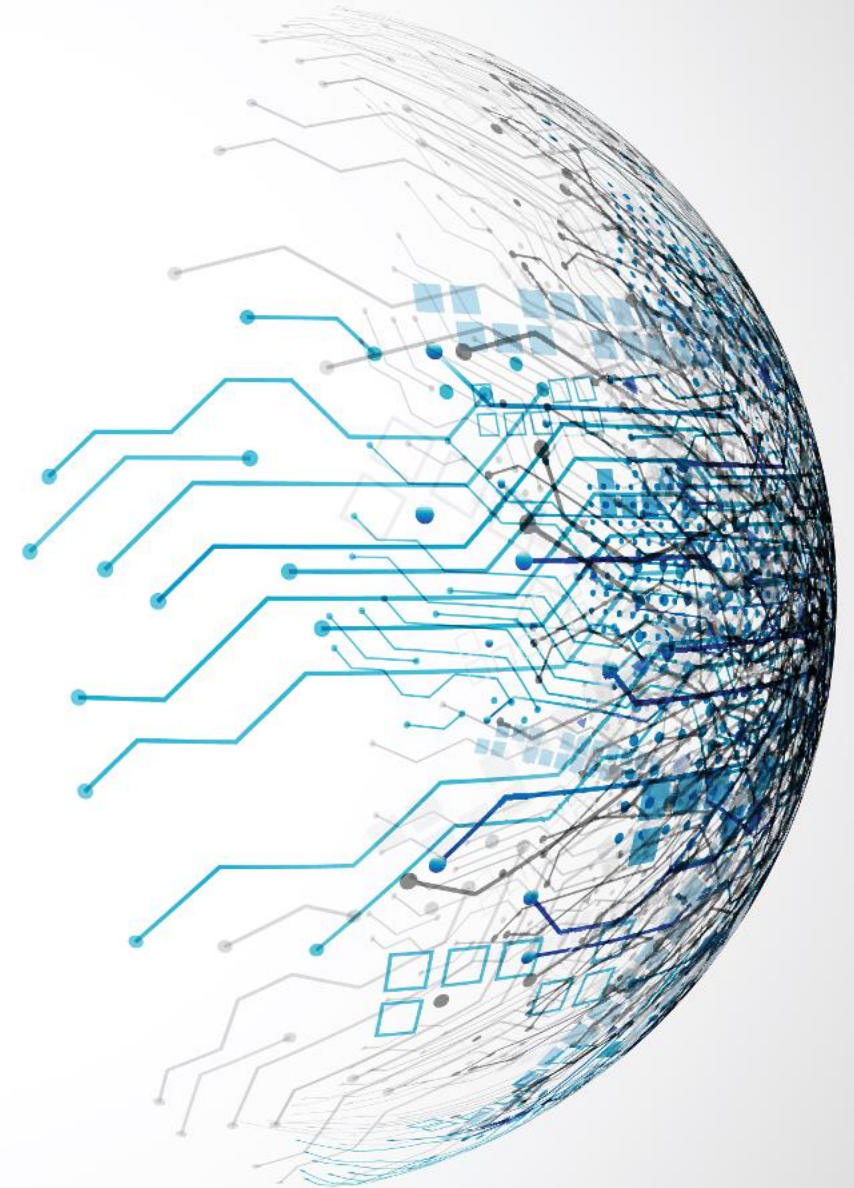


Deep Learning

INTRODUCTION AND SOFTWARE STACK

Dr. Mohammed Salah Al-Radhi

(slides by: Dr. Bálint Gyires-Tóth)



Copyright

Copyright © **Bálint Gyires-Tóth & Mohammed Salah Al-Radhi**, All Rights Reserved.

This presentation and its contents are protected by copyright law. The intellectual property contained herein, including but not limited to text, images, graphics, and design elements, are the exclusive property of the copyright holder identified above. Any unauthorized use, reproduction, distribution, or modification of this presentation or its contents is strictly prohibited without prior written consent from the copyright holder.

No Recordings or Reproductions: Attendees, viewers, and recipients of this presentation are expressly prohibited from making any audio, video, or photographic recordings, as well as screen captures, screenshots, or any form of reproduction, of this presentation, its content, or any related materials, whether during its live presentation or subsequent access. Violation of this prohibition may result in legal action.

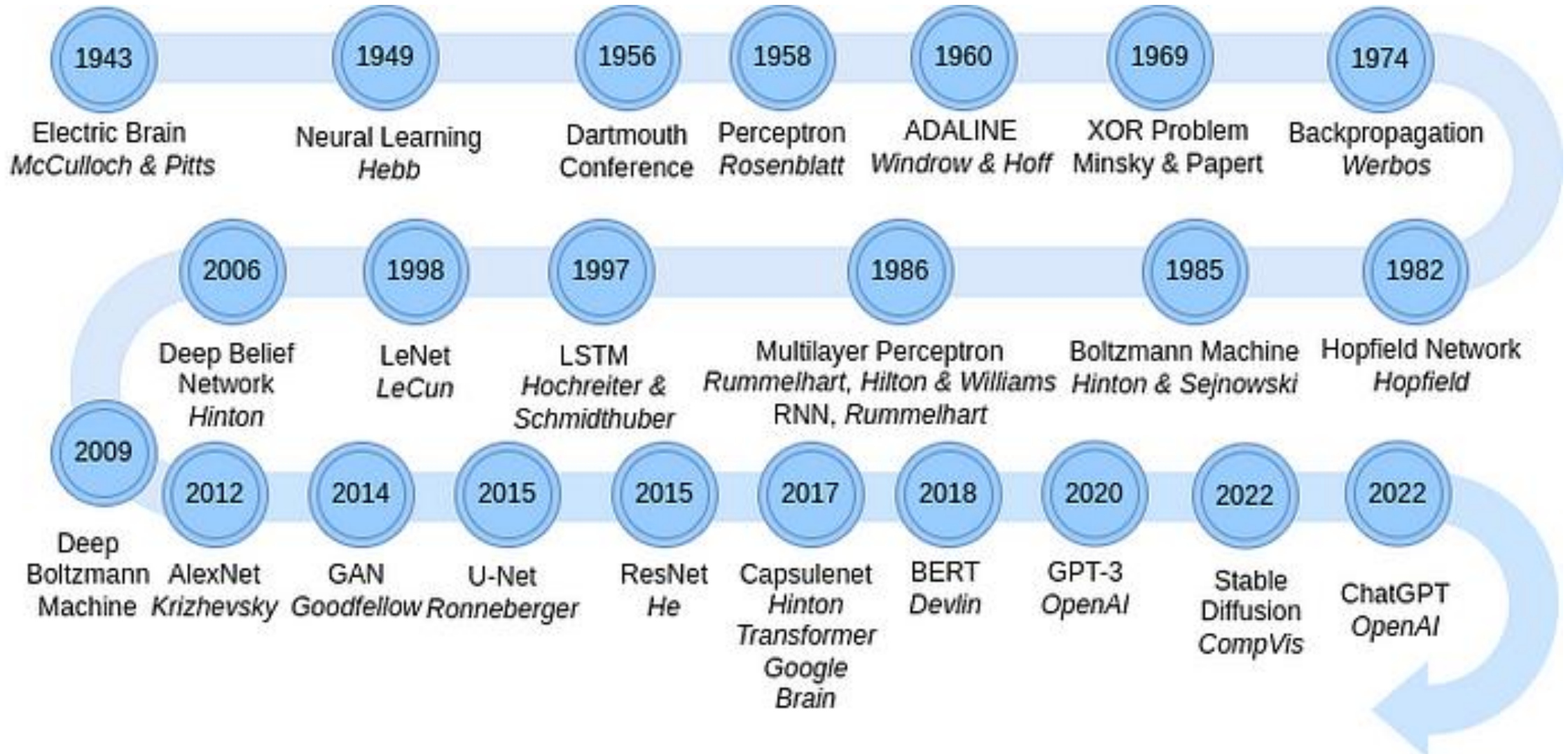
For permissions, inquiries, or licensing requests, please contact: **{toth.b,malradhi}@tmit.bme.hu**

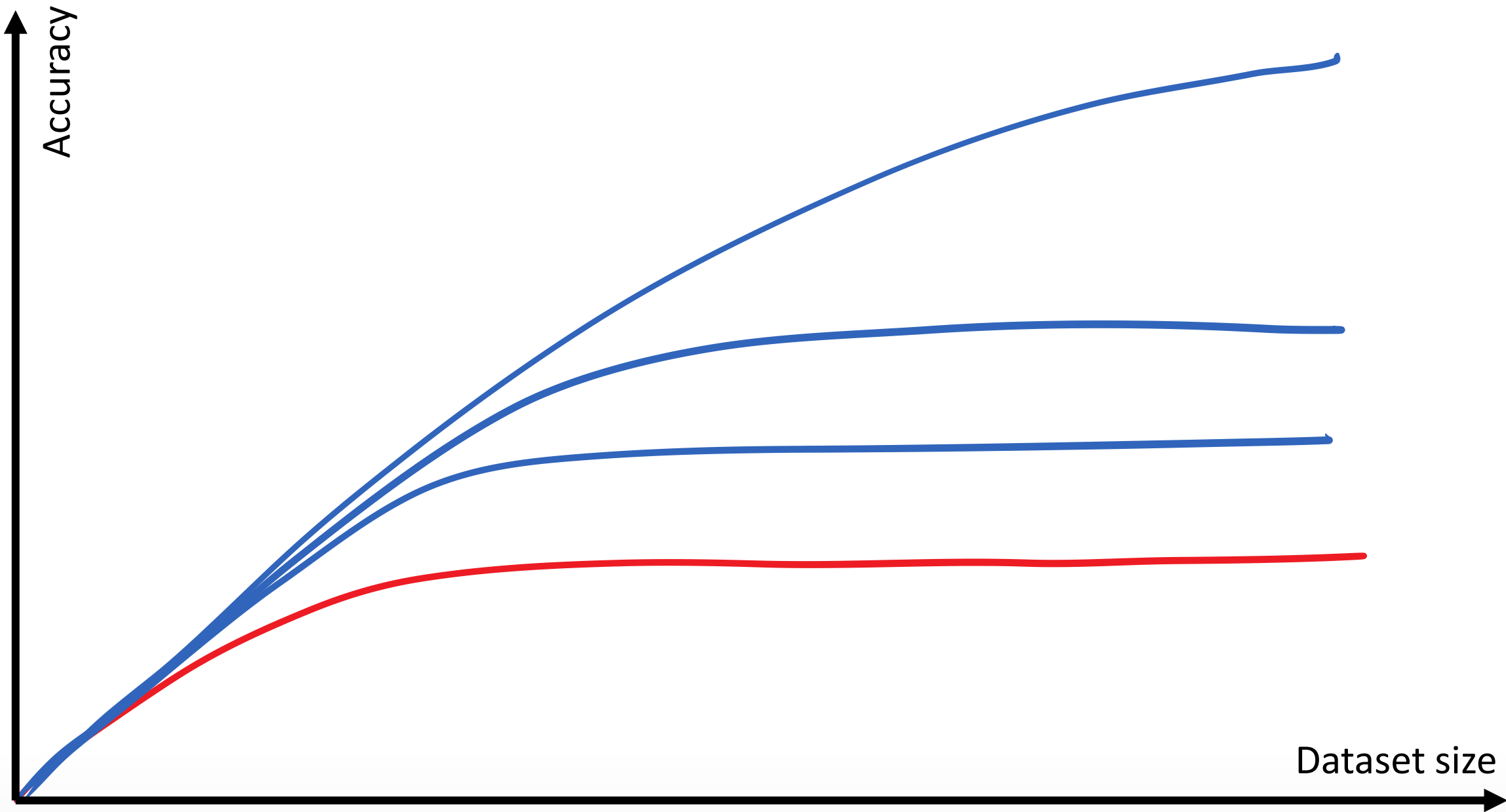
Unauthorized use, distribution, or reproduction of this presentation may result in civil and criminal penalties. Thank you for respecting the intellectual property rights of the copyright holder.

