Bus Inf Syst Eng 63(4):441–456 (2021)
https://doi.org/10.1007/s12599-021-00708-w

CATCHWORD

Artificial Intelligence as a Service

Classification and Research Directions

Sebastian Lins · Konstantin D. Pandl · Heiner Teigeler · Scott Thiebes · Calvin Bayer · Ali Sunyaev

Received: 8 April 2020 / Accepted: 25 May 2021 / Published online: 13 July 2021

Keywords Artificial intelligence as a service · AIaaS · Artificial intelligence · Cloud computing · Machine learning as a service

1 The Roots of Artificial Intelligence as a Service

1.1 The Emergence of Artificial Intelligence Services

Artificial Intelligence (AI) is undoubtedly one of the most actively debated technologies, providing auspicious opportunities to contribute to individuals' well-being, the success and innovativeness of organizations, and societies' prosperity and advancement (Thiebes et al. 2020). The McKinsey Global Institute predicts that the utilization of AI could yield an additional worldwide economic output of USD 13 trillion by 2030 (Bughin et al. 2018). Organizations increasingly employ AI to perform complex tasks that previously only humans were thought to be capable of performing. In some narrow application domains, AI now even surpasses the performance of humans. Examples of such complex tasks include analyzing medical data to assist physicians in making medical treatment decisions faster and more accurately (Madani et al. 2018), or analyzing large amounts of video footage in hours or days instead of months to support criminal investigations (Crawford 2019). However, one major challenge for organizations is the complex and demanding process of adopting and integrating AI, which is rather considered "*a journey and not a destination*" (Dutta 2018). This prevalent reluctance arises from the scarcity of AI experts (Chui and Malhotra 2018); a lack of organizations' abilities and budgets to set up and maintain the extensive IT resources needed (Romero et al. 2019); and limited knowledge on how to deploy and configure the AI-based systems effectively (Yao et al. 2017), among others. As a result, most organizations still fail to adopt AI and harness its full potential (Ransbotham et al. 2019; Zapadka et al. 2020).

To foster AI diffusion and application, cloud providers such as *Amazon*, *Google, IBM, Microsoft, Salesforce,* or *SAP* have started to offer machine learning, deep learning, analytics, and inference as a service, bringing the discussions about provisioning AI capabilities from the cloud into practice. Also, start-ups and small and medium-sized enterprises (SME) are following the trend and providing unique cloud-based AI services tailored to SMEs' needs in various industries. *Incomaker*, for example, offers AI-based sales and marketing automation tools. These services became known as *Artificial Intelligence as a Service* (AIaaS). In its essence, AIaaS combines AI (i.e., the ability of a machine to perform cognitive functions that we

Accepted after two revisions by Ulrich Frank.

S. Lins (✉) · K. D. Pandl · H. Teigeler · S. Thiebes · C. Bayer · A. Sunyaev
Institute AIFB, Research Group Critical Information Infrastructures, Karlsruhe Institute of Technology, Kaiserstr. 89, 76133 Karlsruhe, Germany
e-mail: sebastian.lins@kit.edu

K. D. Pandl
e-mail: konstantin.pandl@kit.edu

H. Teigeler
e-mail: heiner.teigeler@kit.edu

S. Thiebes
e-mail: scott.thiebes@kit.edu

C. Bayer
e-mail: calvin.bayer@gmx.de

A. Sunyaev
e-mail: sunyaev@kit.edu

associate with human minds (Rai et al. 2019)) with the cloud computing model, which is known for "enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources […] that can be rapidly provisioned" (Mell and Grance 2011).

AIaaS has the objective to make AI accessible and affordable across the board, whether or not an organization is big, technologically advanced, or has large budgets to spend on AI. AIaaS guides its users through the process of developing, deploying, or using data analytics models without the need to learn complex algorithms or technologies (Elshawi et al. 2018). Users can then focus on, for example, training and configuring their AI models, thereby pursuing their core competencies and not having to concern themselves with installation, maintenance, and related management problems (Boag et al. 2018).

An illustrative example of how an AIaaS could be used in practice is developing an industrial quality control system based on camera images of a manufactured product. When active, the camera takes images and sends those images to an AIaaS offering computer vision capabilities to predict whether the product condition is sufficient or whether it shows a defect. This way, developers of the visual inspection system do not have to deal with the technical details of the computer vision algorithm's underlying creation and implementation. Instead, concrete hardware or configuration decisions requiring AI experts' knowledge are made by the AIaaS provider.

1.2 The Growing Need for Conceptual Clarity on the Term 'Artificial Intelligence as a Service'

Following the market trends of AIaaS, researchers in diverse disciplines, including information systems, computer science, and management, have started to focus their research on provisioning AI capabilities from the cloud. Diverse research streams on AIaaS recently emerged that deal, for example, with the design and evaluation of AI services (Boag et al. 2018; e.g., Elshawi et al. 2018), the adoption and effective use of AIaaS (e.g., Zapadka et al. 2020; Pandl et al. 2021), uncovering AIaaS misuse by its users (e.g., Javadi et al. 2020), or understanding AIaaS's issues and vulnerabilities (e.g., Truex et al. 2019).

The research field on AIaaS itself is still scattered and combines terminologies and approaches from multiple disciplines. While the term "artificial intelligence as a service" is seldom found in the literature (e.g., Javadi et al. 2020; Zapadka et al. 2020), researchers and practitioners use an ever-increasing amount of different terms to describe the phenomenon. "Machine learning as a service" is certainly most widely encountered in the literature (Duong and Sang 2018; e.g., Yao et al. 2017), but related terms are also, such as "deep learning as a service" (e.g., Boag et al. 2018), "inference as a service" (e.g., Romero et al. 2019), "neural networks as a service" (Huqqani et al. 2014), or "analytics as a service" (e.g., Naous et al. 2017), among others. These terms are mostly driven by practice, innovations, and the ever-increasing number of offerings on the market. In addition, these terms mostly cover AI software and applications, and thus AIaaS literature mostly relates to the conventional software as a service (SaaS) cloud model (e.g., Javadi et al. 2020). On the contrary, cloud providers have already started offering AI developer services and AI infrastructure services, relating to the conventional platform (PaaS) and infrastructure as a service (IaaS) cloud models that have been neglected by prevalent research so far. As a consequence, we still witness no uniform conceptualization of AIaaS in literature and practice.

Our catchword article aims to deepen our understanding of the phenomenon 'AIaaS' and foster conceptual clarity to support both practitioners and researchers. To do so, we first propose a definition of AIaaS and divide AIaaS into three layers hierarchically organized as a stack, based on a literature review on AIaaS research and interviews with experts from the field (Sect. 2). We also discuss core characteristics commonly shared by AIaaS, such as abstracting the complexity of AI services for users and inheriting cloud characteristics (Sect. 3). We then briefly discuss open challenges and future research directions for the BISE community (Sect. 4), followed by a conclusion (Sect. 5).

2 The Artificial Intelligence as a Service Stack

We define **AIaaS as cloud-based systems providing on-demand services to organizations and individuals to deploy, develop, train, and manage AI models.** Reflecting this broad definition reveals that AIaaS not only relates to AI software and applications available on-demand, such as chatbots using natural language processing, but also covers tools and resources needed to develop, operate and maintain AI models. In line with the typical cloud service models (Liu et al. 2011; Mell and Grance 2011), we want to highlight that AIaaS can be divided into three layers, hierarchically organized as a stack according to the abstraction level of the capability provided (Fig. 1):

(1) AI software services that are ready-to-use AI applications and building blocks (relating to the conventional SaaS cloud layer),

(2) AI developer services that are tools for assisting developers in implementing code to bring out AI capabilities (relating the conventional PaaS cloud layer),

