Full Stack AI Projects

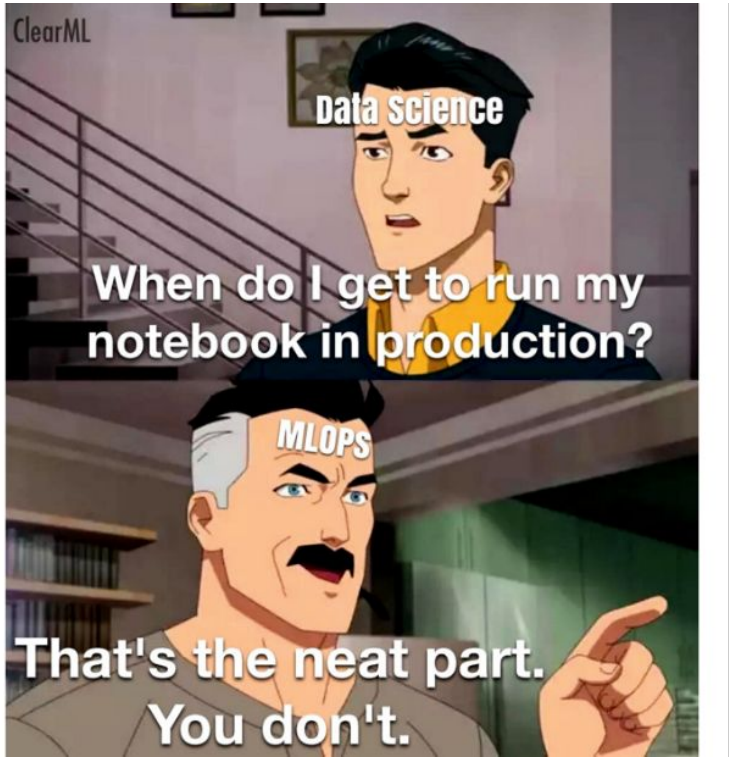
Model Deployment

Henry Ruiz
GDE ML
@devharuiz
https://haruiz.github.io/

What we have discussed so far?

What is machine learning system design?

The process of **defining the <u>interface, algorithms, data, infrastructure, and hardware</u>** for a machine learning system to satisfy specified requirements.

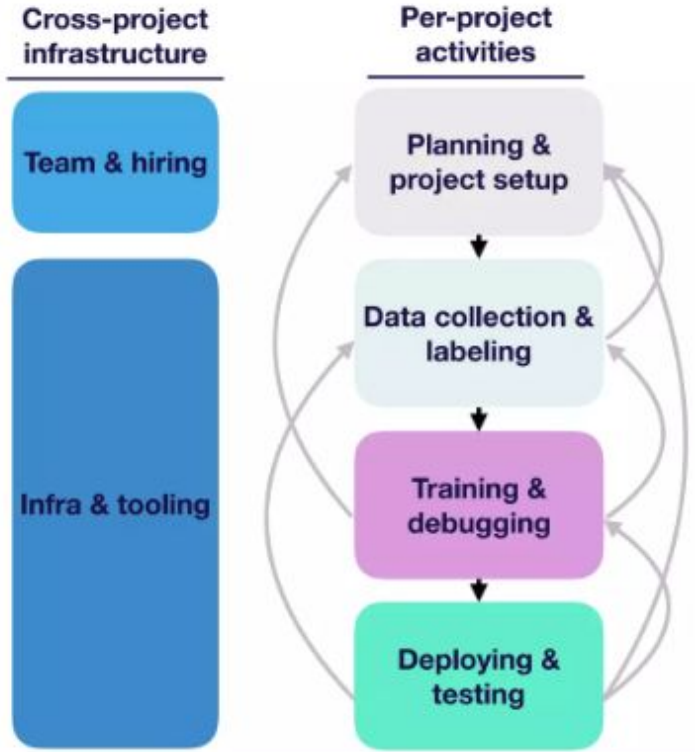


So, What is required to design and build an ML system?

What we have discussed so far?

Define a methodology

Like any other software solution, ML systems require a **well-structured methodology** to maximize the success rate of the implementation.



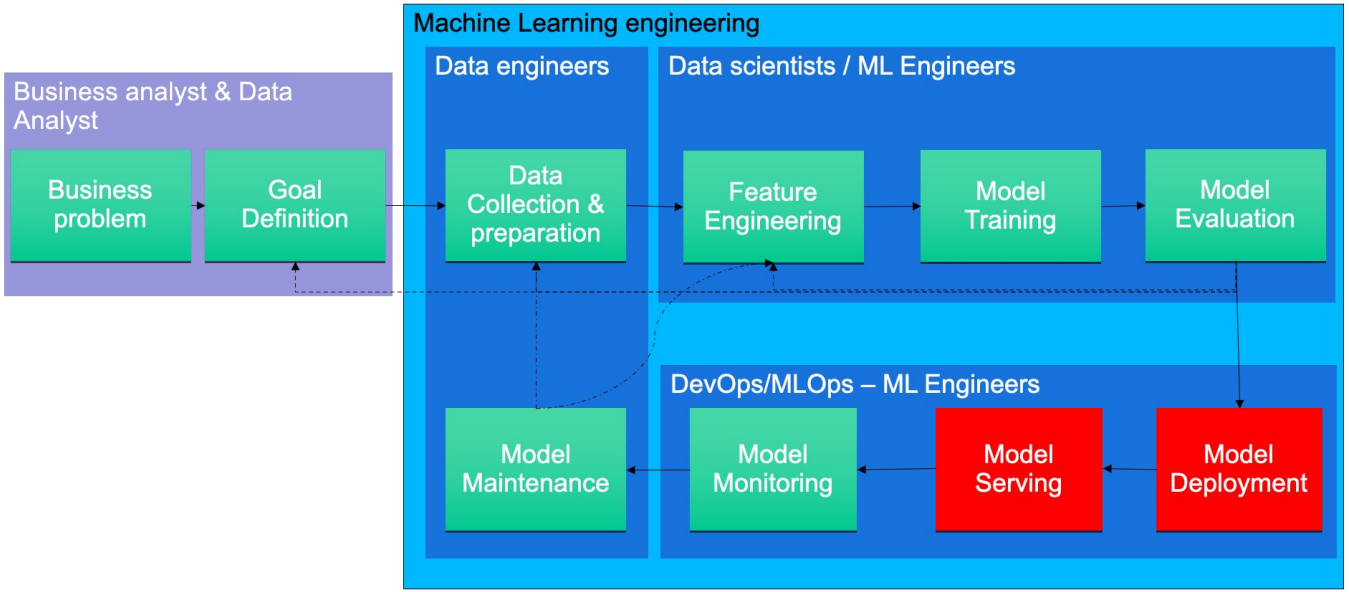
What we have discussed so far?

So, What is required to design and build an ML system?

Set up a team

Managing and leading ML and Data Science teams require unique skills.

ML teams have diverse roles.

