Tensor Primitive OPs Proposal - TPAC'24

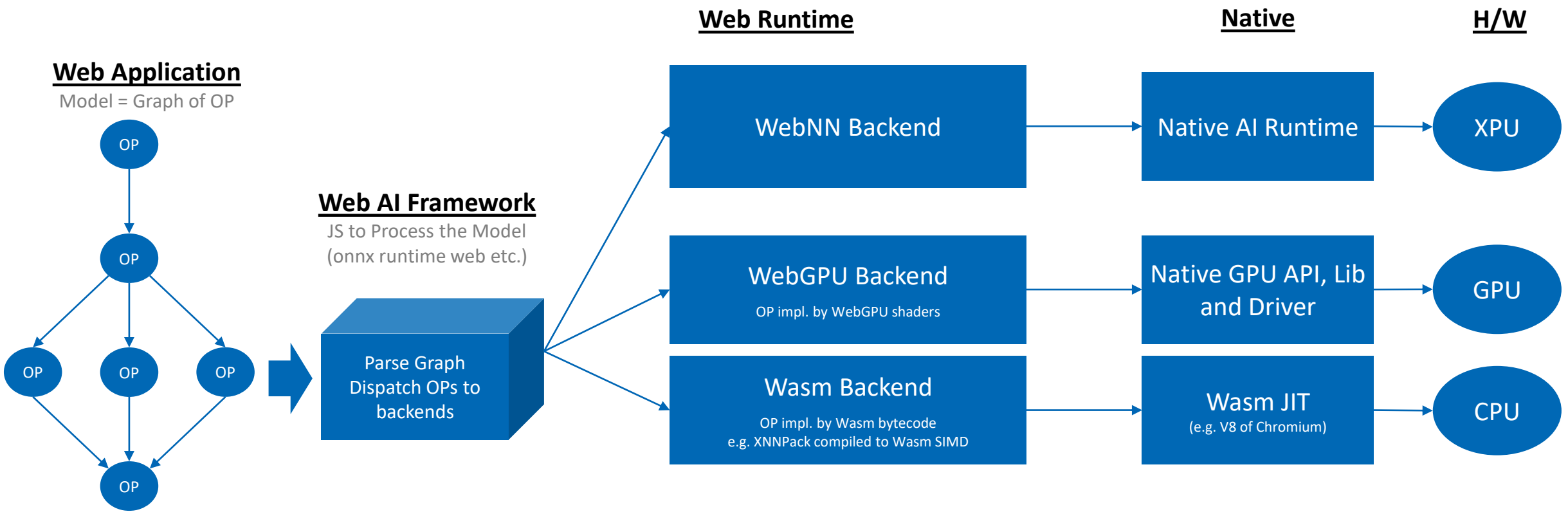
Jonathan Ding, Ningxin Hu, Alexander Heinecke, Yolanda Chen, Jianhui Li, Moh Haghighat