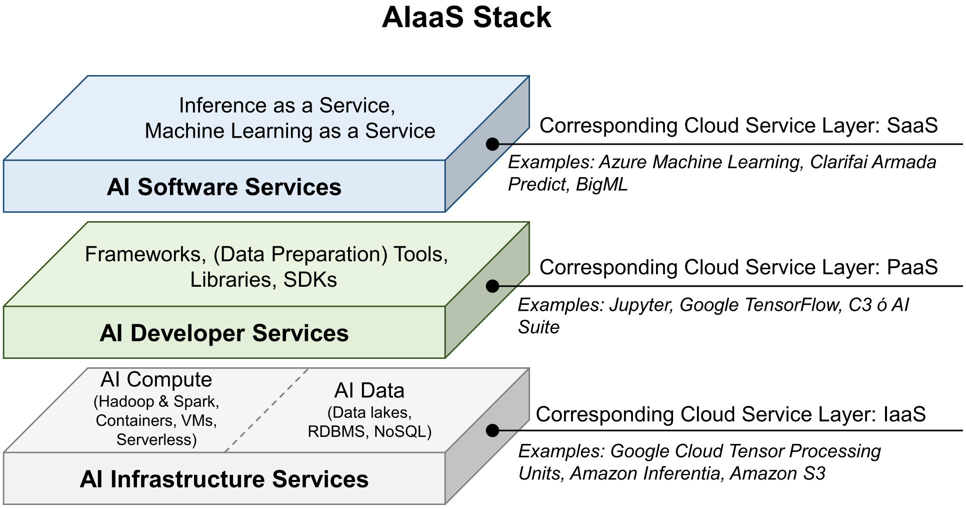
AIaaS Stack
Inference as a Service,
Machine Learning as a Service
AI Software Services
Corresponding Cloud Service Layer: SaaS
Examples: Azure Machine Learning, Clarifai Armada Predict, BigML
Frameworks, (Data Preparation) Tools,
Libraries, SDKs
AI Developer Services
Corresponding Cloud Service Layer: PaaS
Examples: Jupyter, Google TensorFlow, C3 ó AI Suite
AI Compute
(Hadoop & Spark, Containers, VMs, Serverless)
AI Data
(Data lakes, RDBMS, NoSQL)
AI Infrastructure Services
Corresponding Cloud Service Layer: IaaS
Examples: Google Cloud Tensor Processing Units, Amazon Inferentia, Amazon S3

Fig. 1 AIaaS stack in line with the conventional cloud service stack (cf. Liu et al. 2011; Mell and Grance 2011)

(3) AI infrastructure services that comprise raw computational power for building and training AI algorithms, and network and storage capacities to store and share data (relating to the conventional IaaS cloud layer).

It is possible, though not necessary, that organizations can build AI software services on top of AI developer services, which in turn rely on an AI infrastructure service, leading to entangled cloud supply chains. The optional dependency relationships among AI software, developer, and infrastructure services form the AIaaS stack, while each layer can stand by itself (Liu et al. 2011). In the following, we briefly describe each layer in more detail.

2.1 AI Software Services

The most prominent and frequently used types of AIaaS are **AI software services** that are ready-to-use applications or building blocks (Javadi et al. 2020). They relate to the conventional SaaS cloud models (cf. Mell and Grance 2011). Today, most developed, deployed, used AI-based systems are based on machine learning or deep learning methods (Pandl et al. 2020; Thiebes et al. 2020). As such, machine-learning-based techniques are also crucial technologies for the most popular AI software services. These machine-learning-based AI software services are referred to as **inference as a service**, where users can access pre-trained machine learning models, or **machine learning as a service (MLaaS)**, where users can create and customize machine learning models (Table 1). Given the popularity and relevance of MLaaS and inference as a service, we briefly outline their functionalities as prominent examples of AIaaS and AI software services in particular.

Because the development and training of an AI model are expensive and time-consuming, AI models became a form of intellectual property and, therefore, increasingly represent an essential factor in achieving competitive advantages (Haenlein and Kaplan 2019). Efforts to protect competitive advantages can thus lead to situations in which promising AI models are not shared with others (Thiebes et al. 2020). To counteract this issue, a type of AI software service emerged that removes users' burden of setting up and training, and offers pre-trained models, referring to AI models already trained by the AIaaS provider (or other parties) and then made available to users. We refer to this as **inference as a service**; however, the nomenclature of these services depends strongly on the provider as well as the purpose of the service (e.g., prediction application programming interface (API) (Tramèr et al. 2016)). Inference as a services typically provide a query interface to a machine learning classifier trained on existing or user-uploaded datasets (Yao et al. 2017). They thereby simplify running AI models by automatically taking control over data storage, classifier training, and classification, among others.

Different types of inference as a service are accessible on-demand nowadays, such as language services (e.g., text analytics or translation), analytics services (e.g., product recommendations or knowledge inference from big data), speech services (e.g., text-to-speech, speech-to-text), or computer vision services (e.g., analyzing of images and videos in order to find and identify objects, text, and labels) (Javadi et al. 2020; Pandl et al. 2021). It is easy for developers of all skill levels to use machine learning technology by relying on pre-trained models (Ramesh 2017). Users with limited knowledge and related expertise do not have to engage in the time-consuming and labor-intensive aggregation of large amounts of data but can rely on the knowledge representation in the pre-trained AI models. Notably, users whose core competence is not in AI benefit from the access to providers' expert knowledge as

Table 1 Overview of different AI software service types

AI software service	Inference as a service	Machine learning as a service
Definition	Services provide access to pre-trained machine learning models	Service guide users along the machine pipeline to build and configure AI models
Characteristics	Users can query pre-trained AI models to receive inferences; fully automated (black-box) systems; requiring less AI knowledge; typically lacking customizability options	Offering many features and customizability options for users; higher optimal performance; requiring more AI knowledge
Related terms	Prediction API (Romero et al. 2019), recommendations as a service (Baldominos et al. 2015)	Deep learning as a service (Boag et al. 2018), neural networks as a service (Huqqani et al. 2014), training as a service (Zhang et al. 2017)
Implementations	Amazon Transcribe, Clarifai Armada Predict, EPICA, Apache PredictionIO	Azure Machine Learning, IBM Watson Studio, BigML, Domino, Arimo

they do not require scarce AI domain expertise in-house (Truex et al. 2019). These advantages are among the most discussed benefits in prior research and practice because successfully optimizing each machine learning step requires overcoming significant complexity that is difficult without in-depth knowledge and experience (Yao et al. 2017). Offering inference as a service and pre-trained models is thus an efficient means to make promising AI models more widely available to be highly beneficial to society (Thiebes et al. 2020).

Inference as a services are typically black-box systems and offer few or even no means to customize the AI model or underlying data sets (Yao et al. 2017). On the contrary, more control and customizability over AI model configuration allow knowledgeable users to build higher quality models because feature, model, and parameter selection can significantly impact the performance of a machine learning task. As another type of AI software services, **MLaaS** emerged to provide (knowledgeable) users many features and customizability options (Huqqani et al. 2014; Yao et al. 2017; Boag et al. 2018). In its essence, MLaaS guides users in developing and configuring AI models along the machine learning pipeline (Fig. 2). It enables users to focus on training and choosing hyper-parameters, among others, rather than focusing on installation, configuration, and fault tolerance of required AI infrastructure (Boag et al. 2018).

Following the machine learning pipeline, MLaaS assists users in pre-processing their data as a first step. For example, in machine-learning-based image processing, a user could scale down images stored on the cloud storage to a uniform, manageable resolution to prepare these images for further machine learning steps. Afterward, the MLaaS guides users to predefine representations of the data, known as a feature selection step. For example, such a feature could be a vector of the average intensity of the image pixels across different areas in the image. This step can be required for some machine learning models (e.g., support vector machines), whereas other machine learning models may automatically learn such representations (e.g., deep neural networks). This step is closely interrelated with the subsequent classifier choice. A classifier is a hypothesis or discrete-valued function that is used to assign labels to particular data instances. Afterward, the MLaaS guides the user in tuning a set of hyper-parameters of the machine learning classifier, for example, the number of layers of a deep neural network. A training algorithm then optimizes the model parameters to fit the dataset well on a predefined

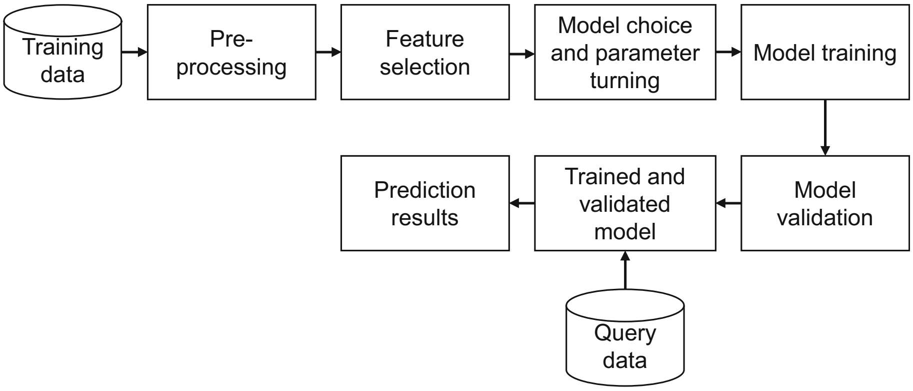

Fig. 2 Steps comprised by the machine learning pipeline and offered by MLaaS (adapted from Yao et al. 2017)

performance metric (e.g., prediction error). When processing large amounts of data, the training process can be computationally expensive. Consequently, AI software services emerged that specifically focus on this training process, referred to as training as a service (Zhang et al. 2017). After training the model, a user can validate the model's performance, for example, by evaluating the performance on a separate test dataset. Once a model passes validation, the user can execute the model in the cloud environment on query data instances and receives individual results.

