Deep Learning Compilers

Abhinav Bhatele, Daniel Nichols

UNIVERSITY OF MARYLAND

Announcements

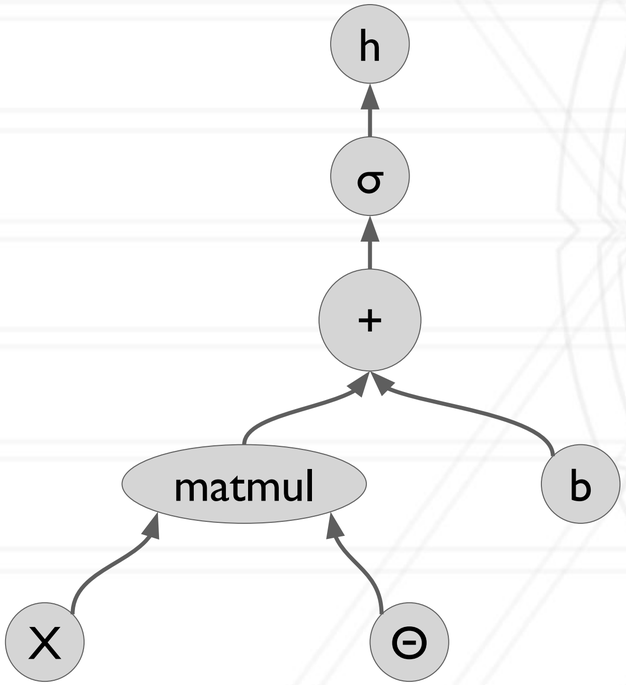
- Assignment 2 due March 14th (with extension to March 17th)
- Assignment 1 grades released. Regrade requests by midnight March 14th.
- Project proposal feedback soon

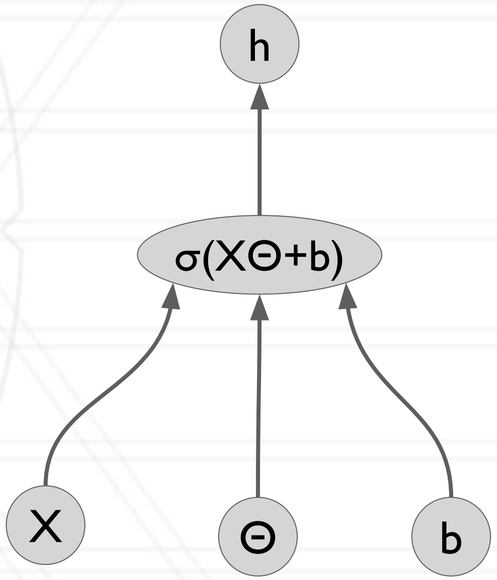
Why DL compilers?

- Enable more optimizations by narrowing to DL kernels
- ML workloads run across a diverse set of hardware
  - Optimally mapping computation to hardware is a hard problem
- Many optimizations and hardware support can be done with any DL library

Why DL compilers?

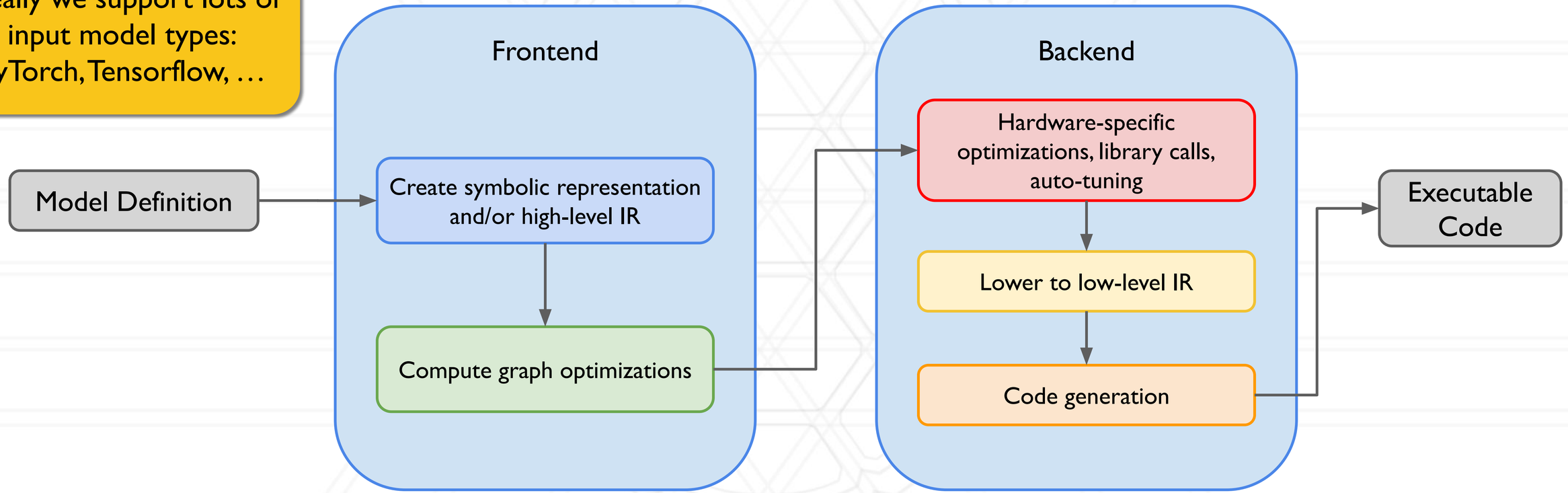
- Automatically detect optimizations for us