9/23/2004

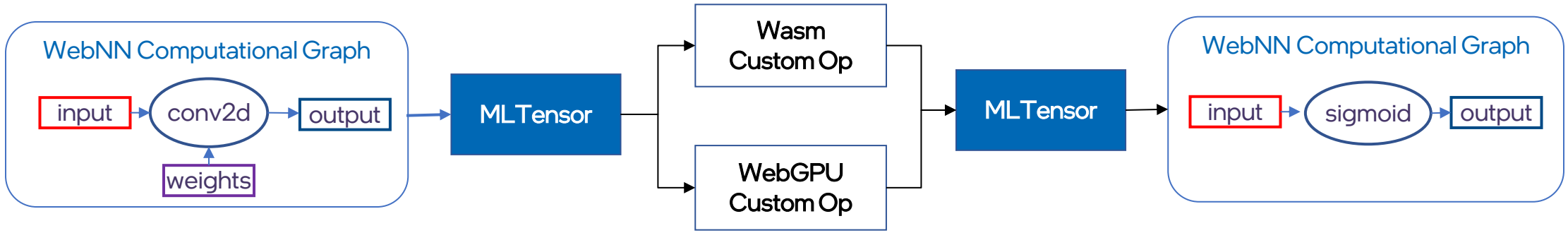
Executive Summary

- AI Models evolve fast and may contain ops that WebNN doesn’t support, e.g., various attentions, today’s frameworks fallback to custom ops written in Wasm or WebGPU
- Problem Statement
  - Non optimal inference performance
    - Lack of accessing dedicated AI accelerators, e.g., CPU/VNNI/AMX, GPU/TensorCore/Systolic Array, NPU
  - Duplicated cost of dev and maintain device specific implementations
- Proposal
  - Composite custom ops by unified Tensor-level Primitive OPs

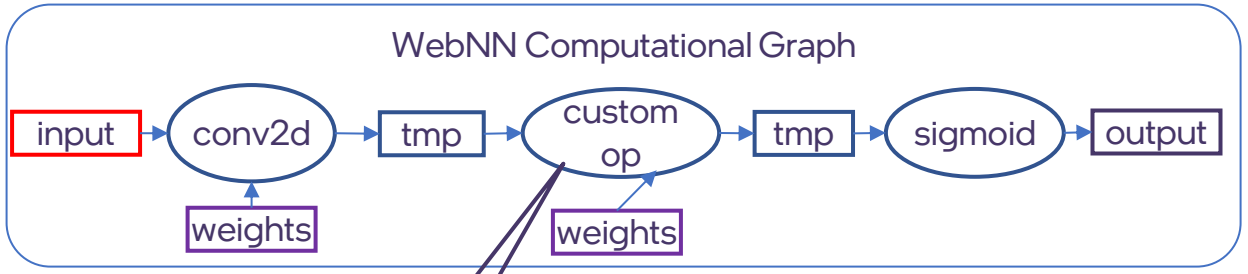
WebGPU and Wasm to complement unsupported OPs in WebNN

