TensorFlow 2.0 Tutorial: Part #1

TensorFlow behind the scenes

Iran University of Science and Technology (IUST)
Department of Computer Engineering

TensorFlow behind the scenes

TensorFlow behind the scenes

What is TensorFlow?

- Open Source Machine Learning Library, Apache 2.0 License

What is TensorFlow?

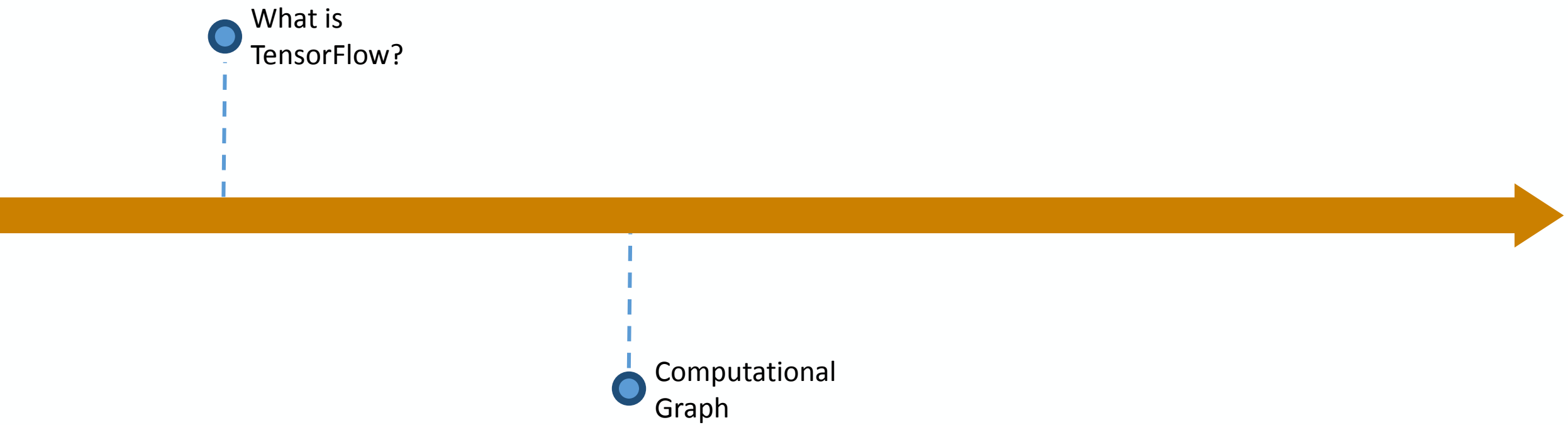
- Open Source Machine Learning Library, Apache 2.0 License
- Useful for Deep Learning

What is TensorFlow?

- Open Source Machine Learning Library, Apache 2.0 License
- Useful for Deep Learning
- Research and Production

TensorFlow behind the scenes

TensorFlow behind the scenes



Motivation

$$(a+b)\times(c+d)$$

- Sequential → (a + b)

Motivation

$$(a+b)\times(c+d)$$

- Sequential → (a + b) (c + d)

Motivation

$$(a+b)\times(c+d)$$

- Sequential → (a + b) × (c + d)

Motivation

$$(a+b)\times(c+d)$$

- Sequential → $\underbrace{(a + b)}_{} \times \underbrace{(c + d)}_{}$

INDEPENDENT!

Motivation

$$(a+b)\times(c+d)$$

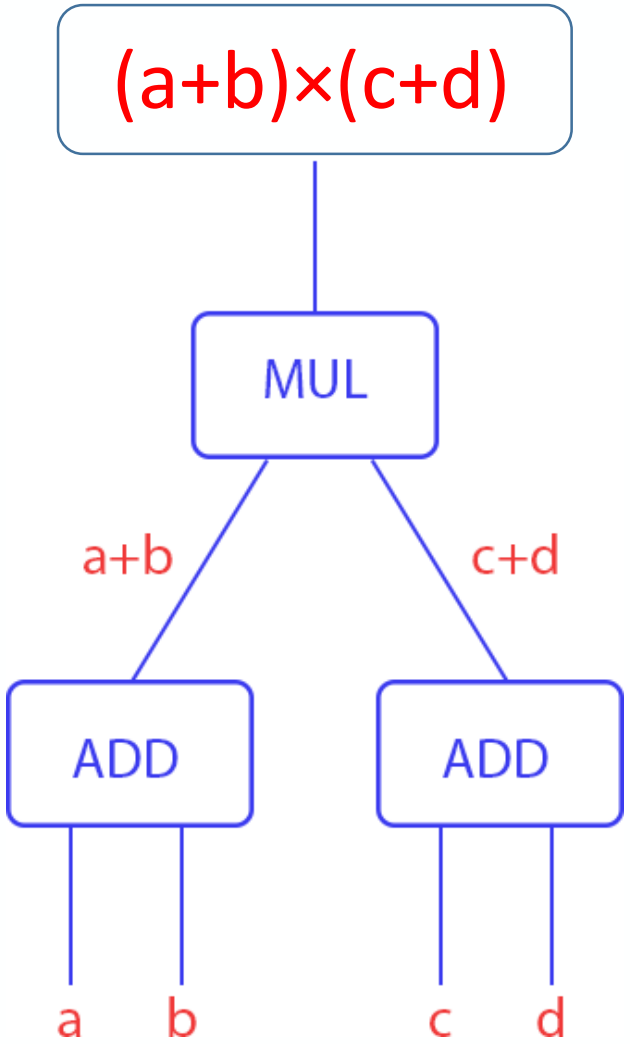
- Sequential → (a + b) × (c + d)
- Parallel → (a + b), (c + d)

