White Paper · Version 1.1 · September 2026

The Universal AI Inference Engine

A sovereign, hardware-agnostic, multimodal runtime that executes any AI model — language, vision, audio, image and video — through a single open-core engine.

One engine. Any model. Any modality. Zero model-specific code.

Technical positioning document · Developed in Alicante, Spain ·neurobrix.org · Apache 2.0

9	49
MODEL FAMILIES, ONE RUNTIME	MODELS PUBLISHED ON THE HUB
4	**0**
INDEPENDENT EXECUTION MODES	THIRD-PARTY LIBS AT INFERENCE TIME

— Executive Summary

The artificial-intelligence inference stack is fragmented and locked in. Today, running different classes of models in production means assembling and maintaining a different software stack for each one — one engine for language models, another set of glue code for image and video diffusion, yet another for speech — and almost all of these tools are tied, in practice, to a single hardware vendor's proprietary software. For European public bodies and regulated organisations, this fragmentation compounds a deeper concern: a structural dependence on non-European technology for one of the most strategic layers of the modern digital economy.

NeuroBrix is a universal AI inference engine that removes this fragmentation. It executes any model — large language models, vision-language models, image and video diffusion, speech recognition and synthesis — through a single runtime, a single portable model format, and a single hardware-placement solver, with no model-specific code. It is open-core (Apache 2.0), designed from the ground up to be hardware-agnostic, and engineered for auditability and on-premise sovereignty rather than dependence on any external cloud or vendor account.

This document sets out the problem NeuroBrix addresses, its technical foundations, the evidence base behind its claims, an honest comparison with existing inference engines, and its alignment with Europe's emerging technological-sovereignty agenda. NeuroBrix

is designed and developed by Hocine Benkelaya in Alicante, Spain, and distributed commercially under licence by WizWorks OÜ, an Estonian company: the engineering and the commercial operation of the project both sit inside the European Union.

---

1. The Problem: A Fragmented, Vendor-Locked Inference Stack

Modern artificial intelligence is no longer a single kind of model. A production system today may combine a large language model for reasoning, a vision-language model for document understanding, a diffusion model for image or video generation, and speech models for transcription and synthesis. Each of these has, in the open-source ecosystem, grown its own serving stack.

The result is structural fragmentation. Language models are served by one family of engines; image and video diffusion are run through a different set of pipelines and user interfaces; speech requires yet other tooling. Each stack carries its own dependencies, its own deployment model, its own failure modes and its own operational expertise. An organisation that wants to deploy several modalities must integrate, secure, patch and staff several unrelated systems at once. There is, today, no unified, industrial-grade runtime that treats all of these the same way.

1.1 The hardware-lock-in problem

Underneath the fragmentation lies a deeper dependency. The dominant inference tooling is, in practice, bound to one hardware vendor's proprietary software ecosystem. The most widely used optimised inference paths assume that vendor's accelerators, its compiler stack, and frequently an online account and registry to obtain optimised model engines. Organisations running — or wishing to run — alternative accelerators face a software-portability barrier: the models and tooling were never designed to leave that ecosystem. This single-vendor concentration is now widely recognised as a strategic risk, not merely a procurement preference.

1.2 Why this matters for Europe and for the public sector

For European institutions, the stakes are explicit. On 3 June 2026 the European Commission presented its European Technological Sovereignty Package, warning that over-reliance on non-EU providers for cloud and AI infrastructure poses a significant risk to Europe's digital autonomy and resilience, and positioning open source as a strategic instrument to reduce vendor lock-in, improve transparency, and build European

alternatives across the technology stack. The accompanying Cloud and AI Development Act introduces a sovereignty-assurance framework in which control over the software supply chain, transparency, and infrastructure location become explicit criteria for public-sector and critical-infrastructure use.

An inference layer that is fragmented by modality, dependent on a single foreign hardware ecosystem, and opaque in its software supply chain is precisely the kind of dependency this agenda seeks to reduce. The need is for an inference engine that is unified across modalities, portable across hardware, auditable in its dependencies, and deployable entirely on-premise under the operator's own control.

NeuroBrix answers that description on grounds that can be checked rather than asserted: it is developed in Alicante, Spain, and managed from Estonia, so both the engineering and the operation sit inside the Union; its engine is Apache 2.0, readable and forkable; it runs on the operator's own machines with no account, no external registry and no outbound call; and its dependency chain at inference is small enough to be enumerated and is verified by the operator's own probe. What follows argues from those properties, not from a flag.