2.2 AI Developer Services

AIaaS also provides easy-to-use **AI developer services** by giving developers access to tools that help them implement code to bring out AI capabilities. AIaaS thereby also relates to the conventional PaaS cloud models (cf. Mell and Grance 2011). For example, *Azure ML Studio* offers a suite of pre-built examples and startup codes; *C3 – AI Suite* comes with standard AI algorithms and models; and *Dataiku* provides tools that allow data scientists, data analysts, and AI developers to work together. These services thereby particularly support AI developers to develop and manage (novel) AI applications.

AI developer services comprise various tools and frameworks that can be used by developers (Table 2). Nowadays, (open-source) AI frameworks are offered as on-demand services comprising various AI algorithms and tools for effective exploitation of respective algorithms, such as *Tensorflow*, *PyTorch*, *Caffe*, *Theano*, *Horovod*, and *MXNet*. These AI frameworks commonly reduce efforts in designing, training, and using AI models (Boag et al. 2018). For example, Google's framework *Tensorflow* is an open-source platform for machine learning and provides a collection of workflows to develop and train models.

AI developer services also offer specific tools enabling faster coding and easier integration of APIs, such as *PyCharm*, *Microsoft VS Code*, *Jupyter*, or *MATLAB*. In particular, AIaaS providers started to offer various data preparation tools since a machine learning model's efficiency depends on the data quality (Yoon and Kang 2017). These data preparation tools assist in extracting, transforming, and loading data, which is then fed into the machine learning algorithm for training and evaluation. Usually, users send their raw data to the AI data preparation service in the given format, and the service automatically handles the pre-processing and post-processing. Such preparation tools make the integration of AIaaS more convenient for users, as they do not have to convert their data into a format demanded by the AI model as input. Therefore, data scientists especially benefit from using such AI preparation services as they can focus solely on the data itself (Ribeiro et al. 2015).

Besides, developers may provision AI libraries and software development kits referring to a set of low-level software functions that help optimize the deployment of an AI framework on a given infrastructure. These libraries are integrated directly into the source code of the AI application and enable developers to interact with the service API by calling methods included in the library. For example, libraries for managing tabular and time-series data (e.g., *pandas*), for leveraging advanced mathematical operations (e.g., *NumPy*), or to add specific cognitive capabilities, such as computer vision (e.g., *OpenCV*) or language translation (e.g., *OpenNMT*), are available on-demand. By using such AI developer services, the barrier of integrating AIaaS into existing software products is reduced, enabling any developer to make use of AI.

2.3 AI Infrastructures Services

Finally, AIaaS offers **AI infrastructure services** referring to the raw computational power for building and training AI algorithms, and network and storage capacities to store and share (training and inference) data. Consequently, AI infrastructure services not only comprise computing resources for efficient deployment and use of AI developer tools and services, relating to the conventional IaaS cloud model (cf. Mell and Grance 2011), but AI infrastructures

Table 2 Overview of example AI developer services

AI developers service	AI frameworks	Developer tools	AI libraries and software development kits
Description	Services provide specific AI algorithms (e.g., Naive Bayes Classification) and tools for effective exploitation of respective algorithms	Services support data preparation or enabling faster coding and easier integration of APIs	Services offer low-level software functions that help optimize the deployment of an AI framework on a given infrastructure
Implementations	Tensorflow, PyTorch, Caffe, Theano, Horovod, and MXNet	PyCharm, Microsoft VS Code, Jupyter, or MATLAB	pandas, NumPy, OpenCV, OpenNMT

also store data relevant for AI model training and inference and provide appropriate data storage and sharing technologies, and respective management processes (Table 3).

First, AIaaS users typically have a wide choice of provisioning computing resources, such as physical servers, virtual machines, containers, or AI-specialized hardware such as graphic processing units (GPUs) or so-called tensor processing units (TPUs) for computations. For example, applying complex deep learning and neural networks might demand complementing central processing units (CPUs) with GPU to enable faster calculations. Providers may offer additional compute services, such as batch and stream processing, container orchestration, and serverless computing, to parallel and automate machine learning steps. Various AI software and developer services nowadays build on *AWS*'s compute, networking, and storage services, enabling them, for instance, to use high-performance machine learning inference chips designed and built by *AWS*. Likewise, *Google's* cloud platform offers access to TPUs, which are specialized hardware for training neural networks using the *TensorFlow* framework.

Second, the AI infrastructure typically provides access to relational or NoSQL databases, or the capability to upload and integrate external data lakes as input to train AI models. Data in its diverse forms and functions constitute the single, most important resource for AI-based systems. However, creating high-quality training data is costly and time-consuming, particularly in situations where experts are required to annotate data (e.g., annotation of large medical data sets). Consequently, large, high-quality data sets are primarily found in data silos of a few large enterprises, and in contrast, there are only a few freely available high-quality data sets, which are limited to a handful of specific application areas (Thiebes et al. 2020). Cloud-based AI infrastructure services may counteract these issues by enabling efficient data storage and sharing for a large amount of AI data and respective models (Pandl et al. 2021). Such data can be used internally to train AI models, and also be provided by data as a service providers on users' request via data APIs or web interfaces with granular authentication and authorization controls and pricing models (e.g., volume-based or data type-based subscriptions) (Javadi et al. 2020). Combining data silos can increase the accuracy of AI-based systems, or enable the application of AI-based systems in the first place (Dorard et al. 2016).

3 Core Characteristics of Artificial Intelligence as a Service

AI software, developer, and infrastructure services share several unique and innovative characteristics that enable organizations to use AI in their contexts effectively. These core characteristics comprise complexity abstraction, automation, customizability, and inherited cloud characteristics (Table 4), which we discuss in detail in the following.

3.1 Complexity Abstraction

AIaaS offers the most advantages for SMEs because they often lack staff with appropriate know-how, and special hardware and software to develop and implement their own AI applications. Best practices, cost estimations, and a suitable amount and quality of training data for developing own AI applications are often not readily available for SMEs. In response to this, AIaaS simplifies the usage of AI technologies and makes AI accessible through **complexity abstraction** (Pandl et al. 2021). Complexity abstraction not only relates to hiding implementation details of an AIaaS and its underlying computing layers but also to handing over the control and responsibility of the service to the AIaaS provider. For example, in AI software services, users do not need to have their own hardware resources, software, and respective know-how because AI services are on the providers' side and are therefore entirely abstracted from the users' point of view. Complexity abstraction applies to each AIaaS stack layer by abstracting the complexity of the respective service layers. Abstraction enables users to achieve a short time-to-market for their AI applications because they do not have to start from scratch and spend a lot of time planning, developing, and setting up the required hardware or developer tools (Javadi et al. 2020).

Table 3 Overview of example AI infrastructure services

AI infrastructure service	AI computing resources	AI data storage and sharing
Description	Services provide AI computing resources, such as physical servers, virtual machines, containers, or AI-specialized hardware such as GPUs for computations	Services provide appropriate storage and sharing technologies, and respective management processes for data relevant for AI model training and inference
Implementations	*Google Cloud Tensor Processing Units, AWS Inferentia*	*Amazon S3, Azure Blob Storage*

Table 4 AIaaS's core characteristics

Core Characteristic	Attribute	Description	Example Benefits for Users
Complexity abstraction	Hardware abstraction	The AIaaS provider deploys and maintains efficient AI infrastructures, and handles performance peaks dynamically	Getting access to AI computing resources and expertise; achieving short time-to-market; focusing on core competencies; optimizing users' core business with the support of cloud-based AI services
	Setup and configuration abstraction	Users are not required to have time or skills to deal with system setup, resource selection, and configuration	
	Maintenance abstraction	The AIaaS provider manages and maintains the underlying hardware and software infrastructure	
Automation	Automatic classifier selection	AIaaS automatically selects a proper classifier, so the user is not required to know or even select which model-variant is most suitable to meet their application's requirements	Deploying AI technologies faster and with higher technical robustness while having little prior knowledge about AI; achieving higher performance and resilience; no need to rely on AI engineers, which may be challenging to find on the job market
	Automatic hyper-parameter tuning	AIaaS performs automated hyper-parameter tuning of the AI model, such as Random search and Bayesian optimization	
	Server-side hardware tuning	AIaaS automatically adapts and optimizes the underlying hardware concerning the unique demands of an AI algorithm	
	Automatic failure handling	AIaaS handles failures automatically and restarts failed (machine learning) tasks	
Customizability	Custom classifier selection	Users can select and experiment with custom classifiers to achieve near-optimal results	Optimizing AI models; achieving higher performance; increasing flexibility; improving cost/benefit ratio
	Custom hyper-parameter tuning	Users can perform manual adjustments on variables that affect the classifier	
	Custom algorithms	Users can integrate their own custom data analysis scripts	
	Customizable and extendable architecture	Users can integrate third-party services, connect with various cloud-based AI infrastructures, and configure these infrastructures to meet their needs	
Cloud characteristics	On-demand self-service	AIaaS users can typically provision cloud capabilities as needed automatically and unilaterally	Easy and anywhere access; parallelization of tasks; increasing flexibility; improving performance; cost savings; increasing cost transparency; using trial subscriptions
	Resource pooling	AIaaS can effectively support multiple concurrent tenants, enabling multiple trainings and executions of different users' AI models in parallel	
	Scalability	AIaaS providers can elastically provision and release hardware resources and scale horizontally following the user-defined configurations and requirements	
	Broad network access	Users may access the AIaaS through APIs or a simple web interface without any programmable integration	
	Measured service and pay-as-you-go	Usage of AIaaS is continuously monitored, enabling pricing models that demand users to pay only for the time using the resources	