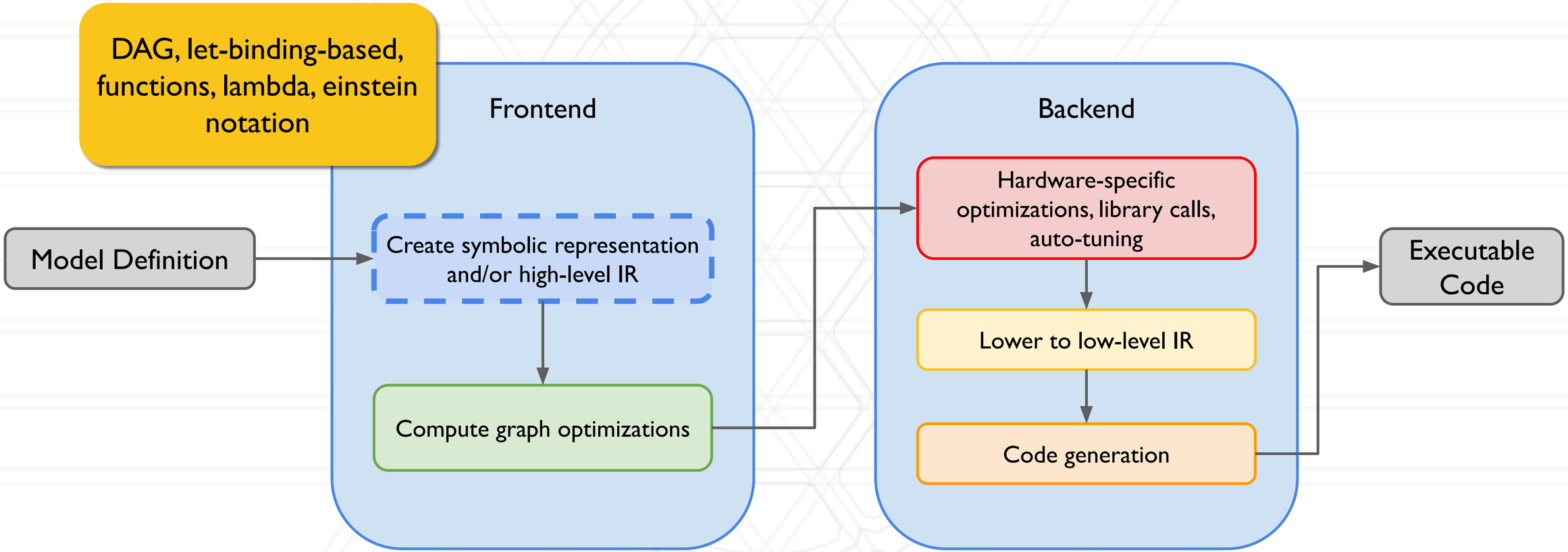


Anatomy of a DL Compiler

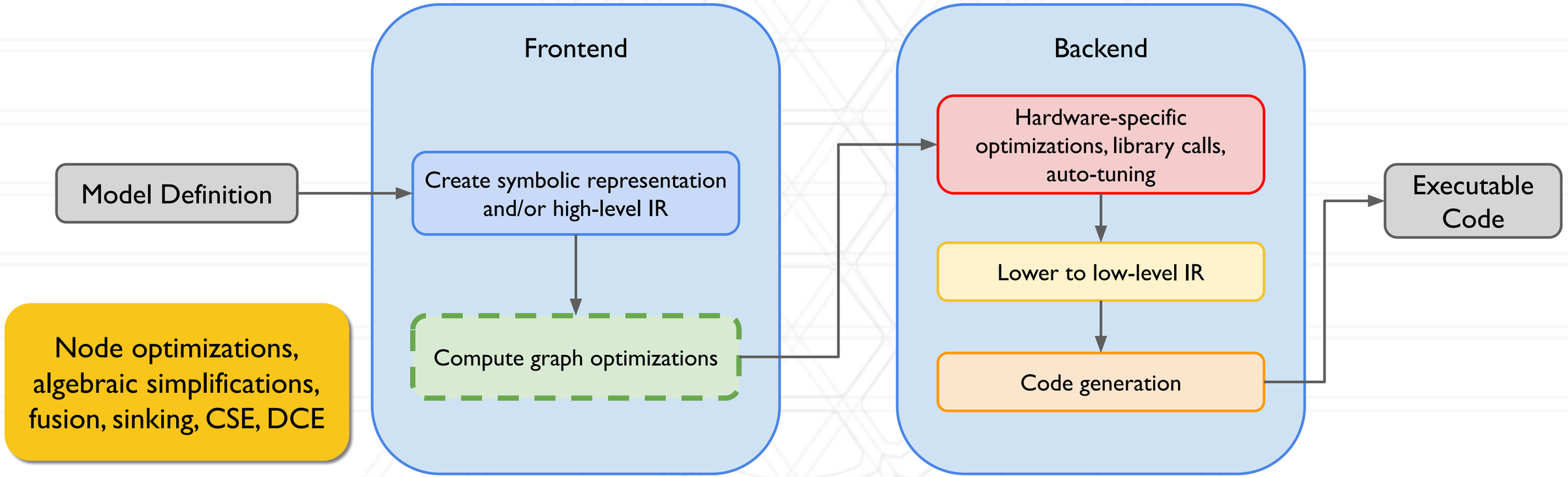
Ideally we support lots of input model types: PyTorch, Tensorflow, …