Outline

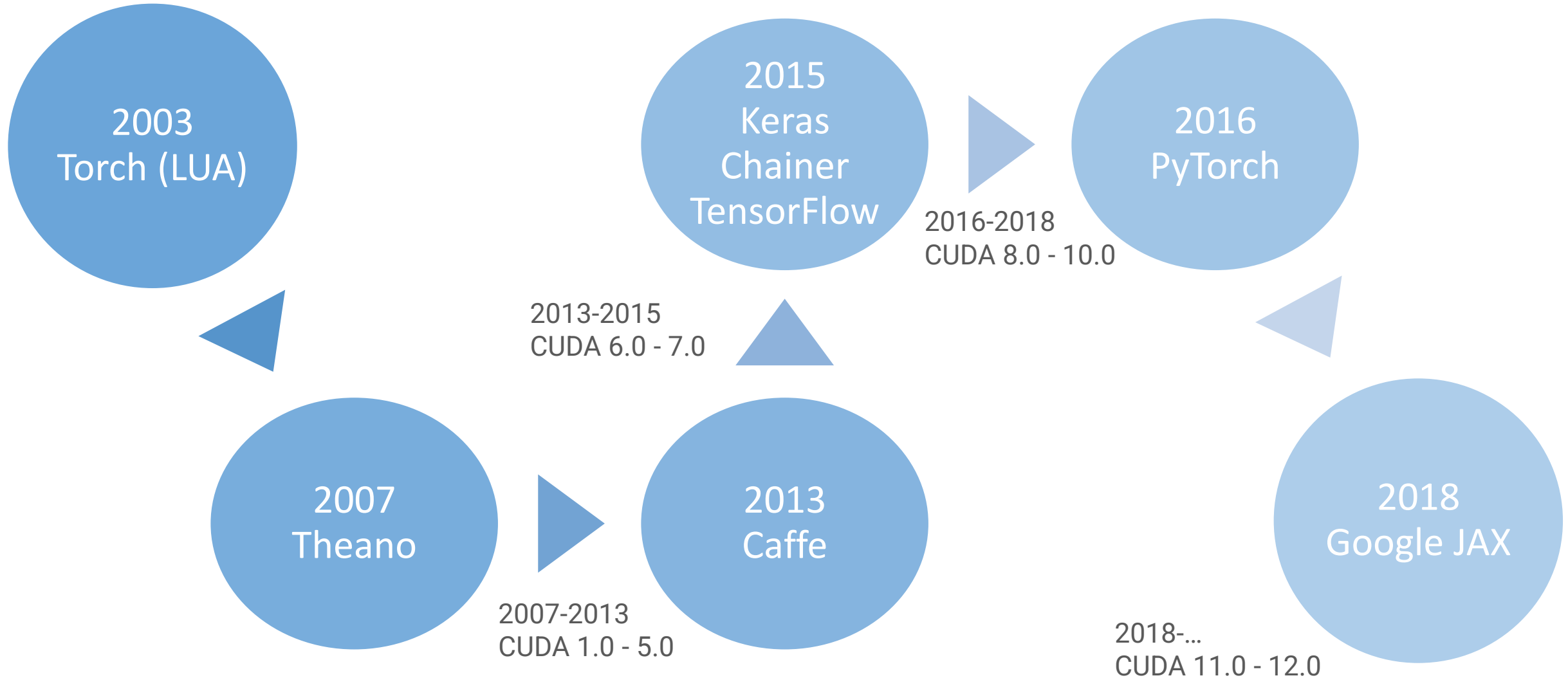
1. Brief history of deep learning
2. CRISP-DM for deep learning
3. Deep learning roles
4. Basic software components
5. Advanced software components

Brief history of deep learning





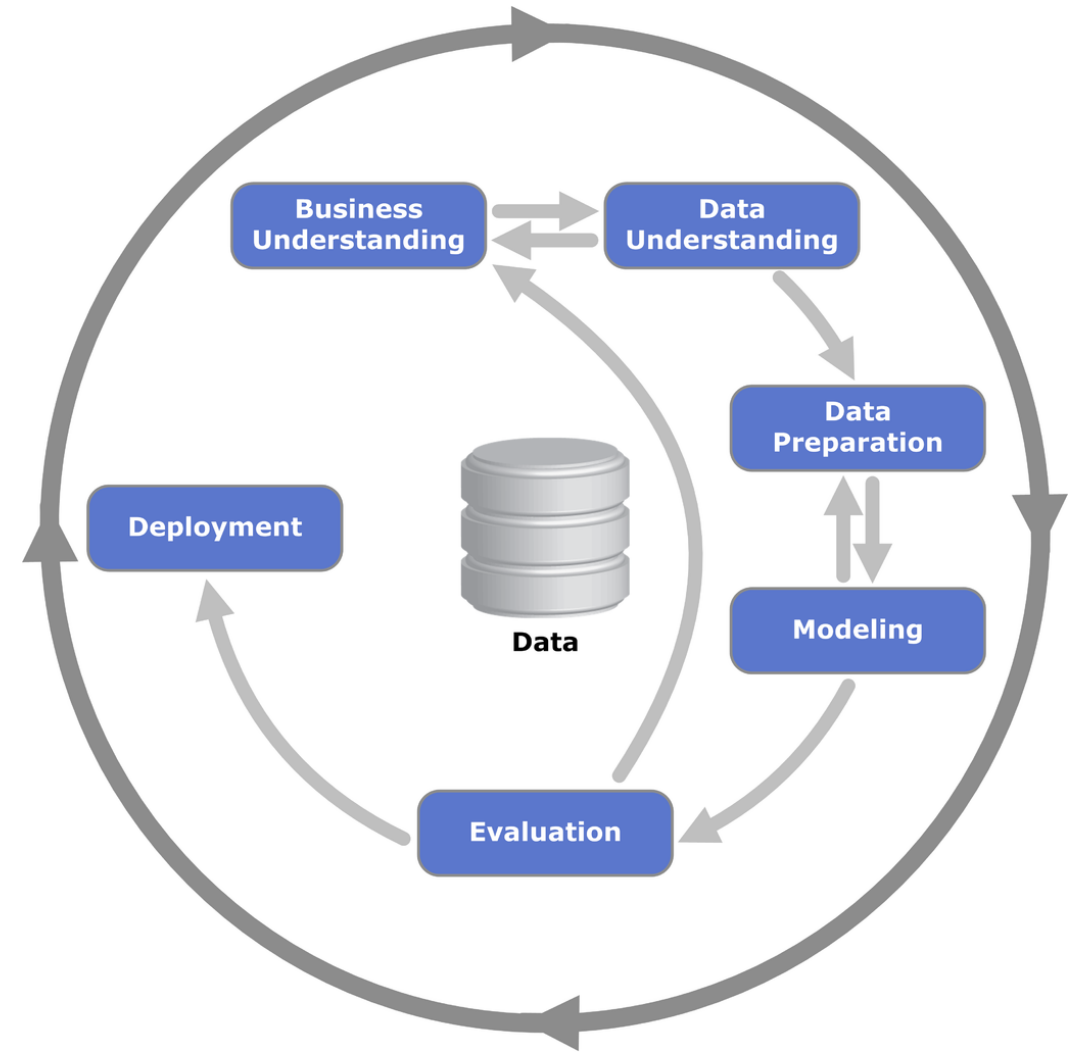
Brief history of deep learning framework



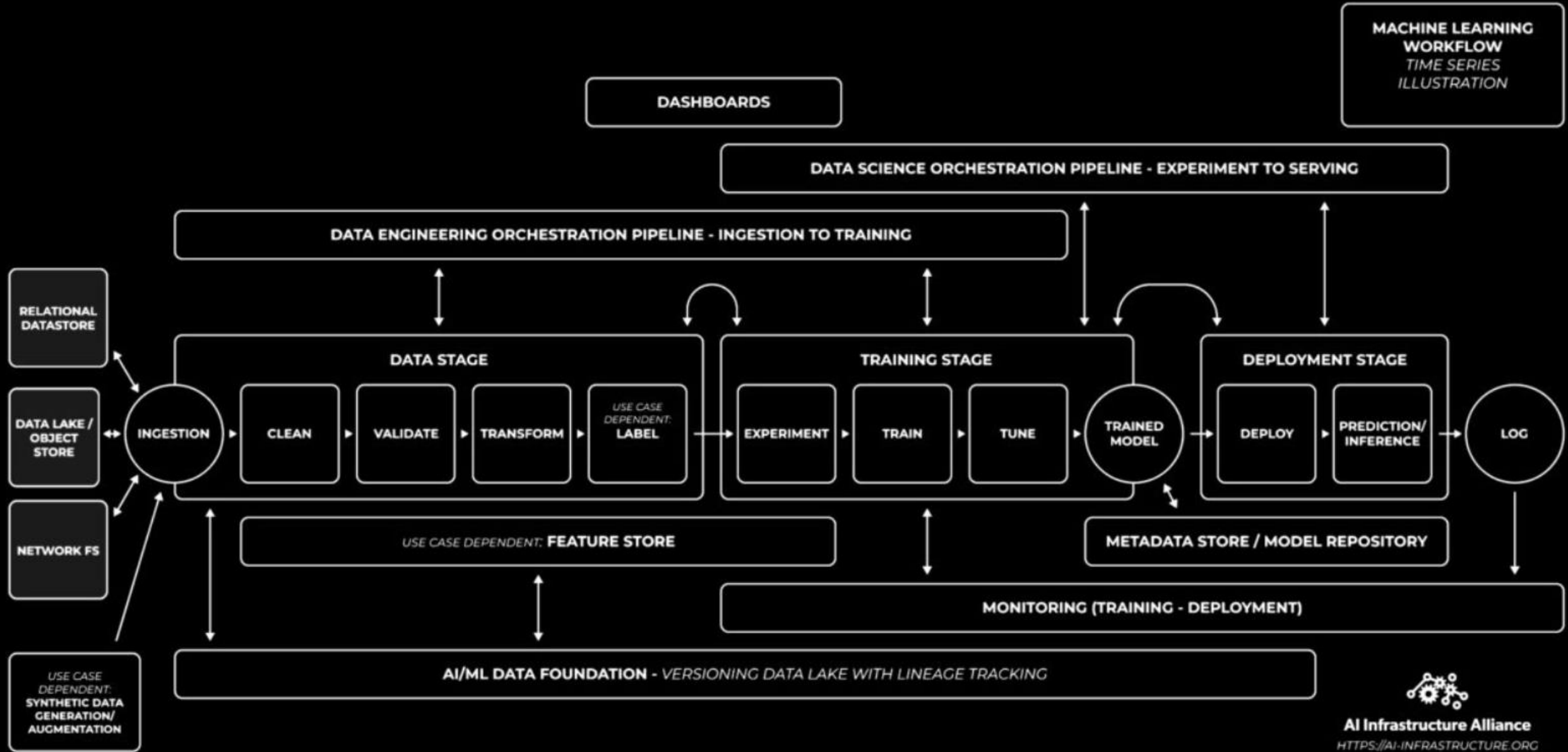
CRISP-DM for deep learning

Cross Industry Standard Process for Data Mining

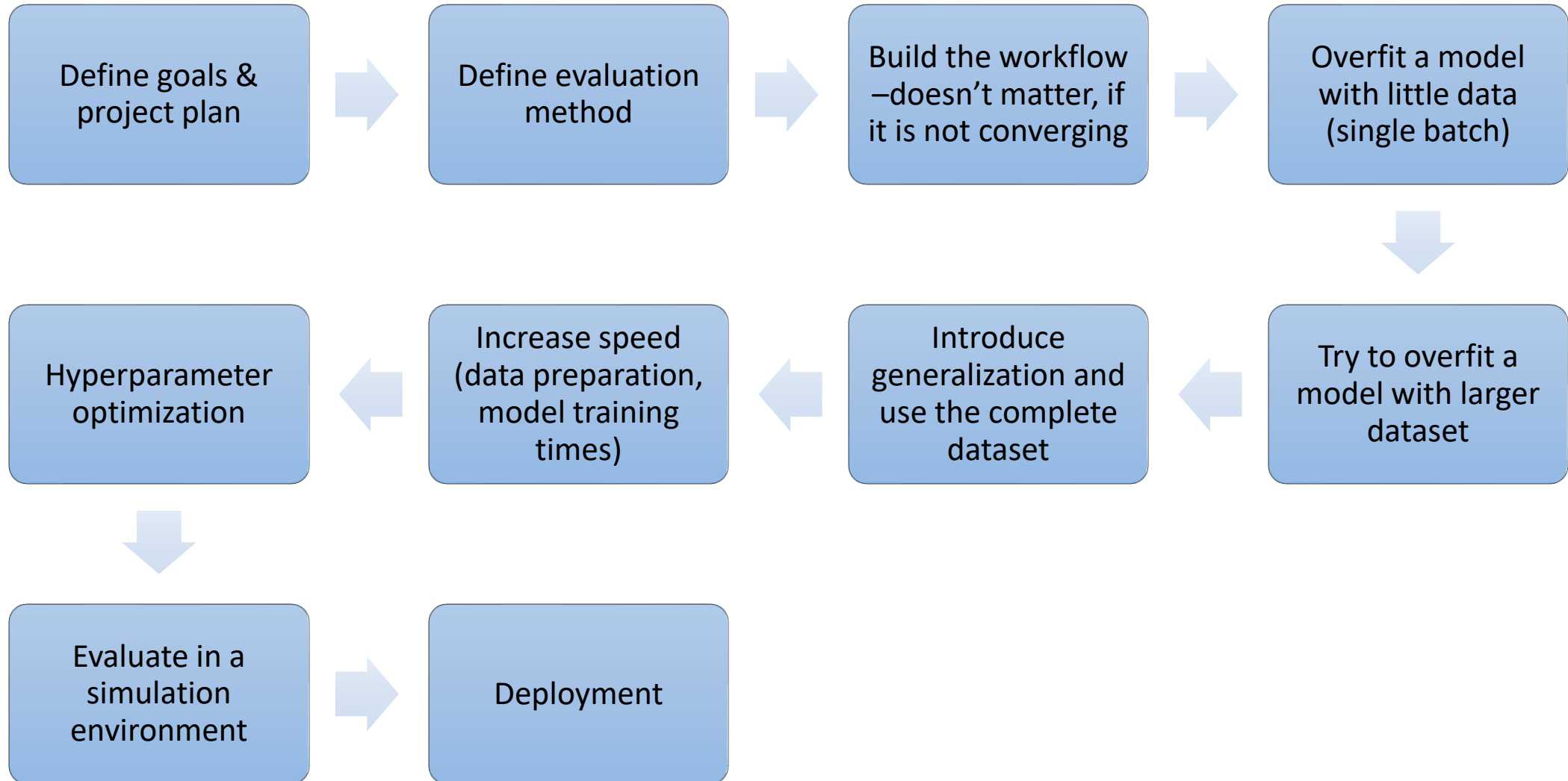
- *Business Understanding*
- *Data Understanding*
- *Data Preparation*
- *Modeling*
- *Evaluation*
- *Deployment*



Source: https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining



Machine learning project main steps





No free lunch theorem

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.

