

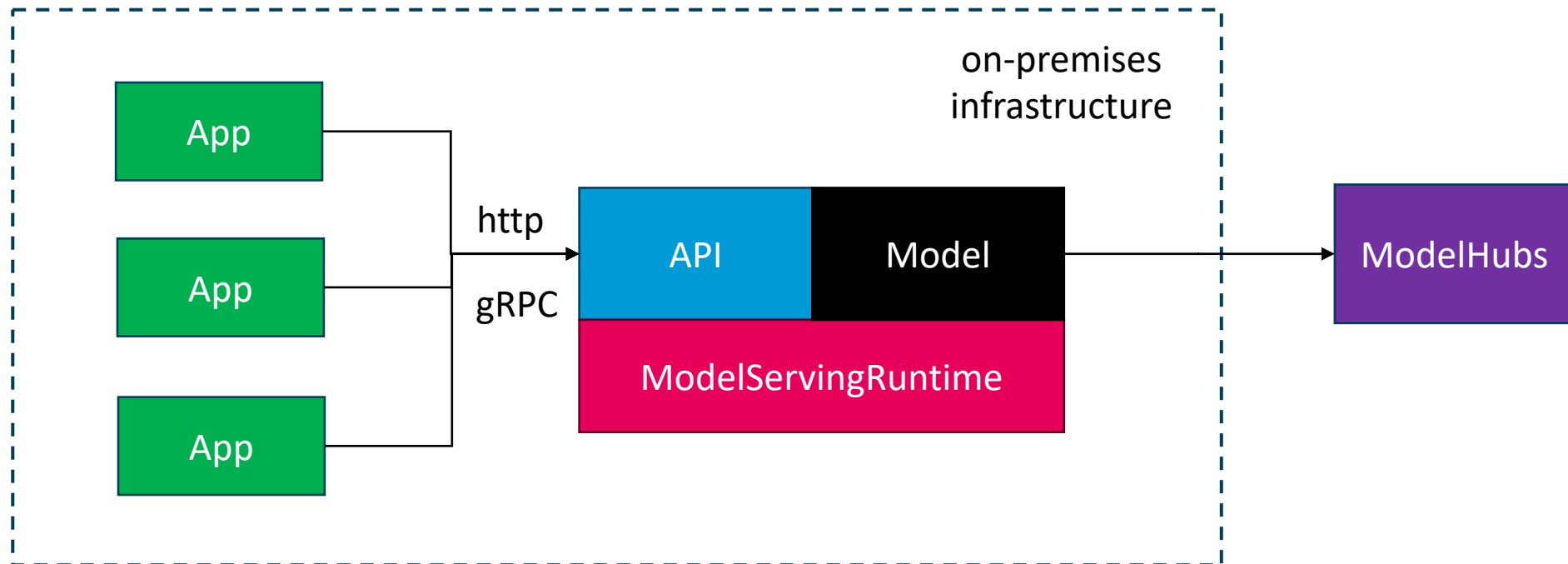


Running Open-Source Machine-Learning Models On-Premises

Malte Groth

This talk in one diagram

“Providing access to Open-Source Machine-learning Models via self-hosted Web-Services for Inference”



What does it have to do with medicine?

This Setup is necessary for many applications in medicine dealing with AI for several reason.

In medicine, there is need for

- **Data Security**, e.g. because of sensitive data
- **Transparency & Reliability**, e.g. for science and traceable services
- **Autonomy & Availability**, e.g. for not being dependent on internet providers
- **Adaptability**, e.g. in terms of fine-tuning
- ...



Running Open-Source Machine-Learning Models On-Premises

Open-Source Machine-Learning Models

... are models available under an Open-Source License (e.g. Apache 2.0)

Sources: HuggingFace, PyTorchHub, Github



Open-Source models differ in terms of transparency, reproducibility and quality control

→ Do not confuse Open-Source with Openness

Advantages of Open-Source Machine-Learning Models

- Transparency and Reliability
- Availability
- Adaptability
- Performance
- Autonomy (avoiding Vendor-Lock-In)
- Cost-Saving



