lies in the diversity of users, for example in the trade-off customizability vs. complexity abstraction. However, it does not pose a specific trade-off since there is no interfering characteristic. When a user interface is deployed on top of the API endpoint the AIaaS provider does not have to sacrifice any of the desired characteristics. It does, however, require more software development resources (i01).

Another reason for the challenge to develop a suitable interface is that AIaaS is used more as a foundation for other applications than immediately as a standalone product (i05). Thus, it is different to, for example, SaaS that is used directly by end-users via a web interface (i05). AIaaS must be integrated into existing processes and applications to actually provide value which is challenging for both, customers and providers (i01, i05). Providers mostly can only deliver a generic API Endpoint and the integration must be done by either a third party or by the customer itself (i01, i05).

CH2: Overcoming Adoption Barriers. Even though AI and AIaaS offerings are on the rise for several years (Lins et al., 2021, pp. 441–442) there are still a multitude of adoption barriers regarding AIaaS:

> "AI is already established enough that every company is applying a little bit of AI. But in my experience, I wouldn't say that AI is already being used on a large scale. And that is actually also a bit of a transformation process for a company, that they are actually ready for AI" (i02, paragraph 62).

Adoption barriers include the lack of the right corporate culture (i02, i05), customers’ difficulty to perceive the value of AI or defining promising use cases (i03, i05), and wrong expectations of AI results (i05). Thus, even if the aim of AIaaS is to enable everyone to leverage AI capabilities, for most use cases customers need at least a basic understanding of AI. This is because providers cannot abstract away the fundamental concepts of AI (i02, i05). Consideration of customer adoption barriers and issues should therefore be an important part of AIaaS design decisions.

CH3: Data-related Challenges. As one of the biggest challenges, several experts repeatedly mentioned the lack of high-quality training data (i01, i04, i05). For example, experts mentioned that "Data quality is always a problem" (i01, paragraph 20) and that companies "have masses of data, but some of it is so disorganized and of such poor quality" (i05, paragraph 10). Collection and storing of high-quality training data pose one of the greatest barriers to effectively use AIaaS (i03). Yet, depending on the specific domain and task, AIaaS providers cannot provide training data or pre-trained models (i05, i06). This might severely impede the value contributions of AIaaS and could be addressed by providing additional functionalities together with AIaaS to enable customers to have the right amount of high-quality training data in the first place. As for now, implementation partners and manual work are still required to combine and pre-process data from multiple sources (i01, i05).

CH4: ML as an empirical field. Lastly, experts mentioned that ML "is really an empirical science in the end" (i04, paragraph 28) and “failure is actually the normality in the development of machine learning models" (i02, paragraph 60). The successful integration of AI is generally based on trial-and-error (i04), lots of experimenting (i02, i04) and several additional factors, such as training data (i05). As a result, AIaaS providers face the problem that neither they nor customers can determine the real value of a final business solution that integrates AIaaS. This problem could be addressed by, for example, providing the possibility of developing a proof-of-concept (POC) (i05). Consequently, this can impact pricing decisions, for example, the provision of a trial period. Hence, this challenge becomes likewise highly relevant for the design of AIaaS services.

5. Conclusion

5.1. Principal Findings

In this thesis characteristics of AIaaS and perceived provider-sided trade-offs within AIaaS service design were derived based on 6 conducted expert interviews. Further, it provided valuable insight into the up-to-date status of AIaaS and presented challenges from AIaaS providers’ viewpoints.

The identified characteristics were grouped into 10 core categories, namely centralization, cloud characteristics, complexity abstraction, customizability, deployment and integration, functionality, genericness, performance, security and privacy and trustworthiness. These core categories were further delineated in various aspects that describe respective core categories in more detail (see Appendix C). The derived characteristics align with previous existent literature (Geske et al., 2021; Lins et al., 2021; Pandl et al., 2021). Additionally, new characteristics and aspects of AIaaS were identified such as centralization, genericness and functionalities, especially including functionalities for MLOps as a newly emerging approach in productizing ML models. Some of the derived characteristics are inherited from the cloud computing paradigm, others from the field of AI. However, the combination of these two technologies leads to the emergence of new characteristics, adding further complexity to service design decisions from a provider’s viewpoint. In particular, there are specific characteristics of AI that challenge the easy-to-use, managed self-service character of cloud services. AIaaS is different from other cloud software services (i05, i06), as its performance is determined by the quality of training data, integration of domain knowledge and manual adjustments, among other factors.

The complexities in service design were approached by identifying trade-offs between AIaaS characteristics, resulting in a total of 8 trade-offs identified (see Figure 2**)**. These trade-offs are ranging from more technical trade-offs (model performance vs. latency and scalability) to socio-technical trade-offs (centralization vs. trustworthiness), highlighting that there is no one-size-fits-all design and the variety of different design dimensions of AIaaS in general. At the center of trade-offs are the characteristics complexity abstraction and performance since they interfere with other characteristics the most. Complexity abstraction is based on generic assumptions, hiding details, and taking on challenges of the customers, which inevitably affects the genericness, trustworthiness, and performance of an AIaaS service. Services trained on generic data can only deliver generic results, consequently one identified service design trade-off is the trade-off between genericness, performance and complexity abstraction. Integrating domain knowledge into the service is not always feasible for the provider, thus, customers must have to ability to customize the service and use custom training data. However, customizability increases service complexity. The automation of domain-dependent steps of the ML workflow remains challenging, leading to a trade-off between automation and performance. Moreover, performance interferes with the provision of resources, as complex, high-performant models are resource-intensive. Ensuring a performant service compromises security and privacy in certain aspects.

Further, providers benefit from the centralization of AI services in terms of IP rights protection, control, and billing, yet centralization aspects interfere with required on-premises deployment options in a few special cases and aspects of trustworthiness.

In addition, challenges could be identified that may also impact the design decisions. These challenges incorporate the development of a suitable inference interface, the overcoming of adoption barriers, data-related challenges, and lastly, the empirical nature of ML.

Ultimately, it is important to note that central trade-offs and user-sided challenges have been recognized by AIaaS providers. To address this issue, AIaaS providers have started to broaden and extend their platforms and offer different services for diverse users at multiple levels of abstraction. Furthermore, they integrate new approaches such as foundation models and MLOPs capabilities, to extend the value contribution of their service offerings and to address trade-offs and challenges.

5.2. Implications for Research and Practice

By identifying prevalent trade-offs and challenges specific to AIaaS from a socio-technological, provider-sided viewpoint, a novel field of research was studied. Thus, findings can be valuable for both, researchers as well as practitioners.

For researchers, this thesis serves as a conceptual overview of trade-offs within AIaaS. The conducted research yields a new perspective on AIaaS characteristics and their dependencies. It extends academic research by providing practical insights in AIaaS providers’ service design choices regarding the combination of AI and the cloud computing paradigm. Consequently, the thesis fosters an understanding of different design choices and their respective limitations and challenges.

For practitioners, the results are beneficial to differentiate service offerings based on characteristics. Additionally, the thesis yields the possibility to assess service design decisions as well as associated risks. The overview of trade-offs can help to identify parts with potential for improvement and guidance in developing more balanced solutions. Moreover, practitioners on the customer side might benefit from the research as well. The deeper understanding of AIaaS and trade-offs is valuable in AIaaS adoption challenges and decisions.