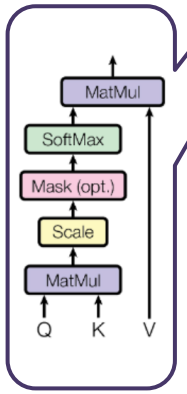


WebNN Custom Op Support with Wasm/WebGPU interop



WebNN Custom Op Support with Tensor Primitive OPs





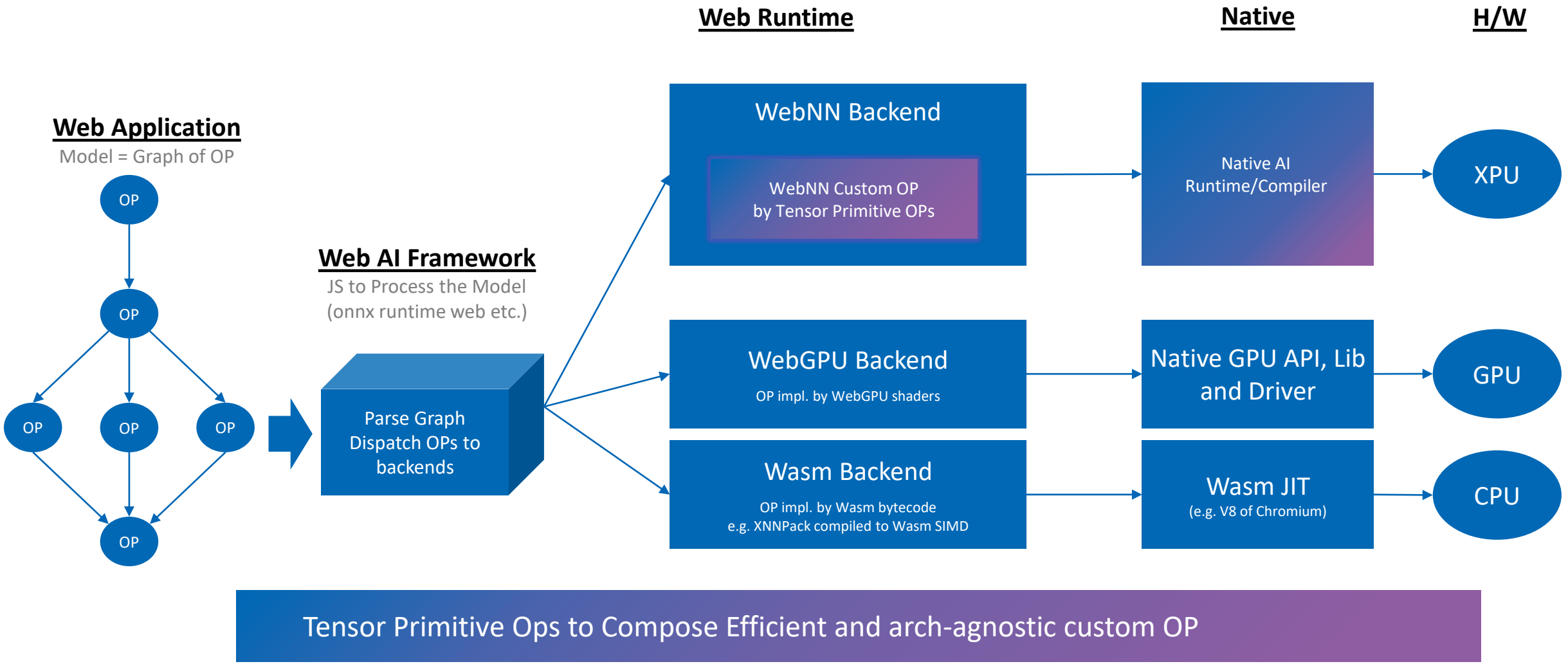
subgraph of primitive ops

Compiled by DML, CoreML, MLIR etc.