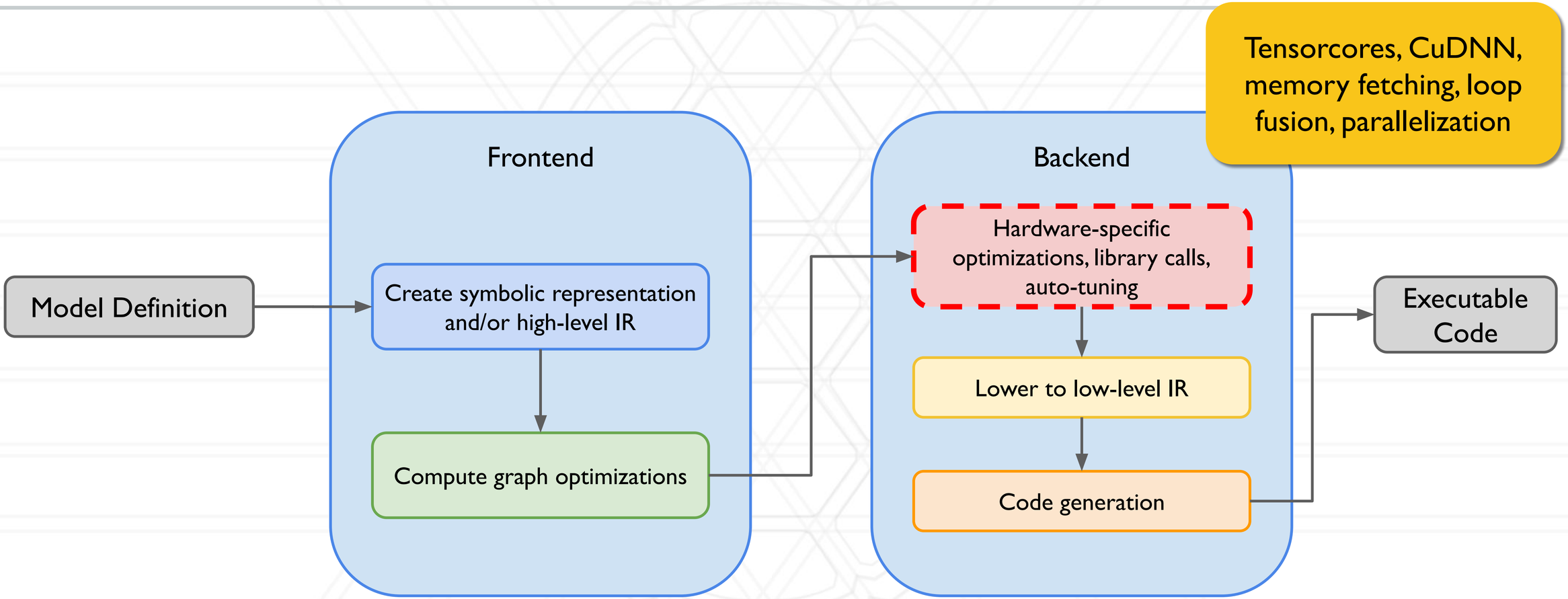
Anatomy of a DL Compiler



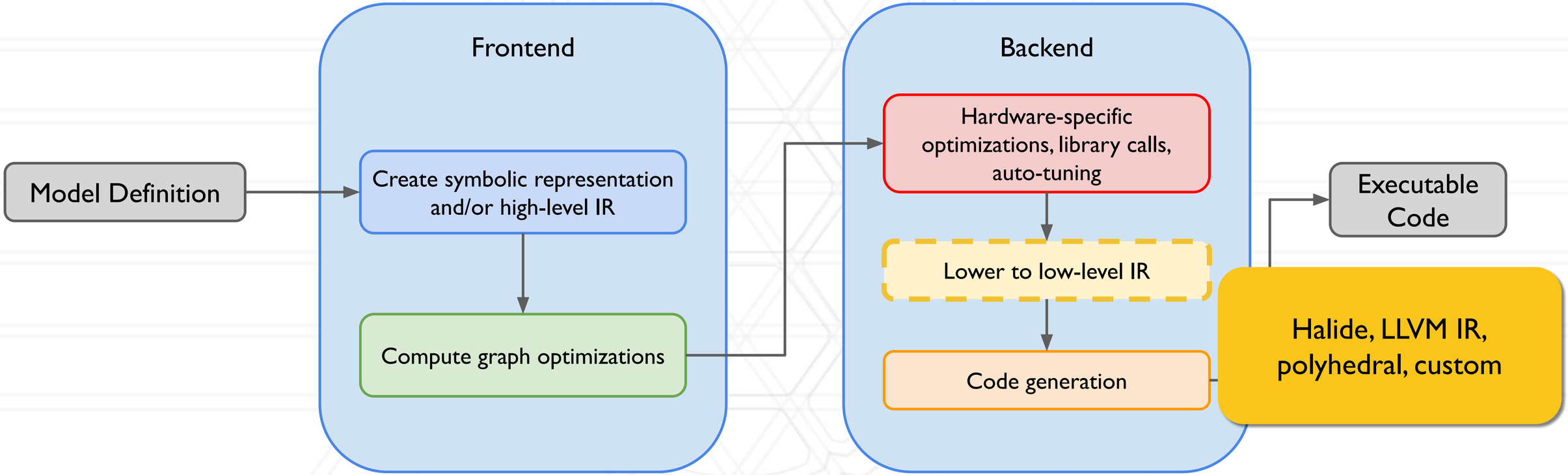
Anatomy of a DL Compiler



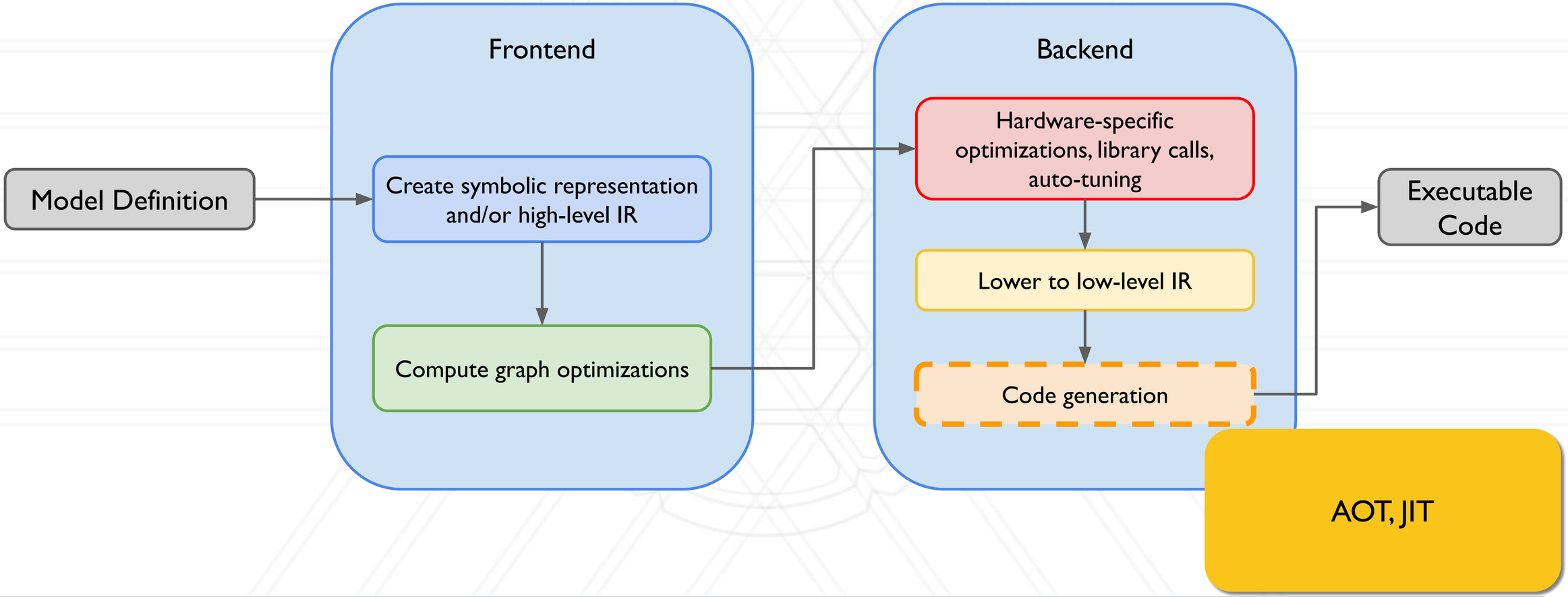