5.3. Limitations

However, this thesis comes not without limitations. First, only German-speaking experts were recruited for the interviews and in total only six experts were interviewed. That might restrict the representativity of the expert interviews as statements could be based on customers' expectations and regulations in Germany, although several experts were employed by international large CSPs.

During the interviews, it became apparent that even for experts it was not trivial to identify concrete trade-offs in AIaaS service design. There is a difference in prevalent characteristics and trade-offs between AI services exclusively for inference and MLaaS services for developing custom ML models. This circumstance was problematic as experts might sometimes considered trade-offs only for one type of AI services. For example, the protection of IP rights of ML models is only a matter in pre-trained services whereas providers don't face this challenge within MLaaS for developing custom models.

Additionally, AIaaS is a very broad field, incorporating diverse technologies, for example AI and cloud computing, as well as different approaches, such as MLOps and AutoML. Consequently, the thesis has covered a broad range of topics. The list of identified characteristics and trade-offs is certainly not complete. Particularly, it was challenging to focus on trade-offs specific to AIaaS, because there are lots of related underlying trade-offs in cloud computing and AI (Baeza-Yates & Liaghat, 2017; Brownlee et al., 2021) that inevitability also affects AIaaS design. However, many of these trade-offs have already been researched. To consider them all and evaluate the consequences of these trade-offs within AIaaS design would have been beyond the scope of this thesis and would have broadened the focus of the work too much.

In addition, the breadth of the topic caused trade-offs to be explored in a somewhat superficial and conceptual way, rather than in depth. Lastly, it was difficult to clearly separate trade-offs from each other because they had multidimensional interdependencies. For example, complexity abstraction was related to customizability and genericness and these characteristics can all have an impact on performance for different reasons. Changing one characteristic is likely to affect several other characteristics and not only the particular characteristic identified in the trade-off. Therefore, the overview derived in this thesis represents only one possible overview of many. Additionally, there are characteristics that are mutually supportive rather than contradictory, making the differentiation of the distinct influences on each characteristic even more complex.

5.4. Further Research

There are several ways to perform further research and address the limitations mentioned above.

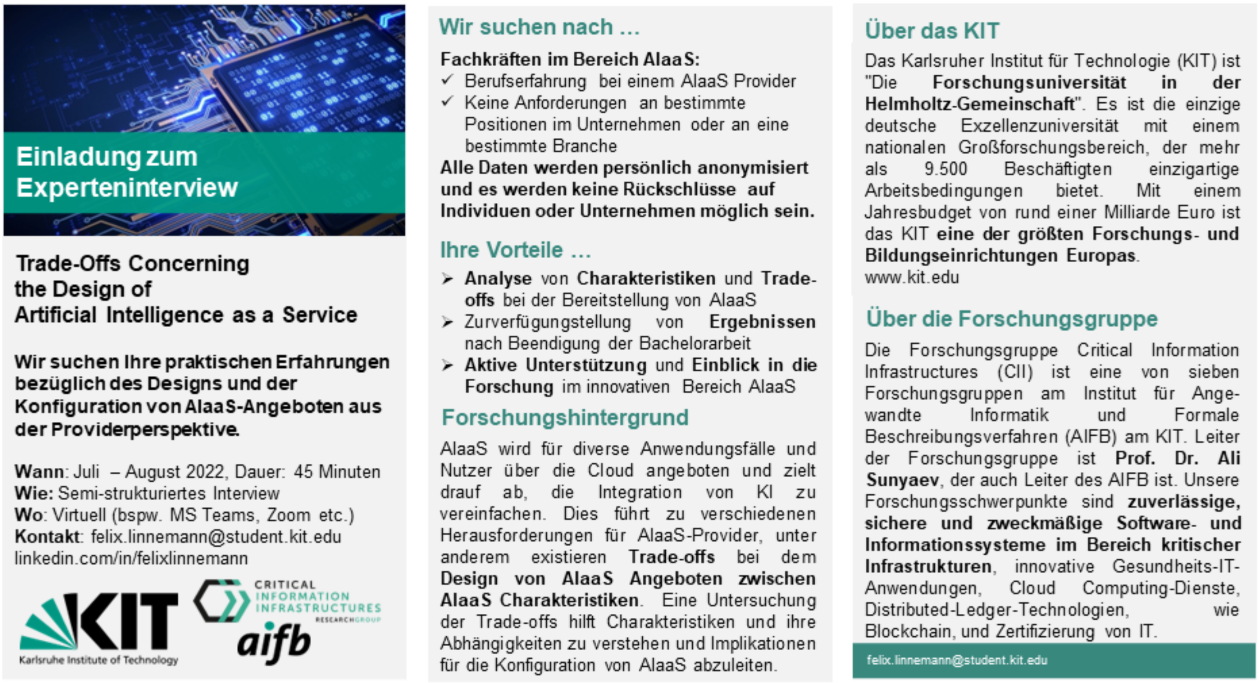
First, researchers could quantify some of the identified trade-offs to understand their real-world importance. Quantifying and focusing on one specific trade-off would foster an in-depth understanding of the trade-off and overcome the challenges mentioned in Section 5.3. Additionally, it might help to identify the most promising design options for further improvement.

While this thesis sets a focus on the provider-sided trade-offs, future research could also address the users’ perspective as most design decisions are not solely made by the provider, rather AIaaS providers offer solutions based on the needs of the users. It could be of great interest to study user needs in-depth, e.g., in terms of the level of automation desired by users or which AI services are used for which tasks. Studying the needs of different customer groups and their perceived user-sided trade-offs could thus lead to promising, balanced and new AIaaS designs.

This can also be achieved by conducting future research to explore the integration of new approaches, drawing from research in the fields of AI and cloud computing, in AIaaS. This potentially resolves existing challenges and trade-offs. For example, research can regard questions on how to preserve customizability in an easy-to-use AIaaS setting or how to make AIaaS services more generic while preserving performance at the same time.

Appendix

A. Appendix A



Einladung zum Experteninterview

Trade-Offs Concerning the Design of Artificial Intelligence as a Service

Wir suchen Ihre praktischen Erfahrungen bezüglich des Designs und der Konfiguration von AIaaS-Angeboten aus der Providerperspektive.

Wann: Juli – August 2022, Dauer: 45 Minuten
Wie: Semi-strukturiertes Interview
Wo: Virtuell (bspw. MS Teams, Zoom etc.)
Kontakt: felix.linnemann@student.kit.edu
linkedin.com/in/felixlinnemann

Karlsruhe Institute of Technology
CRITICAL INFORMATION INFRASTRUCTURES RESEARCHGROUP
aifb

Wir suchen nach …

Fachkräften im Bereich AIaaS:

✓ Berufserfahrung bei einem AIaaS Provider
✓ Keine Anforderungen an bestimmte Positionen im Unternehmen oder an eine bestimmte Branche

Alle Daten werden persönlich anonymisiert und es werden keine Rückschlüsse auf Individuen oder Unternehmen möglich sein.

