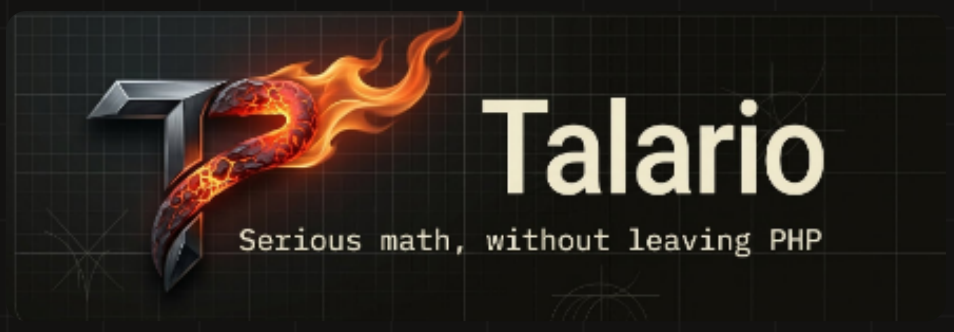


PORTABLE, DETERMINISTIC, IN-PROCESS ML

The Two-Stack Tax.

Your web app runs in one language. Your model runs in another. **You pay for the gap**: in services to run, in hops per prediction, and in results that drift between test and production. Talario closes it: **one model, the same bits on CPU and GPU**, served inside the web stack.

talario.org

swipe >

Shipping ML from a web app usually means two stacks.

1

A second runtime to operate

A Python serving sidecar next to the web server: its own containers, deploys, logs and on-call.

2

A hop on every prediction

Each call leaves the web process, crosses a socket, and serializes data both ways.

3

Two codebases, one feature

Web engineers and ML engineers split across languages, tooling and state.