Motivation

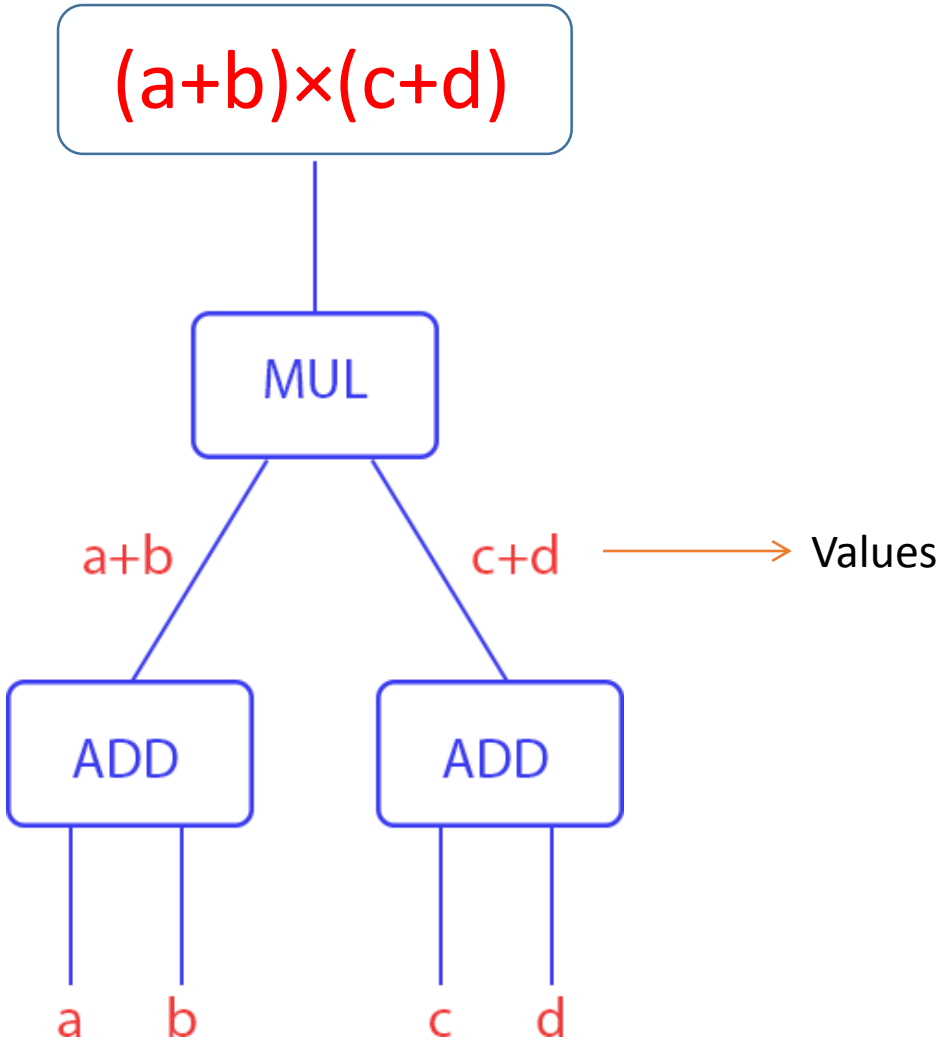
$$(a+b)\times(c+d)$$

- Sequential → (a + b) × (c + d)
- Parallel → (a + b) × (c + d)