---

2. The NeuroBrix Approach: One Unified Runtime

NeuroBrix is built on a single founding premise: *every model is, at bottom, a graph of mathematical operations executed on hardware.* A language model, a diffusion transformer, a speech recogniser and a video generator differ in architecture, but each reduces to a computation graph of standard tensor operations. NeuroBrix is the engine that takes that graph and computes it — efficiently, on whatever hardware is available, regardless of which modality the model belongs to.

Concretely, NeuroBrix replaces a fragmented collection of per-modality stacks with three unified components:

A single portable model format (NBX)

Any model — whatever its origin framework — is converted once into a self-contained .nbx artifact that holds the pure operation graph, the weights, the topology and an execution profile. The artifact is framework-independent and portable: it is the unit that the engine runs.

A single execution runtime

One engine, one command-line interface, executes every .nbx artifact. The runtime does not contain, and does not need, per-model code: it sees tensors, graphs and execution plans, whether it is running a mixture-of-experts language model or a video diffusion pipeline.

A single hardware-placement solver (Prism)

One solver maps a model's computation onto the available hardware — from a single consumer GPU to a multi-GPU server — through a cascade of placement strategies, so the same artifact runs across very different machines without changes.

This unification is the product. Where the prevailing approach bolts modalities together by stitching multiple third-party libraries beneath an interface, NeuroBrix is a single coherent runtime designed for universality from the first line of code.

---

3. Technical Foundations

Five engineering properties distinguish NeuroBrix. They are described here at the level of capability and consequence; the proprietary methods that implement them are part of the project's commercial layer.

3.1 True multimodal universality

A single NeuroBrix runtime executes large language models, vision-language and multimodal models, image and video diffusion, speech-to-text and text-to-speech — without any modality-specific serving stack. The breadth is not aspirational: it is demonstrated across nine model families today. This is the property no other single open inference engine currently provides.

3.2 Hardware-agnostic by design

NeuroBrix is architected so that hardware specifics are data-driven rather than hard-coded. A dedicated type and precision subsystem adapts numerical behaviour to the target accelerator's capabilities, and the placement solver is written to be vendor-neutral. The engine is designed to run the same model on different accelerator families, and to degrade gracefully — down to mixed GPU-plus-CPU or pure-CPU execution — rather than refusing to run when a model does not fit a given device.

3.3 Auditable, dependency-free inference

At inference time, the NeuroBrix engine depends only on its own arsenal: in its default mode, the PyTorch numerical core and the accelerator's standard math libraries; in its dependency-free mode, its own compute kernels over a minimal numeric base. It imports no third-party model libraries at runtime — no external transformer, diffusion, tokenizer or codec libraries. Everything a model needs to run travels inside its .nbx artifact. The property is checked rather than asserted: an execution probe enumerates, at the end of a run, every module resident in the process, and the run fails if a third-party model library is among them. The probe is exercised across several model families, on the shipped artifacts. For regulated and sovereign deployments, this means the software supply chain at inference is small, explicit, and verifiable by the operator on their own machine.

3.4 Reproducible, verifiable artifacts

The model-build process is deterministic: building the same model at the same source revision yields a bit-identical .nbx artifact. Reproducibility of this kind is uncommon in the AI-tracing ecosystem and is a precondition for serious audit, validation and certification of an inference pipeline.

3.5 Model-agnostic dynamic execution

Models are represented with symbolic dimensions, so a single artifact runs at arbitrary input sizes — sequence length, image resolution, video frame count, batch size — without re-tracing or recompilation per input shape. This is the foundation of the engine's model-agnostic behaviour and a meaningful operational advantage over approaches that specialise to fixed shapes.

THE FOUR-MODE VALIDATION METHOD

Every model in NeuroBrix is validated across four independent execution modes, used as mutual oracles: a reference mode (PyTorch with the accelerator's standard math libraries); a component-by-component reference; a dependency-free custom-kernel production mode; and an operation-by-operation introspection mode. A model is considered validated only when all four modes agree and produce coherent output. This cross-checking discipline — rare among inference engines — is what allows NeuroBrix to make correctness claims rather than merely throughput claims.

---

4. Validation and Evidence

NeuroBrix's central claim — universality — is substantiated by breadth of validated coverage, not by a single benchmark. Models across nine runtime families have been validated end-to-end through the four-mode method, and 49 models are published on the public NeuroBrix model hub — the one count this document uses, read from the hub as the page is served. The table below summarises representative validated models per family.