AI/Deep learning roles

Data
engineer

Data
scientist

Business
analyst

DL/ML
Engineer

AI/Deep learning skills

Database

Data engineering

Data vizualization

Storytelling/reporting

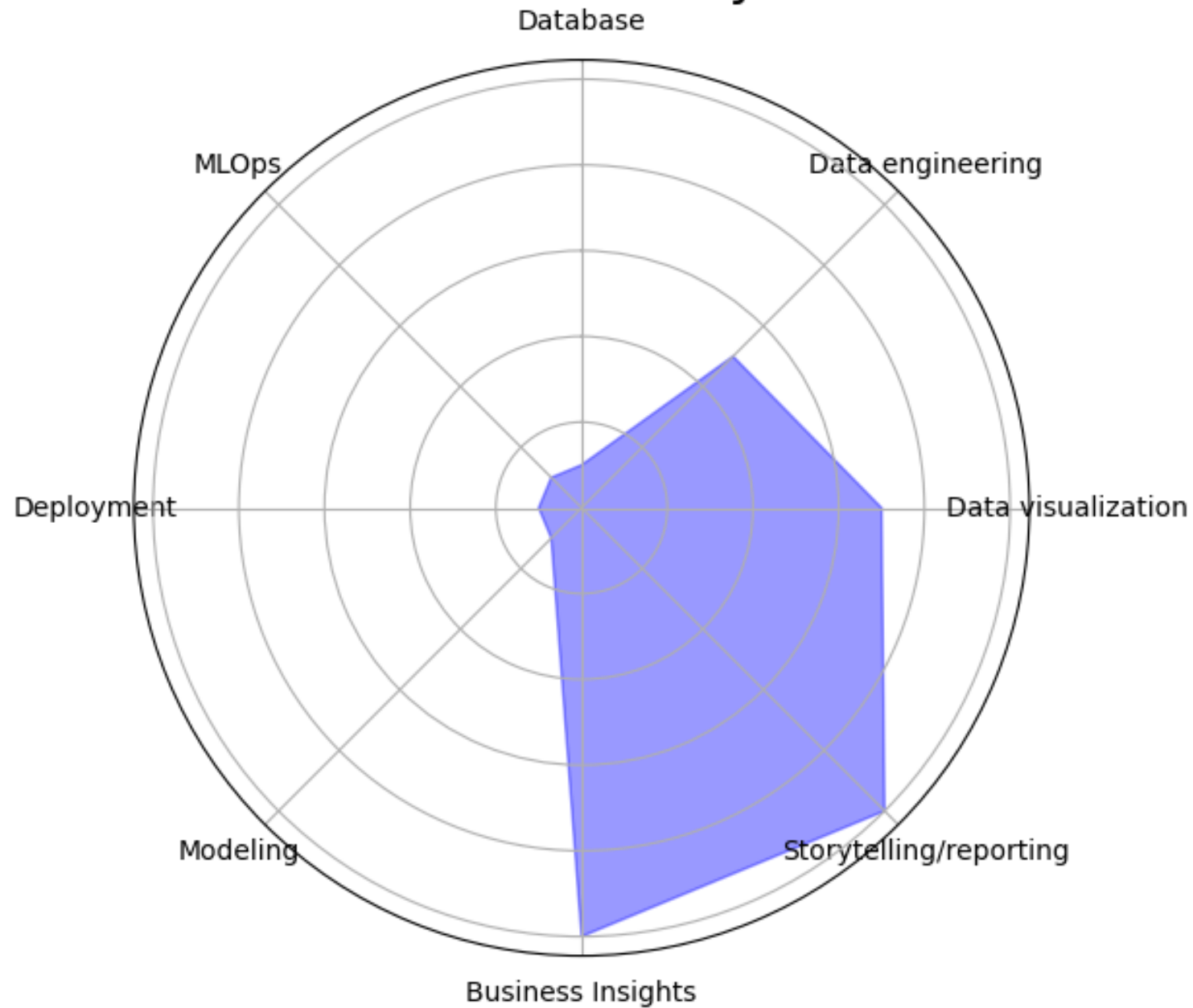
Business Insights

Modeling

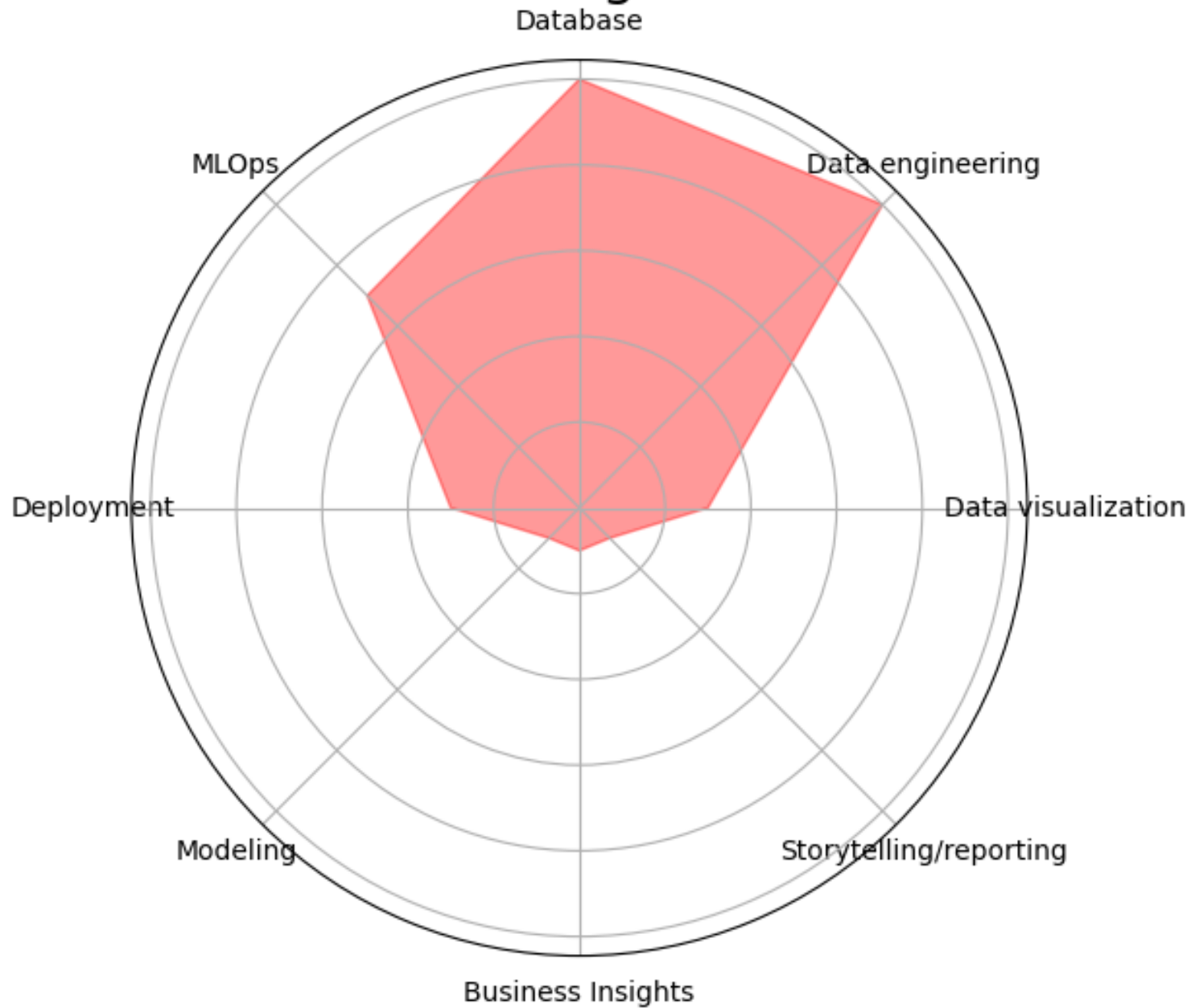
Deployment

MLOps

Business Analyst



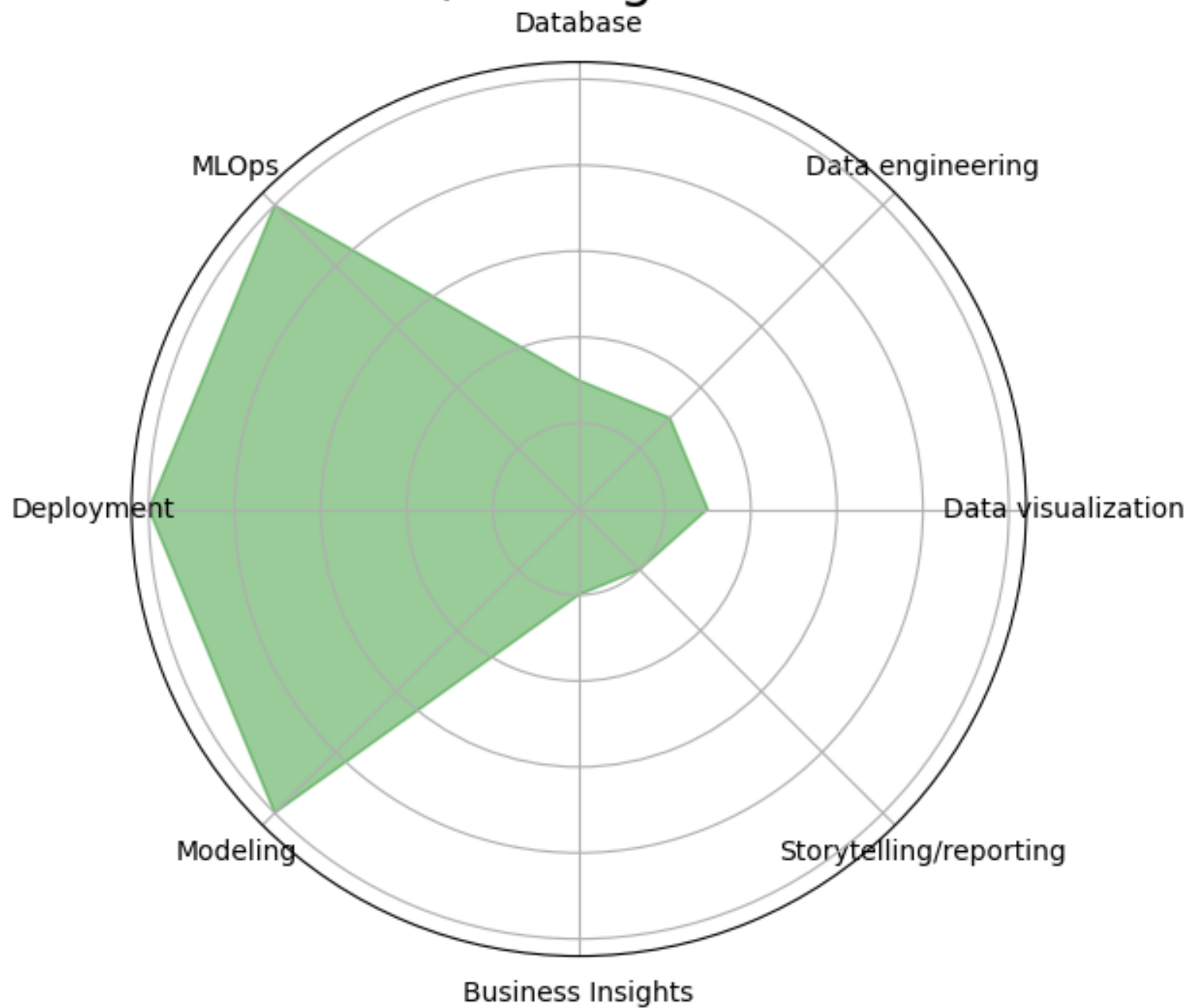
Data Engineer



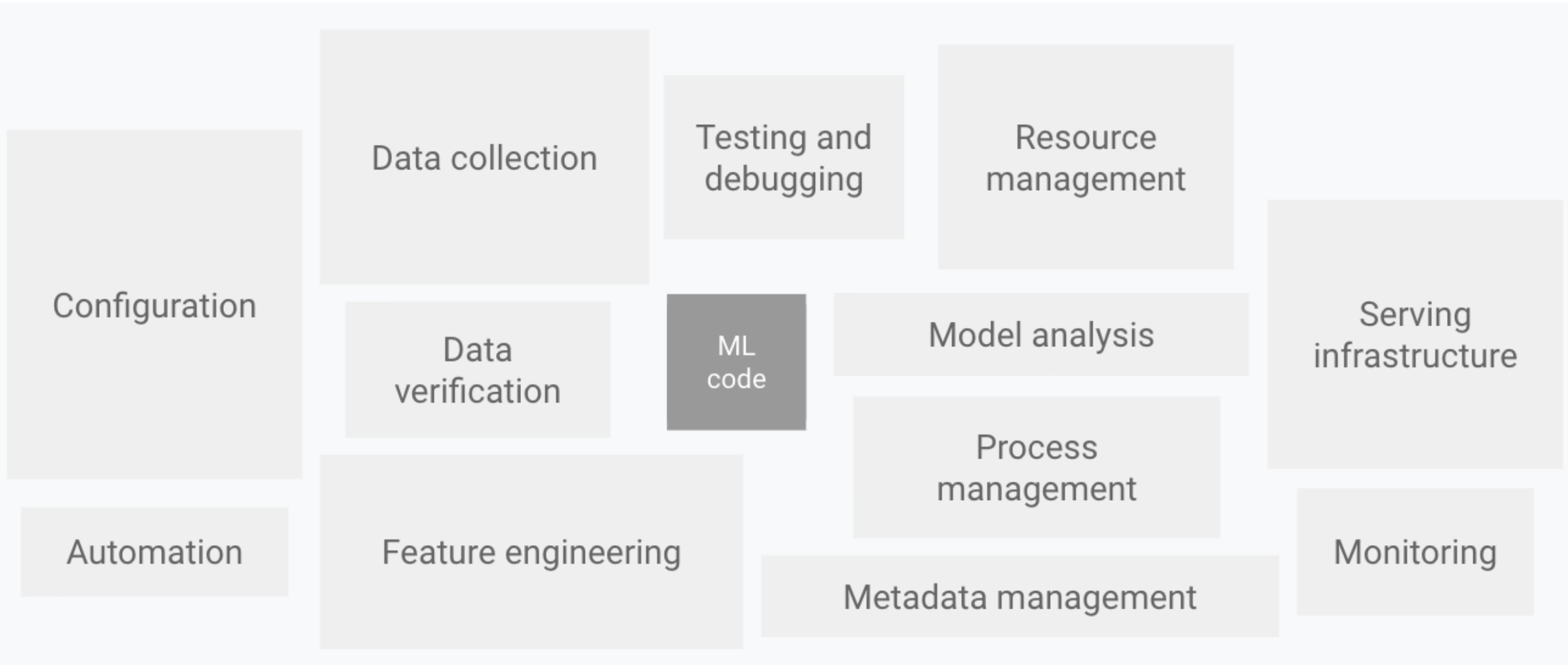
Data Scientist



ML/DL Engineer



General ML related tasks





Deep Learning software stack

Google Colab

<https://colab.research.google.com/>

The screenshot shows the Google Colab interface for a notebook named "VisionTransformer.ipynb". The notebook content includes a copyright notice and a section titled "Vision Transformers and Convolutional Neural Networks". A modal dialog titled "Futtatókörnyezet módosítása" (Change runtime environment) is open, allowing the user to select the runtime type (Python 3) and hardware acceleration (T4 GPU, A100 GPU, V100 GPU, or CPU). The dialog also includes a link to purchase additional compute units.

Copyright

The present Jupyter notebook was created as a teaching material at the Budapest University of Technology and Economics. Reproduction or publication of any part of this notebook is only allowed with the written consent of the authors.

The following source has been used: https://d21.ai/chapter_attention

2023 (c) András Kalapos (kalapos.andras@edu.bme.hu)

Vision Transformers and Convolutional Neural Networks

This notebook compares the performance of Vision Transformers (ViT) and Convolutional Neural Networks (CNN) on the Imagenette dataset.

The Imagenette dataset is a subset of the larger ImageNet dataset, consisting of 100 classes of images. It is used to evaluate the capabilities of different models with limited resources.

```
[ ] !pip install pytorch-lightning timm torchmetrics wandb > null
```

```
[ ] import os # for path and environment variables
from pathlib import Path # for path
from torchvision.datasets import ImageFolder # for loading images
from torchvision.datasets.utils import download_and_extract_archive, verify_str_arg # for downloading and extracting dataset
import logging # for logging
import numpy as np
import torch # deep learning framework
import torch.utils.data # dataloader
from typing import Optional
import albumentations as A # for data augmentation

import torchvision.transforms as transforms # for data augmentation
```



Grafana

PyTorch

Keras



Prometheus



docker



kubernetes



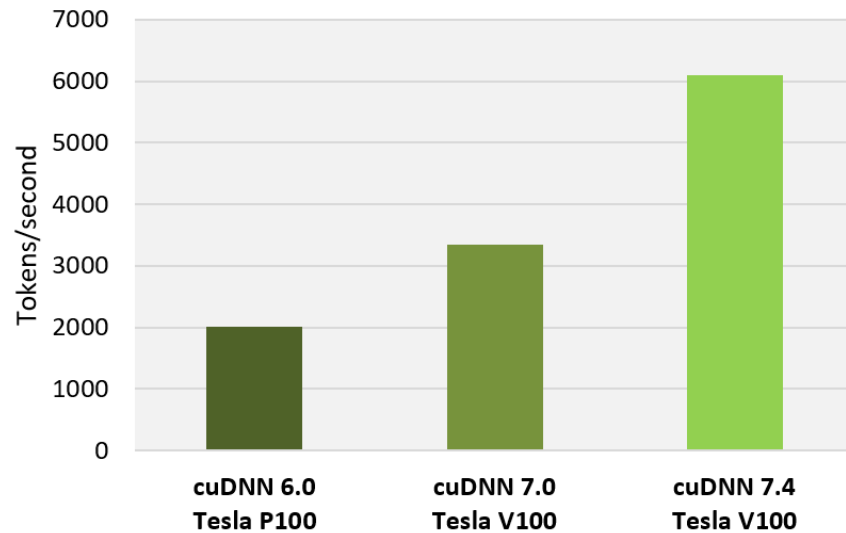