While the time to develop their own solutions is a factor of uncertainty that prevents organizations from experimenting with AI applications, organizations can provision ready-to-use AIaaS and thus focus on their core business. Users can then better position themselves in the market by generating competitive advantages and optimizing their core business with the support of cloud-based AI services, developer tools, or infrastructures.

AIaaS particularly decreases the efforts when implementing AI applications by conceptualizing, setting up, and maintaining the underlying hardware and software infrastructure. Primarily, the advantages in abstraction originate from users requiring no hardware resources because the provider manages resources. This hardware abstraction is highly valuable in the context of AI because an efficient conceptualization of hardware architectures for the execution of AI models requires the optimal composition of complementary hardware components, such as combining CPUs and GPUs, and therefore extensive knowledge of the properties, benefits, and boundary conditions of various hardware components (Romero et al. 2019). The AIaaS provider has the expertise required to develop and maintain efficient AI infrastructures and is also able to deploy expensive, specialized hardware (i.e., GPUs or TPUs) and handle performance peaks dynamically due to efficient utilization of the hardware and economics of scale, in contrast to users deploying AI in-house. AIaaS providers also rely on cost-effective storage for large amounts of data that are, concerning AI, reflected in training datasets or the results of batch processing tasks by building on cloud computing storage concepts such as *Amazon S3* or *Microsoft Azure Blob Storage* (Arnaldo et al. 2015). Central data stores also enable fast read and write operations of AI algorithms and prevent large-scale data redundancies, for example, several AI algorithms use a shared set of training data, thereby saving time and resources (Dorard et al. 2016).

Besides the procurement of the required hardware, users do not have to concern themselves with the proper setup, configuration, and maintenance of these computing resources. Apart from the fact that users should have comprehensive knowledge and in-depth experience for an optimal setup and configuration, the process itself is challenging and time-consuming (Duong and Sang 2018). Users need to manage physical and virtual machines and install required AI libraries, which is more challenging in the context of AI because users have to ensure the resilience of the training jobs and facilitate consistent response times for inference requests, among others (Bhattacharjee et al. 2017). Consequently, AIaaS spares users considerable complexity as they bypass setup and configuration and transfer this task (and related risks) to the AIaaS provider. Finally, the employment of AIaaS transfers maintenance responsibilities to the provider, which is very challenging in the context of AI, given a high pace of updates to AI frameworks in the open-source communities (Bhattacharjee et al. 2017).

3.2 Automation

AIaaS also achieves high degrees of **automation** because AIaaS enables users to optimize their AI models automatically, provides a selection of the most suitable hardware architectures, and handles hard- and software failures in an automated manner (Zapadka et al. 2020; Pandl et al. 2021). Thereby, automation impacts each AIaaS stack layer.

When using AI software services or AI frameworks offered by AI developer services, classifier selection and hyper-parameter tuning become crucial for optimizing AI models. The selection of different classifiers can lead to varying degrees of accuracy on a given dataset. No universal recommendation can be made for arbitrary data as to which classifier will perform best (Reif et al. 2014). It is, therefore, difficult for users to determine a suitable classifier. AI software and developer services often automate the selection of an optimal classifier and, thus, shift this difficulty from the user to the provider side. The user is only required to upload the training data onto the platform, which then uses server-side tests to determine the classifier promising the highest accuracy, often differentiating between linear and non-linear classifiers (Yao et al. 2017). Although these tests occasionally err and choose non-optimal classifiers, the classifier's automated adaption to the dataset and automated optimizations in the background provide better performance on average than services using statically defined classifiers (Yao et al. 2017). In addition to the fundamental choice of the classifier for an AI algorithm, fine-grained adjustments of the hyper-parameters can have a large influence on the performance of the AI model (Reif et al. 2014), and appropriate settings are considered crucial for the accuracy of the prediction (Chan et al. 2013). AI software and developer services also support users through automatic hyper-parameter tuning, hence further optimizing the performance of the AI algorithm. Besides popular automatic tuning approaches, such as Random search and Bayesian optimization (Wang et al. 2018), accessing observations of performance and characteristics of previously trained models allows providers to improve the automatic tuning of hyper-parameters even more (Bhattacharjee et al. 2017). Hereby, providers analyze historical data (across their users) to understand which hyper-parameter configurations yielded satisfactory results in the past.

Aside from improving the AI model's accuracy, there are enhancements in speed and efficiency due to automation relating to the AI infrastructure. AI infrastructure services automatically adapt and optimize the underlying hardware concerning the unique demands of an AI algorithm. Each hardware architecture is unique in terms of its performance potential and optimization requirements and thus significantly impacts cost and processing time. For instance, small batch sizes and low requirements towards a low latency make the use of CPUs appealing as they are cost-effective (Hazelwood et al. 2018), whereas using GPUs allows for a more than ten-fold higher throughput, especially for large batch sizes (Romero et al. 2019). Additional hardware options include field-programmable gate arrays (FPGAs) and innovative training accelerators, such as *Google's* TPUs, or inference accelerators, such as *AWS's Inferentia*. Accordingly, users benefit from an automated optimization by leveraging AIaaS's unique hardware resources depending on users' AI-model-specific needs that users cannot achieve when deploying AI in-house.

AIaaS is also perceived as being more resilient than in-house AI applications due to automated handling of failures in the infrastructure and software stack, including physical machine crashes, loss of network connectivity, crashes of containers, or failures of sub-services on which the AIaaS depends (Bhattacharjee et al. 2017). Preventing these failures and effective recovery is especially crucial for AI-based systems since, for example, training a deep neural network with a large dataset may take days and losing the progress due to failure would be critical. If failures are based on user input errors, AIaaS automatically provides meaningful error messages in the log (Bhattacharjee et al. 2017). Furthermore, an AIaaS can retry failing tasks automatically a certain number of times before they are marked as failed (Bhattacharjee et al. 2017). Finally, many other advantages of conventional cloud solutions further strengthen the resilience of AIaaS, such as automatic backups of AIaaS applications and data.

3.3 Customizability

AIaaS not only provides glaring opportunities for organizations with limited AI expertise or resources but also provides users having this expertise and experience in the domain of AI with the functionality to individually create, configure, modify, and control their AI models. Such **customizability** enables them to optimize their AI models to their needs fully. Prior research has shown a correlation between increasing configurability and higher optimal performance of AI models (Yao et al. 2017). Likewise, recent research highlights that organizations with high internal AI capabilities use AIaaS, particularly for internal process improvements and complementing their knowledge base (Zapadka et al. 2020). Several providers have emerged on the market to serve users with different levels of knowledge, which differ, among other aspects, in the scope of possible configuration and customizability options. *BigML,* for instance, offers users a choice between four classifiers, while *Microsoft Azure ML Studio* allows the user to control everything except for the implementation of the program and therefore may outperform other services when configurations of the model are carefully tuned (Yao et al. 2017).

The most frequently addressed aspect of customizability of AI software and developer services is selecting a custom classifier. Classifier choice accounts for much of the benefits of customization, and users can achieve near-optimal results by experimenting with a small random set of classifiers (Yao et al. 2017). By using multiple AIaaS instances in parallel, users run multiple algorithms, each using a different classifier, and compare their performances, so the most suitable one can be identified (Ribeiro et al. 2015). There are also fine-granular adjustment options, such as individually tuning the hyper-parameters applied to a model. A dashboard may be provisioned to monitor and evaluate the service intuitively and graphically to visualize some analytics performed over the data (Baldominos et al. 2014). The visualization provides users with easy-to-understand feedback, allowing them to gain potentially relevant insights about the data and taking corrective measures. For example*,* key performance indicators (KPIs) such as mean absolute errors, mean square errors, or the run time are displayed graphically by the AIaaS and can thus be compared with the respective KPIs from another AI model to select the best possible configuration of hyper-parameters (Ribeiro et al. 2015). Although this might significantly improve a model's accuracy, it requires rich experience and is tedious and therefore contrasts the automated tuning of hyper-parameters.