Ihre Vorteile …

- **Analyse** von **Charakteristiken** und **Trade-offs** bei der Bereitstellung von AIaaS
- Zurverfügungstellung von **Ergebnissen** nach Beendigung der Bachelorarbeit
- **Aktive Unterstützung** und **Einblick in die Forschung** im innovativen Bereich AIaaS

Forschungshintergrund

AIaaS wird für diverse Anwendungsfälle und Nutzer über die Cloud angeboten und zielt drauf ab, die Integration von KI zu vereinfachen. Dies führt zu verschiedenen Herausforderungen für AIaaS-Provider, unter anderem existieren **Trade-offs** bei dem **Design von AIaaS Angeboten zwischen AIaaS Charakteristiken**. Eine Untersuchung der Trade-offs hilft Charakteristiken und ihre Abhängigkeiten zu verstehen und Implikationen für die Konfiguration von AIaaS abzuleiten.

Über das KIT

Das Karlsruher Institut für Technologie (KIT) ist "Die **Forschungsuniversität in der Helmholtz-Gemeinschaft**". Es ist die einzige deutsche Exzellenzuniversität mit einem nationalen Großforschungsbereich, der mehr als 9.500 Beschäftigten einzigartige Arbeitsbedingungen bietet. Mit einem Jahresbudget von rund einer Milliarde Euro ist das KIT **eine der größten Forschungs- und Bildungseinrichtungen Europas**.
www.kit.edu

Über die Forschungsgruppe

Die Forschungsgruppe Critical Information Infrastructures (CII) ist eine von sieben Forschungsgruppen am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) am KIT. Leiter der Forschungsgruppe ist **Prof. Dr. Ali Sunyaev**, der auch Leiter des AIFB ist. Unsere Forschungsschwerpunkte sind **zuverlässige, sichere und zweckmäßige Software- und Informationssysteme im Bereich kritischer Infrastrukturen**, innovative Gesundheits-IT-Anwendungen, Cloud Computing-Dienste, Distributed-Ledger-Technologien, wie Blockchain, und Zertifizierung von IT.

felix.linnemann@student.kit.edu

B. Appendix B

Karlsruher Institut für Technologie

Interviewleitfaden zur Erhebung
qualitativer Forschungsdaten

Trade-offs Concerning the Design of Artificial-Intelligence-as-a-service

Datum: ___________________________________

Interviewpartner: ___________________________________

Dauer des Interviews: ___________________________________

Inhaltsverzeichnis

1. Einführung in den Interviewprozess ... 41

1.1. Begrüßung der Interviewpartner ... 41

1.2. Geplante Dauer des Interviews ... 41

1.3. Geplante Agenda des Interviews ... 41

1.4. Aufnahme des Interviews ... 42

2. Das Interview ... 43

2.1. Background des Interviewpartners ... 43

2.2. Gesprächseinstieg: Konzeptionelle Grundlagen ... 44

2.2.1. AIaaS – Schaffung einer gemeinsamen Gesprächsgrundlage ... 44

2.2.2. Einleitung in das Forschungsvorhaben ... 44

2.3. Charakteristiken und Trade-offs bei der Konfiguration von AIaaS ... 45

2.3.1. Charakteristiken ... 45

2.3.2. Trade-offs und Implikationen ... 47

2.3.3. Weitere Persönliche Erfahrungen ... 50

3. Abschluss des Interviews und nächste Schritte ... 50

1. Einführung in den Interviewprozess

5 Minuten / 5 Minuten (Totalzeit)

Dieser erste Abschnitt dient der Einführung des Interviewpartners in den Interviewprozess. Die ganzheitliche Vorstellung des Vorhabens hat zum Ziel sowohl eine thematische Einführung zu bieten als auch organisatorische Fragen zu klären und auf Bedenken einzugehen.

1.1. Begrüßung der Interviewpartner

- Vorstellung eigene Person:
    - Name, Bachelorstudent Wirtschaftsingenieurwesen KIT, Erhebung qualitativer Interviewdaten im Rahmen der Bachelorarbeit am AIFB in der Forschungsgruppe Critical Information Infrastructures
- Ausgiebige Vorstellung des Interviewpartners im Rahmen des weiteren Interviews (siehe: Erhebung von Background Informationen)

1.2. Geplante Dauer des Interviews

- Geplante Länge: 45 Minuten
- Gibt es von Ihrer Seite zeitliche Begrenzungen oder Präferenzen?
- Ist eine Überschreitung dieses zeitlichen Rahmens ihrerseits möglich (z.B. für den Fall offener Fragen, oder aufgrund von aufkommendem Diskussionsbedarf)?
- Offene Kultur - Sie können jederzeit im Interview Fragen stellen oder Kommentare einbringen, die wir direkt diskutieren.

1.3. Geplante Agenda des Interviews

a. Persönliche Erfahrungen und Background des Interviewpartners
    - Erster lockerer Gesprächseinstieg, um den Interviewpartner kennenzulernen
    - Informationen zu Unternehmen und Position

b. Artificial Intelligence as a Service – Schaffung einer gemeinsamen Gesprächsgrundlage. Verständnis für den Forschungsgegenstand. An den Flyer anknüpfen.
    - Der Begriff Artificial Intelligence as a Service wird in der Praxis uneinheitlich verwendet, daher soll zunächst eine gemeinsame Gesprächsgrundlage geschaffen werden.
    - Klarmachen, was das Ziel des Interviews und das Forschungsvorhaben ist

c. Charakteristiken und Trade-offs von AIaaS

    - Bildet das Kernstück des Interviews. Hierbei wird der Interviewpartner zunächst allgemein befragt welche typischen Charakteristiken von AIaaS existieren.
    - Dies erfolgt zunächst im Rahmen einer Brainstorming-Phase. Der Interviewpartner legt in dieser Phase seine grundlegenden Gedanken und Argumentationsketten im Zusammenhang mit der übergeordneten Fragestellung dar. Diese Phase schafft die Möglichkeit Gedanken ungefiltert darzulegen, ohne diese durch eventuell nicht passende Fragestellungen zu beeinflussen.
    - Anschließend erfolgt die Bearbeitung explizit formulierter Fragestellungen nach verschiedenen Charakteristiken angepasst auf den bisherigen Gesprächsablauf
    - Zuletzt werden Trade-offs zwischen den genannten Charakteristiken diskutiert. Dies erfolgt zunächst auch in einer Brainstorming-Phase. Es besteht die Möglichkeit ein beispielhaften Trade-off für das Verständnis des Experten aufzuzeigen. Anschließend sollte der Interviewpartner jedoch auch eigene Trade-offs ergänzen falls seiner Meinung nach noch weitere Trade-offs in der Praxis existieren.
    - Für erwähnte Trade-offs werden die Implikationen für die Praxis und eventuelle Lösungsoptionen diskutiert.

d. Abschluss des Interviews und nächste Schritte

    - Abschließend werden wir Ihnen die nächsten Schritte dieser Arbeit erläutern.