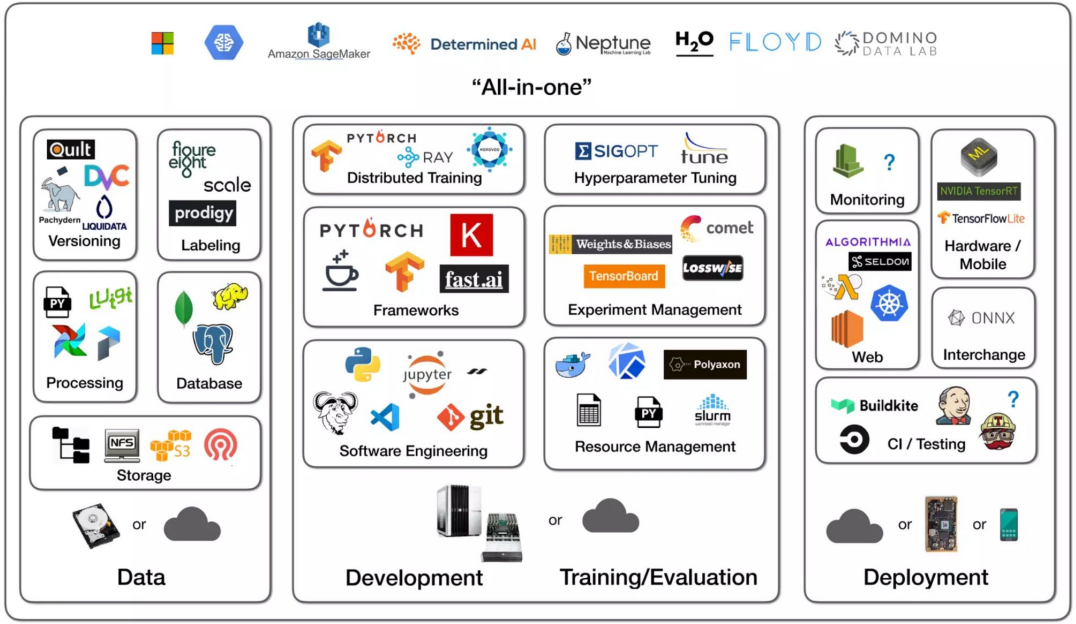


What we have discussed so far?

Define Development infrastructure & Tooling

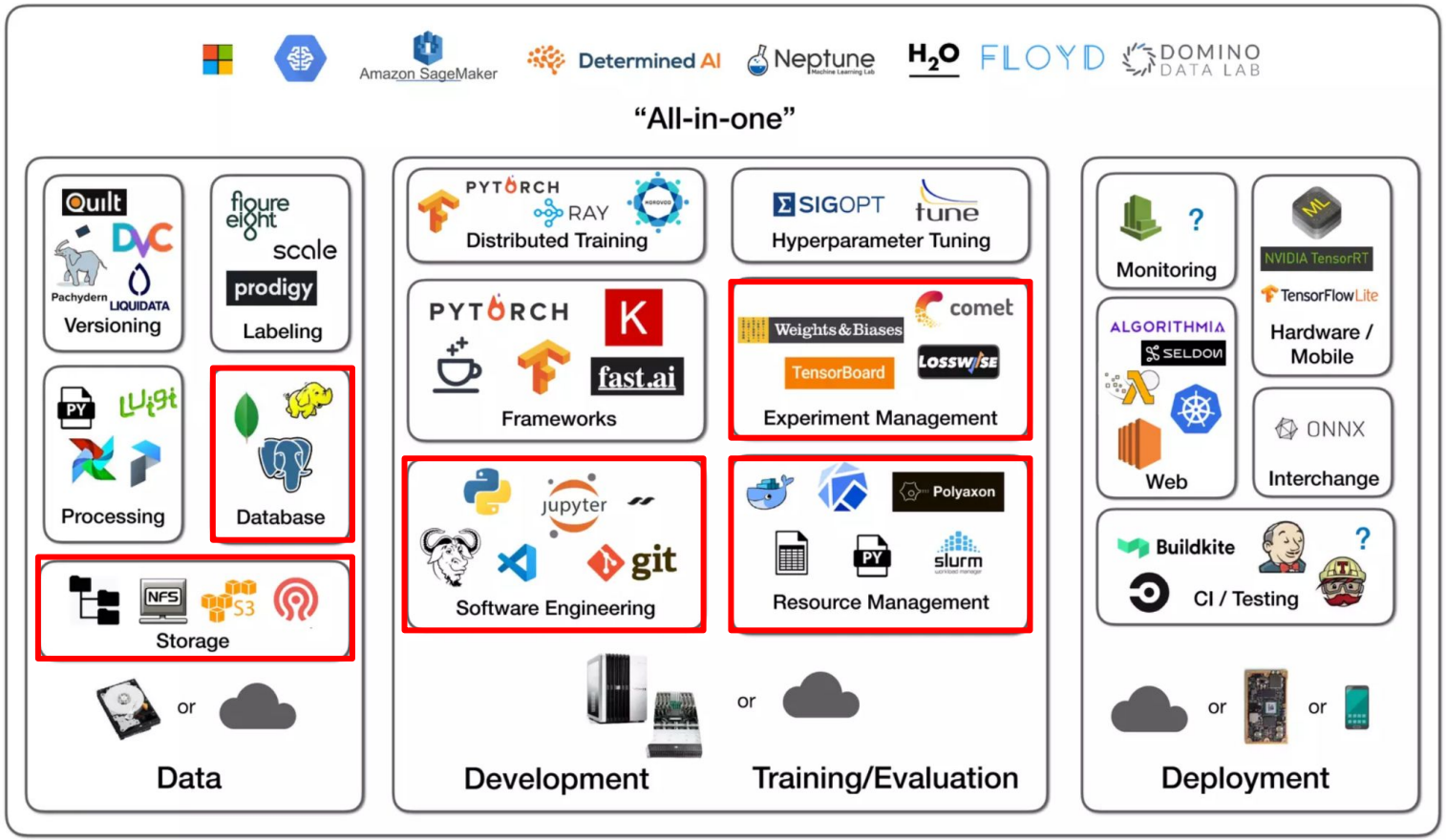
The number of **available tools to work with ML seems endless.**

Selecting the appropriate tools depends on:
the kind of problem, **type of solution, deployment scenario, capacity building**, team experience, cost, hardware and software infrastructure, etc.