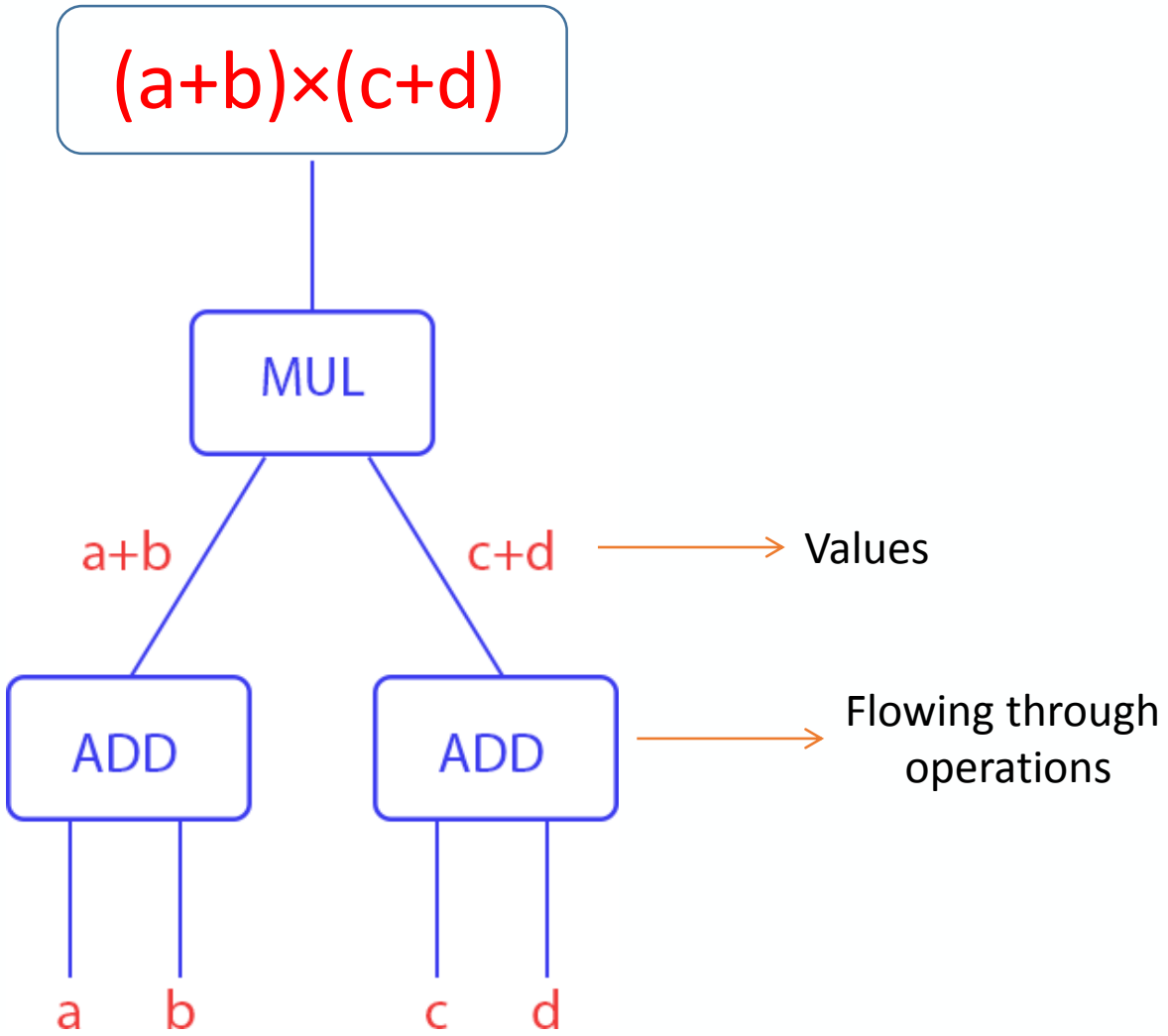
Computational Graph



Computational Graph



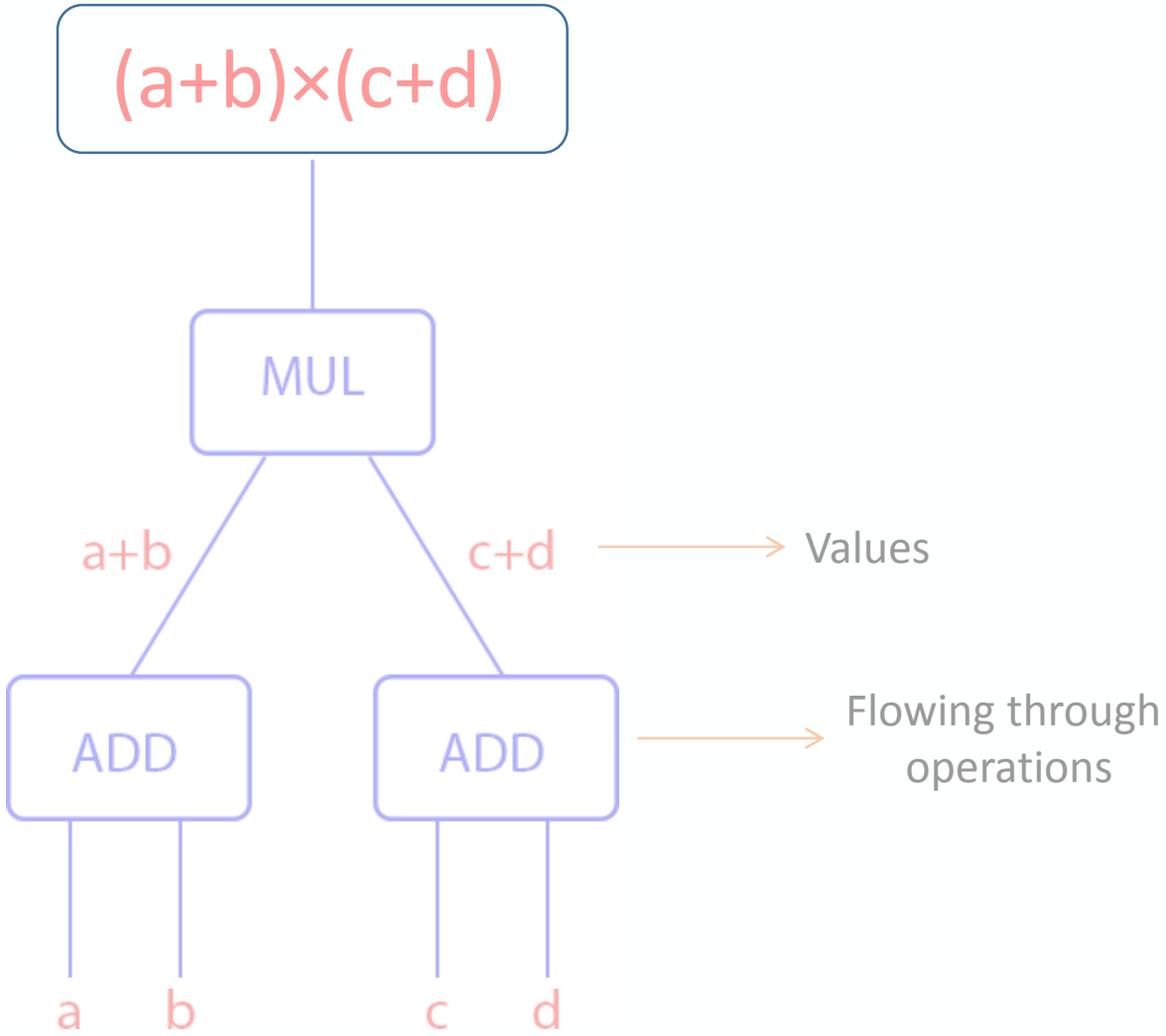