1.4. Aufnahme des Interviews

- Zur erleichterten Auswertung der Interviewergebnisse wird der Ton des Interviews aufgezeichnet.
- Alle Daten werden anonymisiert und es werden keinerlei Rückschlüsse auf Personen oder Unternehmen möglich sein
    - Bei Bedenken bezügliche der Aufnahme:
        - Gründe erfragen warum Bedenken bestehen
        - Möglichkeit Aufnahme anzuhalten oder im Nachhinein das Transkript zur Verfügung zu stellen
        - Falls Aufnahme nicht erlaubt: Notizen anfertigen
- Die Ergebnisse des Forschungsvorhabens werden Ihnen selbstverständlich gerne zur Verfügung gestellt.

Haben Sie weitere Fragen oder Wünsche für das Interview?

2. Das Interview

2.1. Background des Interviewpartners

5 Minuten / 10 Minuten

1) In welcher Branche sind Sie tätig?		
□ Banken/Versicherungen □ Automobilindustrie □ Chemie/Pharmaindustrie	□ Services □ Handel □ Logistik	□ Telekommunikation/IT □ Andere Industrie
2) Wie groß ist ihr Unternehmen in Bezug auf (a) Mitarbeiterzahl und (b) Umsatz?		
□ 1-10 Mitarbeiter (a) □ 11-50 Mitarbeiter □ 51-250 Mitarbeiter □ > 250 Mitarbeiter	□ < 2 Mio. € Umsatz (b) □ ≤ 10 Mio. € Umsatz □ ≤ 50 Mio. € Umsatz □ > 50 Mio. € Umsatz	
3) In welcher Industrie agieren Ihre Kunden?		
□ Banken/Versicherungen □ Automobilindustrie □ Chemie/Pharmaindustrie □ Gesundheitswesen	□ Services □ Handel □ Logistik	□ Telekommunikation/IT □ Andere Industrie
4) Welche Position bekleiden Sie im Unternehmen und was sind ihre Aufgaben?		
□ Head of IT □ Consultant □ Product Manager □ IT-Service Manager	□ CTO □ CFO □ CEO □ Andere	
5) (a) Wie lange sind Sie bereits in Unternehmen tätig und (b) wie lange arbeiten Sie bereits im IT-Umfeld?		
□ weniger als 1 Jahr (a) □ 1 - 3 Jahre □ 3 – 5 Jahre □ 5 Jahre und mehr	□ weniger als 1 Jahr (b) □ 1 – 3 Jahre □ 3 – 5 Jahre □ 5 Jahre und mehr	

2.2. Gesprächseinstieg: Konzeptionelle Grundlagen

5 Minuten / 15 Minuten

2.2.1. AIaaS – Schaffung einer gemeinsamen Gesprächsgrundlage

In der Praxis gibt es viele verschiedene Begriffe (bspw. MLaaS, Training-aaS, ML services, AI services, Inference services) auf unterschiedlichen Abstraktions-Ebenen (Infrastruktur, Plattform, Software, API). Daher soll zunächst ein gemeinsames Verständnis geschaffen werden.

- Was **verstehen** Sie unter **AIaaS**?
- Wie lange beschäftigen Sie sich schon mit AIaaS?
- Welche **Art von AI-Services** bietet Ihr Unternehmen an?

2.2.2. Einleitung in das Forschungsvorhaben

- AIaaS soll AI einfacher verfügbar machen, indem es die Komplexität und den Aufwand von der Entwicklung bis zur Bereitstellung von AI-Anwendungen reduziert.
- AIaaS wird für diverse Nutzergruppen und Unternehmen angeboten, teils branchenübergreifend und für unspezifische Aufgaben. Zusätzlich wird versucht, den normalerweise sehr manuellen und spezifischen Prozess ein ML-Modell zu erstellen, zu automatisieren bzw. weniger aufwendig zu machen. Dabei entstehen Herausforderungen für die Provider.
- **Eine spezielle Herausforderung**: Trade-offs bei dem Design von AIaaS zwischen unterschiedlichen Charakteristiken (Bsp.: Ein benutzerfreundliches und einfaches System kann nur bedingt gleichzeitig eine hohe Anpassbarkeit besitzen). Ein AIaaS Service lässt sich nicht durch den Provider auf alle Nutzer und diversen Anwendungsfälle gleichzeitig optimal anpassen. Daher müssen Provider während des Designs/der Architektur eines AI-Services statische Entscheidungen über zentrale Charakteristiken ihres Services treffen, welche im Nachhinein nicht oder nur schwer geändert werden können.
- **Ziel:** Herauszufinden, welche Trade-offs zwischen welchen Charakteristiken in der Praxis existieren und Implikationen für das Design von AIaaS abzuleiten.
- **Relevanz:** Abhängigkeiten zwischen Charakteristiken verstehen, Chancen und Risiken beurteilen, Konfigurationstypen ableiten

Ihre Meinung ist uns an dieser Stelle als Forschungsinput sehr wichtig, um vor allem einen praxisnahen Einblick erhalten zu können.

2.3. Charakteristiken und Trade-offs bei der Konfiguration von AIaaS

25 Minuten/ 40 Minuten

2.3.1. Charakteristiken

Zunächst sollen zentrale Charakteristiken von AIaaS und deren Beschreibung identifiziert werden, anhand welcher sich unterschiedliche AIaaS Services beschreiben lassen. Diese Charakteristiken dienen als Basis für vorhandene Trade-offs und die nachfolgenden Fragen dienen dazu ein Verständnis für die Charakteristiken zu entwickeln.

Brainstorming-Phase:
Unverzerrte Darstellung der zentralen Eigenschaften des Service, ohne vorher Einfluss zu nehmen. Diese Fragen können im Rahmen der Brainstorming Phase simultan beantwortet werden.

- Wie lässt sich ihr AIaaS Angebot beschreiben?
- Was sind zentrale Charakteristiken oder Eigenschaften ihres AIaaS Angebots?
- Was hebt Sie von anderen AIaaS Providern ab?
- Was ist das zentrales Wertversprechen Ihres AIaaS Angebots?

Offene Fragen:
Beschreibung und Verständnis von Charakteristiken von AIaaS. Eventuelle weitere Charakteristiken besprechen, falls diese im ersten Teil nicht genannt worden sind.

Komplexitätsreduzierung und Benutzerfreundlichkeit

- Inwiefern ist das System benutzerfreundlich?
- Gibt es Funktionen, die die Komplexität der Entwicklung von KI-Anwendungen reduzieren? Wie reduzieren diese die Komplexität?
- Inwieweit können User ohne KI-Expertise das System gewinnbringend nutzen?
- Inwieweit hilft es Usern, wenn diese KI-Expertise mitbringen?
- Inwiefern müssen sich User um die Systemumgebung/Infrastruktur kümmern?
- Was ist ihre Meinung zu no-code oder low-code Systemen?

Anpassbarkeit

- Können Sie etwas über die Anpassbarkeit von Ihren AI-Services erzählen?
- Inwiefern können die User ihr System individuell anpassen? Sind die User damit zufrieden?
- Was sind Vor- und Nachteile, wenn User ihr KI-Algorithmus selbst wählen können?
- Können User Hyperparameter selbst einstellen? Inwiefern ist dies sinnvoll und warum wird diese Funktion von den Usern genutzt?
- Inwieweit können User bestimmte Restriktionen an bestimme Eigenschaften festlegen, beispielsweise hinsichtlich der Fehlertoleranz, Latenz, Dauer des Trainings usw.?

