Wasi-nn Status Spring 2025

Wasm ML Workgroup Meeting

(06/11/25)

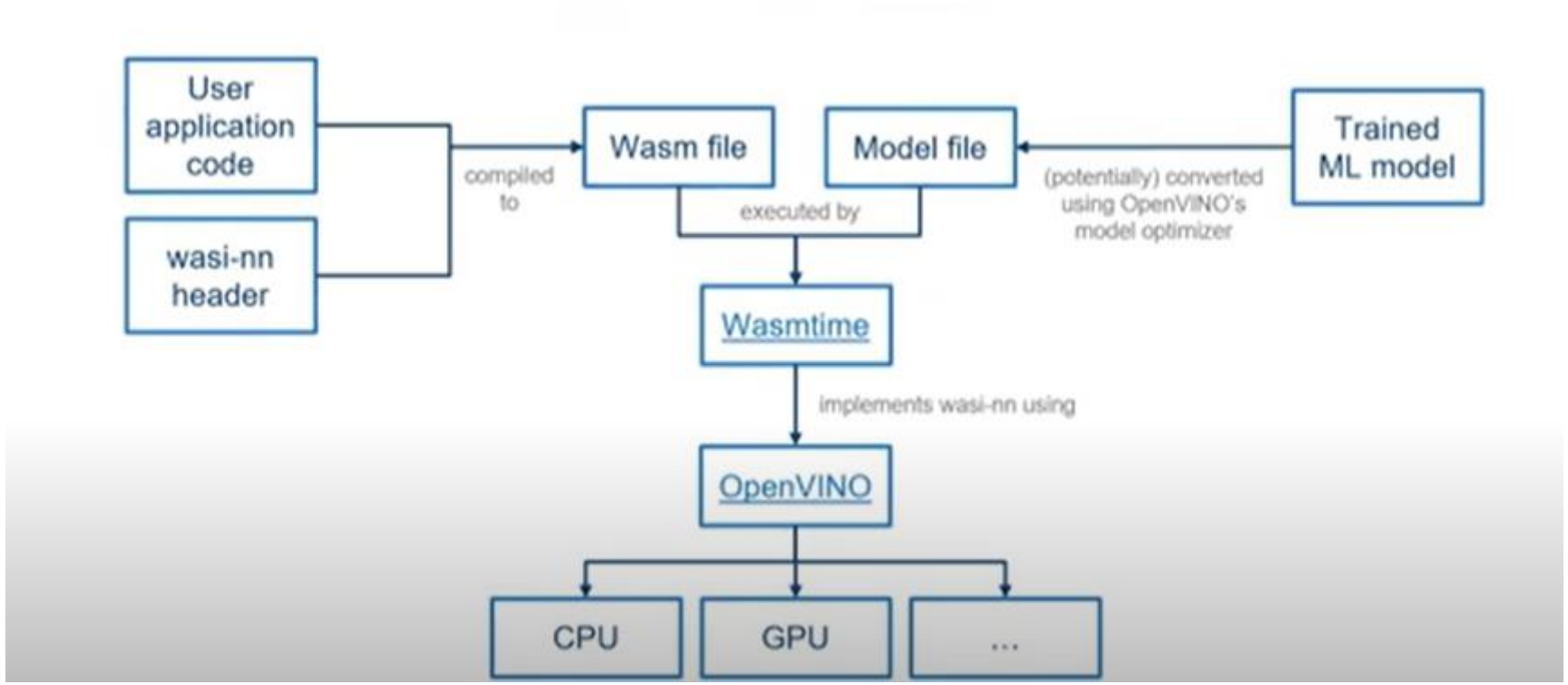
Johnnie Birch

Meeting goals

- Introduce and/or reestablish progress
- Wasi-nn status summary
- Wasi-nn discussion items in 2024
  - Phase 3 requirements
- Review conformance test suite status
- Propose wasi-nn needs in 2025
- Other topics to discuss now or future meetings
- Meeting time and cadence

Wasi-nn Goals

- Proposed in 2020 to allow performant Wasm based ML inferencing applications.



Wasi-nn Implementations

Runtime	Primary Owner(s)	Witx Support	Wit Support	ML-Backend	ML-Backend Status
Wasmtime	Bytecode Alliance	Yes	Yes	OpenVINO	Available
Wasmtime	Bytecode Alliance	Yes	Yes	Pytorch	Available
Wasmtime	Bytecode Alliance	Yes	Yes	Onnx	Available
Wasmtime	Bytecode Alliance	Yes	Yes	WinML	Available
Wasmtime	Bytecode Alliance	Yes	?	Llama.cpp	Experimental
Wasmtime	Bytecode Alliance	Yes	Yes	TFLite	Experimental
Wasmtime	Bytecode Alliance	Yes	Yes	Candle	Experimental
Wamr	Intel, Bytecode Alliance	Yes	No	OpenVino	Available
Wamr	Intel, Bytecode Alliance	Yes	No	TFLite	Available
Wamr	Intel, Bytecode Alliance	Yes	No	Llama.cpp	Available
WasmEdge	SecondState	Yes	No	Llama.cpp	Available
WasmEdge	SecondState	Yes	No	MLX	Available
WasmEdge	SecondState	Yes	No	OpenVINO	Available
WasmEdge	SecondState	Yes	No	Pytorch	Available

ML Workgroup 2024 Progress

- Participation from Microsoft, Fastly, Cosmonic, Second State, with lots of discussion on use cases and spec improvements.
  - API change to replace “load” with “load-by-name”
  - Add a prompt interface
- Discussion and planning for phase 3 acceptance requirements
  - Testimonials and use survey: https://github.com/WebAssembly/wasi-nn/issues/83#issuecomment-2595144573
  - Portability criteria must be met (or present a plan)
- Is this enough in 2025 or do we want to rethink our package for phase 3?
  - Do component model solutions undermine the need for wasi-nn?
  - Do we need to rethink / enhance the wasi-nn API to handle future use cases?
  - Do we need more support from the community?