Data management tools

Credits: https://fullstackdeeplearning.com/



..Last class

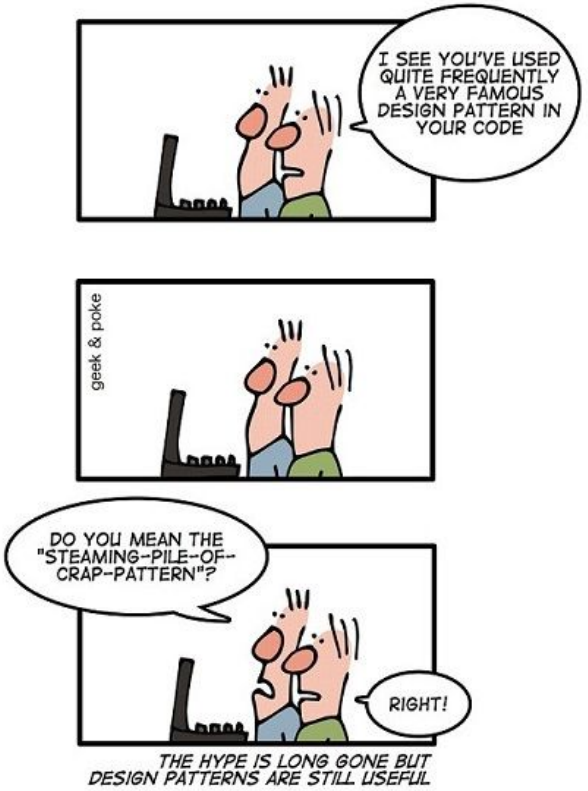
Training, Debugging and Design patterns

What are Design patterns?

In engineering disciplines, design patterns **capture best practices and solutions to commonly occurring problems** to provide **a standard approach to solve them**. They **codify the knowledge and experience of experts** into advice that all practitioners can follow.

Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

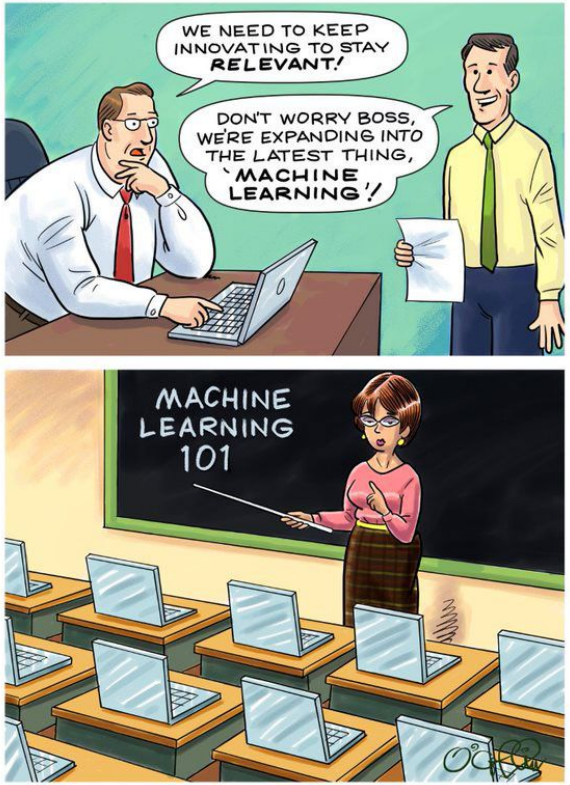
— *A Pattern Language* (Oxford University Press, 1977)



Design Patterns in ML

Developing machine learning models for production is becoming more of an engineering practice, where established ML techniques from research settings are utilized to solve business-related issues.

With the growing popularity of machine learning, it's crucial for professionals to utilize proven methods to tackle recurrent challenges.

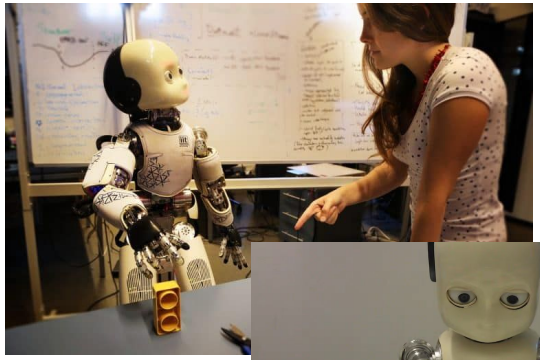


Model's deployment

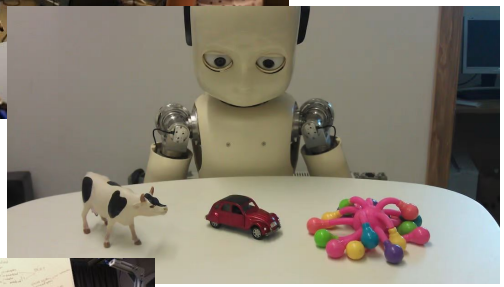
Model deployment is one of the most important steps in the ML pipeline. ***We have spent a lot of time and effort playing around with different models, training and tuning our model hyperparameters, so after evaluating its performance and obtaining that long-awaited score, now is the time to release it to the world, exposing this to real use.***

It sounds like graduation time!!.

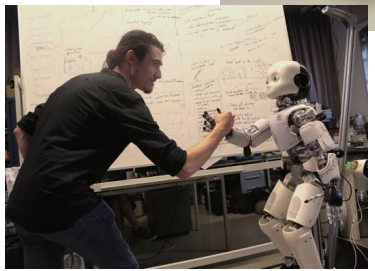
Training



Testing



Evaluation

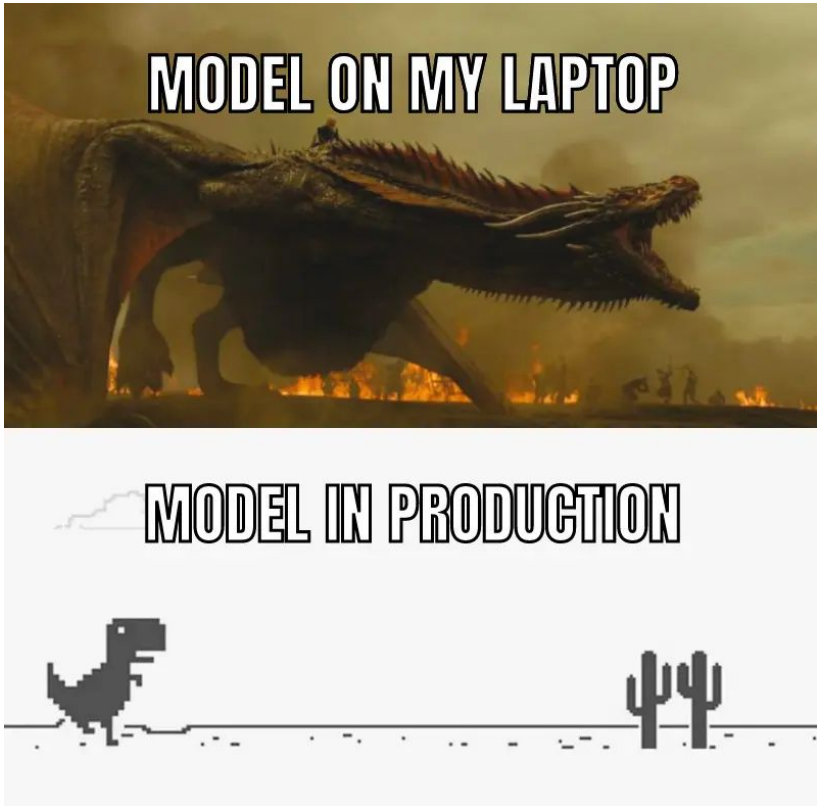


Deployment

Model Deployment

Machine Learning (ML) model deployment refers to the process of making a trained ML model available for use in a production environment. This involves taking a trained ML model and integrating it into a software application or system where it can be used to make predictions or classifications on new data.