Automatisierung

- Welche Schritte eines klassischen ML-Prozesses (auch „ML-Pipeline“) sind teil- oder vollautomatisiert? (Training Data, Pre-processing, Feature selection, Model choice and Parameter tuning, Model training, Model validation, Query, Prediction results (Yao et al., 2017))
- Was sind Ihrer Meinung nach Nachteile und Vorteile der Automatisierung?
- Inwiefern wird die Hardware bezüglich der Nutzung automatisch angepasst und optimiert?

KI-Modelle und Daten

- Bieten Sie den Usern vortrainierte KI-Modelle an? Falls ja, welches Ziel verfolgen Sie mit dem Angebot von vortrainierten Modellen (Aufwandsreduzierung, Vermeidung von verzerrten oder schlechten Modellen durch schlechte Qualität der Trainingsdaten, kein Training mehr notwendig)?
- Inwiefern stellen Sie sicher, dass die Trainingsdaten (von vortrainierten Modellen) repräsentativ sind für die Anwendungen Ihrer Kunden?
- Inwiefern können vortrainierte Modelle noch angepasst werden?
- Werden die Modelle mit den Daten von Kunden weiter verbessert oder trainiert?

Erklärbarkeit/Transparenz/Vertrauenswürdigkeit

- Inwieweit haben Sie Funktionen (ModelOps, Übersicht über Trainingsdaten, Erklärungen des Outputs) integriert, welche die Transparenz des Systems erhöhen sollen? Falls ja, welchen Funktionen sind dies und inwiefern verbessern sie die Transparenz?
- In welchem Umfang lässt sich nachvollziehen wie Ihr System Modelle und Anfragen automatisch optimiert? Lässt sich nachvollziehen welcher Algorithmus und welche Parameter verwendet wurden?
- Viele KI-Modellen werden als Blackbox wahrgenommen. Es gibt jedoch verschiedene Ansätze den Output eines Modells zu erklären. Inwiefern sind solche Ansätze in AIaaS integrierbar? Inwiefern ist es in Ihren Service integriert?

Performance und Erreichbarkeit

- Welche Faktoren beeinflussen die Genauigkeit/Richtigkeit des Modells? Welche Funktionen gibt es, um die Genauigkeit zu verbessern und zu evaluieren?
- Welche Faktoren beeinflussen die Latenz für eine Abfrage?
- Inwiefern stellen Sie sicher, dass die Modelle jederzeit zur Abfrage erreichbar sind?

Privacy und Security

- Für die Vorhersage müssen Inputdaten an eine Schnittstelle des Modells übertagen werden. Wie stellen Sie die Privatsphäre sicher? (Insbesondere bei kritischen Daten im Gesundheitswesen, Banken, usw.)
- Inwiefern stellen KI-Modelle in der Cloud eine Gefahr für den Datenschutz dar? Was unternehmen Sie dahingehend, dass keine Inferenz auf Trainingsdaten durch gezieltes Abfragen des Modells möglich ist?

Weitere mögliche Charakteristiken

- **komplementäre Funktionen** (Data Factory, Model Sharing, human in the Loop, Data Labeling bei der Datenaufbereitung, MLOps)
- **Cloud Charakteristiken**: pay-as-you-go, Skalierbarkeit, On-demand, Resource pooling
- **Kosten**: Wie ist das Pricing bei Ihnen aufgebaut? Gibt es unterschiedliche Preismodelle?

2.3.2. Trade-offs und Implikationen

Eben haben wir über bestimmte Charakteristiken von AIaaS gesprochen. In diesem Teil sollen nun die Beziehungen und Abhängigkeiten zwischen Charakteristiken analysiert werden um eventuell existierende Trade-offs aufzudeckend.

Brainstorming Phase:
Zunächst werden in dieser Phase wieder abstrakte Fragen nach Trade-Offs gestellt, ohne vorher Einfluss auf den Interviewpartner zu nehmen.

- Inwieweit gab es während der Entwicklungs- und Design-Phase Ihres AIaaS-Angebotes Situationen, in denen Sie für die Verbesserung einer Charakteristik eine Verschlechterung einer anderen Charakteristik in Kauf nehmen mussten?
- Denken Sie nochmals an die vorher genannten Charakteristiken. Welche Charakteristiken sind konträr zueinander?
- Gibt es Ihrer Meinung nach mögliche denkbare weitere, konkurrierende Charakteristiken, auch wenn Sie diese nicht auf Ihren AI-Service/auf Ihr Produkt zutreffen?

Offene Fragen:
Diese Fragen können verwendet werden, um die Diskussion über bestimmte Trade-Offs anzustoßen und dadurch Einblicke in die Praxis zu erhalten.

Benutzerfreundlichkeit vs. Anpassbarkeit

Ein auftretender Konflikt könnte zwischen Benutzerfreundlichkeit und Anpassbarkeit von AIaaS entstehen. Dabei könnte ein System mit vielen individuellen Einstellungsmöglichkeiten besonders Nutzer mit wenig Erfahrung im KI-Bereich überfordern und deshalb nicht mehr so benutzerfreundlich sein.

- Inwiefern ist dieser Trade-off auch auf Ihr System übertragbar?
- Falls, Sie diesen Trade-off eingehen mussten, welche Balance bzw. Lösung haben sie gefunden? Welche Abwägung haben Sie angestellt?
- Wie sinnvoll wäre ihrer Meinung nach eine Unterscheidung zwischen verschiedenen Nutzern? Wäre dies auch technisch umsetzbar?
  (Bsp.: AWS SageMaker bietet mittlerweile 3 unterschiedliche Rollen an: SageMaker Canvas, SageMaker for Data Scientists, SageMaker for ML-Engineers)
  - Welche Versionen gibt es und inwiefern unterscheiden sich diese?
  - Warum haben Sie sich dagegen entschieden unterschiedliche Versionen anzubieten

Automatisierung und Komplexitätsabstraktion vs. Performance u. Anpassbarkeit

Ein System, welches die unterschiedlichen Aufgaben entlang der ML-Pipeline vollständig oder teilweise automatisiert kann nicht gleichzeitig manuell anpassbar sein. Forschungsergebnisse haben gezeigt, dass das manuelle Tuning von Features, Classifier und Modell-Parameter durch erfahrene Benutzer bessere Ergebnisse im Vergleich zu dem automatischen Parameter Tuning liefern kann. Jedoch liefert manuelles Tuning umso schlechterer Ergebnisse bei schlechter Auswahl der Einstellungsmöglichkeiten (Yao et al., 2017)