AI infrastructure services commonly exhibit a customizable and extendable architecture that allows users to easily select and configure the infrastructure and integrate their own modules or third-party services into them. For example, users can use custom algorithms performing tasks, such as pre-processing and post-processing of data, and rely on third-party developer libraries, which are integrated as modules into the workflow of the AIaaS (Dorard et al. 2016; Elshawi et al. 2018). This is attractive for data scientists interested in using their own AI models, but do not want to concern themselves with all workflow tasks or the underlying infrastructure. Thus, they focus on developing and optimizing their algorithms, and the AIaaS handles the remaining part. An extendable architecture also enables the formation of large AI communities that focus on steadily extending AIaaS's functionalities. Likewise, several AIaaS are designed to connect to and work in harmony with major cloud-based seamlessly AI infrastructures, such as *C3 AI Suite* that enables developers to deploy their applications on *Microsoft Azure*, *Amazon*, *Google Cloud*, *Intel,* and *NVIDIA* infrastructure services or use pre-built connectors to access cloud and on-premise data sources. Existing AI infrastructures in turn can be customized depending on users' needs, for example, regarding users' latency, scalability, performance, or security requirements.

3.4 Inheriting Cloud Characteristics

With AIaaS being a cloud service, it inherits the strengths and typical **cloud characteristics** that have transformed cloud services into a critical information infrastructure for our everyday life, including (1) on-demand self-service access to (2) virtualized, shared, and managed IT resources that are (3) scalable on-demand, (4) available over a network, and (5) priced on a pay-per-use basis (Mell and Grance 2011). These characteristics have already rendered cloud computing an attractive alternative to traditional information technologies for organizations in diverse industries (i.e., healthcare (Gao et al. 2018)) while, nevertheless, challenging contemporary security and privacy risk-assessment approaches (Benlian et al. 2018; Lins et al. 2018). For example, a multi-tenant and virtualized approach seems promising from a cloud provider's perspective in terms of profit but increases the risk of co-location attacks due to inappropriate logical and virtual isolation.

On-demand self-service. A cloud user can typically provision cloud capabilities, such as additional storage for training data or further users of an AI application, as needed automatically and unilaterally without requiring human interaction with each AIaaS provider (Mell and Grance 2011). In the case of AI software services, for example, this is reflected by the action of the user sending a request to the AI software service, dynamically creating an instance on-demand that is used for, for instance, querying the addressed AI algorithm and responding with the result (Arnaldo et al. 2015). Potential users can even test AIaaS easily by using trial subscriptions in advance (Pandl et al. 2021). For example, *Microsoft Azure* gives potential users a trial to test a conversational question-and-answer bot build on their existing content for three days.

Virtualized, shared, and managed IT resources (resource pooling). Cloud service resources are commonly pooled to serve multiple users using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to user demand (Mell and Grance 2011). Inheriting this multi-tenancy capability, AIaaS can perform computations in parallel and enables thousands of users to continuously and concurrently access the services (Lu and Sakuma 2018). This is particularly advantageous regarding parameter configuration and classifier selection because data scientists typically experiment with parameters and classifiers to identify the best performing setting. Executing these experiments in parallel and comparing the results of differently configured models can thus significantly decrease the time required before a model can be deployed. Also, training data and configured AI models can be easier shared across different users if needed, reducing redundancies and fostering general AI model availability. Easier sharing and pre-trained models also provide the foundations for transfer learning, referring to a method in which a model and associated data developed for a particular task are used as a building block to solve a different problem (Samreen et al. 2020).

Scalability. The most dominant advantage is scalability because AIaaS providers can elastically provision and release hardware resources available to the platform and thus scale horizontally in accordance with the user-defined configurations and requirements if the consumption of computing resources for the defined AI model has increased (Boag et al. 2018; Elshawi et al. 2018; Pandl et al. 2021). The scalability of the cloud, combined with the number of available hardware resources, results in a large amount of processing power provisioned by the cloud and enables the AIaaS to respond to extensive requests with scalable and responsive utilization of CPUs and GPUs (Bao et al. 2018). Since AI algorithms are based on the knowledge inferred from a substantial quantity of data, the processing is performed by allocating significant computational resources that require the cloud's capability (Rouhani et al. 2018). Scalability is also beneficial because when using AI, organizations' hardware requirements change frequently and quickly. For example, the training of machine learning models can require powerful GPU resources for a certain period of time (e.g., weeks), while the hardware requirements for the inference of machine learning models are typically much less. However, they can also strongly vary with a varying load of inference requests. With cloud-based AIaaS, organizations can share hardware resources using the same cloud environment, thus, utilizing the hardware resources more efficiently (Shaukat et al. 2018).

Broad network access. Cloud capabilities are typically available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (Mell and Grance 2011). AIaaS are mostly offered through an API or graphical user interface (GUI). Standardized service APIs enable users to integrate the services into existing products using various programming languages (Xu et al. 2015). For example, a user requesting an inference for a particular data record would send that data to the API in a format accepted by the interface (e.g., JavaScript Object Notation format). The AI software service would then perform an inference task based on the received data record using the AI model and send the prediction back to the user, who can further process the result in her/his program. Nevertheless, most AI software service providers also offer a GUI to select, tune, and deploy appropriate machine learning algorithms, thereby simplifying operation (Chan et al. 2013). Some providers go even further and offer services not necessitating any programming knowledge by offering userfriendly interfaces with simple drag and drop functionality (Elshawi et al. 2018) or the functionality of analyzing data based on spreadsheets, which users can process using simple web interfaces (Yoon and Kang 2017). In these cases, users do not integrate the functionality of AIaaS into their programs through an API but perform all interactions using the provider's website, while both input and output are uploaded or downloaded through the website.

Measured service. Cloud services automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts) (Mell and Grance 2011). Resource usage can be monitored, controlled, and reported, providing transparency for both providers and users. Such a measured service also enables '*pay-as-you-go*' pricing models, which are common in the context of AIaaS. For example, inference as a services charge their users per request, whereas MLaaS that allows users to train their models charge for their services on an hourly basis (Kaplunovich and Yesha 2017; Javadi et al. 2020). With such a pricing model, AIaaS offers disruptive potential against researching and developing its own AI applications. No upfront costs are incurred for hardware because the user is neither responsible for the procurement, nor for the ongoing operation and maintenance (Bhattacharjee et al. 2017; Boag et al. 2018). AIaaS providers offer their services at a low cost due to economics of scale, compared to the cost of acquiring an equally powerful in-house server (Shaukat et al. 2018; Zapadka et al. 2020), hence increasing the attractiveness for users to rely on AIaaS. Some AI algorithms require ongoing research and maintenance to be state-of-the-art, retain a representative underlying dataset for current application scenarios, and ensure good performance. For this purpose, AI experts must be continuously assigned to maintain these algorithms, which is the provider's obligation and saves costs for the user. Especially for SMEs, this is convenient and economically reasonable because hiring AI experts is cost-intensive and challenging because the market currently lacks well-trained AI experts. It is hardly possible for organizations to estimate the costs based on existing KPIs or prior experiences in traditional AI projects. This is counteracted by the '*pay-as-you-go*' pricing model of AIaaS because organizations can calculate the costs of short- and long-term usage based on transparent payment structures. Therefore, users are not confronted with unexpected costs and can adapt their resource utilization to their budget as opposed to own solutions where costs for maintenance must be paid regardless of whether the hardware is used. Offering AIaaS trial subscriptions benefits users, especially SMEs with no AI expertise, because organizations can test in a short test period whether the offered services are useful for their use cases and which business advantages are associated with them.

