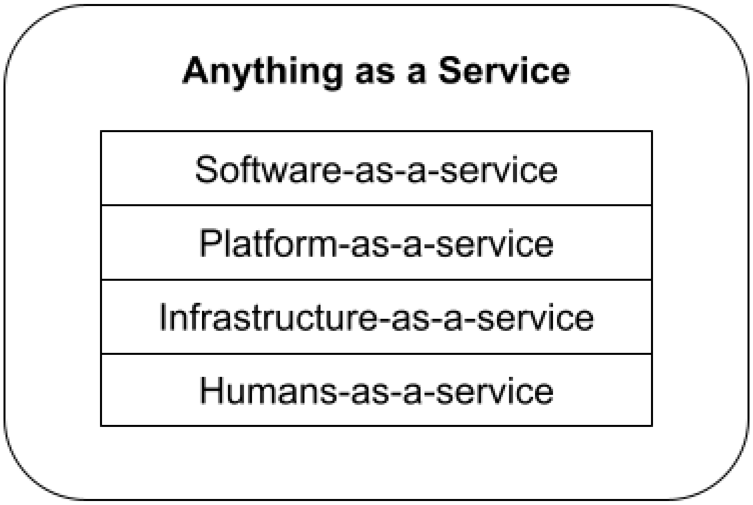
Anything As a Service



Overview

Anything as a Service (XaaS) refers to a wide range of products and tools that can be purchased through a "as a service" consumption model. It is very useful for enabling easy ML deployments at scale and in this context we will be talking about TinyML as a Service. But first, we will have to learn about Anything as a Service. To talk about the anything-as-a-service (XaaS) paradigm, we first have to become familiar with the concept of cloud computing.

Cloud computing refers to the use of externally hosted computational resources. Common platforms for this include the Google Cloud Platform, Microsoft Azure, and Amazon Web Services. Things that can be hosted on the cloud are applications (e.g., Dropbox, Slack), development environments (e.g., Amazon Sagemaker, Azure ML Studio), and even entire virtual machines (e.g., hosted websites, network devices). Cloud computing is an incredibly powerful idea, especially in the business world, as it reduces the need to have on-site computational resources. Essentially, this eliminates the need to spend capital on expensive servers and networking infrastructure, as they can instead be managed externally by a cloud provider. This essentially transfers capital expenses to operational expenses, making infrastructure management much cheaper. In the modern-day, most of our favorite websites are hosted by one of these cloud providers (e.g., Facebook, Netflix, LinkedIn, and Twitch are all hosted on Amazon Web Services).

Anything-as-a-service

Anything-as-a-service (XaaS) is a relatively new computing paradigm that refers to hosted cloud computing resources that are *externally accessible to a customer*. If you have ever used Google Docs, watched a video on Netflix, or hosted a website on Amazon Web Services, you have interacted with some form of XaaS. Although this concept may sound intimidating initially, the service essentially removes any complication to the user by providing them with the correct level of computational abstraction to perform the service they require. There are three common forms of XaaS that we will briefly describe.

Software-as-a-service (SaaS) is conceptually the simplest, and most common, manifestation of the as-a-service paradigm. In SaaS, the entire computing ecosystem is abstracted at the level of the application, with which the user is able to interact. The rest of the virtual machine is hosted externally through cloud resources, which are not visible to the user. This is why an internet connection is needed to use these services, which may only intermittently interact with the cloud resources (as in data storage and text editing applications), or maybe continuously streamed (as in video streaming applications). SaaS is great for many end-user applications, but provides relatively little flexibility, meaning it is not useful in developmental environments. This is where the other XaaS types come into play.

Platform-as-a-service (PaaS) refers to a set of resources that are contained within a virtual machine. For example, a suite of coding resources hosted on a virtual machine might be classed as PaaS. Naturally, this is harder to use than a single hosted application but offers more flexibility to the end-user, and is thus more useful for certain developmental applications (e.g., video editing or video game design).

Infrastructure-as-a-service (IaaS) is the lowest-level abstraction of XaaS, which is the level of the virtual machine. This is what most companies use to host their resources, as it offers maximal flexibility since the entire system (apart from the hardware itself) can be managed externally by the end-user. This is ideal for hosting websites, and even for an individual to host their own PaaS or SaaS applications on top of the infrastructure.

Humans-as-a-service (HaaS) is an emerging concept that is quite relevant to machine learning systems. In the vast majority of cases, we need humans to label the ground truth datasets for machine learning. These “human” services can be outsourced to the cloud through HaaS using existing platforms such as Amazon Mechanical Turk (AMT) and ClickWorker. Thanks to AMT we can readily tap into a wide range of people that would be willing to perform tasks that our computer systems cannot (yet) perform as efficiently as humans.

Of course, there are pros and cons to using the HaaS approach in practice that one should be aware of being considering it as a resource. For example, AMT offers a low rate of pay, and provides no incentives for providing high-quality responses, which may lead to an overall lower-quality of work. Additionally, services that require some form of specialization (such as AMT workers pretending to be a judge and deciding whether bail is granted in a particular legal case, testing the effectiveness of an algorithm-in-the-loop analysis) may not provide representative results of the actual implementation. Despite this, HaaS has already proven to be a highly useful resource for many datasets, such as labeling larges swatches of images, text, and audio data.

Conclusion

Why is XaaS important in the context of TinyML? Well, because we are now beginning to see applications that are described as **machine-learning-as-a-service (MLaaS)**, which leverage cloud resources to perform machine learning tasks. For example, the Google Colaboratory, which we are all familiar with from this and previous courses, can be described as MLaaS. Naturally, this can be extended to TinyMLaaS in the context of TinyML, which depending on the context could mean several things. The remainder of this section will discuss the concepts of MLaaS and its extension to TinyMLaaS, as well as the unique challenges that come with them.