Anatomy of a DL Compiler



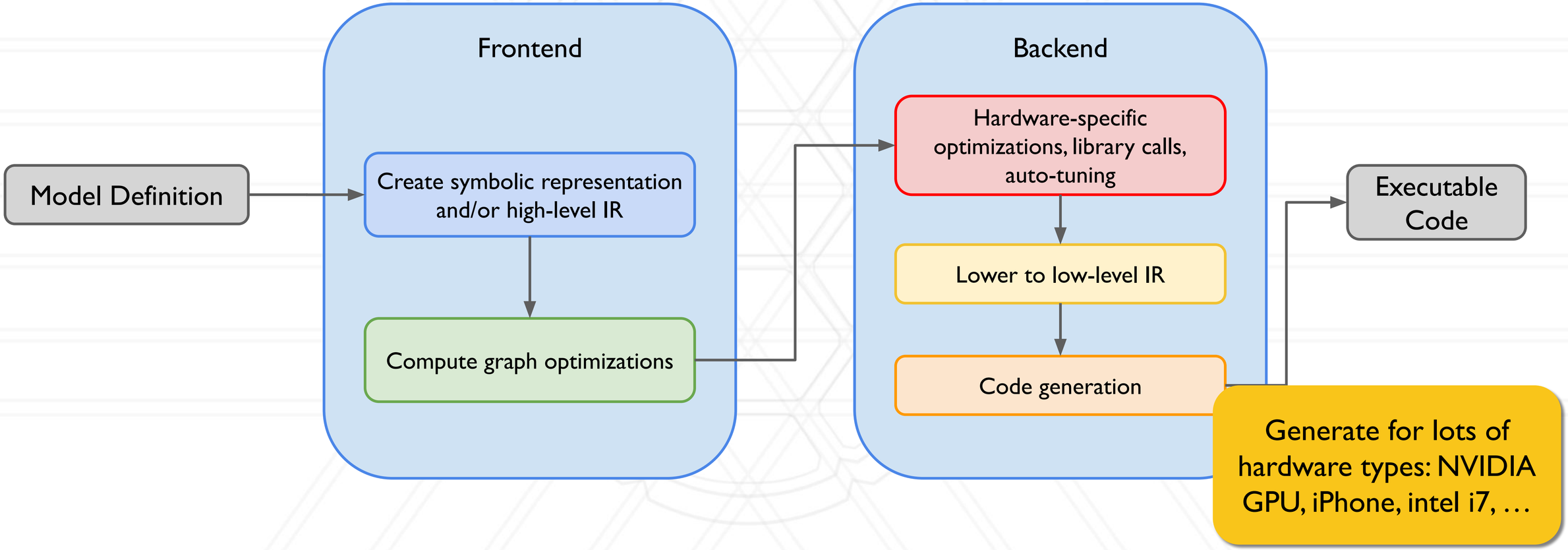
Anatomy of a DL Compiler



Anatomy of a DL Compiler



Anatomy of a DL Compiler



Examples

- TVM
- nGraph
- Glow
- XLA and OpenXLA
- Torch FX
- Torch Dynamo and Inductor
- Tensor Comprehensions (TC)
- TACO

UNIVERSITY OF
MARYLAND