4 Open Socio-technical Challenges and Future Research Directions for the BISE Community

While the emergence of AIaaS offers manifold opportunities, AIaaS on the one hand inherits a variety of issues and challenges relating to AI in general, such as the presence of racial bias in a widely used AI in the health care industry (Obermeyer et al. 2019). On the other hand, AIaaS also possesses common cloud computing risks, including users' lack of control and security concerns (Weinhardt et al. 2009; Trenz et al. 2019). Yet, AIaaS also intensifies these issues, such as AIaaS being perceived as a black-box, thereby further decreasing accountability, trustworthiness, and explainability of offered AI services (Javadi et al. 2020; Pandl et al. 2021). AIaaS also leads to various novel socio-technical challenges and issues that may severely impede its value contributions if not handled appropriately by the BISE community. To maximize the benefits of AIaaS while at the same time mitigating or even preventing its risks, AIaaS should fulfill the manifold guidelines of **Trustworthy AI (TAI)**, for example, issued by the European Union and the Independent High-Level Expert Group (HLEG) on Artificial Intelligence of the European Commission (European Commission 2019). AI users (e.g., individuals, organizations, society) perceive AI as trustworthy "when it is developed, deployed, and used in ways that not only ensure its compliance with all relevant laws and its robustness but especially its adherence to general ethical principles" (Thiebes et al. 2020). AIaaS needs to fulfill several requirements by applying technical and non-technical means to be perceived as trustworthy. For example, the HLEG proposes seven key requirements that AI systems should meet in order to be trustworthy: (#1) support human agency and oversight, (#2) be technically robust and safe, (#3) provide privacy and data governance, (#4) be transparent, (#5) support diversity, non-discrimination, and fairness, as well as (#6) societal and environmental well-being, and (#7) provide accountability (European Commission 2019). While the TAI requirements apply to AI in general, fulfilling them gains high importance in the context of AIaaS because the service operations are not under the control of or transparent for users. Future research is required that provides best practices for each TAI requirement in the context of AIaaS. In the following, we will briefly outline four example requirements and the need for future research on AIaaS.

First, the requirement' *support of human agency and oversight'* (#1) requires that AIaaS empowers users to make informed decisions and fosters their fundamental rights, particularly when overseeing and controlling the AIaaS. However, AIaaS providers face a **trade-off between user control and ease of use** (Yao et al. 2017). In the case of pure black-box inference as a service, for example, that lacks customizability and oversight to achieve higher ease of use, the users know only about the input and output formats of the respective API, but the model and the dataset on which the model is trained remains private to the provider (Truex et al. 2019). Users thus are unable to determine whether the training dataset on which the queried model is based represents their data in a meaningful manner. Furthermore, they have no control over the classifier and the hyper-parameters used and thus may not have the ability to adapt the model to their specific requirements. However, higher customizability and system control require higher domain knowledge and might thus not be suitable for every user. Future research, therefore, 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.

Second, following the requirement' *technical robustness and safety'* (#2), AIaaS needs to be resilient and secure, ensuring a fallback plan in case something goes wrong, as well as being accurate, reliable, and reproducible. While AIaaS is generally perceived as being more resilient than in-house AI applications, the history of cloud computing has shown that even the dominant cloud providers may fail in providing reliable services. In addition, more and more start-ups are entering the market offering innovative AI services to SMEs but may lack technical means to ensure high degrees of security and reliability. Finally, AIaaS is characterized by entangled supply-chains because it is operated in an interdependent ecosystem of providers, complementors, and other stakeholders, bearing the risk of cascading and escalating failures (Fig. 3).

To foster technical robustness and mitigate the adverse effects of service failures, the **interoperability of AIaaS** is required. Interoperable AIaaS should make it possible to securely and efficiently move data in, out, and among AIaaS providers and allow to port applications from one AI service to another. The interoperability of AIaaS, therefore, not only enables the integration of different AIaaS to unleash its full potential but also prevents vendor lock-in effects, allowing users to easily switch services, such as swiftly switching an AI infrastructure in case of outages. However, the question of how to achieve cloud interoperability (effectively) still remains unanswered. Initial best practices and standards for cloud interoperability have been proposed recently, such as the *Open Virtualization Framework*, the *Cloud Infrastructure Management Interface*, *SWIPO* (Switching Cloud Providers and Porting Data), or the standard *ISO/IEC 19,941:2017*. Promising initiatives like the European project *GAIA-X*, which aims to establish a federated data infrastructure by integrating cloud and edge services and required data centers across Europe, might boost AIaaS diffusion and provide the foundation for seamless integration of AIaaS and exchange of data between providers and users. Future research is required to understand not only technological means to integrate AIaaS and data sources but also organizational governance structures for an ecosystem fostering mutual and trustworthy exchanges, thereby achieving TAI requirements and creating a flourishing ecosystem.

Third, in line with the requirement *'provide privacy and data governance'* (#3), researchers and practitioners demand that AIaaS must **implement adequate data governance and protection mechanisms** in order to prevent invasion of individuals' privacy when collecting and generating data about them and to allow users to understand the consequences of data disclosure better. For example, the AIaaS might leak information about its training data (Tramèr et al. 2016). By querying a pre-trained model in a purposeful way, an adversary may determine whether a given data record was part of the model's training data, called a membership inference attack (Truex et al. 2019). For example, this concerns a scenario of a black-box inference as a service that was trained with large amounts of a cancer treatment center's patient records and that predicts cancer-related health outcomes when given an individual's health information as input (Truex et al. 2019). An adversary could then provide health information of another individual and, based on the model's output, try to infer whether this individual was a patient at the treatment center. Such a membership inference attack would raise concerns about patients' privacy, as their health information would be made publicly available through the publication of the trained model in the form of AIaaS.

Extant research has started to propose several approaches to protect people's privacy during the training and operation of an AI in the cloud, such as training AI models using encrypted data, making encrypted predictions, as well as returning the predictions in an encrypted form (e.g., Hesamifard et al. 2017). Further emerging research aims to provide AIaaS based not on a centralized cloud computing platform, but on trusted hardware-enabled, scalable distributed ledger technology (Pandl et al. 2020), potentially increasing robustness and trustworthiness. Potential benefits include a resilient system with high uptime and a transparent and comprehensible system architecture. Finally, third-party attestations and related certifications are promising means to assess whether an AIaaS has implemented adequate data governance and protection mechanisms (Lins et al. 2018). However, it remains unclear whether these approaches are suitable for practice, and

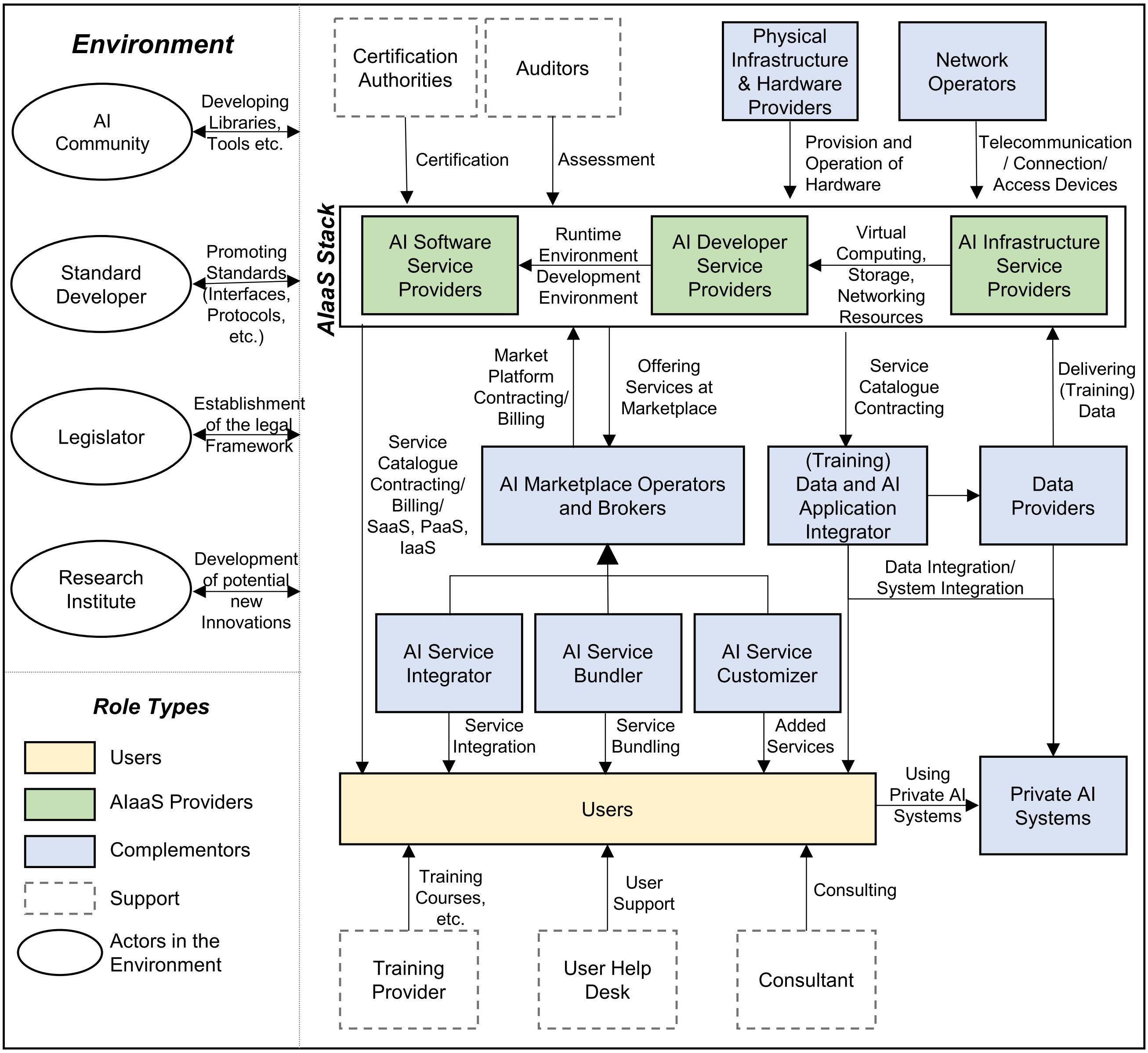
Environment
AI Community
Developing Libraries, Tools etc.
Standard Developer
Promoting Standards (Interfaces, Protocols, etc.)
Legislator
Establishment of the legal Framework
Research Institute
Development of potential new Innovations
Role Types
Users
AIaaS Providers
Complementors
Support
Actors in the Environment
Certification Authorities
Auditors
Physical Infrastructure & Hardware Providers
Network Operators
Certification
Assessment
Provision and Operation of Hardware
Telecommunication / Connection/ Access Devices
AIaaS Stack
AI Software Service Providers
Runtime Environment Development Environment
AI Developer Service Providers
Virtual Computing, Storage, Networking Resources
AI Infrastructure Service Providers
Market Platform Contracting/ Billing
Offering Services at Marketplace
Service Catalogue Contracting
Delivering (Training) Data
Service Catalogue Contracting/ Billing/ SaaS, PaaS, IaaS
AI Marketplace Operators and Brokers
(Training) Data and AI Application Integrator
Data Providers
Data Integration/ System Integration
AI Service Integrator
AI Service Bundler
AI Service Customizer
Service Integration
Service Bundling
Added Services
Users
Using Private AI Systems
Private AI Systems
Training Courses, etc.
User Support
Consulting
Training Provider
User Help Desk
Consultant