- In welchem Umfang rechtfertigt die Vereinfachung durch Automatisierung von AIaaS den Verzicht auf bessere Performance des Modells?
- Inwiefern können Sie diesen Trade-off auf ihr AIaaS-Design beziehen? Was waren Implikationen für Ihren Service?
- Insbesondere die Auswahl des „richtigen“ Classifiers (Achtung: nur bei Classification-Tasks!) hat großen Einfluss auf die Performance. Ist ein Training verschiedener Classifier und anschließender Auswahl des am besten performenden ihrer Meinung nach sinnvoll und umsetzbar? Welche Nachteile ergeben sich dadurch?
- Wenn automatisiert: Wie gehen sie mit „geerbten“ AI spezifischen Trade-offs um?
  - Explainability vs. Performance
  - Latency vs. Accuracy

Generalisierbarkeit vs. Richtigkeit und Fairness des Modells

Einer der wichtigsten Faktoren für das Trainieren von KI-Modellen ist der zugrunde liegende Trainingsdatensatz. AIaaS Provider müssen versuchen repräsentative Trainingsdaten zu konzeptualisieren, jedoch gibt es nicht ein Modell, welches alle möglichen Variationen von Daten enthält. Daher können Modelle, die auf generischen Daten trainiert wurden, unter anderem schlechtere Vorhersagen liefern oder sogar diskriminierend bezüglich bestimmter Features sein. Dies hängt davon ab wie gut die Trainingsdaten die Inputdaten repräsentieren. Bei Anwendungen in unterschiedlichen Industrien/Bereichen, welche nicht in den Trainingsdaten vorhanden waren, könnte dieses Problem nochmals verstärkt werden.

- Inwieweit ist AIaaS generalisierbar und welche Nachteile bringt dies mit sich?
- Haben Sie dieses Problem bei der Bereitstellung Ihres AI-Services? Inwieweit ist ihr AIaaS-System davon betroffen?
- Wie versuchen sie diese Problematik zu adressieren?
- Was sind denkbare Lösungen für dieses Problem?

Confidentiality vs. Transparency

Einige Unternehmen könnten das Veröffentlichen von automatisierten Prozessen der ML-Pipeline oder von Informationen über genutzte Trainingsdaten verhindern (bspw. Wie Hyperparameter getunt werden, welches Model verwendet wird). Dies reduziert die Transparenz bei der Erstellung eines KI-Modelles und verstärkt das Problem, dass KI als „black-Box“ wahrgenommen wird (insbesondere bei vortrainierten Modellen, welche über eine API abgerufen werden).

- In welchem Umfang trifft das auch auf Ihre Platform/Services zu?
- Wenn sie die Transparenz erhöhen wollen: Wie würden Sie vorgehen ohne vertrauliche Daten, Prozesse und Technologien zu veröffentlichen?

Privacy vs. Performance des Modells

- Inwiefern existiert ein Trade-off zwischen Privacy und Performance des AIaaS Angebots? Verhindern beispielsweise Datenschutzrechte, dass Inputdaten zum weiteren Training des AI-Modells genutzt werden?
- Werden User-Daten zum Verbessern des Modells genutzt?

Geerbte Trade-Offs von AI:

Explainability vs. Performance

Es gibt ML-Modelle, welche von Natur aus besser erklärbar sind (bspw. Tree-based models oder lineare Regression) jedoch besonders bei komplexen Aufgaben nicht so eine gute Performance besitzen wie beispielsweise Neuronale Netze. Außerdem könnte es sein, dass post-hoc Erklärungen eigene Berechnungen benötigen, weshalb die Latenz höher liegt.

- Inwiefern existiert bei Ihrem Angebot ein Trade-off zwischen der Erklärbarkeit eines AI-Services und der Performance hinsichtlich Latenz als auch der Genauigkeit der Vorhersage?
- Wie versuchen sie diese Problematik zu adressieren?

Latenz vs. Genauigkeit

Die Latenz könnte dadurch erhöht werden, dass Modelle mit mehr Parametern zur Vorhersage genutzt werden, was von die Vorhersagegenauigkeit erhöhen kann (Halpern et al., 2019) (Ausführen mehrerer Modelle und aggregieren der Ergebnisse kann ebenfalls Latzen und Genauigkeit erhöhen)

- Inwiefern existiert dieser Trade-Off in der Praxis?
- Inwiefern kann automatische Skalierung dazu beitragen die Genauigkeit zu erhöhen, bei gleichzeitig niedriger Latenz?

Weitere mögliche Trade-Offs:

Ressourcennutzung (Energienutzung) vs. Accuracy

Synergieeffekte

- Gibt es ihrer Meinung nach, im Gegensatz zu Trade-offs, Charakteristiken von AIaaS die sehr einfach zusammen realisiert werden können?
- Inwiefern gibt es Charakteristiken welche als Voraussetzung für andere Charakteristiken erfüllt sein müssen?

2.3.3. Weitere Persönliche Erfahrungen

- Welche weiteren persönlichen Erfahrungen haben sie bezüglich Charakteristiken von AIaaS Angeboten? Beispielweise von vorherigen Positionen in anderen Unternehmen, Recherchen oder Mitbewerbern?
- Inwiefern haben Sie persönliche Erfahrungen mit weiteren Trade-Offs für AIaaS-Anbietern, auch wenn diese nicht zwingend auf Ihren Anwendungsfall beziehungsweise Ihre aktuellen Projekte zutreffen?
- Was sind die größten Herausforderungen als AIaaS Anbieter?

3. Abschluss des Interviews und nächste Schritte

5 Minuten/ 45 Minuten

Vielen Dank für Ihre Teilnahme!

- Wir werden zeitnah weitere Interviews durchführen und versuchen die ermittelten Ergebnisse in Bezug zueinander zu setzen. Stehen Sie in diesem Rahmen für Rückfragen zu Verfügung?
- Kennen Sie weitere Kollegen, die an einer Teilnahme interessiert sein könnten?
- Wie empfanden Sie vorangegangene Brainstorming-Phase in Ergänzung zu den darauffolgenden offenen Fragen?
- Wie empfanden Sie die offen formulierten Fragen? War Ihnen die Formulierung dennoch zu direkt?
- Sollen wir Ihnen die Forschungsergebnisse zukommen lassen?
- Haben Sie weitere Fragen, Anregungen oder Verbesserungen in Bezug auf das Interview oder das Projekt im Allgemeinen?

Vielen Dank für Ihre Zeit und Ihre Unterstützung!

C. Appendix C

Complexity Abstraction	Simplification of the usage, development, deployment, and operation of AI capabilities
Managed AI services	Ready to use services for inference, no needed customization upfront.
Pre-configured AI components	Task-specific development kits and pipelines developed and provisioned by the AIaaS provider.
User-friendly Interface	Services provide easy-to-use and intuitive web interfaces, sometimes even with a no-code or low-code interface.
Automation	Automation of the whole or parts of the ML workflow.
Managed Infrastructure and platform	The provider configures the necessary underlying infrastructure and computing environment.
Customizability	Ability of a service to be customizable to specific needs of a user.
Custom Models	Users can train and develop their own custom models from scratch on MLaaS platforms. This includes custom modelling, custom algorithm selection, custom model tuning and custom deployment.
Custom AutoML	Users can customize AutoML processes and functionalities of the platform to adapt the AutoML process to specific requirements or increase efficiency and performance.
Domain Adaption	Ability to adapt the underlying model of a service to a different domain by leveraging transfer learning techniques.