Computational Graph



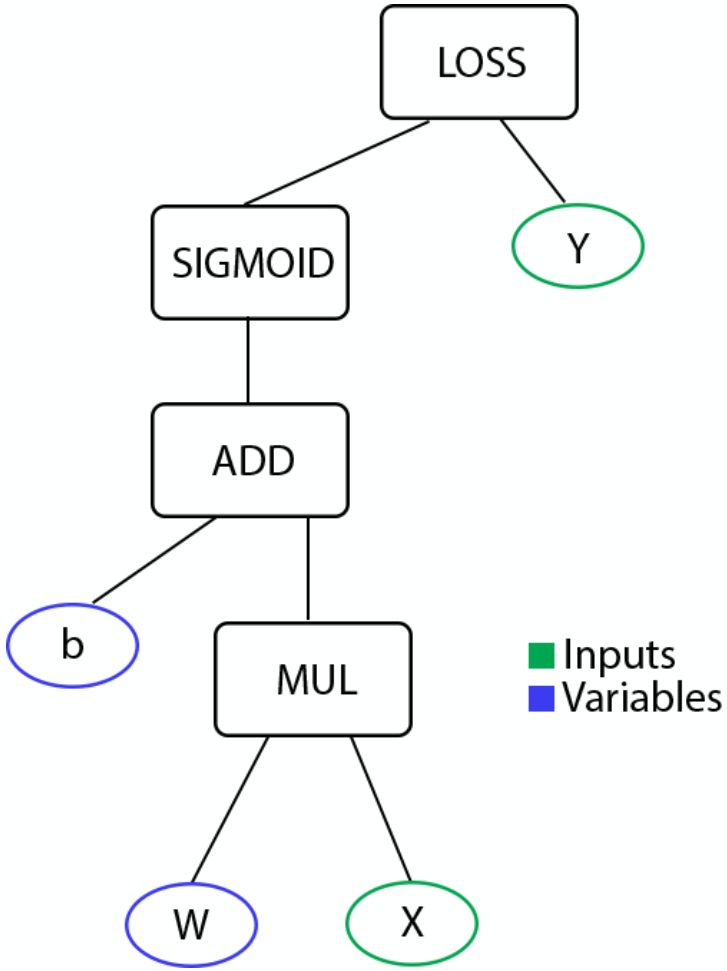
Computational Graph

TensorFlow

BAM!