Fig. 3 AIaaS ecosystem comprising various stakeholders (adapted from Floerecke et al. 2020)

whether these will impact users' trust perceptions towards AIaaS, ultimately requiring further research.

Finally, AIaaS needs to fulfill the requirement' *support diversity, non-discrimination and fairness'* (#5) to avoid unfair bias since this could have multiple negative implications, ranging from the marginalization of vulnerable groups to the exacerbation of prejudice and discrimination (Feuerriegel et al. 2020). Concerning AIaaS, prior research has already started to discuss the disadvantages and fairness risks of inferences based on pre-trained models or models transferred via transfer learning. While AIaaS providers aim to serve the broadest possible range of users, they are forced to make static design-time decisions based on generic user needs (Halpern et al. 2019). As the most decisive factor for general applicability is the model's underlying dataset, providers aim to conceptualize a set of generic training data. The associated challenge is to correctly categorize new user data that differ from those used for training (Bishop 2006) and is known as generalization in the domain of machine learning. In practice, there is no one-model-fits-all solution, thus, there is no single model or algorithm that can handle all dataset varieties (Elshawi et al. 2018), and therefore, using a dataset with general data potentially leads to low prediction accuracy and discrimination (Wang et al. 2018). An illustrative example of the generalization challenge and resulting discrimination is an AIaaS offering a pre-trained model to predict a person's weight, which receives characteristics such as age, gender, and height as input. The problem is that people from North America may have a significantly different distribution of body weight to the mentioned characteristics than, for instance, people in Asia, and hence the training data could be unrepresentative, which will not only adversely affect the models' ability to handle unseen test data but may also lead to biases (Chung et al. 2018). Consequently, AIaaS providers enter **a trade-off between accuracy and fairness vs. generalizability** (Halpern et al. 2019). For users, this means they knowingly have to accept non-optimal results when querying these generalized models. In addition, AIaaS providers may not provide sufficient information to users about the training data and assumptions made to prevent discrimination and related biases (e.g., that training data stem from another culture). Future research is required to ultimately create a balance between accuracy and generalizability and ensure model diversity and fairness, particularly in the case of pre-trained and transferred models.

5 Conclusion

Organizations do not have to decide between adopting or not adopting AI but between adopting it now or deferring that decision. The critical question of how to implement and use AI currently overrides any of the promised benefits that this technology offers (Phillips 2018). The latest discussions emphasize that AIaaS could be a promising alternative for organizations dealing with the difficulty of adopting in-house AI because it overcomes major adoption barriers. As more and more providers offer AIaaS, more organizations from every industry will be able to find solutions that fit their specific use-cases, making AI adoption more global and AIaaS even more compelling. Besides inheriting valuable cloud characteristics (i.e., on-demand provisioning, resource-pooling, and scalability), AIaaS comes with unique and innovative features, such as complexity abstraction and pre-trained and customizable AI models, thus enabling companies to achieve AI's full potential. Given these benefits and growing external market pressures (Zapadka et al. 2020), organizations are likely to adopt AIaaS in the future frequently (i.e., it is expected that the AIaaS market will grow by more than 42% in 2020 (Infiniti Research Ltd 2020)).

With this catchword article, we aim to provide a foundation for future discussions by proposing an AIaaS definition and three-layered service stack, highlighting important characteristics of AIaaS, and revealing further research directions to motivate researchers to engage with AIaaS. While computer scientists are strongly driving prevalent research on AIaaS, we call for more interdisciplinary research taking a socio-technical perspective on AIaaS to foster diffusion and application.

Acknowledgements The authors would like to thank Jakub Szefer, Yale University, for his fruitful comments and suggestions on earlier versions of this manuscript. The authors further gratefully acknowledge the feedback provided by Ulrich Frank, University of Duisburg-Essen, and the two anonymous reviewers that helped us to improve the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Arnaldo I, Veeramachaneni K, Song A, O'Reilly U (2015) Bring your own learner: a cloud-based, data-parallel commons for machine learning. IEEE Comput Intell Mag 10:20–32. https://doi.org/10.1109/MCI.2014.2369892

Baldominos A, Albacete E, Saez Y, Isasi P (2014) A scalable machine learning online service for big data real-time analysis. In: Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Big Data, pp 1–8

Baldominos A, Saez Y, Albacete E, Marrero I (2015) An efficient and scalable recommender system for the smart web. In: Proceedings of the 11th International Conference on Innovations in Information Technology, pp 296–301

Bao B, Xiang Y, Li Y, Lyu S, Munshi H, Zhu H (2018) Scalable cloud service for multimedia analysis based on deep learning. In: Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, pp 1–4

Benlian A, Kettinger WJ, Sunyaev A, Winkler TJ (2018) The transformative value of cloud computing: a decoupling, platformization, and recombination theoretical framework. J Manag Inf Syst 35:719–739. https://doi.org/10.1080/07421222.2018.1481634

Bhattacharjee B, Boag S, Doshi C, Dube P, Herta B, Ishakian V, Jayaram KR, Khalaf R, Krishna A, Li YB, Muthusamy V, Puri R, Ren Y, Rosenberg F, Seelam SR, Wang Y, Zhang JM, Zhang L (2017) IBM deep learning service. IBM J Res Dev 61:1–11. https://doi.org/10.1147/JRD.2017.2716578

Bishop CM (2006) Pattern recognition and machine learning. Information science and statistics. Springer, New York

Boag S, Dube P, El Maghraoui K, Herta B, Hummer W, Jayaram KR, Khalaf R, Muthusamy V, Kalantar M, Verma A (2018) Dependability in a multi-tenant multi-framework deep learning as-a-service platform. In: Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, pp 43–46

Bughin J, Seong J, Manyika J, Chiu M, Joshi R (2018) Notes from the AI frontier: modeling the impact of AI on the world economy. https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-modeling-the-impact-of-ai-on-the-world-economy. Accessed 13 July 2020

Chan S, Stone T, Szeto KP, Chan KH (2013) PredictionIO: a distributed machine learning server for practical software development. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp 2493–2496

Chui M, Malhotra S (2018) AI adoption advances, but foundational barriers remain. https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain. Accessed 13 July 2020

Chung Y, Haas PJ, Upfal E, Kraska T (2018) Unknown examples & machine learning model generalization. https://arxiv.org/pdf/1808.08294.pdf. Accessed 31 March 2020

European Commission (2019) Ethics guidelines for trustworthy AI. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai. Accessed 2 April 2020

Crawford K (2019) Halt the use of facial-recognition technology until it is regulated. Nature 572:565. https://doi.org/10.1038/d41586-019-02514-7