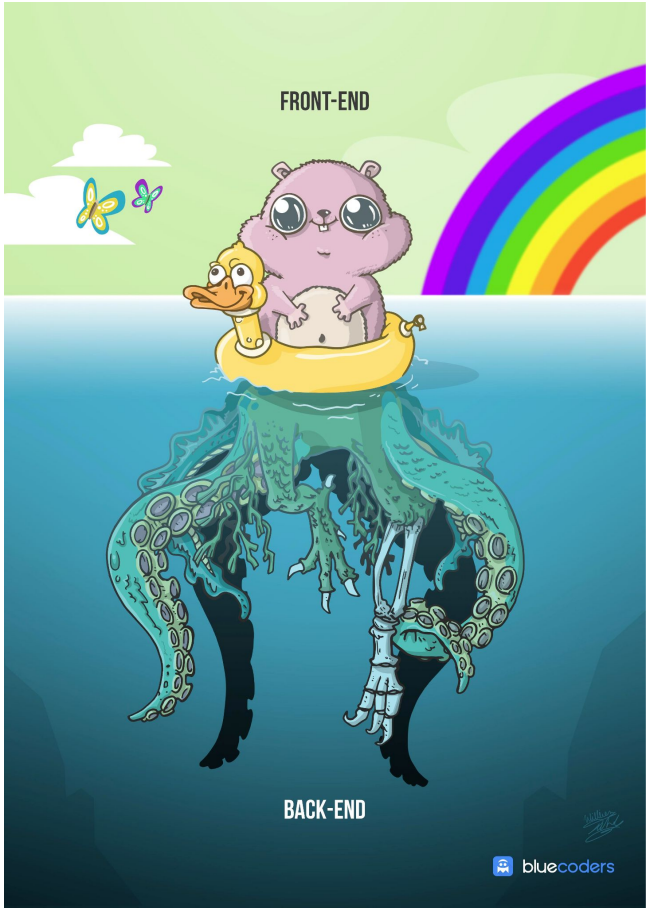
A machine learning model can only begin to add value to an organization when that model's insights routinely become available to the users for which it was built.



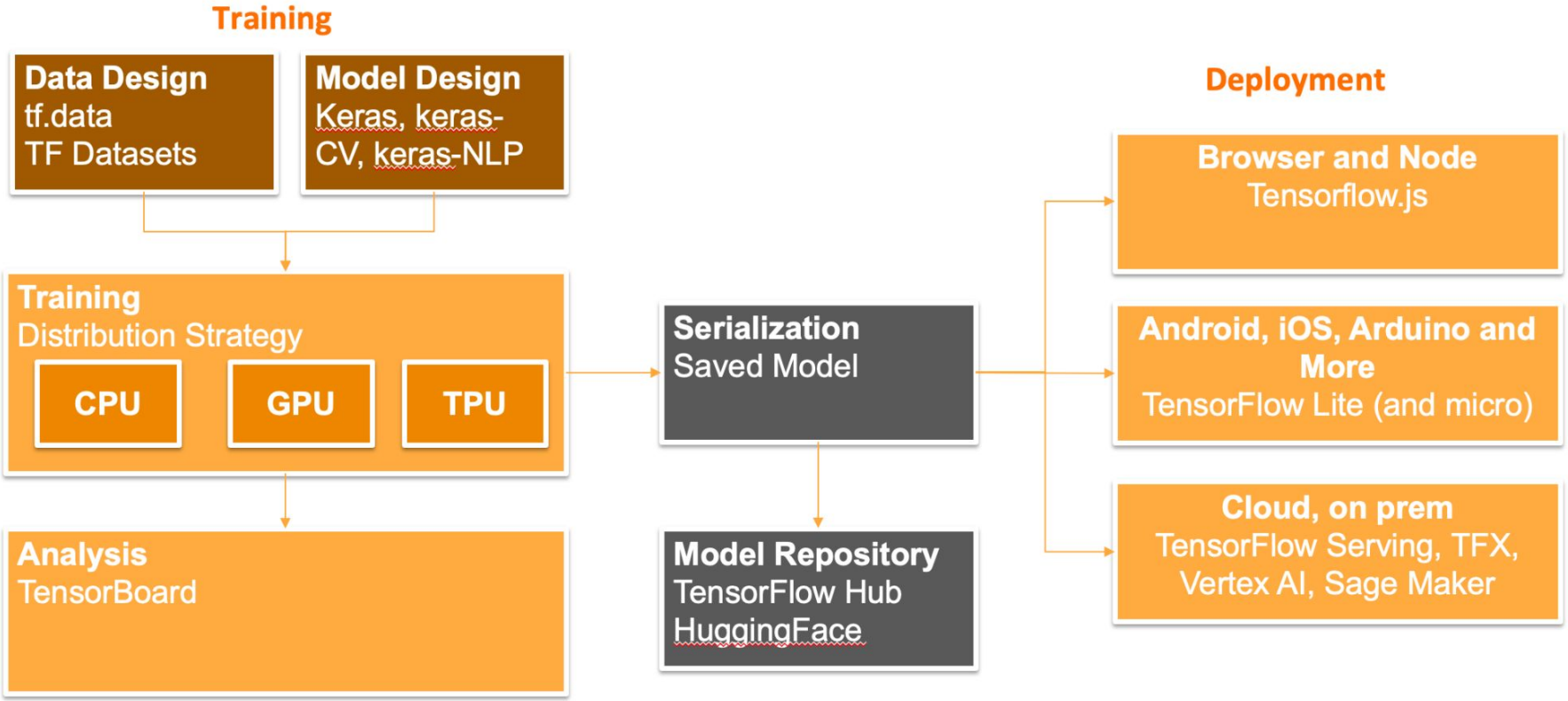
Model Deployment

Deploying an ML model involves several steps, including:

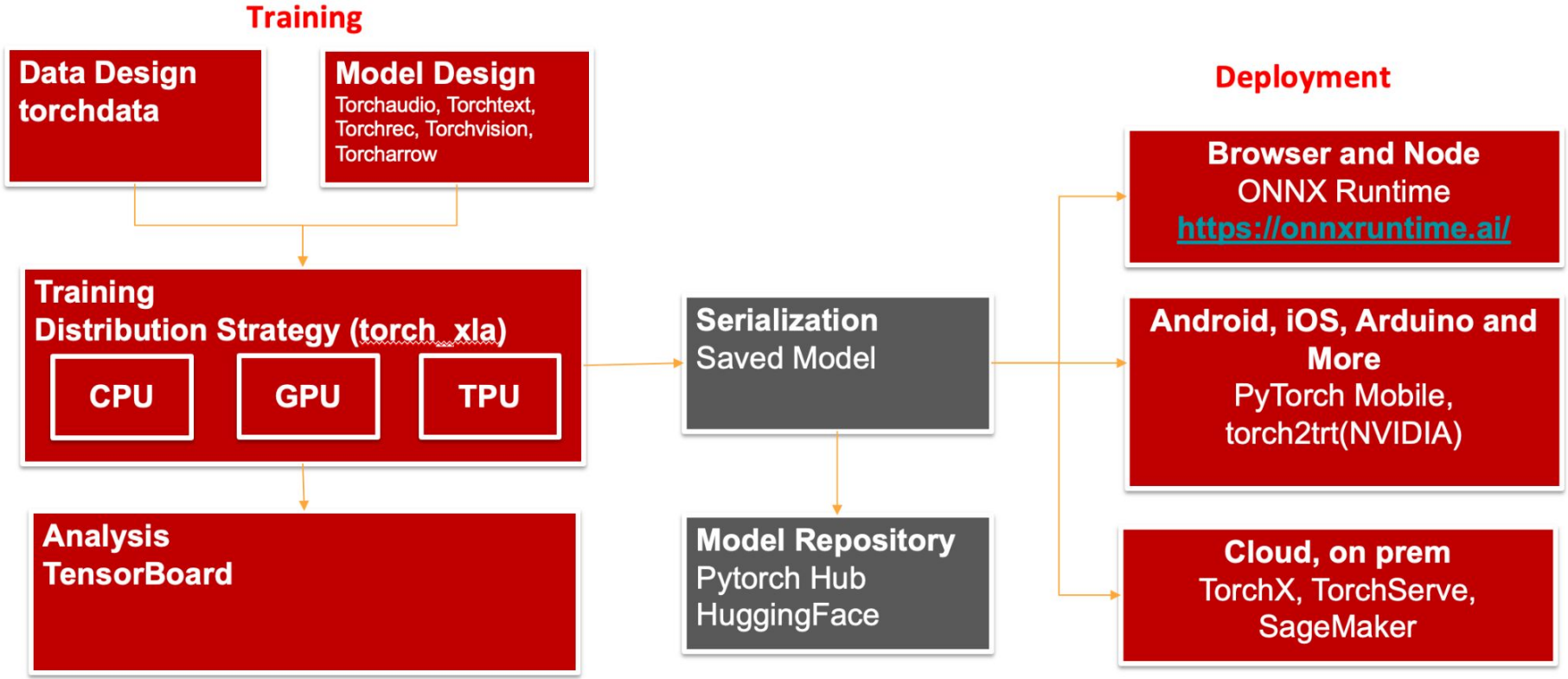
1. Converting the trained model into a format that can be easily loaded and used by the target application or system.
2. Setting up the necessary infrastructure to host the model, including servers, storage, and network connectivity required for the application to run.
3. Developing the interface for users to interact with the ML system.
4. Testing the deployed model to ensure that it works correctly and provides accurate predictions.

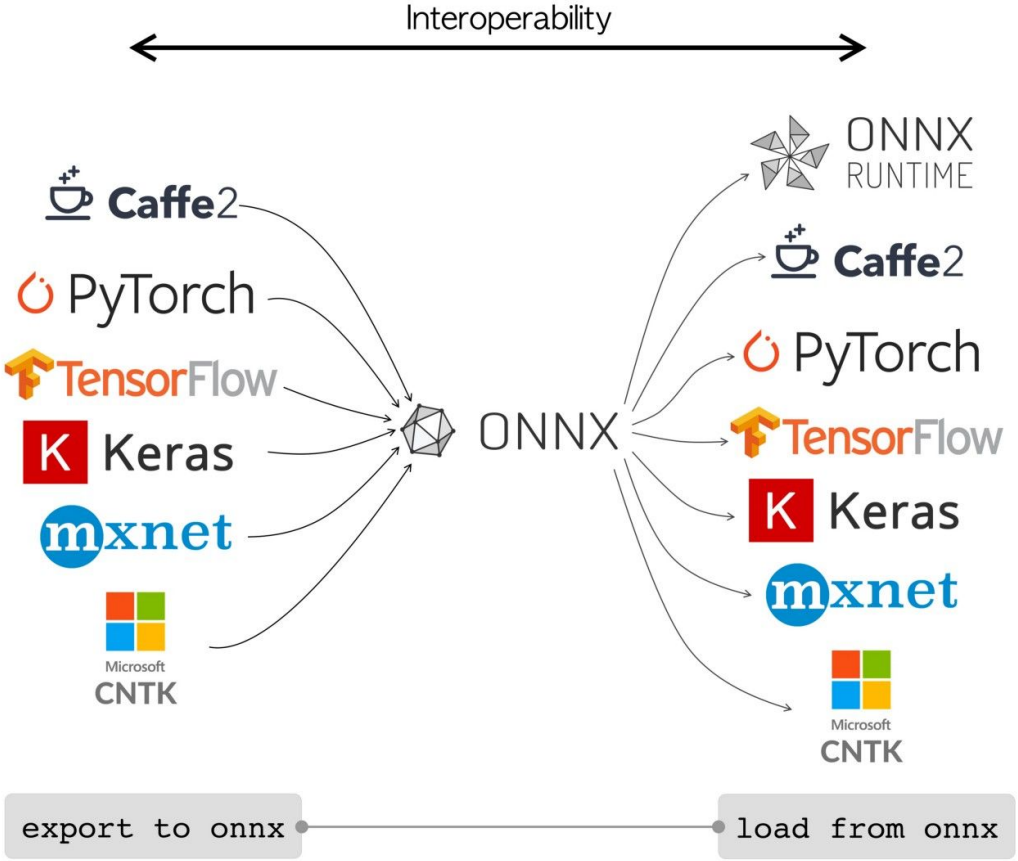


TensorFlow Ecosystem



PyTorch Ecosystem



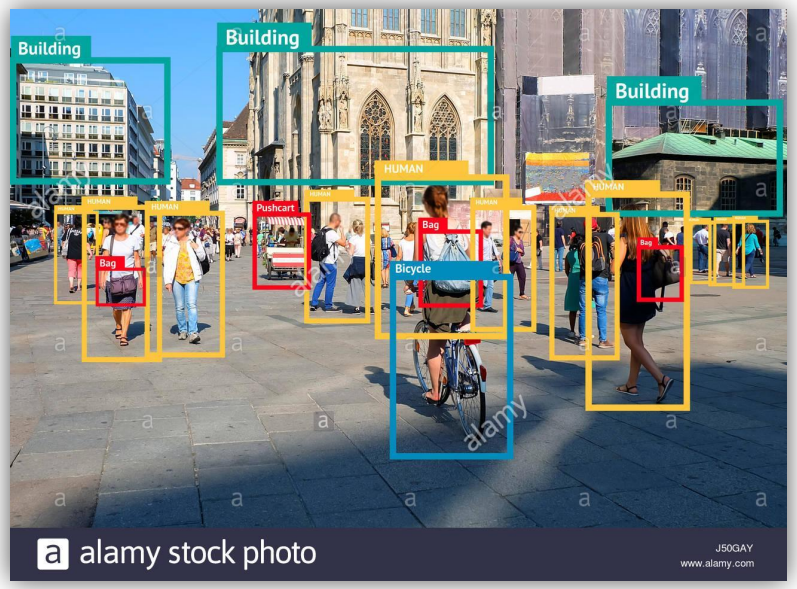

Interoperability
Caffe2
PyTorch
TensorFlow
Keras
mxnet
Microsoft
CNTK
ONNX
ONNX
RUNTIME
Caffe2
PyTorch
TensorFlow
Keras
mxnet
Microsoft
CNTK
export to onnx
load from onnx