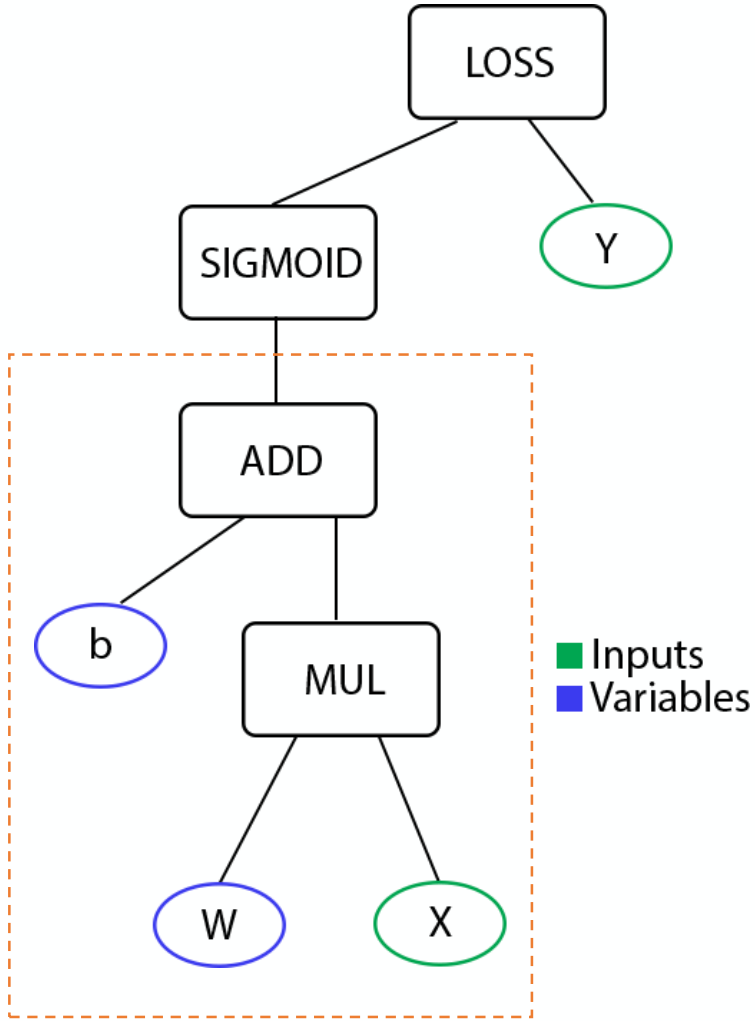


Example: Logistic Regression

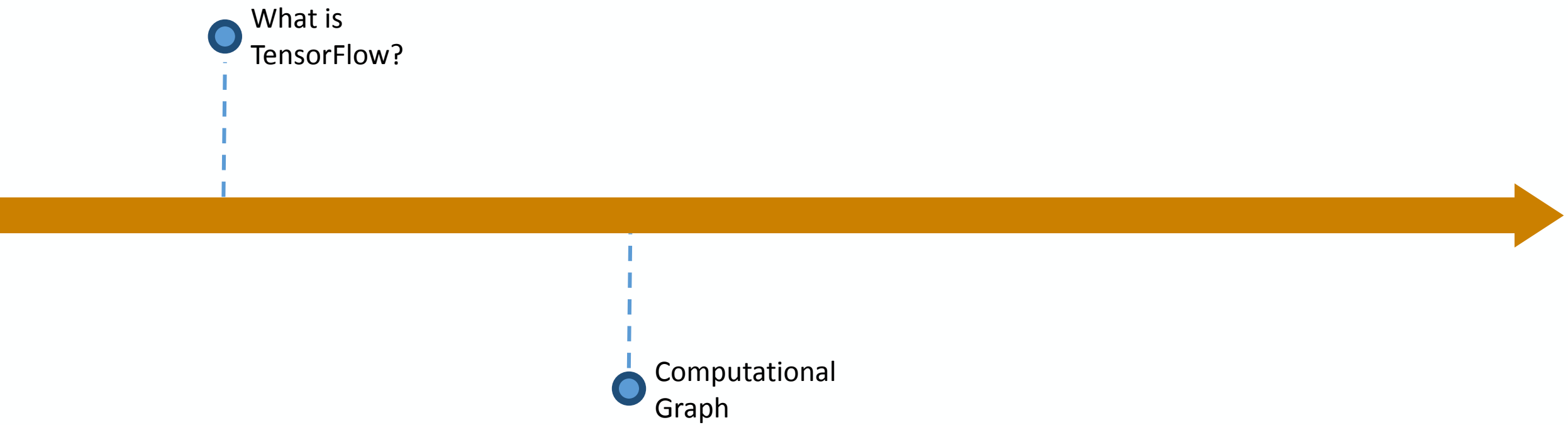


Example: Logistic Regression

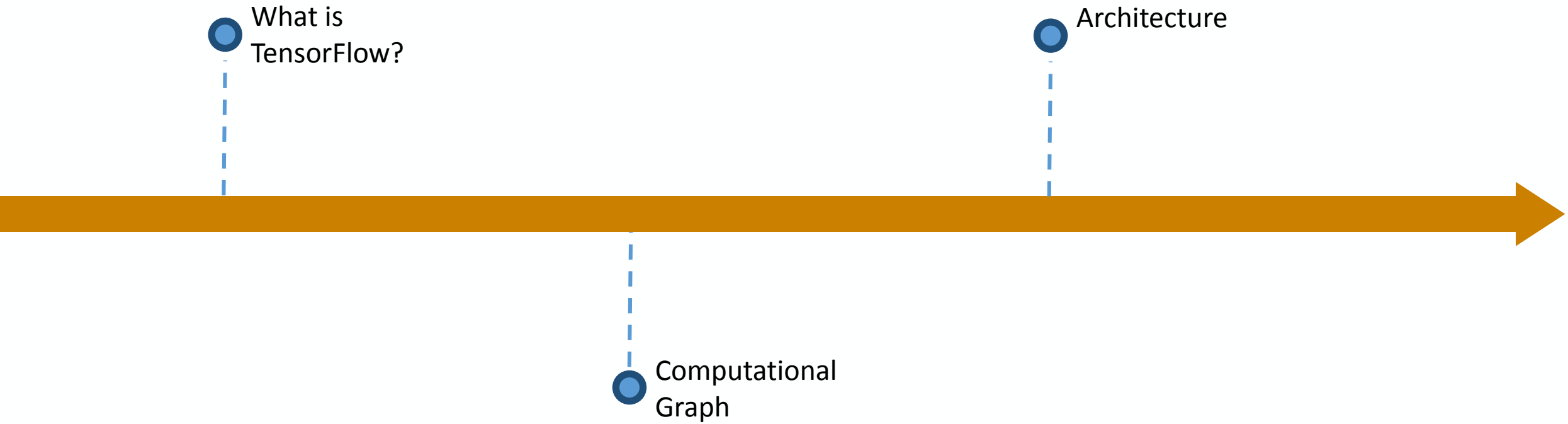