ANSIBLE



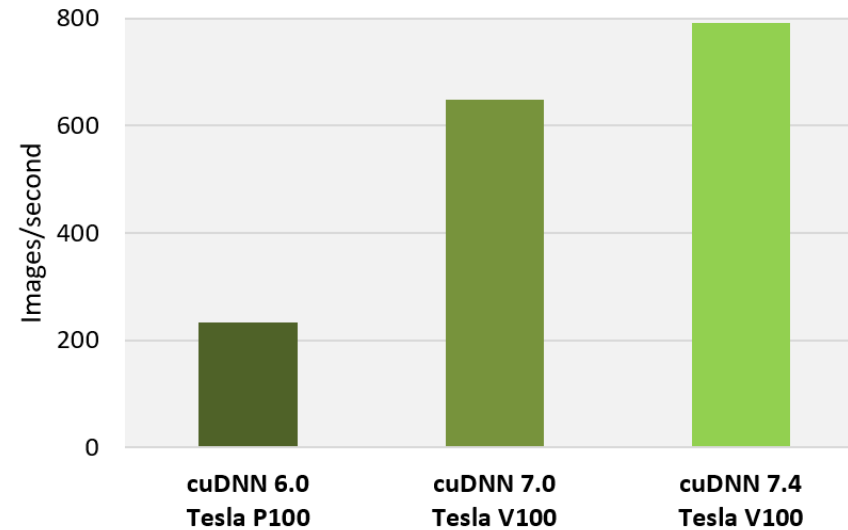
Basic software components: CUDA driver

Required: NVIDIA Driver

Main elements: cuBLAS, cuSPARSE, cuDNN, NCCL, NVVP, debugger, memcheck.



TensorFlow performance (tokens/sec), Tesla P100 + cuDNN 6 (FP32) on 17.12 NGC container, Tesla V100 + cuDNN 7.0 (Mixed) on 18.02 NGC container, Tesla V100 + cuDNN 7.4 (Mixed) on 18.10 NGC container, OpenSeq2Seq (GNMT), Batch Size: 64



TensorFlow performance (images/sec), Tesla P100 + cuDNN 6 (FP32) on 17.12 NGC container, Tesla V100 + cuDNN 7.0 (Mixed) on 18.02 NGC container, Tesla V100 + cuDNN 7.4 (Mixed) on 18.10 NGC container, ResNet-50, Batch Size: 128

CUDA version

nvidia-smi

```
Fri Jul 28 19:17:05 2023
```

NVIDIA-SMI 525.125.06		Driver Version: 525.125.06		CUDA Version: 12.0	
GPU	Name	Persistence-M	Bus-Id	Disp A	Volatile Uncorr. EC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory	GPU-Util Compute M.
0	NVIDIA A100-SXM ...	On	00000000:07:00.0	Off	0
N/A	53C	P0	349W / 400W	76249MiB / 81920MiB	100% Default Disabled
1	NVIDIA A100-SXM ...	On	00000000:0F:00.0	Off	0
N/A	57C	P0	394W / 400W	76873MiB / 81920MiB	100% Default Disabled
2	NVIDIA A100-SXM ...	On	00000000:47:00.0	Off	0
N/A	61C	P0	370W / 400W	76615MiB / 81920MiB	99% Default Disabled
3	NVIDIA A100-SXM ...	On	00000000:4E:00.0	Off	0
N/A	59C	P0	404W / 400W	76615MiB / 81920MiB	100% Default Disabled
4	NVIDIA A100-SXM ...	On	00000000:87:00.0	Off	0
N/A	74C	P0	397W / 400W	76619MiB / 81920MiB	100% Default Disabled
5	NVIDIA A100-SXM ...	On	00000000:90:00.0	Off	0
N/A	67C	P0	353W / 400W	76601MiB / 81920MiB	100% Default Disabled
6	NVIDIA A100-SXM ...	On	00000000:B7:00.0	Off	0
N/A	71C	P0	389W / 400W	76631MiB / 81920MiB	99% Default Disabled
7	NVIDIA A100-SXM ...	On	00000000:BD:00.0	Off	0
N/A	71C	P0	336W / 400W	76687MiB / 81920MiB	100% Default Disabled

Processes:						
GPU	GI ID	CI ID	PID	Type	Process name	GPU Memory Usage
0	N/A	N/A	342870	C	/opt/conda/bin/python3	76246MiB
1	N/A	N/A	342871	C	/opt/conda/bin/python3	76870MiB
2	N/A	N/A	342872	C	/opt/conda/bin/python3	76612MiB
3	N/A	N/A	342873	C	/opt/conda/bin/python3	76612MiB
4	N/A	N/A	342874	C	/opt/conda/bin/python3	76616MiB
5	N/A	N/A	342875	C	/opt/conda/bin/python3	76598MiB
6	N/A	N/A	342876	C	/opt/conda/bin/python3	76628MiB
7	N/A	N/A	342877	C	/opt/conda/bin/python3	76684MiB

nvcc --version

```
nvcc: NVIDIA (R) Cuda compiler driver  
Copyright (c) 2005-2022 NVIDIA Corporation  
Built on Wed Sep 21 10:33:58 PDT 2022  
Cuda compilation tools, release 11.8, V11.8.89  
Build cuda_11.8.r11.8/compiler.3183396_0
```