AI at the edge

What is AI at Edge

- The edge means local (or near local) processing
- Not Just Anywhere in the cloud
- NO need to send data to the cloud
- Edge applications are often used:
  - where low latency is necessary
  - Where a network may not always be available
  - Real-time decision making(autonomous driving)

Applications



Edge vs. Cloud

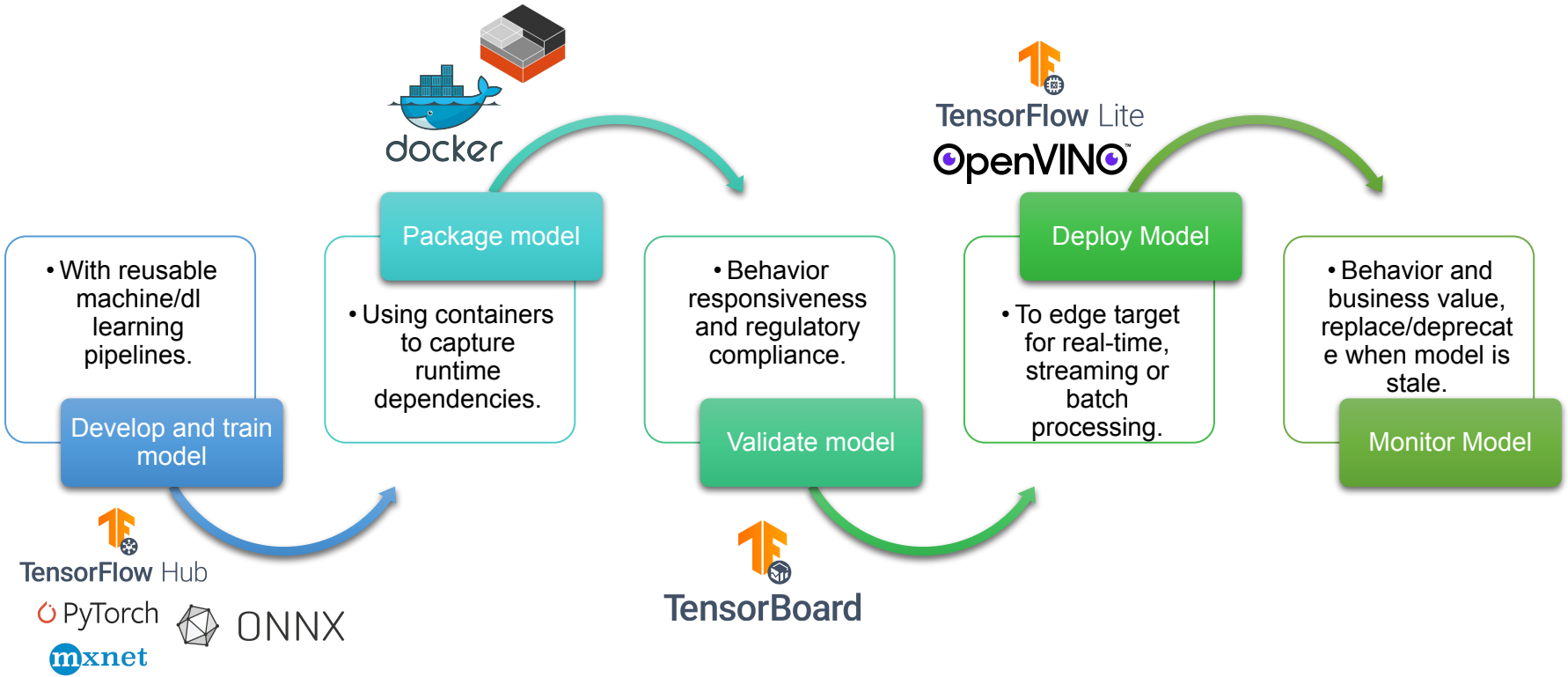
Why deploy AI on the edge?

- Near real time decision making
- Security concerns
- Avoid significant data transfer costs to the cloud
- Heterogeneous execution across hardware

Why deploy AI on the cloud?

- Large scale parallel compute for training models
- Amount of frameworks supported
- Streamlined experience for training models
  - Not software downloads
  - Not configuration
  - No installations

AI from development to deployment



Hardware

CPU (Central Processing Unit)

- A **CPU** or **Central** electronic circuitry that executes the instructions of a computer program.

IGPU (*integrated GPU)*

- A GPU that is located on a processor alongside the CPU cores and shares memory with them.

Vision Processing Units (VPUs)

- are accelerators that are specialized for AI tasks related to computer vision—such as Convolutional Neural Networks (CNNs) and image processing.

FPGAS - Application-Specific Integrated Circuits (ASICs)

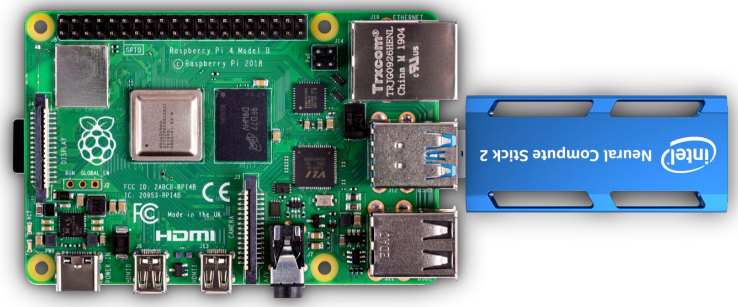
- *are chips that are hardwired during manufacturing in order to be optimally efficient for a specific need.*

Optimization techniques

Quantization	Model compression	Model Pruning	Replacing inefficient layers	Knowledge distillation
High precision weights are converted to low precision weights The ability to lower the precision of a model from FP32 to INT8	Model size is reduced by reducing the number of weights that need to be stored.	Neurons or connections between neurons are removed in the model	A computationally expensive layer is replaced with a computationally simple one	A larger "teacher" model trains a smaller model.