Genericness	The extent how generic a service is in terms of applicability to different use cases and solving diverse ML tasks.
Task-applicability	A service can be task-specific and focus on one specific ML task or task-agnostic to support the development of a variety of AI capabilities.
Domain-applicability	Whether the service includes business specific knowledge (domain-specific) or is not targeted at one industry (domain-agnostic).
Trustworthiness	Functionalities and properties that aim to increase the customer's and end-user's trust in AIaaS.
Fairness	Functionalities to monitor the fairness of an AI service.
Ethical AI	To what extent the services are ethical and methods to ensure the ethical use of AI services.
Explainability and Interpretability	Functionalities that make inferences as well as the development process of services more interpretable and explainable.
Degree of Transparency	To what extent the services are transparent to the user.
Security and Privacy	Implemented measures to assure data security and privacy.
Security	AIaaS providers are responsible for security measures that are often more secure than customer's own security measures.
Privacy	Methods and standards to increase the privacy within AIaaS. Importance of privacy depends on application area.
Deployment and Integration	How the service is made available to the customer and end user and how an AI service can be integrated into existing applications.
Inference interface	Provision of the service is possible via different interfaces including for example REST APIs or web interfaces.
Deployment model	Different deployment types of AI services are possible, ranging from public clouds to on-premises deployment.
Type of Processing	The way inference requests are processed by the system, distinguishing between batch-inference or stream-inference.
Latency	Amount of time which is needed to process an inference request by the AIaaS service.
Performance	Concerning the quality and speed of inference requests to the AI model and the training process.
Training performance	Duration and computational cost of training.
Model performance	The quality of the model in terms of context-specific metrics.

Model complexity	Size of the model regarding adjustable parameters and needed computing resources for training and inference, e.g., storage and compute.
Centralization	AIaaS is centralized and thus under the control of the AIaaS provider, who benefits from the centralization.
Intellectual Property	IP includes large and complex models that require high investments in time, computing power and expertise. Additionally, training data is another part of IP since it is required for pre-trained models.
Control and observability	Control and track service usage for billing or improvement of services.
Interoperability	Interoperability with other products and services of the same service provider or of third parties.
Functionalities	Central features and functionalities that are provided within MLaaS platforms to increase value proposition and extend the functionality of AI service.
MLOPs	Functionalities for ML lifecycle management.
Data Capabilities	Functionalities to collect and annotate training data or increase its quality.
Integration of third-party providers	Build up an ecosystem around the platform to support customers in complementary tasks.
Cloud	AIaaS inherits typical cloud characteristics.
Scalability	Availability of lots of computing resources to scale services according to the requirements and usage.
On-demand	Off-the shelf AI components are provisioned automatically.
Pay-per-use	Billing based on actual usage of the endpoint or platform.

References

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 291–300), Montreal, QC, Canada, May 25-31.

Ammanath, B., Mittal, N., Saif, I., & Anderson, S. (2021). *Becoming an AI-fueled organization: Deloitte's State of AI in the Enterprise*. Accessed on 01.06.22. Refer to https://www2.deloitte.com/content/dam/insights/articles/US144384_CIR-State-of-AI-4th-edition/DI_CIR-State-of-AI-4th-edition.pdf.

Arnold, M., Bellamy, R. K. E., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R., Ramamurthy, K. N., Olteanu, A., Piorkowski, D., Reimer, D., Richards, J., Tsay, J., & Varshney, K. R. (2019). FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development*, *63*(4/5), 6:1-6:13.

Arqane, A., Boutkhoum, O., Boukhriss, H., & El Moutaouakkil, A. (2021). A Review of Intrusion Detection Systems: Datasets and machine learning methods. In *The 4th International Conference on Networking, Information Systems and Security* (pp. 1–6), Kenitra, Morocco, April 01-02.

Baeza-Yates, R., & Liaghat, Z. (2017). Quality-efficiency trade-offs in machine learning for text processing. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 897–904), Boston, MA, USA, December 11-14.

Bishop, C. M. (2006). *Pattern recognition and machine learning* (1st ed.). *Information Science and Statistics.* Springer.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., Arx, S. v., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., . . . Liang, P. (2021). *On the Opportunities and Risks of Foundation Models*. Accessed on 31.10.22. Refer to http://arxiv.org/pdf/2108.07258v3.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J [Jeffrey], Winter, C., . . . Amodei, D. (2020). *Language Models are Few-Shot Learners*. Accessed on 31.10.22. Refer to http://arxiv.org/pdf/2005.14165v4.

Brownlee, A. E., Adair, J., Haraldsson, S. O., & Jabbo, J. (2021). Exploring the Accuracy – Energy Trade-off in Machine Learning. In *2021 IEEE/ACM International Workshop on Genetic Improvement (GI)* (pp. 11–18), Madrid, Spain, May 30.

Chui, M., Hall, B., Singla, A., & Sukharevsky, A. (2021). *The state of AI in 2021*. Accessed on 15.09.22. Refer to https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Analytics/Our%20Insights/Global%20survey%20The%20state%20of%20AI%20in%202021/Global-survey-The-state-of-AI-in-2021.pdf.

Cobbe, J., & Singh, J. (2021). Artificial Intelligence as a Service: Legal Responsibilities, Liabilities, and Policy Challenges. *SSRN Electronic Journal*, 1–61.

Corbin, J. M., & Strauss, A. L. (2015). *Basics of qualitative research: techniques and procedures for developing grounded theory* (4. ed.). Sage.

Cristofaro, E. de (2021). A Critical Overview of Privacy in Machine Learning. *IEEE Security & Privacy*, *19*(4), 19–27.

Ding, H., Gao, R. X., Isaksson, A. J., Landers, R. G., Parisini, T., & Yuan, Y. (2020). State of AI-Based Monitoring in Smart Manufacturing and Introduction to Focused Section. *IEEE/ASME Transactions on Mechatronics*, *25*(5), 2143–2154.

Dresing, T., & Pehl, T. (2018). *Praxisbuch Interview, Transkription & Analyse: Anleitungen und Regelsysteme für qualitativ Forschende* (8. Auflage). Eigenverlag.

Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N. C., & Hu, B. (2015). Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends. In *2015 IEEE 8th International Conference on Cloud Computing* (pp. 621–628), New York City, NY, USA, June 27-July 2.

Elshawi, R., & Sakr, S. (2017). *Big Data Systems Meet Machine Learning Challenges: Towards Big Data Science as a Service*. Accessed on 31.10.22. Refer to http://arxiv.org/pdf/1709.07493v1.

Geske, F., Hofmann, P., Lämmermann, L., Schlatt, V., & Urbach, N. (2021). Gateways to Artificial Intelligence: Developing a Taxonomy for AI Service Platforms. In *29th European Conference on Information Systems - Human Values Crisis in a Digitizing World, ECIS 2021,* Marrakech, Morocco, June 14-16.

Gogas, P., & Papadimitriou, T. (2021). Machine Learning in Economics and Finance. *Computational Economics*, *57*(1), 1–4.