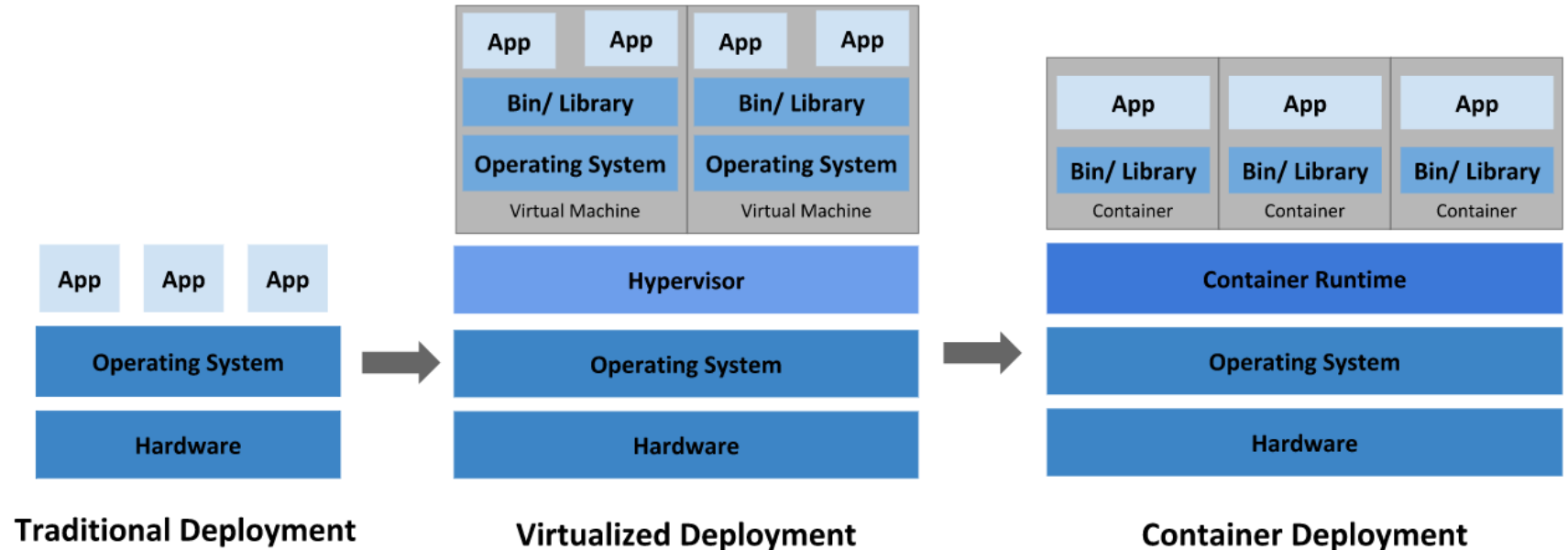
Basic software components: Containerization

Runs a virtual machine on the host and shares resources.

Encapsulations of system environments.

Advantages:

- Reproducibility
- Portability
- Isolation
- Integration



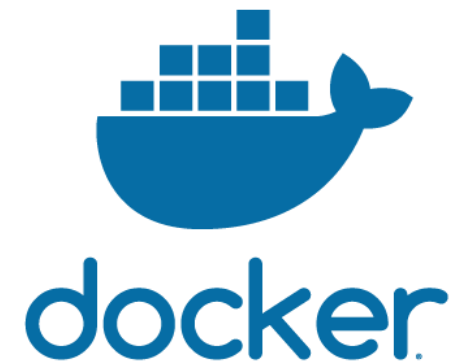
Source: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>

Basic software components: Containerization

- PaaS (Platform-as-a-Service) - Primarily for microservices
- OS and GPU lightweight virtualization
- Isolates Dev and Ops
- Docker images are stored in a local cache and can be interacted with by commands
- Large ecosystem, Linux, MacOS and Windows support

Difference compared to VM: e.g. the system is 1 GB

- 1000 VM $\sim 1000 * 1$ GB
- 1000 application container ~ 1 GB
- Container is refreshed \rightarrow Everything is refreshed



Basic software components: DL frameworks

- TensorFlow and TensorFlow Keras (Google)
- PyTorch (Meta AI)
- JAX (Google)
- MXNet (Apache)
- Gluon (Amazon)
- Chainer
- PaddlePaddle

Depricated

- Sonnet (DeepMind)
- CNTK (Microsoft)

Advanced components: monitoring

Metrics logging tools is required:

- nvidia-smi dmon
- Prometheus + NVML (NVIDIA Management Library)

Open source tools:

- Grafana
- Zabbix



Prometheus



Grafana

ZABBIX



GPU 1937b558-347d-0f30-105b-893b98985668 ▾

Name
NVIDIA GeForce RTX 2080 SUPER

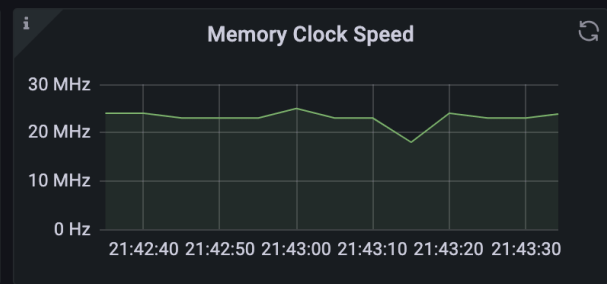
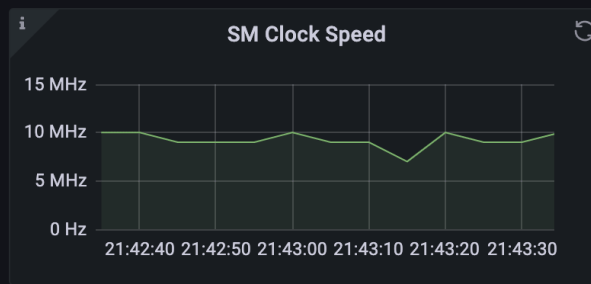
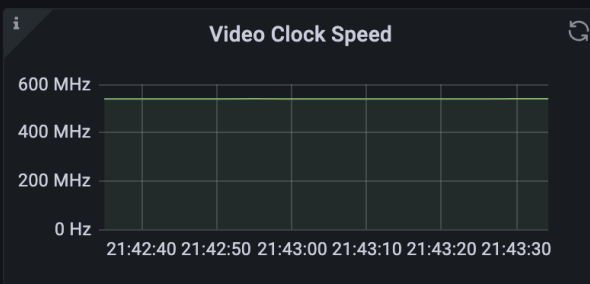
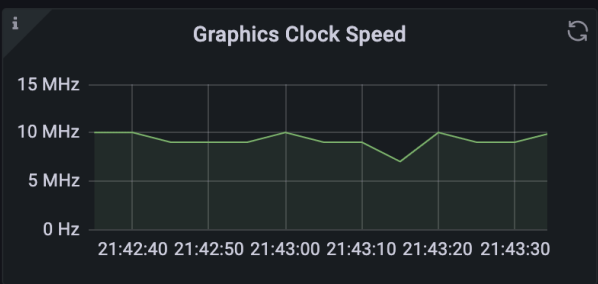
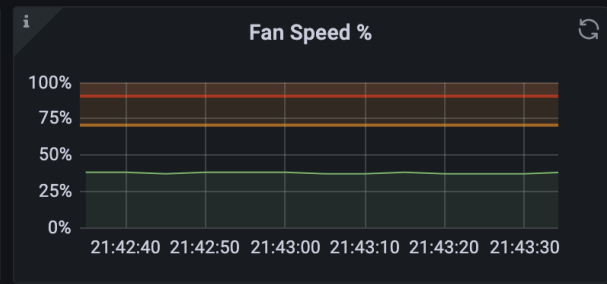
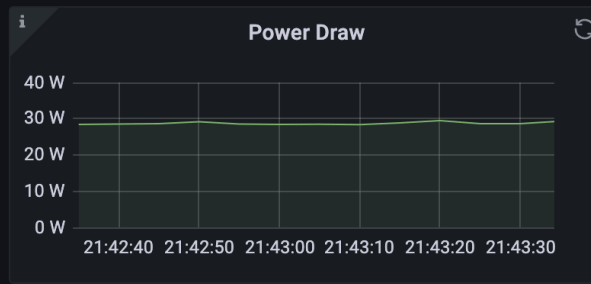
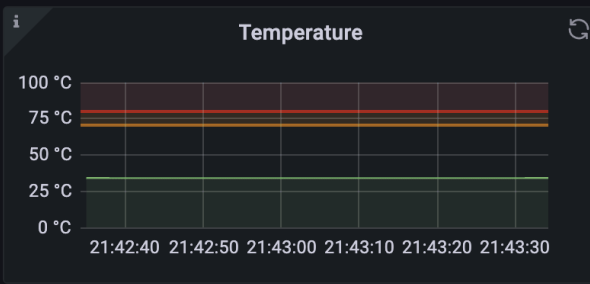
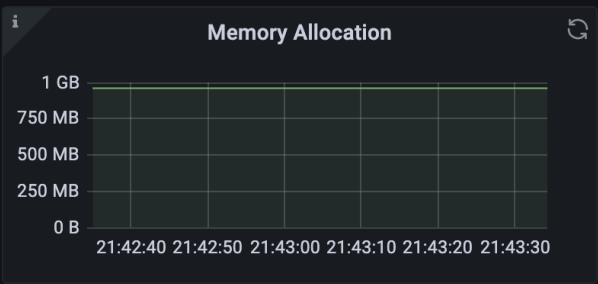
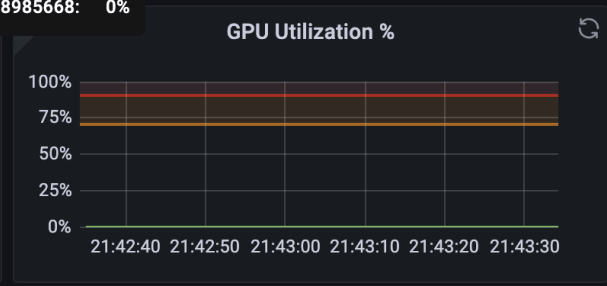
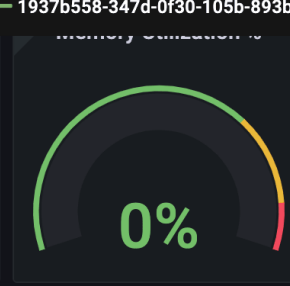
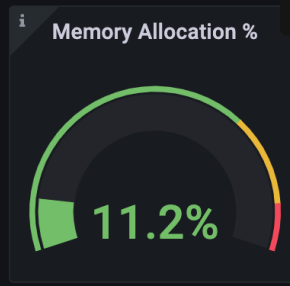
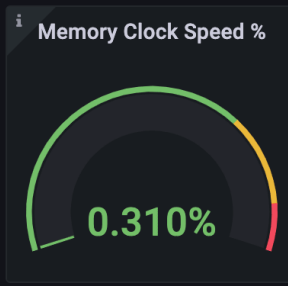
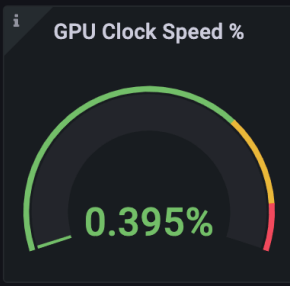
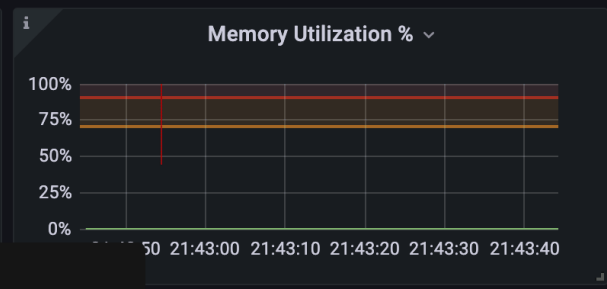
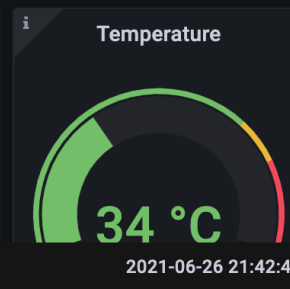
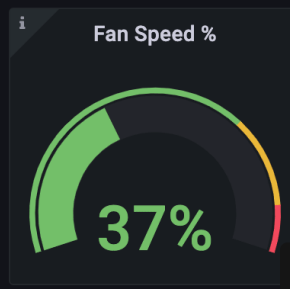
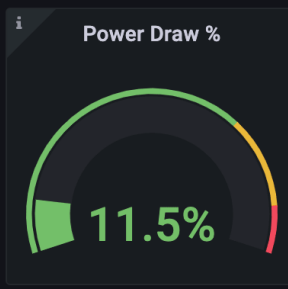
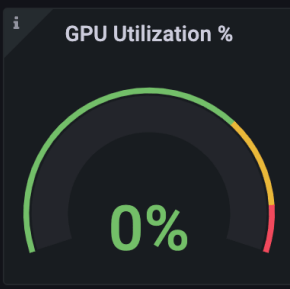
P-State
P8