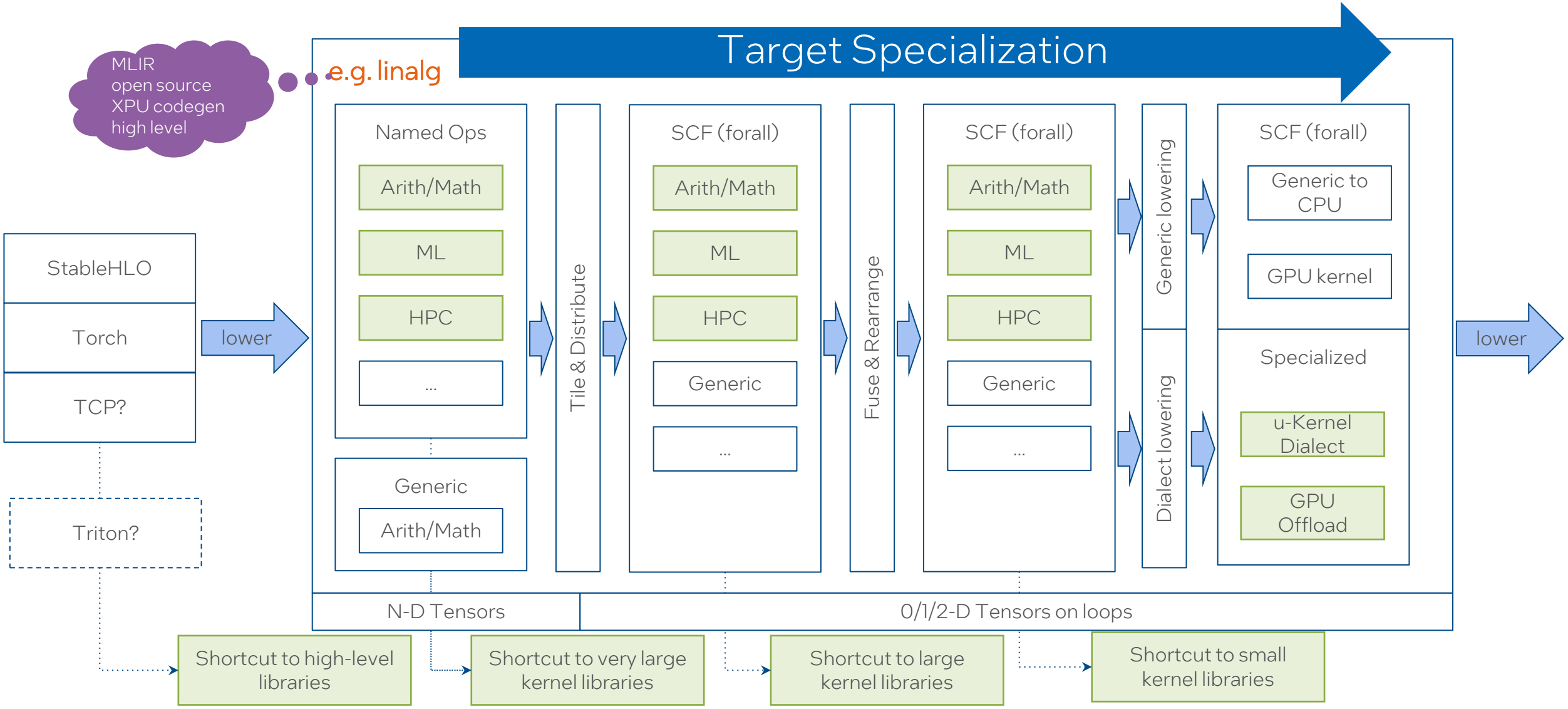
Native Implementation (MLIR as example, can be DXIL, MIL, ...)

```
 // NOTE: only skeleton below, omitted many lines in between
func.func @attention(%X: tensor<?x1024xf32>, %W_Q, %W_K, %W_V // ... ...
  ) -> tensor<?x1024xf32> {
  %Q = linalg.matmul ins(%X,%W_Q : tensor<?x1024xf32>, tensor<1024x1024xf32>)
         outs(%Q_with_bias: tensor<?x1024xf32>) -> tensor<?x1024xf32>
  %K = linalg.matmul ins(%X,%W_K : tensor<?x1024xf32>, tensor<1024x1024xf32>)
         outs(%K_with_bias: tensor<?x1024xf32>) -> tensor<?x1024xf32>
  %V = linalg.matmul ins(%X,%W_V : tensor<?x1024xf32>, tensor<1024x1024xf32>)
         outs(%V_with_bias: tensor<?x1024xf32>) -> tensor<?x1024xf32>
  %QK = linalg.matmul_transpose_b  ins(%Q, %K: tensor<?x1024xf32>,
         tensor<?x1024xf32>) outs(%QK_fill_zero: tensor<?x?xf32>) -> tensor<?x?xf32>
  %QK_scaled = linalg.generic {
                indexing_maps = [affine_map<(d0, d1) -> (d0, d1)>, // ... ...
                iterator_types = ["parallel", "parallel"]
              }
  %QK_softmax = call @softmax(%QK_scaled) : (tensor<?x?xf32>) -> tensor<?x?xf32>
  %output = linalg.matmul ins(%QK_softmax, %V: tensor<?x?xf32>,
          tensor<?x1024xf32>) outs(%output: tensor<?x1024xf32>) -> tensor<?x1024xf32>
```