Keras Dense Layer



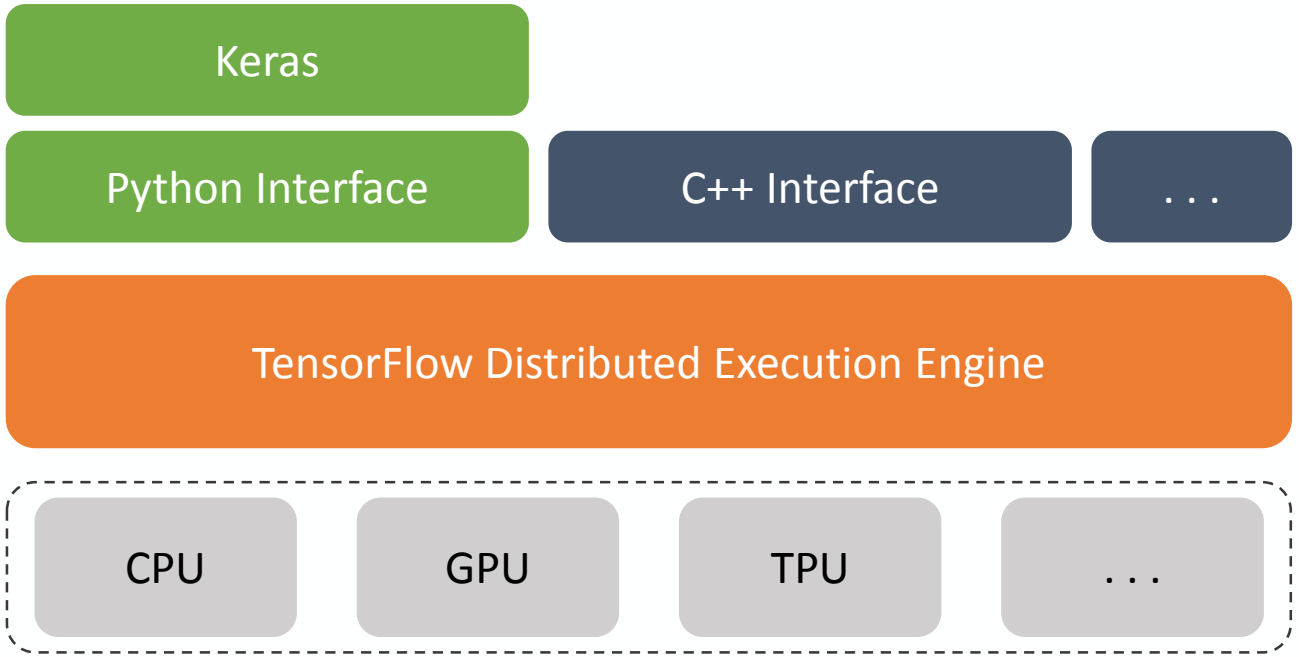
TensorFlow behind the scenes



TensorFlow behind the scenes

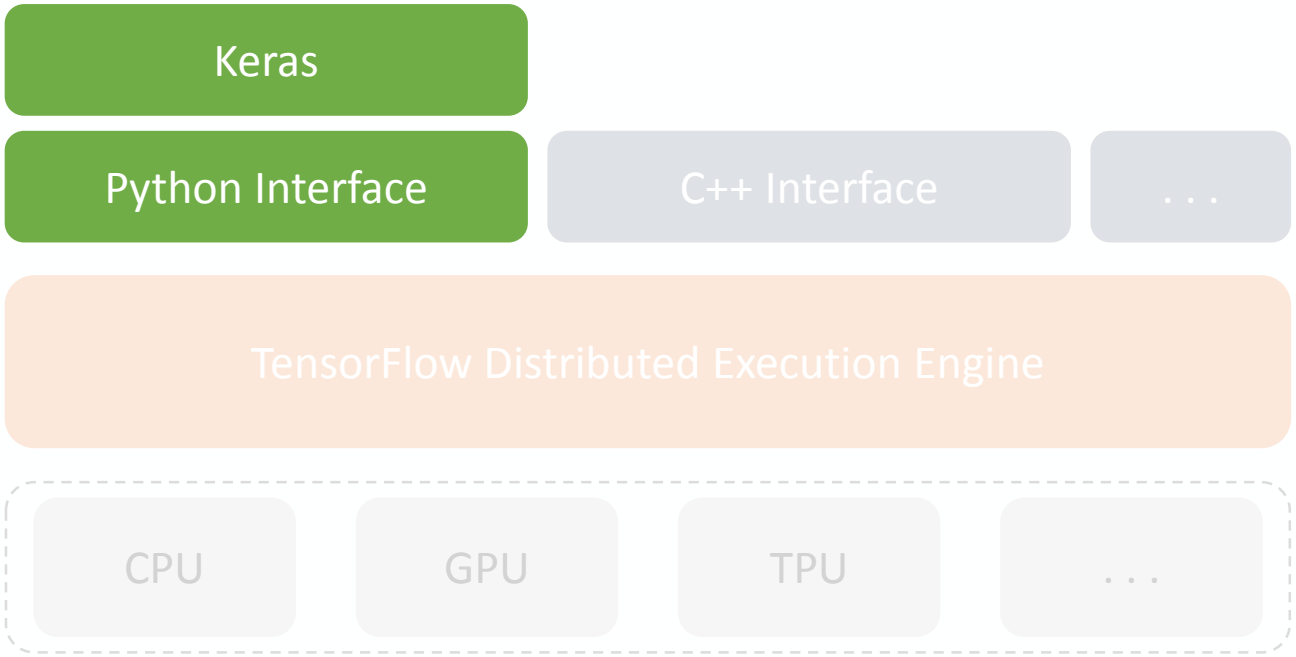