Dorard L, Reid MD, Martin FJ (2016) AzureML anatomy of a machine learning service. In: Proceedings of the 2nd International Conference on Predictive APIs and Apps, pp 1–13

Duong TNB, Sang NQ (2018) Distributed machine learning on IAAS clouds. In: Proceedings of the 5th IEEE International Conference on Cloud Computing and Intelligence Systems, pp 58–62

Dutta S (2018) An overview on the evolution and adoption of deep learning applications used in the industry. Wires Data Min Knowl Discov 8:1–12. https://doi.org/10.1002/widm.1257

Elshawi R, Sakr S, Talia D, Trunfio P (2018) Big data systems meet machine learning challenges: towards big data science as a service. Big Data Res 14:1–11. https://doi.org/10.1016/j.bdr.2018.04.004

Feuerriegel S, Dolata M, Schwabe G (2020) Fair AI. Bus Inf. Syst Eng 62:379–384. https://doi.org/10.1007/s12599-020-00650-3

Floerecke S, Lehner F, Schweikl S (2020) Cloud computing ecosystem model: evaluation and role clusters. Electron Mark. https://doi.org/10.1007/s12525-020-00419-2

Gao F, Thiebes S, Sunyaev A (2018) Rethinking the meaning of cloud computing for health care: a taxonomic perspective and future research directions. J Med Internet Res 20:e10041. https://doi.org/10.2196/10041

Haenlein M, Kaplan A (2019) A brief history of artificial intelligence: on the past, present, and future of artificial intelligence. Calif Manag Rev 61:5–14. https://doi.org/10.1177/0008125619864925

Halpern M, Boroudjerdian B, Mummert T, Duesterwald E, Reddi VJ (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: Proceedings of the 19th International Symposium on Performance Analysis of Systems and Software, pp 1–14

Hazelwood K, Bird S, Brooks D, Chintala S, Diril U, Dzhulgakov D, Fawzy M, Jia B, Jia Y, Kalro A, Law J, Lee K, Lu J, Noordhuis P, Smelyanskiy M, Xiong L, Wang X (2018) Applied machine learning at facebook: a datacenter infrastructure perspective. In: Proceedings of the 2018 IEEE International Symposium on High Performance Computer Architecture, pp 620–629

Hesamifard E, Takabi H, Ghasemi M, Jones C (2017) Privacy-preserving machine learning in cloud. In: Proceedings of the Workshop on Cloud Computing Security, ACM, pp 39–43

Huqqani AA, Schikuta E, Mann E (2014) Parallelized neural networks as a service. In: Proceedings of the 2014 International Joint Conference on Neural Networks, pp 2282–2289

Infiniti Research Ltd (2020) Artificial intelligence-as-a-service (aiaas) market by end-user and geography - forecast and analysis 2020–2024. https://www.technavio.com/report/artificial-intelligence-as-a-service-market-industry-analysis. Accessed 13 July 2020

Javadi SA, Cloete R, Cobbe J, Lee MSA, 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

Kaplunovich A, Yesha Y (2017) Cloud big data decision support system for machine learning on AWS: analytics of analytics. In: Proceedings of the 2017 IEEE International Conference on Big Data, pp 3508–3516

Lins S, Schneider S, Sunyaev A (2018) Trust is good, control is better: creating secure clouds by continuous auditing. IEEE Trans Cloud Comput 6:890–903. https://doi.org/10.1109/TCC.2016.2522411

Liu F, Tong J, Mao J, Bohn R, Messina J, Badger L, Leaf D (2011) NIST cloud computing reference architecture. National Institute of Standards and Technology. Gaithersburg, MD, USA

Lu W, Sakuma J (2018) More practical privacy-preserving machine learning as a service via efficient secure matrix multiplication. In: Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp 25–36

Madani A, Arnaout R, Mofrad M, Arnaout R (2018) Fast and accurate view classification of echocardiograms using deep learning. NPJ Digit Med. https://doi.org/10.1038/s41746-017-0013-1

Mell PM, Grance T (2011) The NIST definition of cloud computing. National Institute of Standards and Technology. Gaithersburg, MD, USA

Naous D, Schwarz J, Legner C (2017) Analytics as a service: cloud computing and the transformation of business analytics business models and ecosystems. In: Proceedings of the 25th European Conference on Information Systems, pp 1–16

Obermeyer Z, Powers B, Vogeli C, Mullainathan S (2019) Dissecting racial bias in an algorithm used to manage the health of populations. Science 366:447–453. https://doi.org/10.1126/science.aax2342

Pandl KD, Thiebes S, Schmidt-Kraepelin M, Sunyaev A (2020) On the convergence of artificial intelligence and distributed ledger technology: a scoping review and future research agenda. IEEE Access 8:57075–57095. https://doi.org/10.1109/ACCESS.2020.2981447

Pandl KD, 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, pp 1–10

Phillips JM (2018) Integrating machine learning in law: a precis of best practices for initial law firm adoption. J Bus Entepreneurship Law 11:321–328

Rai A, Constantinides P, Sarker S (2019) Next-generation digital platforms: toward human-AI hybrids. MIS Q 43:iii–ix

Ramesh R (2017) Predictive Analytics for banking user data using AWS machine learning cloud service. In: Proceedings of the 2017 2nd International Conference on Computing and Communications Technologies, pp 210–215

Ransbotham S, Khodabandeh S, Fehling R, LaFountain B, Kiron D (2019) Winning with AI. MIT Sloan Manag. Rev. 61180

Reif M, Shafait F, Goldstein M, Breuel T, Dengel A (2014) Automatic classifier selection for non-experts. Pattern Anal Appl 17:83–96. https://doi.org/10.1007/s10044-012-0280-z

Ribeiro M, Grolinger K, Capretz MAM (2015) MLaaS: machine learning as a service. In: Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp 896–902

Romero F, Li Q, Yadwadkar NJ, Kozyrakis C (2019) INFaaS: managed & model-less inference serving. https://arxiv.org/pdf/1905.13348.pdf. Accessed 31 March 2020

Rouhani BD, Hussain SU, Lauter K, Koushanfar F (2018) ReDCrypt: real-time privacy-preserving deep learning inference in clouds using FPGAs. ACM Trans Reconfigurable Technol Syst 11:1–21. https://doi.org/10.1145/3242899

Samreen F, Blair G, Elkhatib Y (2020) Transferable knowledge for low-cost decision making in cloud environments. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2020.2989381

Shaukat Z, Fang J, Azeem M, Akhtar F, Ali S (2018) Cloud based face recognition for Google Glass. In: Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, pp 104–111

Thiebes S, Lins S, Sunyaev A (2020) Trustworthy artificial intelligence. Electron Mark. https://doi.org/10.1007/s12525-020-00441-4

Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T (2016) Stealing machine learning models via prediction APIs. In: Proceedings of the 25th USENIX Security Symposium, pp 601–618

Trenz M, Huntgeburth J, Veit D (2019) How to succeed with cloud services? Bus Inf Syst Eng 61:181–194. https://doi.org/10.1007/s12599-017-0494-0

Truex S, Liu L, Gursoy ME, Yu L, Wei W (2019) Demystifying membership inference attacks in machine learning as a service. IEEE Trans Serv Comput Forthcom. https://doi.org/10.1109/TSC.2019.2897554

Wang W, Gao J, Zhang M, Wang S, Chen G, Ng TK, Ooi BC, Shao J, Reyad M (2018) Rafiki: machine learning as an analytics service system. Proc VLDB Endow 12:128–140. https://doi.org/10.14778/3282495.3282499

Weinhardt C, Anandasivam A, Blau B, Borissov N, Meinl T, Michalk W, Stößer J (2009) Cloud computing – a classification, business models, and research directions. Bus Inf Syst Eng 1:391–399. https://doi.org/10.1007/s12599-009-0071-2

Xu D, Wu D, Xu X, Zhu L, Bass L (2015) Making real time data analytics available as a service. In: Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, pp 1–10

Yao Y, Xiao Z, Wang B, Viswanath B, Zheng H, Zhao BY (2017) Complexity vs. performance: empirical analysis of machine learning as a service. In: Proceedings of the 2017 Internet Measurement Conference, pp 384–397

Yoon C, Kang S (2017) A study on machine learning web service using spreadsheets. In: Proceedings of the 2017 International Conference on Information and Communication Technology Convergence, pp 760–765

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

Zhang W, Feng M, Zheng Y, Ren Y, Wang Y, Liu J, Liu P, Xiang B, Zhang L, Zhou B, Wang F (2017) GaDei: on scale-up training as a service for deep learning. In: Proceedings of the 2017 IEEE International Conference on Data Mining, pp 1195–1200