

Model Versioning

<https://github.com/WebAssembly/wasi-nn/issues/65>

Any given model will have an embedded assumption about the version(s) of backends that will be able to consume it. There may be differences in of:

- serialization format or intermediate representations
- tensor data type support
- operator versions

When a model is loaded in the API, the behavior of mismatch is undefined. There is an "invalid-encoding" error type, but this does not disambiguate between the model actually being invalid or if the requested backend cannot handle the version of the model provided given the requested execution target or (Is this supposed to be an "invalid-operation"?, we should document the errors expected for each API).

One possible solution would be to simply default to the new proposal of extended error information supplied along with the error code. This would be specific, requiring the caller to understand the syntax of what is returned. In the case of autodetected execution targets, discriminating information may exist.

An alternate solution would be to **provide an enumeration API** which defines the supported backends. A supported-encodings() API could provide an encoding value and an extended version string. For example, the onnx encoding would be paired with a string specifying the SemVer of the equivalent e.g. "1.14.1"

If this enumeration API may work before deployment, it fits our IoT/Edge AI requirements

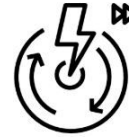
Edge AI Use case

- There are multiple places where AI inference & Pre/Post processing can run
- Where is the most efficient place? considering
 - Device capability
 - Data bandwidth
 - Power consumption
 - Latency
 - Privacy

<https://www.midokura.com/>



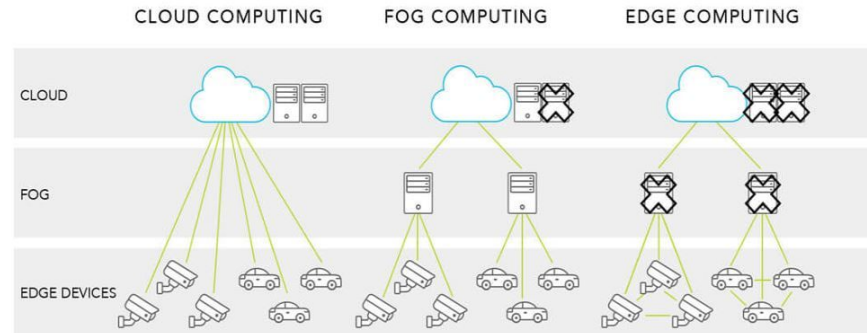
To store privacy data
on local



Real-time
responsiveness




Remote
programmability



Device Capability

Static information

- Supported WASI including version  Component model's "World"
- Supported encoding type (including version)
- Supported IO (image, text, sound..)
- Supported HW accelerator (GPU, TPU, DSP, CPU..)

Runtime information

- Usable Memory (not only for large Model)

I want to know the possibility in advance instead of detecting errors during deployment.
[Enumeration](#) API or [Runtime config](#) might be used regardless of use case.