Halpern, M., Boroujerdian, B., Mummert, T., Duesterwald, E., & Janapa Reddi, V. (2019). One Size Does Not Fit All: Quantifying and Exposing the Accuracy-Latency Trade-Off in Machine Learning Cloud Service APIs via Tolerance Tiers. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (pp. 34–47), Madison, WI, USA, March 24-26.

Hogan, M., Liu, F., Sokol, A., & Jin, T. (2011). *Nist-SP 500-291, NIST Cloud Computing Standards Roadmap*. Accessed on 31.10.22. Refer to https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=909024.

Javadi, S. A., Cloete, R., Cobbe, J., Lee, M. S. A., & Singh, J. (2020). Monitoring Misuse for Accountable 'Artificial Intelligence as a Service'. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (pp. 300–306), New York, NY, USA, February 7-9.

Javadi, S. A., Norval, C., Cloete, R., & Singh, J. (2021). Monitoring AI Services for Misuse. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 597–607), Virtual Event, USA, May 19-21.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science (New York, N.Y.)*, *349*(6245), 255–260.

Karmaker, S. K., Hassan, M. M., Smith, M. J., Xu, L., Zhai, C., & Veeramachaneni, K. (2022). AutoML to Date and Beyond: Challenges and Opportunities. *ACM Computing Surveys*, *54*(8), 1–36.

Kaur, D., Uslu, S., Rittichier, K. J., & Durresi, A. (2023). Trustworthy Artificial Intelligence: A Review. *ACM Computing Surveys*, *55*(2), 1–38.

Kreuzberger, D., Kühl, N., & Hirschl, S. (2022). *Machine Learning Operations (MLOps): Overview, Definition, and Architecture*. Accessed on 31.10.22. Refer to https://arxiv.org/ftp/arxiv/papers/2205/2205.02302.pdf.

Lapan, S. D., Quartaroli, M. T., & Riemer, F. J. (2012). *Qualitative research : an introduction to methods and designs* (1st ed.). Jossey-Bass.

Leroux, S., Simoens, P., Lootus, M., Thakore, K., & Sharma, A. (2022). TinyMLOps: Operational Challenges for Widespread Edge AI Adoption. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)* (pp. 1003–1010), Lyon, France, May 30-June 03.

Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., & Zhou, B. (2022). Trustworthy AI: From Principles to Practices. *ACM Computing Surveys*, Article 3555803. Advance online publication. https://doi.org/10.1145/3555803

Linneberg, M. S., & Korsgaard, S. (2019). Coding qualitative data: a synthesis guiding the novice. *Qualitative Research Journal*, *19*(3), 259–270.

Lins, S., Pandl, K. D., Teigeler, H., Thiebes, S., Bayer, C., & Sunyaev, A. (2021). Artificial Intelligence as a Service. *Business & Information Systems Engineering*, *63*(4), 441–456.

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. Gaithersburg, MD, USA. Accessed on 31.10.22. Refer to https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

Mo, F., Haddadi, H., Katevas, K., Marin, E., Perino, D., & Kourtellis, N. (2021). PPFL: Privacy-Preserving Federated Learning with Trusted Execution Environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (pp. 94–108), New York, NY, USA, June 24-July 02.

Myers, M. D. (2013). *Qualitative research in business & management* (2 ed.). Sage.

National Institute of Standards and Technology (2019). *U.S. LEADERSHIP IN AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools*. Accessed on 04.06.22. Refer to https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf.

Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022). Challenges in Deploying Machine Learning: a Survey of Case Studies. *ACM Computing Surveys*, Article 3533378. Advance online publication. https://doi.org/10.1145/3533378

Pandl, K. D., Teigeler, H., Lins, S., Thiebes, S., & Sunyaev, A. (2021). Drivers and Inhibitors for Organizations' Intention to Adopt Artificial Intelligence as a Service. In *Proceedings of the 54th Hawaii International Conference on System Sciences* Grand Wailea, Maui, Hawaii, January 5 - 8.

Pessach, D., & Shmueli, E. (2023). A Review on Fairness in Machine Learning. *ACM Computing Surveys*, *55*(3), 1–44.

Phaphoom, N., Oza, N., Wang, X., & Abrahamsson, P. (2012). Does cloud computing deliver the promised benefits for IT industry? In *Proceedings of the WICSA/ECSA 2012 Companion Volume on - WICSA/ECSA '12* (p. 45), Helsinki, Finland, August 20-24.

Philipp, R., Mladenow, A., Strauss, C., & Völz, A. (2020). Machine Learning as a Service. In *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services* (pp. 396–406), Chiang Mai Thailand, November 30-December 02.

Pushkarna, M., Zaldivar, A., & Kjartansson, O. (2022). Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (pp. 1776–1826), Seoul Republic of Korea, June 21-24.

Ribeiro, M., Grolinger, K., & Capretz, M. A. (2015). MLaaS: Machine Learning as a Service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* (pp. 896–902), Miami, FL, USA, December 09-11.

Russell, S., & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall Press.

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2, 160*(3), 1–21.

Savu, L. (2011). Cloud Computing: Deployment Models, Delivery Models, Risks and Research Challenges. In *2011 International Conference on Computer and Management (CAMAN)* (pp. 1–4), Wuhan, China, May 19-21.

Tanuwidjaja, H. C., Choi, R., Baek, S., & Kim, K. (2020). Privacy-Preserving Deep Learning on Machine Learning as a Service—a Comprehensive Survey. *IEEE Access*, *8*, 167425–167447.

Thoppilan, R., Freitas, D. de, Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., . . . Le, Q. (2022). *LaMDA: Language Models for Dialog Applications*. Accessed on 31.10.22. Refer to https://arxiv.org/pdf/2201.08239.pdf.

Wang, D., Liao, Q. V., Zhang, Y [Yunfeng], Khurana, U., Samulowitz, H., Park, S., Muller, M., & Amini, L. (2021). *How Much Automation Does a Data Scientist Want?* Accessed on 10.10.22. Refer to http://arxiv.org/pdf/2101.03970v1.

Xiao, Z [Zhifeng], & Xiao, Y. (2013). Security and Privacy in Cloud Computing. *IEEE Communications Surveys & Tutorials*, *15*(2), 843–859.

Yao, Y., Xiao, Z [Zhujun], Wang, B., Viswanath, B., Zheng, H., & Zhao, B. Y. (2017). Complexity vs. performance. In *Proceedings of the 2017 Internet Measurement Conference* (pp. 384–397), London, United Kingdom, November 01-03.

Zapadka, P., Hanelt, A., Firk, S., & Oehmichen, J. (2020). Leveraging "AI-as-a-Service" - Antecedents and Consequences of Using Artificial Intelligence Boundary Resources. In *Proceedings of the 41st International Conference on Information Systems, ICIS 2020, Making Digital Inclusive: Blending the Local and the Global* Hyderabad, India, December 13-16.

Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M. P., Huang, H., & Molloy, I. (2018). Protecting Intellectual Property of Deep Neural Networks with Watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 159–172), Incheon, Republic of Korea, June 04 - 08.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76.

Declaration about the Thesis

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Quellen und Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben

F. Linnemann

Karlsruhe, den 7. November 2022

FELIX LINNEMANN