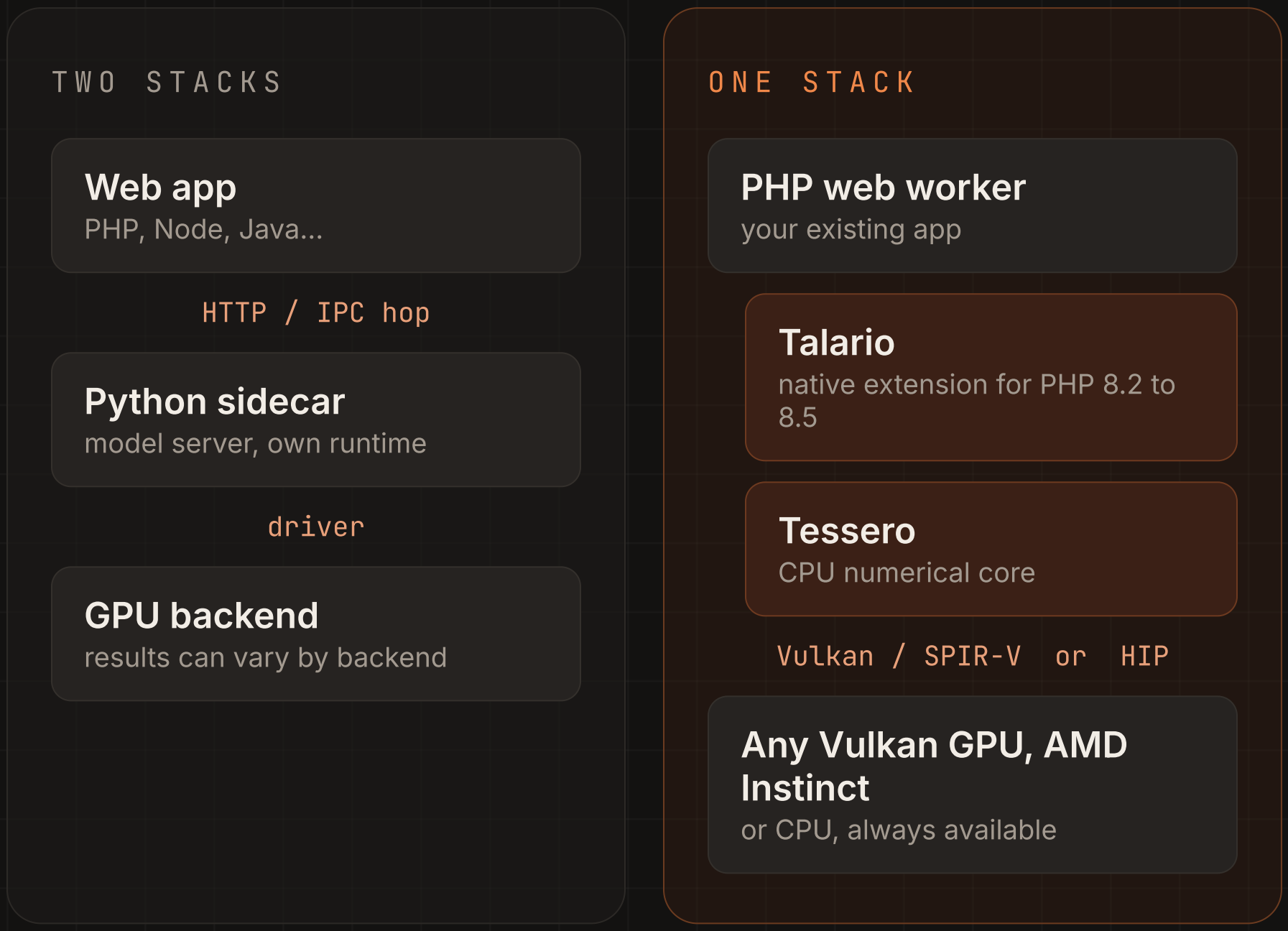
4

Results that move

The numbers you validated on a laptop CPU are not guaranteed to be the bits a cloud GPU returns.

02 / THE APPROACH

Put the model inside the web process.



// **In-process inference.** The model lives in the web worker; no serving sidecar.

// **One engine across GPUs.** Vendor-neutral SPIR-V for any Vulkan GPU, HIP for AMD Instinct, CPU fallback always on.

Train in PyTorch.
Ship in Talario.

Keep your research stack. Export to ONNX, and Talario imports it natively in PHP for the supported operation set, with no onnxruntime and no PyTorch on the server.

```
# research side, unchanged
torch.onnx.export(model, x, "model.onnx")

// production side, inside the PHP app
// import once, compile to .tal,
// reload bit-identical, serve warm
```

// A full **nn.TransformerEncoderLayer** imports and matches PyTorch to **~1e-7**, on CPU and GPU.

// Persistent inference service and warm HTTP endpoint, so other tools can call your models.

CPU == GPU. Training too.

Same output bytes on CPU and GPU, forward and gradients. Checked by SHA-256, not tolerance, at 1024 × 1024 on real silicon.

26

GPU models, all 35 checks bit-identical, softmax, GELU and attention included: 22 NVIDIA (Turing to Blackwell) and 4 AMD Instinct

100

training steps of a transformer block, SGD and Adam: every gradient, every step, CPU == GPU, on x86 and Arm hosts

1

training run, shared bit for bit by the CPU and 19 GPU models across 5 NVIDIA generations

7

NVIDIA generations, Pascal through Blackwell, for the exact ops; drivers 555 to 615, same digests

Why it matters: you can reproduce a result by re-running it, on CPU or GPU, and comparing a hash.

What it is, and what it isn't.

PRECISION	**f32; no Tensor Cores on the bit-identical path.** Determinism is the design priority, not peak throughput.
VENDORS	Attested at full scale on **NVIDIA and AMD Instinct** (MI300X, MI325X, MI350X, MI355X). AMD Radeon and Intel integrated GPUs: exact ops at small sizes.
OPS	A **focused op catalogue**, not PyTorch's thousands. Transformers, GNNs, RL and ONNX import included.
EDGES	**Subnormals and NaN bits** can differ: NVIDIA GPUs flush values near 1e-38 to zero.
STAGE	**Pre-release. There is no public build yet.** Engine built, validated on real NVIDIA and AMD silicon.

Portable. Deterministic. In-process.

Products and services for teams running ML next to a PHP app: talario.org