

PORTABLE, DETERMINISTIC, IN-PROCESS ML

The Two-Stack Tax.

Your web app runs in one language and your model 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 and Tessero: an engineering briefing

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.

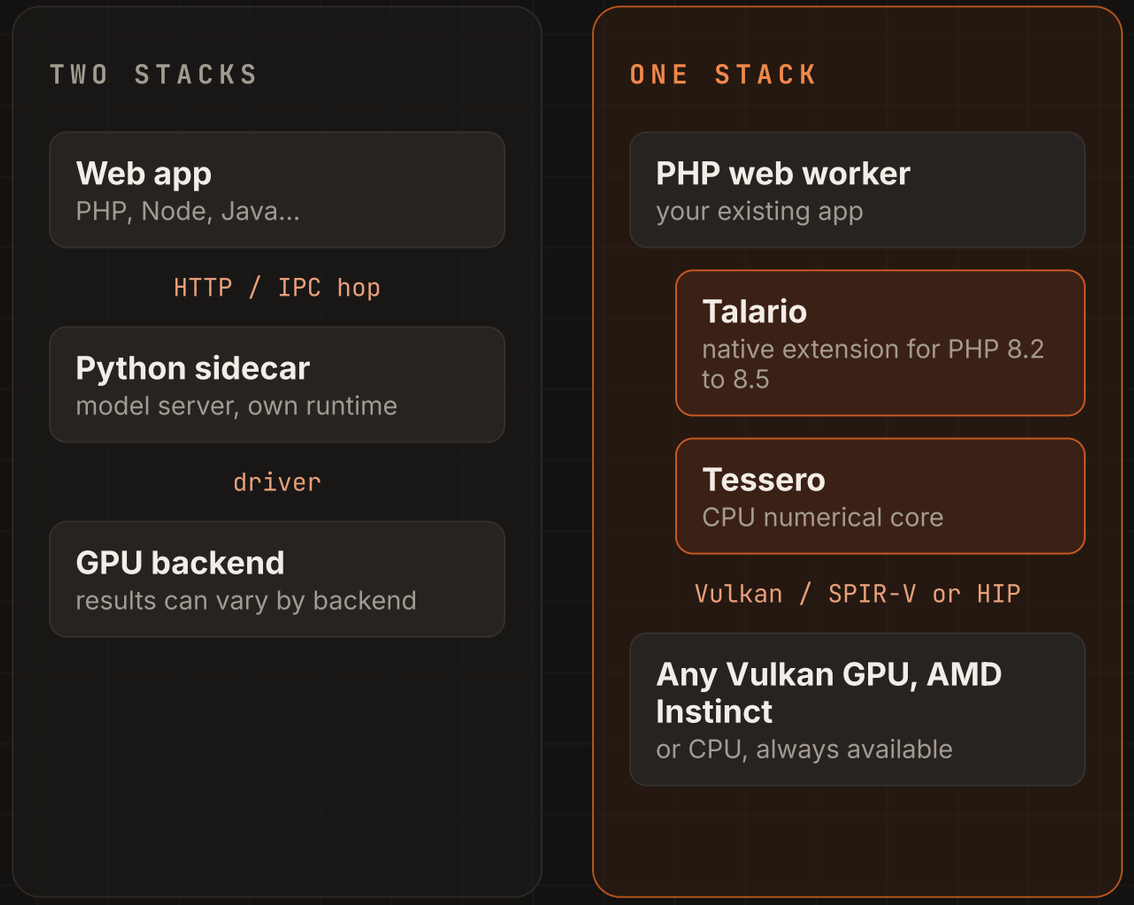
Web and ML engineers split across languages, tooling and state.

4

Results that move.

The numbers validated on a laptop CPU are not guaranteed to be the bits a cloud GPU returns, so a result cannot simply be re-checked somewhere else.

Put the model inside the web process.



In-process inference. The model runs inside the PHP web worker as a native extension. No serving sidecar, no network hop per call.

Talario engine. A native C/C++ runtime with PyTorch-style reverse-mode autograd, layers and optimizers, embedded in the web process.

Tessero numerical core. A native C numerical library, NumPy/SciPy-class (ND arrays, linear algebra, FFT), supplying the CPU kernels in-process.

One engine across GPUs. Kernels compile once to vendor-neutral SPIR-V and run on any Vulkan GPU; a HIP backend reaches AMD Instinct, which has no Vulkan driver. The CPU fallback is always available and, where attested, returns the same bits.

Train in PyTorch. Ship in Talario.

```
# 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
```

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

// **Import once, serve warm.** Compile to a .tal artifact, reload it bit-identical, and serve from a persistent inference service or a warm HTTP endpoint.

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

// **Callable over HTTP.** A warm HTTP endpoint lets other tools call your models.

04 / THE PROOF

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

Results reproduce on attested hardware.

Reproduce by re-running.

Reproduce a result by re-running it on attested hardware and comparing a hash, with no tolerance to argue about.

No train/serve drift.

Where attested, the model validated in testing returns the same bits in production, on a different machine and a different processor.

Audit.

A past result can be reproduced exactly by re-running it on attested hardware.

Debugging.

A training run can be replayed step by step on other hardware, so a divergence is a real change, not noise.

06 / HONEST SCOPE

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.
OPERATIONS	**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

`talario.org`