Train a model or find a trained one → Run the model optimizer → Intermediate representation → Deploy using the inference engine

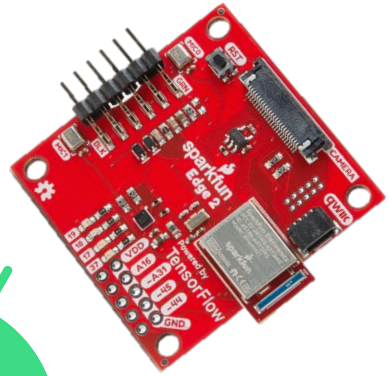
```
python3 mo_tf.py --input_model <INPUT_MODEL>.pb
```

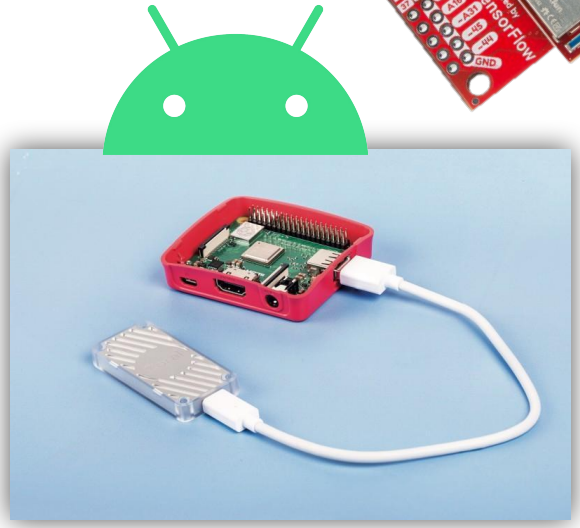


OpenVINO™

Train a model or find a trained one → Convert the model → tflite model → Deploy at the edge

```python
import tensorflow as tf

# Construct a basic model.
root = tf.train.Checkpoint()
root.v1 = tf.Variable(3.)
root.v2 = tf.Variable(2.)
root.f = tf.function(lambda x: root.v1 * root.v2 * x)

# Save the model in SavedModel format.
export_dir = "/tmp/test_saved_model"
input_data = tf.constant(1., shape=[1, 1])
to_save = root.f.get_concrete_function(input_data)
tf.saved_model.save(root, export_dir, to_save)

# Convert the model.
converter = tf.lite.TFLiteConverter.from_saved_model(export_dir)
tflite_model = converter.convert()

# Save the TF Lite model.
with tf.gfile.GFile('model.tflite', 'wb') as f:
  f.write(tflite_model)
```