Driver Version
471.11

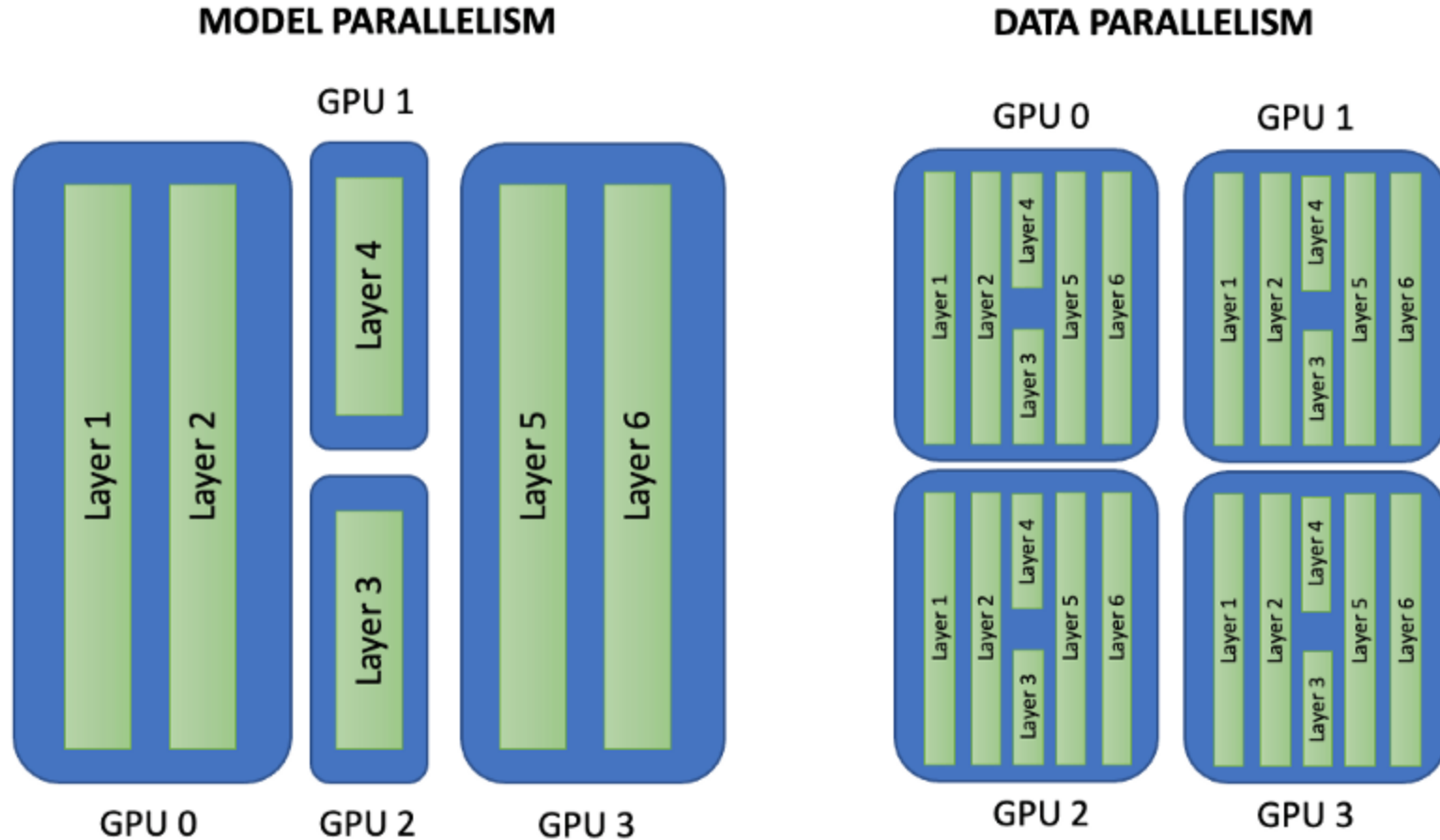
Vbios Version
90.04.7a.40.73

Throttle Reasons

Reason	Status
Idle	Active
HW Thermal Slowdown	Not Active
SW Power Cap	Not Active
App Clocks Setting	Not Active
HW Power Brake	Not Active
SW Thermal Slowdown	Not Active
Sync Boost	Not Active

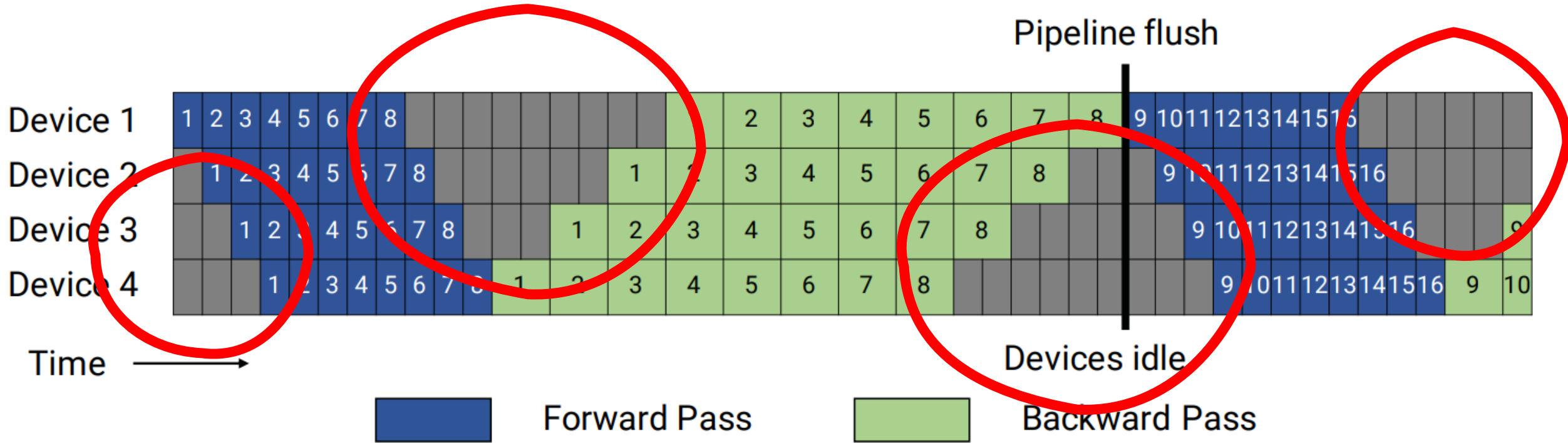


Advanced components: multi GPU, multi node



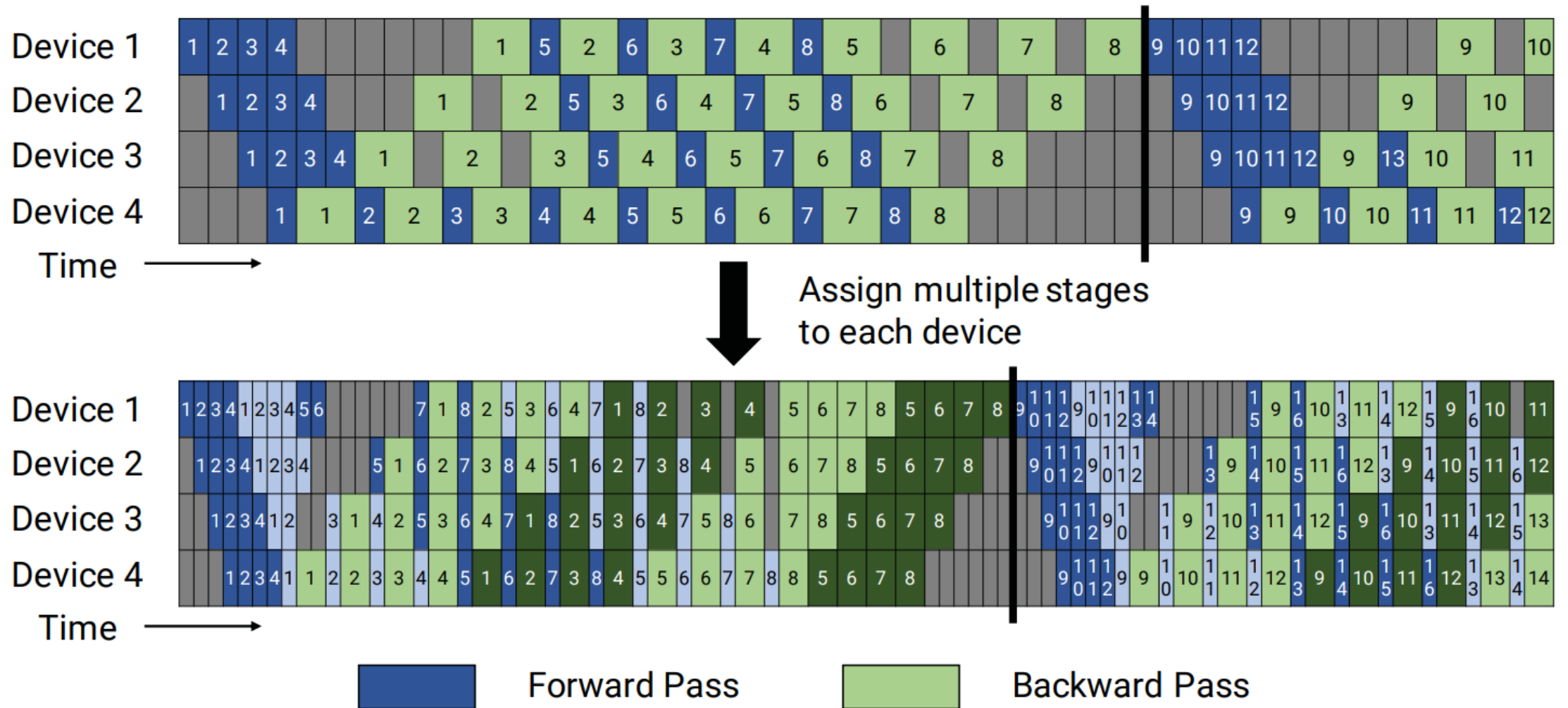
Source: <https://neptune.ai/blog/distributed-training-frameworks-and-tools>

Multi-GPU training (microbatching)



Source: Narayanan, Deepak, et al. "Efficient large-scale language model training on gpu clusters using megatron-lm." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021.

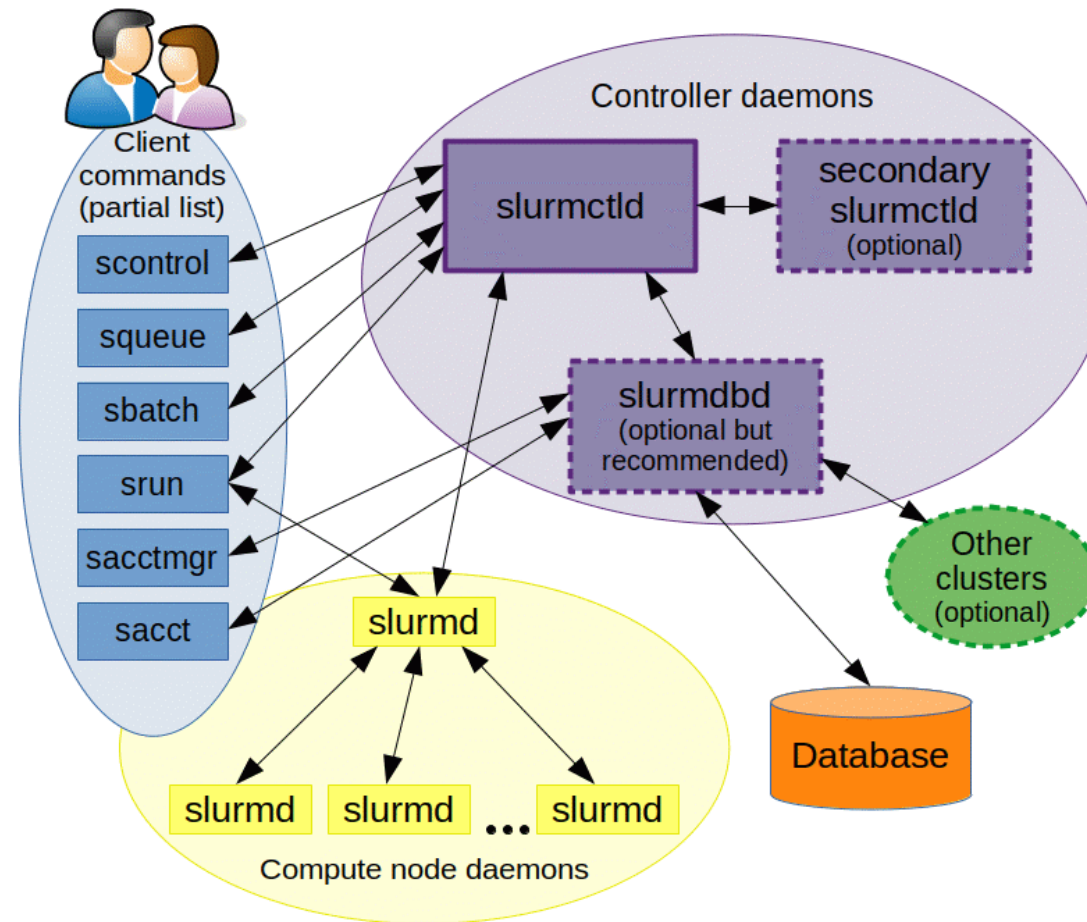
Multi-GPU training pipeline



Source: Narayanan, Deepak, et al. "Efficient large-scale language model training on gpu clusters using megatron-lm." *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021.