Running Open-Source Machine-Learning Models **On-Premises**

On-Premises

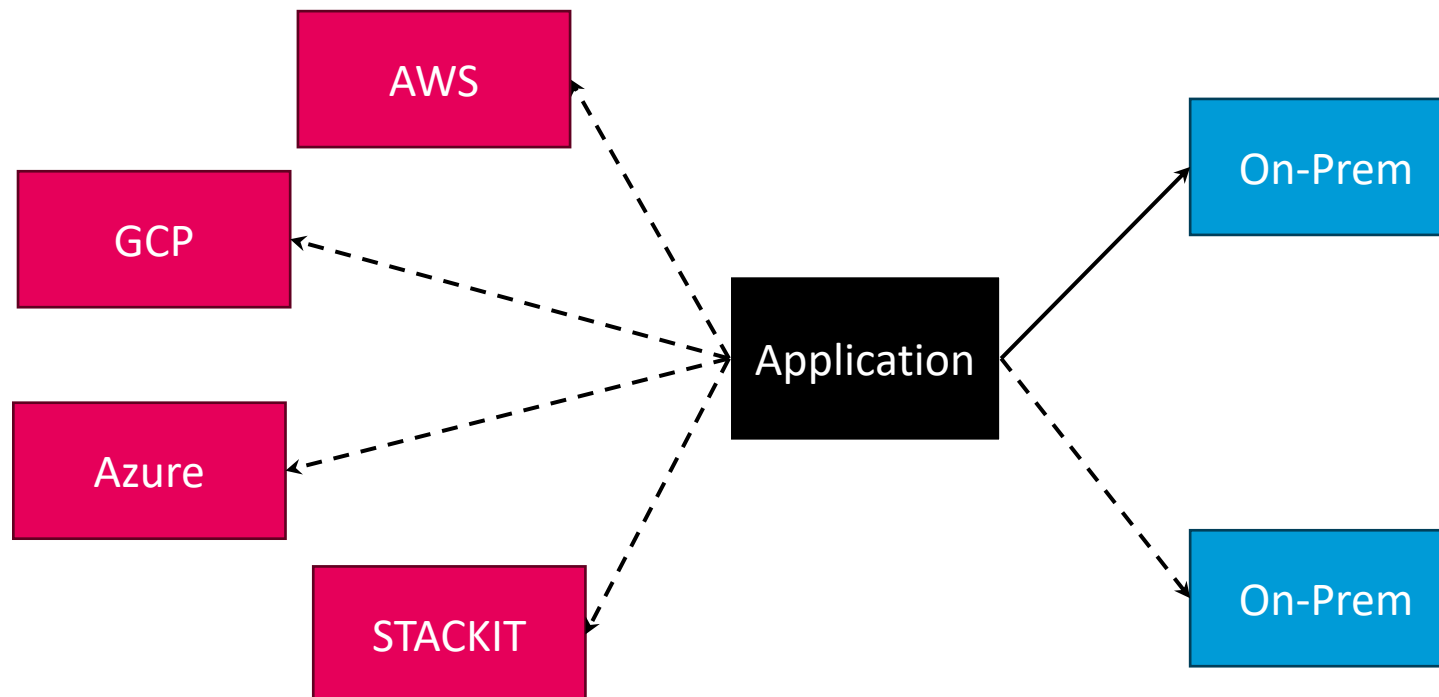
- Deployments and operations are located within physical premises of a company/organization
- Opposite of the cloud
- Full control over IT infrastructure, data and applications

Motivation for running Models on-premises

- Data Security and Privacy
- Latency and Performance
- Cost-Saving
- Offline Access
- Control over intellectual property
- Flexibility → especially true if following a ...

Cloud-agnostic deployment strategy

Designing your Applications, tools and services in a way so they can migrate seamlessly between multiple cloud platforms and on-premises.



Kubernetes

... is a key technology for implementing a cloud-agnostic deployment strategy because it