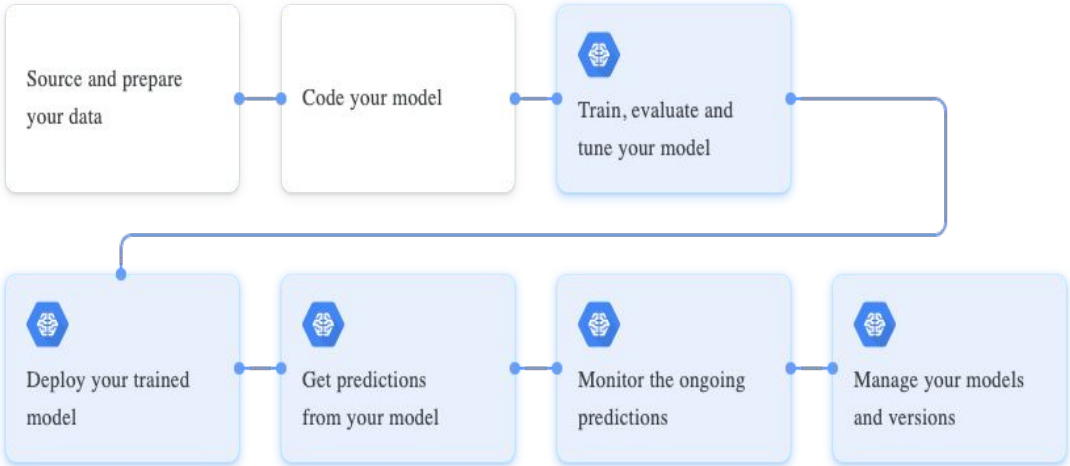


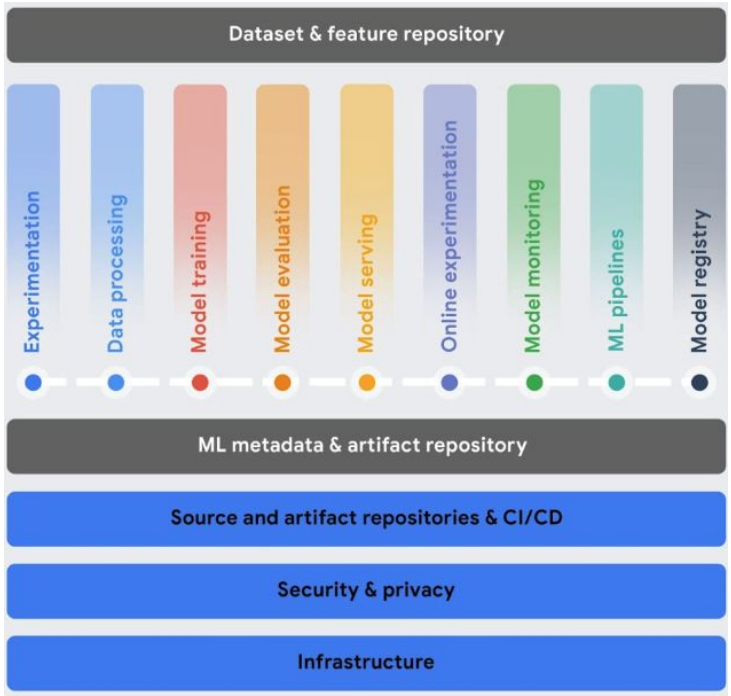


TensorFlow Lite

AI at the cloud

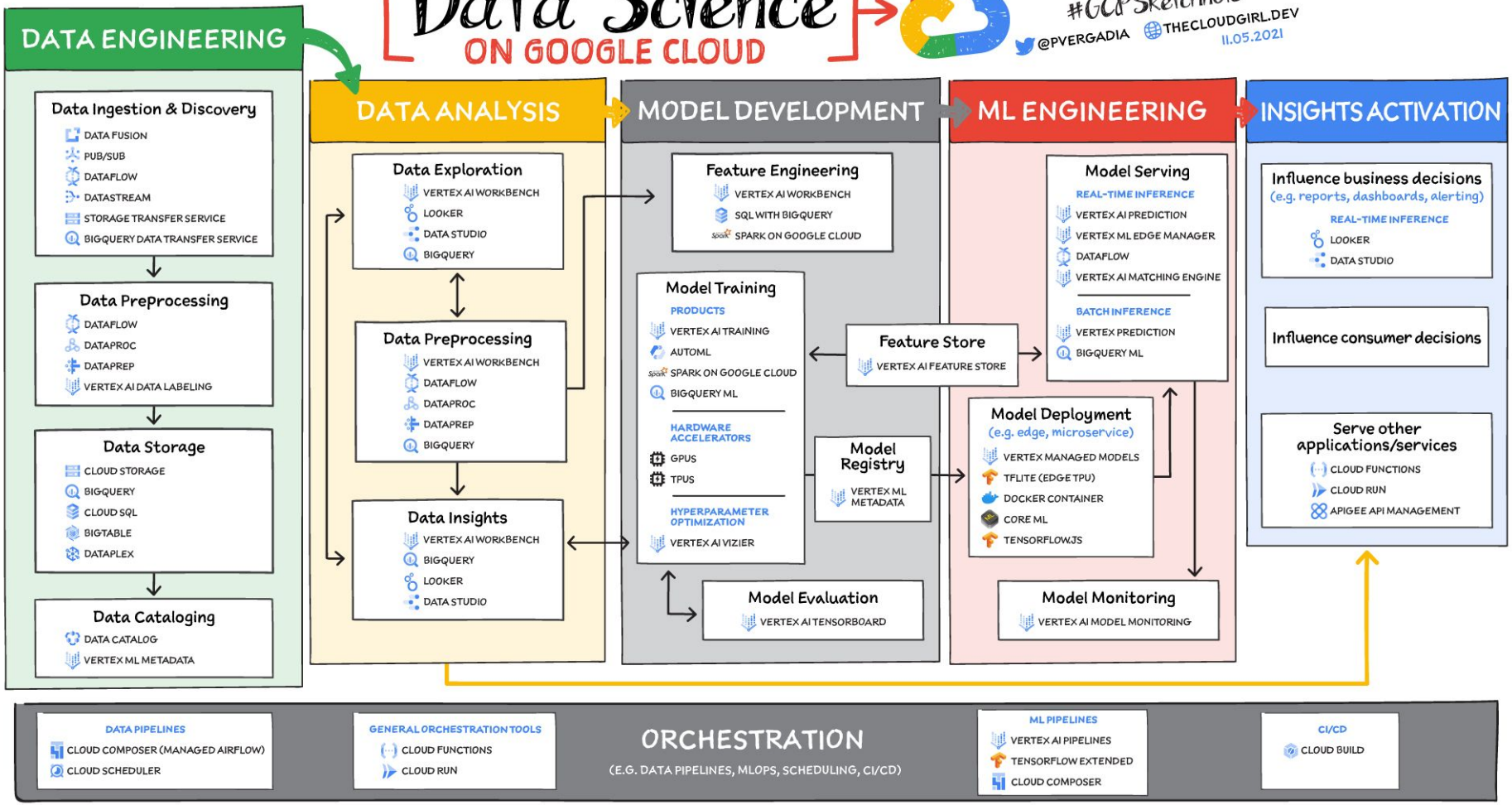
GCP ecosystem for AI



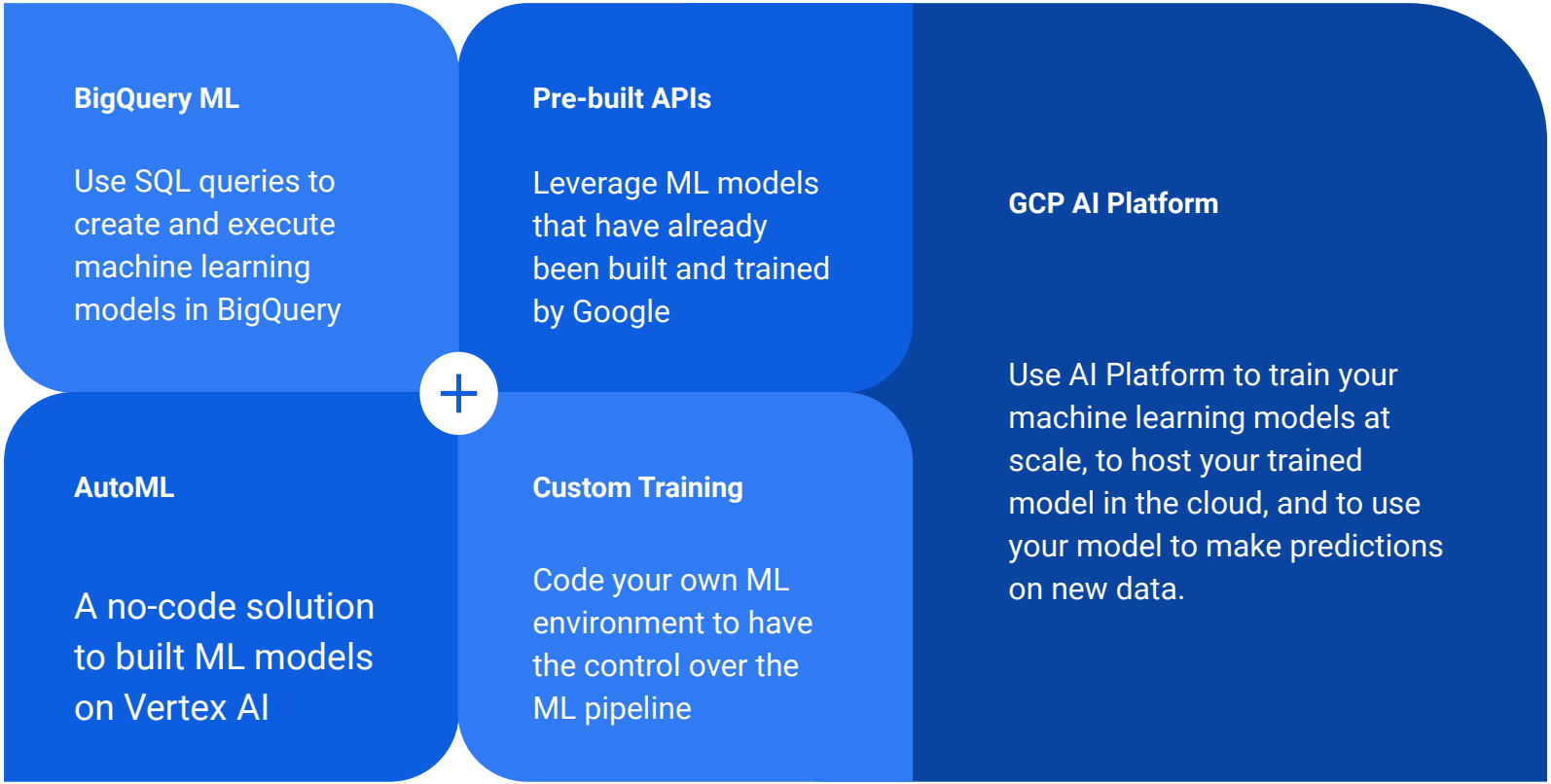


Data Science ON GOOGLE CLOUD

#GCPSketchnote
@PVERGADIA THECLOUDGIRL.DEV
11.05.2021



GCP ecosystem for AI



GCP ecosystem for AI

	BigQuery ML	Prebuilt APIs	Auto ML	Custom Training
Data type	Tabular	Tabular, image, text and video	Tabular, image, text and video	Tabular, image, text and video
Training data size	Medium to Large	No data required	Small to Medium	Medium to Large
ML and Coding expertise	Medium	Low	Low	High
Flexibility to tune hyperparameters	Medium	None	None	High
Time to train a model	Medium	None	Medium	Long

MLOps