Model family	Representative validated models	Status
Large language models (incl. mixture-of-experts)	TinyLlama, Qwen3-30B-A3B (MoE), DeepSeek-MoE	Validated
Vision-language / multimodal	Janus (understanding + generation)	Validated
Speech-to-text (STT)	Whisper-large-v3-turbo, Whisper-large, Parakeet	Validated
Speech understanding (audio-LLM)	Voxtral-Mini-3B, Canary-Qwen-2.5B, Granite-Speech-3.3-8B	Validated
Text-to-speech (TTS)	Kokoro, VibeVoice, Chatterbox, OpenAudio, Orpheus	Validated
Image generation (diffusion)	SANA, PixArt-α, PixArt-Σ, Flex.1, Janus	Validated
Super-resolution / upscaling	Swin2SR, Real-ESRGAN, SwinIR, HAT	Validated
Video generation	Wan2.1-T2V, SANA-Video, CogVideoX-5b-I2V, CogVideoX-2b, mochi-1, Wan2.1-VACE	In active integration

The audio domain (speech-to-text, speech understanding and text-to-speech together) is fully validated: eleven of eleven models pass all four execution modes. The super-resolution family is validated to numerical agreement with reference output. Video generation — the most demanding family, where the engine extends its diffusion machinery into the temporal dimension, with text-to-video and image-to-video, classifier-free guidance, dual-expert denoisers and spatio-temporal VAE tiling — is the one family still in active integration. Its state, model by model, is reported with the global benchmark of the stable release (see the benchmarks page). This document states no figure of its own on speed or capacity.

4.1 What the breadth demonstrates

Running a mixture-of-experts language model, a diffusion image generator, a speech recogniser and a video generator *through the same runtime and the same artifact format* is the concrete proof of universality. Each family exercises different parts of the engine; passing all of them under one runtime is evidence that the unification is real and not a façade over per-modality special cases. The dependency-free property has been demonstrated across this entire set in a clean-room environment, and the build reproducibility holds artifact-by-artifact.

5. Competitive Landscape

NeuroBrix is best understood by contrast with the leading inference tools. Each of them is strong within its scope; the point of difference is that each solves one part of the problem, while NeuroBrix is designed to solve the whole. The comparison below is drawn from each project's own current documentation.

Capability	vLLM (core)	NVIDIA NIM	llama.cpp	ONNX Runtime	Neurol
Language models	Yes	Yes	Yes	Graph	Yes
Image & video generation	No*	No	No	No	Yes
Speech (STT + TTS)	No*	Partial	No	No	Yes
All modalities, one runtime	No	No	No	No	Yes
Hardware-vendor-neutral	Partial	No	Yes	Yes	By des
No external account required	Yes	No	Yes	Yes	Yes
No third-party model libs at runtime	No	No	Yes	Partial	Yes
Reproducible model artifacts	No	No	Partial	Partial	Bit-ider
Open source	Yes	No	Yes	Yes	Apache cor

* Core vLLM is optimised for autoregressive language-model serving; image, video and audio generation are provided through a separate companion project rather than the core engine. As that companion project's own technical paper notes, mainstream serving frameworks are specialised for a single generation paradigm — language-model serving on one side, diffusion on the other — which is the fragmentation NeuroBrix is built to remove.

5.1 The honest reading

Several tools are excellent at one thing. **llama.cpp** runs language models across an exceptionally wide range of hardware, but only language and vision-language models, in a single specialised format. **vLLM** is the reference for high-throughput language-model serving and has extended toward multimodality through a separate companion project that stitches together many third-party libraries beneath it. **NVIDIA NIM** delivers highly optimised inference but is bound to one hardware vendor and, in its standard mode, to that vendor's online account and registry. **ONNX Runtime** is a portable graph executor with many hardware back-ends, but it is not a multimodal serving runtime: the multimodal pipelines, schedulers and memory management must be assembled around it.

NeuroBrix's distinctive position is the combination, in one system, of four properties that no competitor offers together: *all modalities under one runtime; hardware-vendor-neutral by design; zero third-party model dependencies at inference; and reproducible, auditable artifacts* — delivered as an open-core engine.