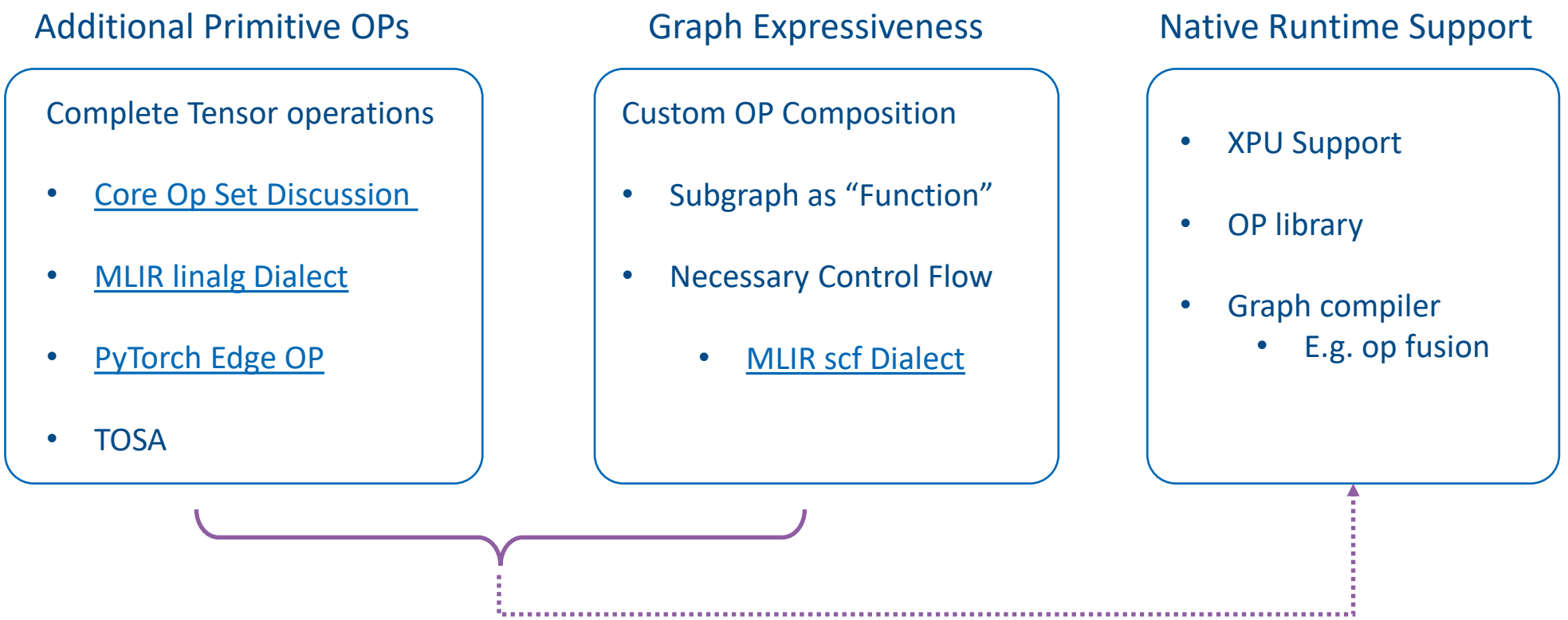
Explicit Low-Level Tensor Primitive OPs Can Enhance AI APIs on the Web