Advanced components: scheduler

- HPC/AI solution: SLURM Workload Manager



SLURM batch script

```
#!/bin/bash
```

```
#SBATCH -J sample
```

```
#SBATCH -t 15:00
```

```
#SBATCH -N 2
```

```
#SBATCH -n 8
```

```
#SBATCH --gres=gpu:4
```

Job name

Requested time

Number of nodes

Number of CPUs

Number of GPUs

```
module load singularity OpenMPI/3.1.6-GCC-8.3.0
```

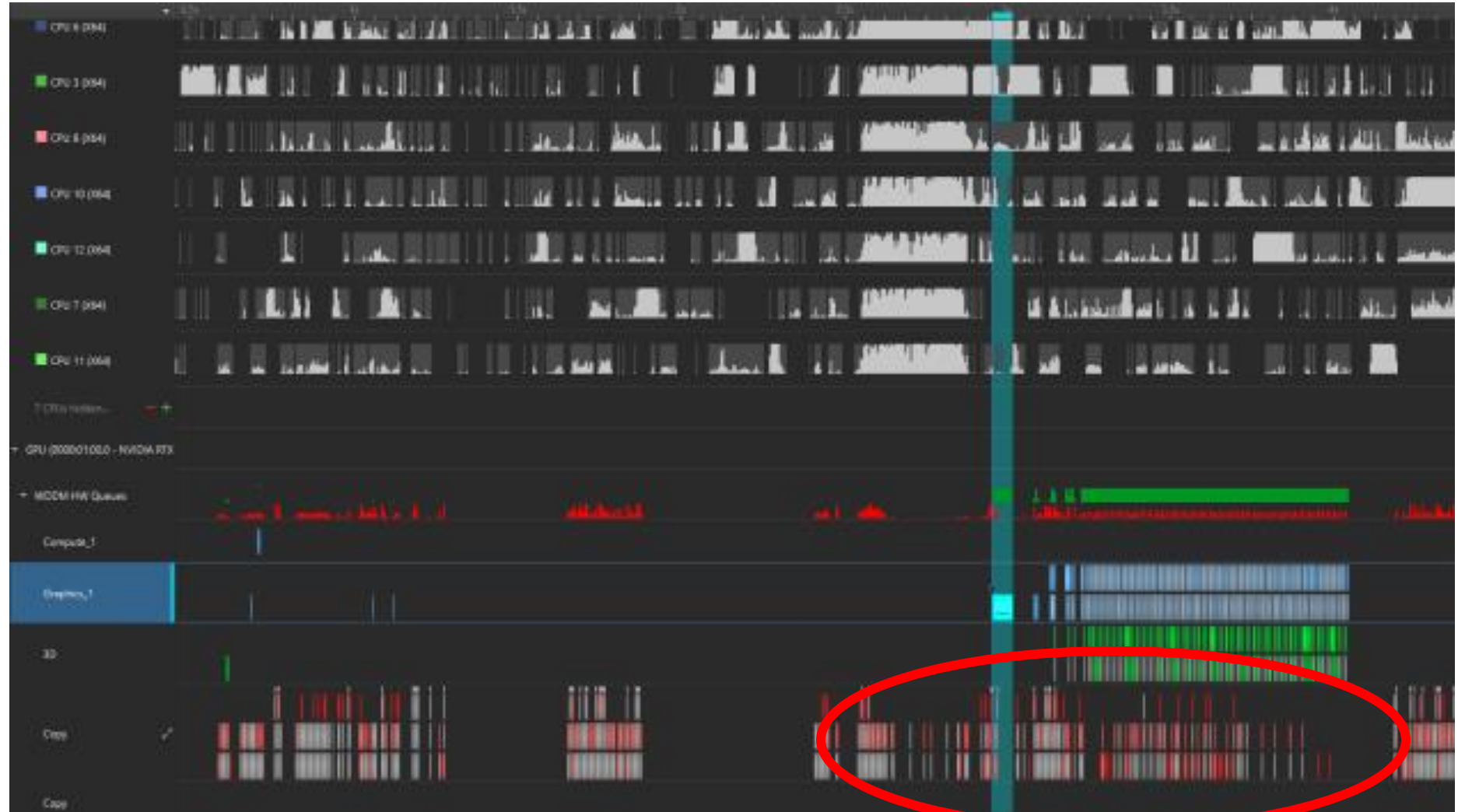
```
mpirun singularity run --nv horovod.sif python test.py
```

```
$ sbatch batchfile.sh
```


Advanced component: performance analytics

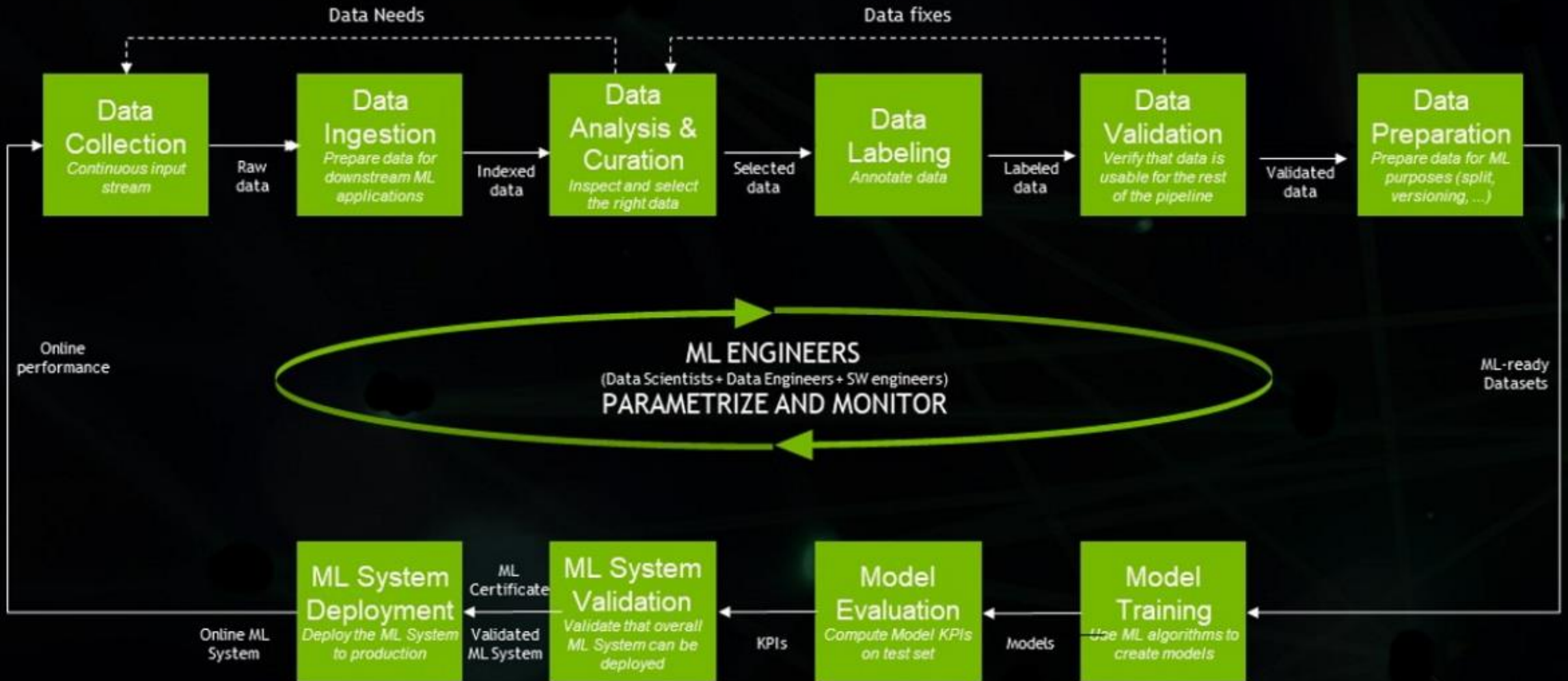
NVIDIA Nsight Systems:

- Monitor CPU and GPU usage
- Identify bottlenecks
- Get most out of the HW component