---

6. Alignment with European Digital Sovereignty

NeuroBrix maps directly onto the priorities set out in the European Technological Sovereignty Package of June 2026 and the policy direction it represents. The alignment is structural, not rhetorical:

European policy objective	How NeuroBrix contributes
Reduce dependence on a single foreign hardware ecosystem	Hardware-agnostic by design: the engine is built to run the same models across accelerator families, removing the software-portability barrier that ties workloads to one vendor.
Strengthen the software supply chain & transparency (Cloud and AI Development Act assurance criteria)	Dependency-free inference: no third-party model libraries are imported at runtime; the inference supply chain is small, explicit and clean-room-verified.
Promote open source as strategic infrastructure (EU Open Source Strategy)	Open-core under Apache 2.0, reusable by public administrations and private operators alike, with no licence cost to adopt the engine.
Enable sovereign, on-premise deployment & data control	Runs entirely on the operator's own infrastructure, with no mandatory external account, registry or cloud call; data never leaves the operator's perimeter.
Support auditability and certification	Bit-identical reproducible artifacts and the four-mode validation discipline provide a foundation for independent audit and verification.
Build European capability and talent	Developed in Alicante, Spain, and managed from Estonia: the engineering and the operation of the project both sit inside the Union.

WHAT THE ALIGNMENT RESTS ON

Five things, each of them checkable: the engine is developed in Alicante, Spain, and managed from Estonia, so engineering and operation are both inside the Union; the core is Apache 2.0, readable and forkable; it deploys on the operator's own hardware with no account, no external registry and no outbound call; its dependency chain at inference is short enough to enumerate and is verified by an execution probe; and its artifacts rebuild bit-identically. That is the inference layer of the AI stack — where models meet hardware, and where today's dependencies are most acute.

7. Target Deployments and Use Cases

NeuroBrix is positioned for organisations and operators for whom control, portability and breadth matter more than dependence on a managed external service.

On-premise and sovereign deployments

Public administrations, healthcare, finance, defence and other regulated sectors that require data residency, full infrastructure control and an auditable software supply chain. NeuroBrix runs entirely within the operator's perimeter and serves multiple modalities from one installation.

Non-mainstream and heterogeneous hardware

Teams operating accelerators outside the dominant vendor ecosystem, or mixed fleets, who face the software-portability barrier. NeuroBrix's hardware-agnostic design is intended precisely to make such hardware usable without rewriting models.

Cost-sensitive and multi-modality inference

Organisations running several model types who would otherwise maintain several stacks. Consolidating onto one engine reduces integration surface, operational overhead and the number of systems to secure and patch.

Compute providers

Infrastructure operators offering alternative accelerators, for whom software portability is the principal barrier to adoption by their customers. An integrated NeuroBrix offering can remove that barrier directly.

Accessibility across the hardware spectrum

Because the placement solver degrades gracefully rather than refusing to run, the same engine serves a single-GPU workstation and a multi-GPU server. This widens access to capable inference beyond well-funded datacentres.

---

8. Open-Core Model and Governance

NeuroBrix follows an open-core model. The inference engine — the runtime that executes .nbx artifacts — is fully open source under the Apache 2.0 licence, freely available on public repositories and the Python package index, and free to adopt, inspect and self-host. Around this open core sits a commercial layer: the proprietary toolchain that converts and prepares models, and the model hub that distributes

validated artifacts. This is a well-established structure for deep-technology infrastructure companies: an open, trustable, adoptable core, with sustainable commercial tooling and distribution on top.

The open core is deliberately the part that matters most for sovereignty and trust. An organisation can read the engine's source, verify what it does, and run it without depending on the project's commercial services. Licensing is clean: the engine carries no model licence of its own; each model retains the licence of its original author, recorded transparently in the hub.

The public presence is live today: the engine is distributed through public source repositories and the Python package index, and the model hub (neurobrix.org) serves validated artifacts with per-model licence metadata.

---

9. Origin, Status and Roadmap

9.1 Origin and structure

NeuroBrix is designed and developed by Hocine Benkelaya in Alicante, Spain, on a privately operated GPU cluster, and NeuroBrix and its intellectual property belong to Hocine Benkelaya personally. The project is supported and funded by Neural Network Holding Ltd, a company incorporated in the United Kingdom, and distributed commercially under licence by WizWorks OÜ, a company incorporated in Estonia.

Development therefore takes place in Spain and operations are run from Estonia — both within the European Union — under United Kingdom ownership of the operating company. This document announces no other domiciliation: where the project may be established in future is not decided, and an undecided thing is not a claim.

9.2 Current status

The engine is at public release 0.6.4 (Apache 2.0). Nine model families are validated end-to-end through the four-mode method; 49 models are published on the public hub; and the dependency-free property and build reproducibility hold across the validated set. Empirically, validation to date has been performed on NVIDIA Volta-class accelerators and on Apple Silicon. The architecture is vendor-neutral and the additional accelerator paths are engineered work, not a research question.