Tensor Primitives at Different Levels

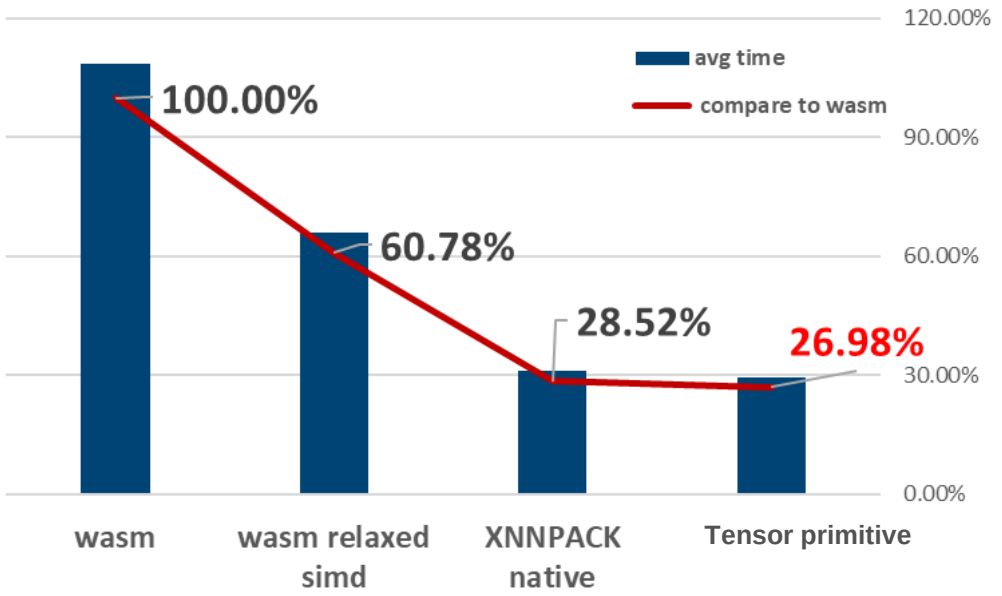


What are needed

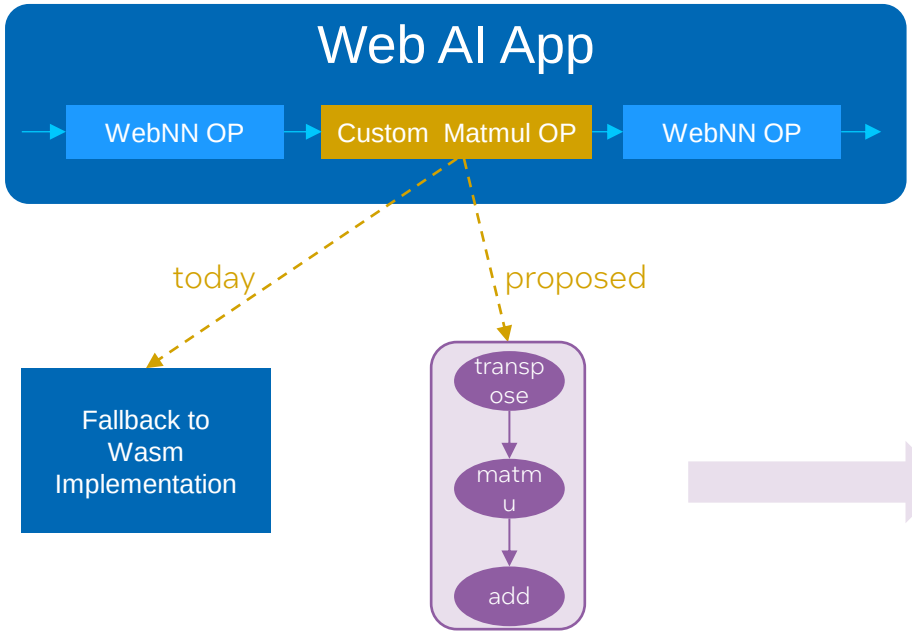


Custom OP composed by Primitives should have comparable perf. as if it is implemented as a high-level OP
Easy to map to native backends, e.g. MLIR, DirectML, CoreML, etc.

PoC WIP



Fused Matmul OP used a POC
Other examples could be MHA, LSTM variants etc.



- H/W-aware partition / tiling + kernel
- Fully utilize native CPU advanced inst./ capability
- Efficient mem re-layout / packing (e.g. for conv)
- Native parallelism

Native Runtime

MLIR toolchain in POC

XPU binary

Subgraph by Tensor Primitive OPs

* Assume OPs are linalg like

Subgraph → linalg Dialect → XPU Code

* Assume MLIR as native runtime

Demonstrated 2X perf. on CPU vs. Wasm fallback