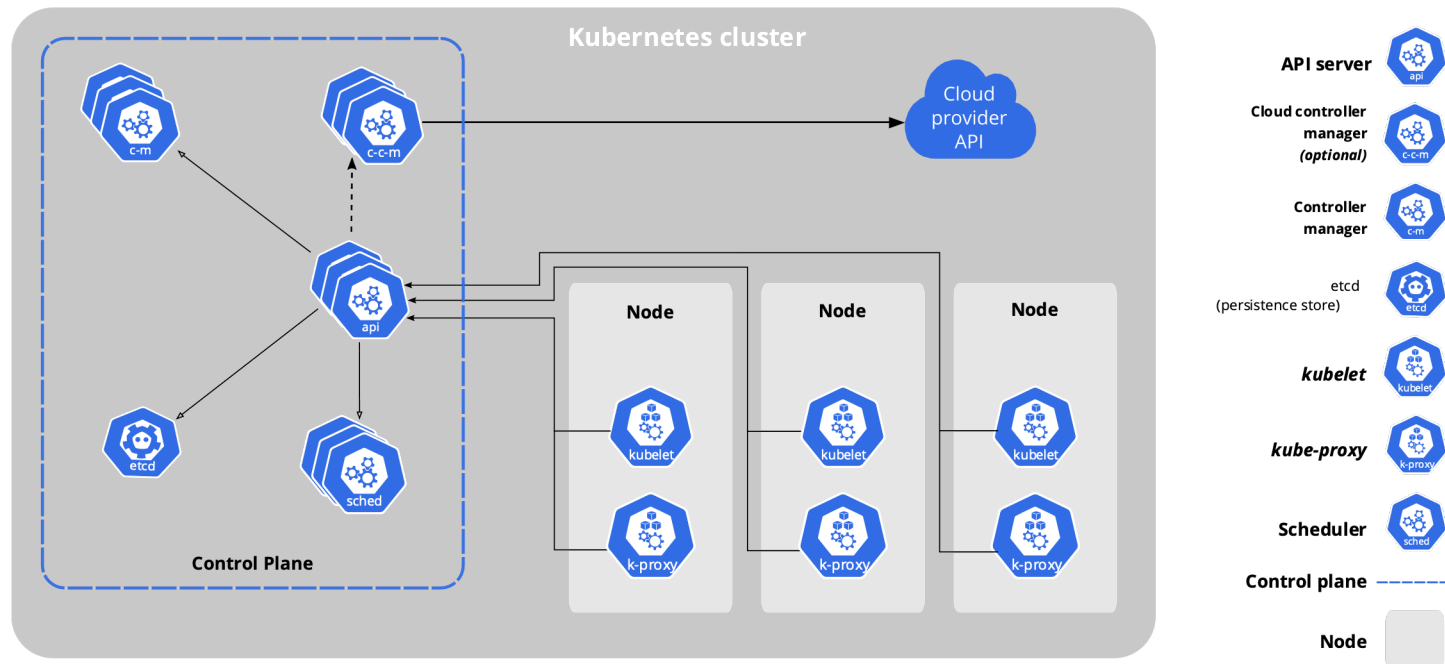
- **abstracts** the underlying infrastructure
- has a **consistent** tooling and user interface
- provides **scaling at ease**


"Kubernetes (k8s) is a ...

- **open-source**
- **container** orchestration system
- for **automating** software deployment, **scaling** and management."

Kubernetes

Kubernetes Cluster simplified and from a user perspective:
web service that can be used to manage and automate processes on many computers





Running Open-Source Machine-Learning Models On-Premises **with Kubernetes**

KServe: Model Inference Platform on Kubernetes

KServe

- offers (auto)scaling, e.g.
 - if number of requests increases
 - if there is no load at all (scale to zero)
- **standardized inference protocol** across ML frameworks and Model Serving Runtimes
- **simplifies** model deployment



Running Open-Source Machine-Learning Models On-Premises **In Practice**

Demo: Steps to run a Model for ImageClassification from torch.hub on Kubernetes

Setup:

- local Kubernetes-Cluster based on Minikube
- KServe is already installed
- Model for Image Recognition: resnet18 from torch.hub
- Model files were already downloaded

Steps:

- Preparing the model files: create a MAR file so that torchserve can work with the model
- Uploading the MAR file
- Creating an InferenceService with references to the location of the model and the model serving runtime to be used
- Preparing images and classifying them via HTTP requests

Summary

- Open-Source Machine-Learning Models have a lot of advantages.
- There are many reasons why it may be appropriate or even necessary to deploy models on-premise for a medical use case.
- You can gain high flexibility by choosing a cloud-agnostic approach.
- Kubernetes is a key technology for implementing a cloud-agnostic deployment strategy.
- KServe is a inference platform suited for deploying models in production on Kubernetes.

A large, abstract graphic on the left side of the slide. It features a complex network of glowing blue lines and dots, resembling a molecular structure or a digital network, set against a dark blue background. The graphic is partially obscured by a white diagonal shape that separates it from the main text area.

**Thank you for your
attention**

Why Kubernetes?

- High Availability:
 - K8s clusters consists of multiple nodes
 - K8s controllers enable service (pod) replication
 - K8s controllers provide self-healing mechanisms
- Resource Efficiency:
 - K8s comes with a powerful scheduling

Link: <https://dzone.com/articles/kubernetes-advantages-and-disadvantages>

InferenceService

InferenceServices

- provide Inference APIs out-of-the-box
- support multiple ML frameworks/Model Serving Runtimes
- support for obtaining models from different storage locations
- provide Autoscaling, incl. Scale-To-Zero

"Since your model is being deployed as an InferenceService, not a raw Kubernetes Service, you just need to provide the storage location of the model and it gets some super powers out of the box 🚀."

InferenceService