One of those paths is further along than the others and is reported here as work in progress rather than as a delivered feature. Triton execution on Apple GPUs is proven on a public development branch: first light is passed, a complete language model has been executed end to end, and the memory allocator and the kernel launcher are written. It is not on the main branch and it is not in the release above; it is evidence that the vendor-neutral design holds outside the platform it was built on, and nothing more than that yet.

9.3 Roadmap

- **Complete the video-generation family.** Video generation is the one family still in active integration, the largest models depending on the next item. Its state is reported model by model once the global benchmark of the stable release is published, rather than summarised here.
- **General multi-GPU placement and long-sequence capability.** A unified, hardware-agnostic capability for distributing very large models across multiple accelerators and for long-clip / native-resolution video — the principal capability work now in progress.
- **Empirical validation on additional accelerator families.** Extending the demonstrated hardware coverage beyond the current validation platforms, exercising the engine's vendor-neutral design on alternative accelerators.
- **Extend the published measurements.** One global benchmark, run from the engine entry point and published with the stable release, is the planned measurement. Extending it to further hardware is continuing work.
- **Ecosystem engagement.** Engaging with the national and European AI ecosystem — public research, compute providers and public-sector operators — around on-premise, auditable inference.

> **A NOTE ON HONESTY**
>
> This document deliberately distinguishes what is validated today from what is in progress. The universality, the dependency-free property, the reproducibility and the breadth of validated models are demonstrated facts. The full multi-vendor hardware matrix and the completion of the largest video models are engineering work in progress, presented as such. The project's discipline is to claim correctness only where it has been independently cross-checked.

10. Conclusion

The inference layer of the AI stack is where models meet hardware — and it is today both fragmented by modality and concentrated on a single foreign hardware ecosystem. NeuroBrix offers a different foundation: one open-core engine that runs any model of any modality, on a hardware-agnostic basis, with an auditable and dependency-free inference path and reproducible artifacts. It is developed in Alicante, Spain, and managed from Estonia — engineering and operation both inside the Union — aligned by construction with Europe's technological-sovereignty objectives, and validated across nine model families, with 49 models published on the public hub.

We welcome the opportunity to demonstrate the engine and to explore where a sovereign, universal inference engine developed in Spain can contribute across the national and European AI ecosystem. Technical teams, deployment operators and compute partners are invited to engage with the project directly — to evaluate the engine on their own models and hardware, and to discuss concrete deployments.

Project coordinates

Project	NeuroBrix — Universal AI Inference Engine
Licence	Apache 2.0 (open core)
Website & model hub	neurobrix.org
Distribution	Public source repositories · Python package index (neurobrix)
Development and ownership	Hocine Benkelaya — Alicante, Spain — owns NeuroBrix and its intellectual property — https://www.linkedin.com/in/hocine-benkelaya/
Support and funding	Neural Network Holding Ltd (United Kingdom) — https://neuralnetworkholding.com
Commercial distribution	WizWorks OÜ (Estonia), under licence — https://wizworks.io
Contact	contact@neurobrix.org

References

1. European Commission, "Strengthening Europe's Tech Sovereignty", press material, 3 June 2026. commission.europa.eu

2. European Commission, "Cloud and AI Development Act", Shaping Europe's Digital Future, 2026. digital-strategy.ec.europa.eu
3. European Commission, press release IP/26/1187, "Commission proposes tech sovereignty package", 3 June 2026.
4. vLLM-Omni project, technical paper and documentation, "Disaggregated serving for any-to-any multimodal models", 2026; vLLM core documentation, "Supported Models", 2026. docs.vllm.ai
5. NVIDIA, "NIM Microservices" product and deployment documentation, 2026. nvidia.com / docs.nvidia.com
6. llama.cpp project documentation and build guide, 2026. github.com/ggml-org/llama.cpp
7. ONNX Runtime documentation, execution providers, 2026. onnxruntime.ai
8. NeuroBrix public documentation and model hub, 2026. neurobrix.org

---

This document is a technical positioning white paper. It describes the capabilities and design of NeuroBrix at the level of what the engine does and why; the proprietary methods of the project's commercial toolchain are not disclosed here. It states no figure of its own on speed or capacity — none is published at this time; one global benchmark will be published with the stable release. The count of published models is read from the hub at the moment the page is served. Version 1.1, September 2026.

© 2025–2026 Hocine Benkelaya. NeuroBrix engine released under the Apache License 2.0.
Developed by Hocine Benkelaya · Supported and funded by Neural Network Holding Ltd · Commercial distribution by WizWorks OÜ