Overview

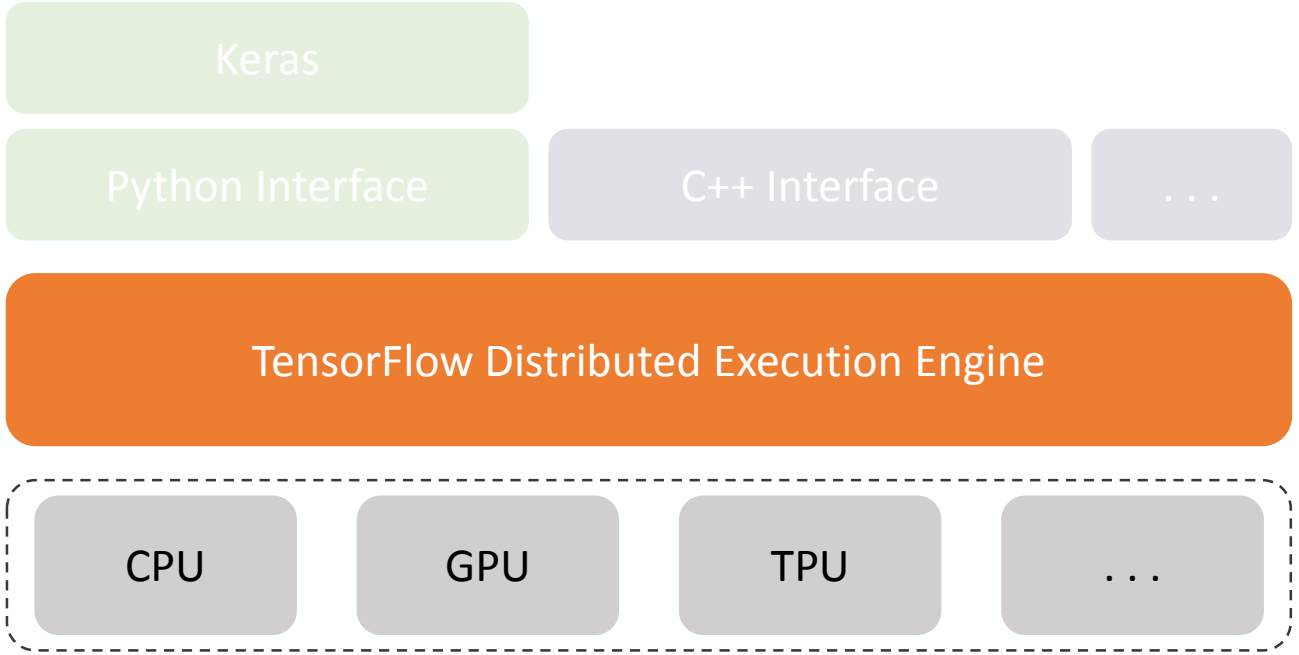

Keras
Python Interface
C++ Interface
. . .
TensorFlow Distributed Execution Engine
CPU
GPU
TPU
. . .