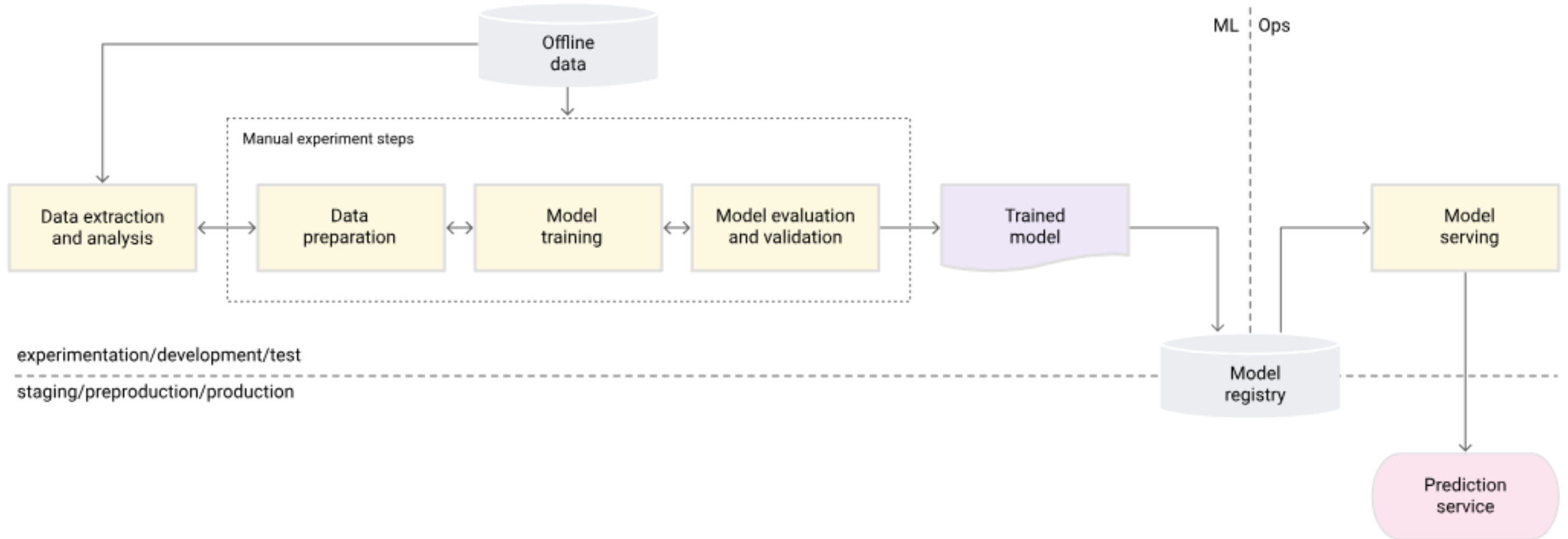


CPU-GPU interactions

MLOPS: THE AI LIFECYCLE FOR IT PRODUCTION

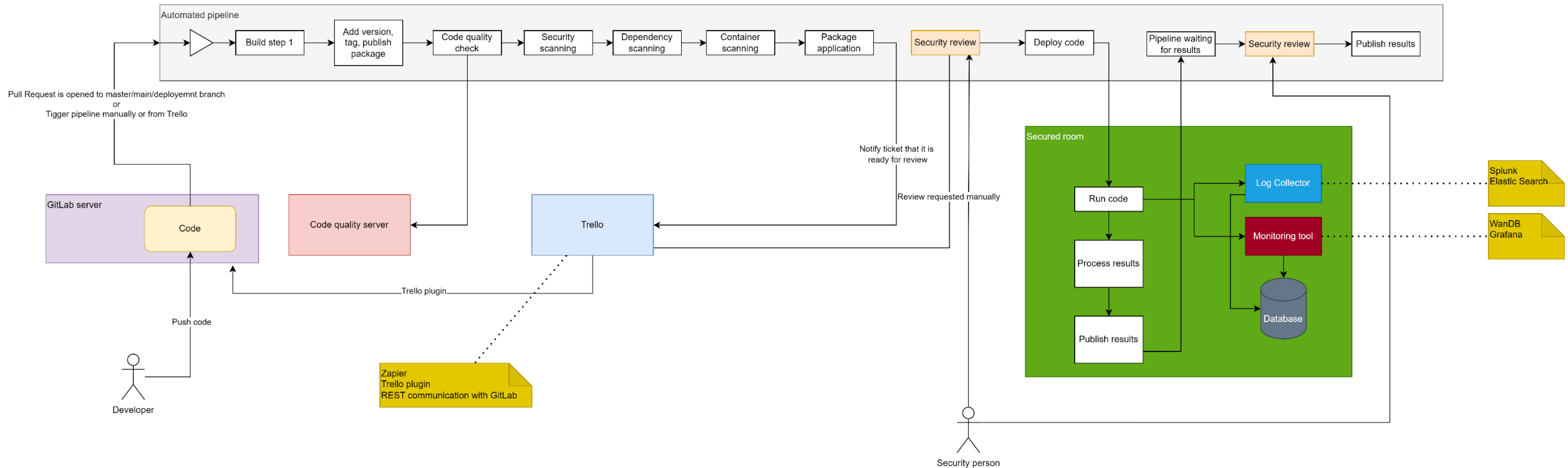


MLOps pipeline: level 0 – manual process

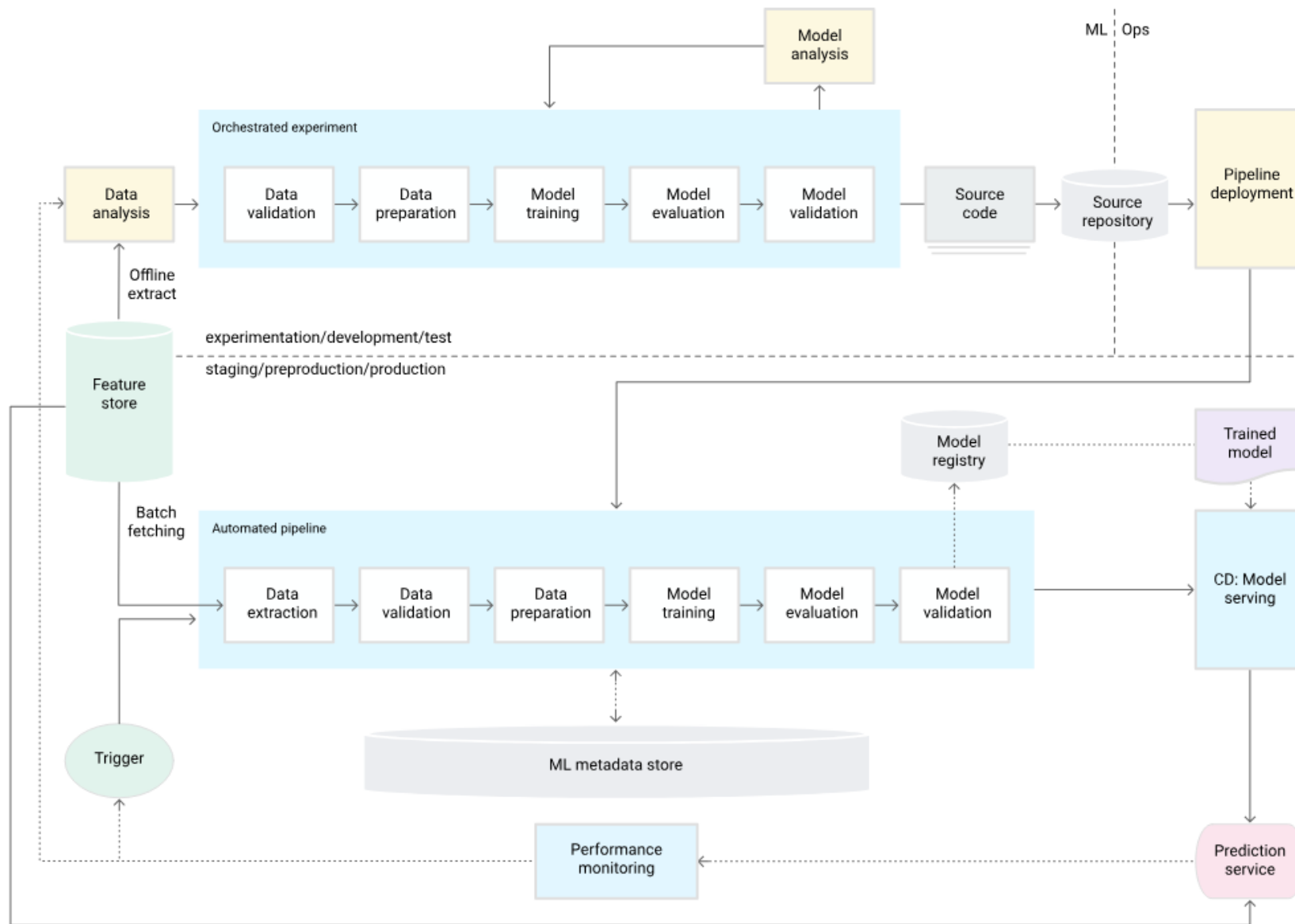


Source: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>

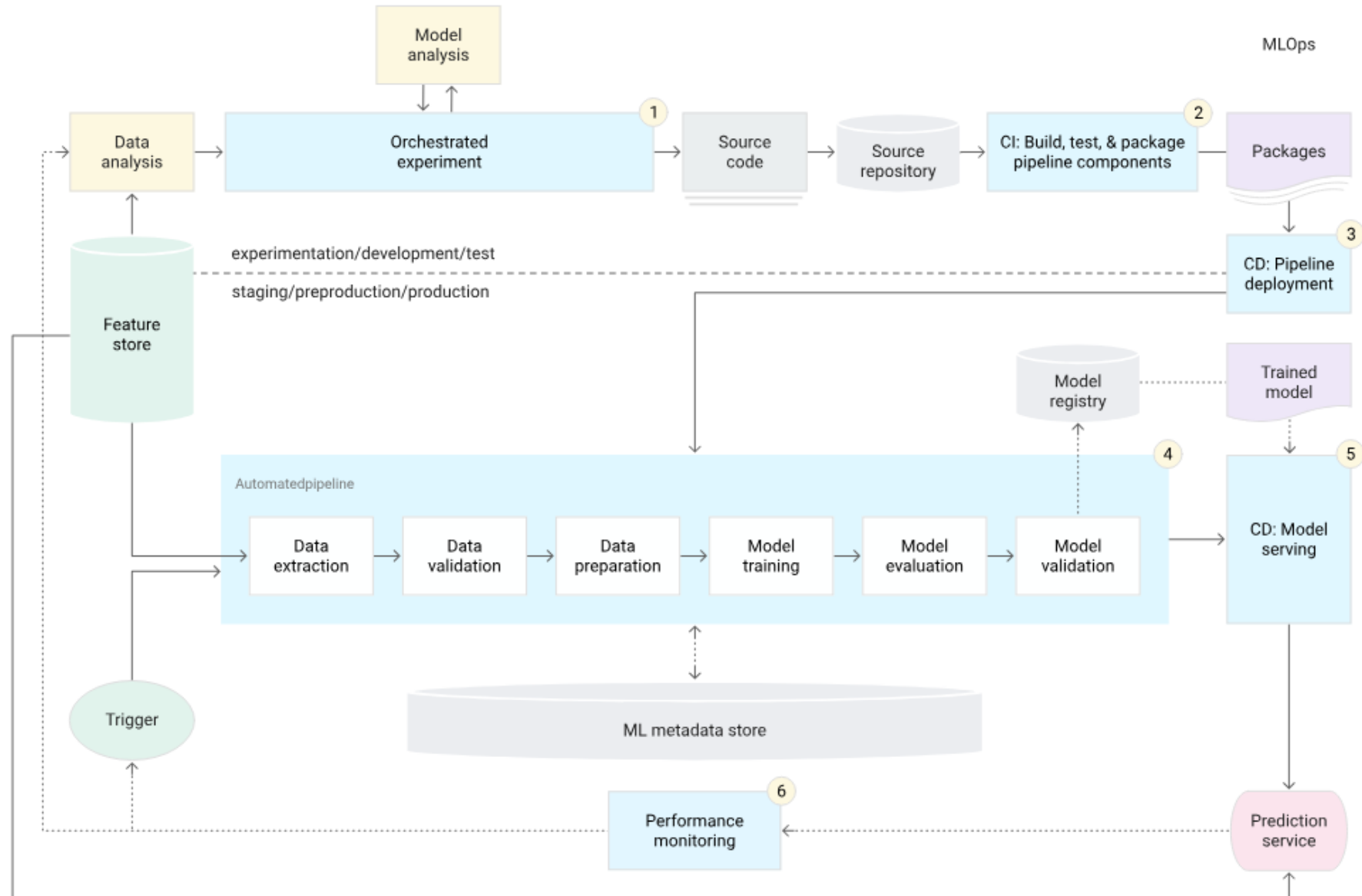
MLOps pipeline: with manual intervention



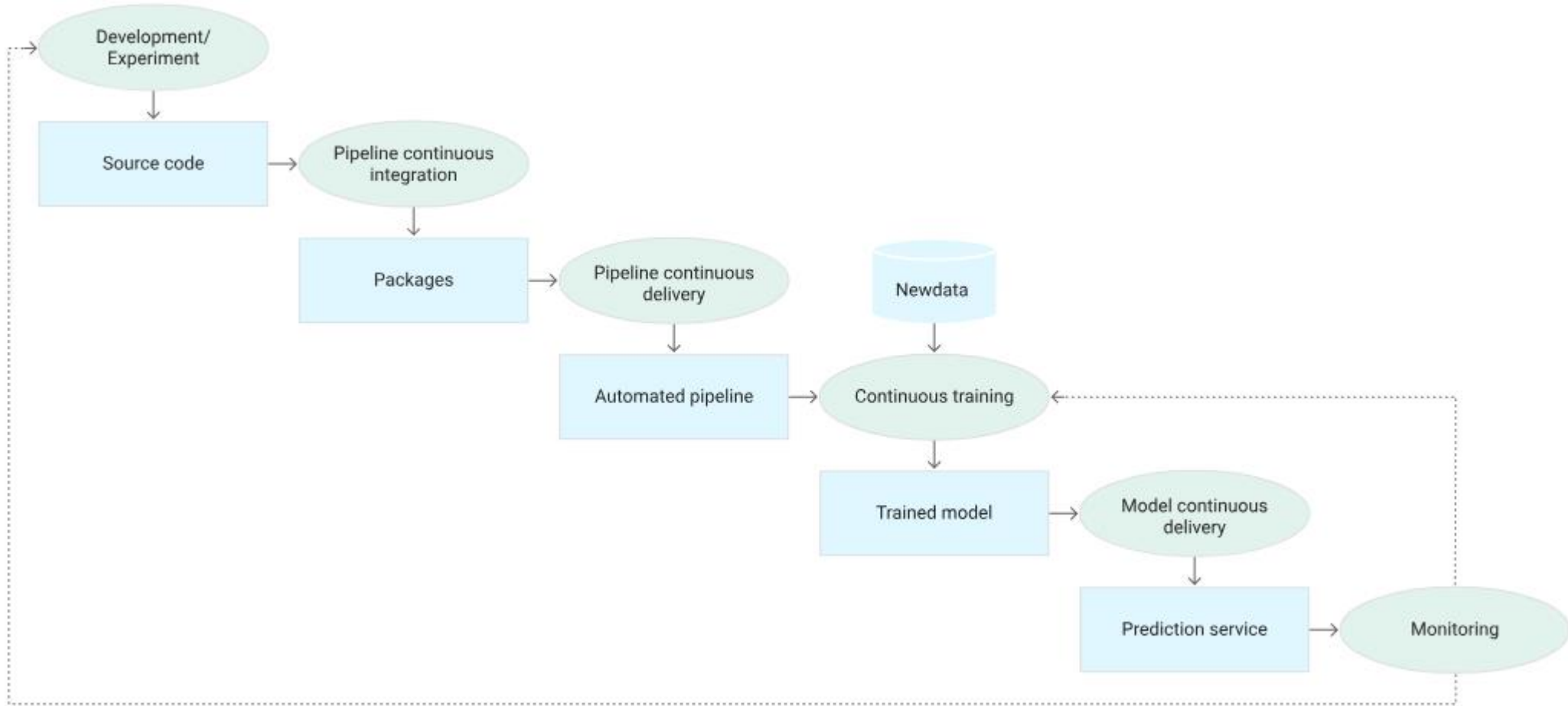
MLOps pipeline: level 1 – ML pipeline automation



MLOps pipeline: level 2 – CI/CD



MLOps pipeline: level 2 – CI/CD



Source: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>

References

- Google Colab: <https://colab.research.google.com/>
- MLOps pipeline: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>

Please, don't forget
to send feedback:

<https://bit.ly/bme-dl>



Thank you for your attention

Dr. Mohammed Salah Al-Radhi
malradhi@tmit.bme.hu

(slides by: Dr. Bálint Gyires-Tóth)

03 September 2024