Overview

What we cover in this course



Overview



First lets gain some insight about this part

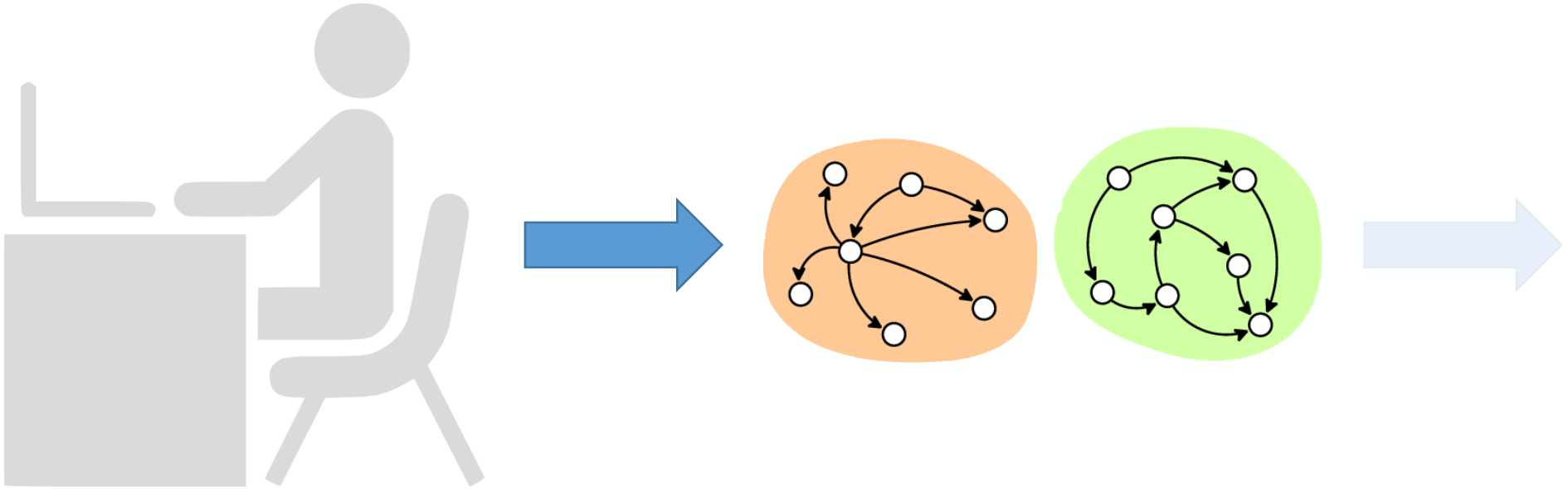
Under The Hood

1. High Level API

Programmer makes the computational graph using the high level API

Under The Hood

2. Distributed Master

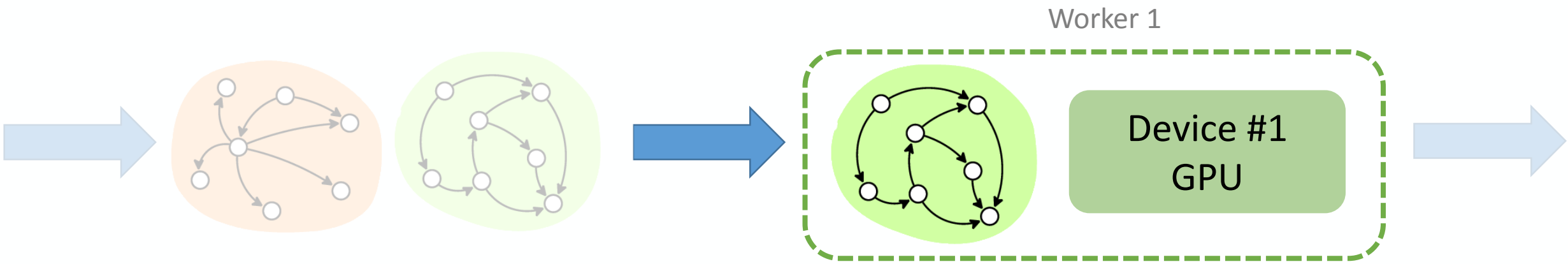


The Master agent:

1. Prunes the graph to obtain the subgraph required to evaluate the requested nodes
2. Finds graph pieces which are independent and can be evaluated in parallel

Under The Hood

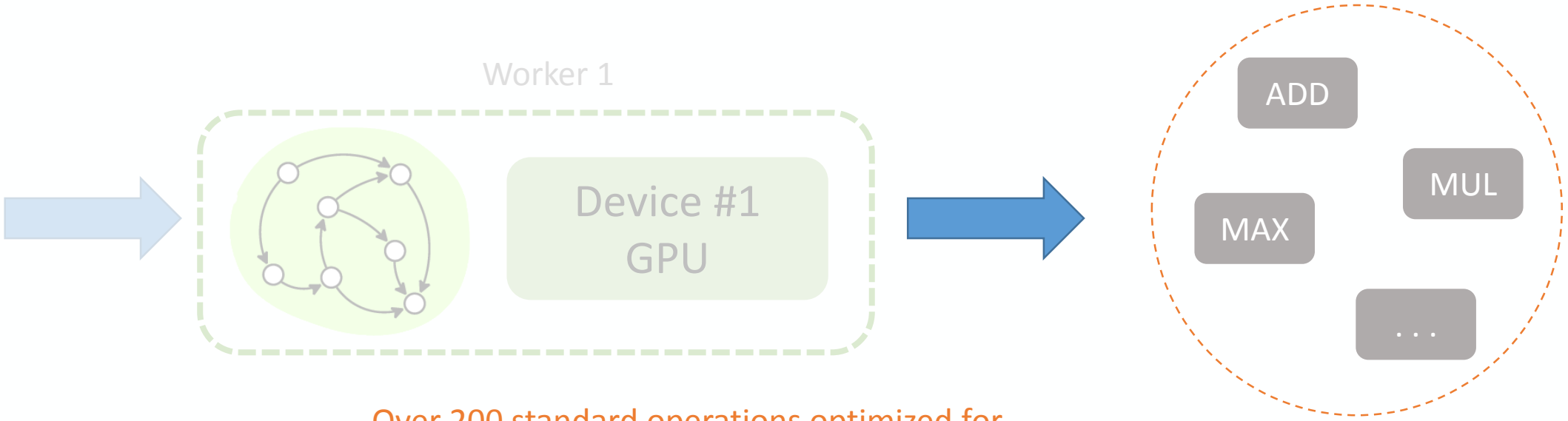
3. Worker Service



Schedules the execution of the kernels for the operations that comprise a local subgraph

Under The Hood

4. Kernel Implementations



Over 200 standard operations optimized for
a variety of devices (C++ for CPU, CUDA for GPU, …) :

- Mathematical
- array manipulation
- control flow
- state management

Thank you!