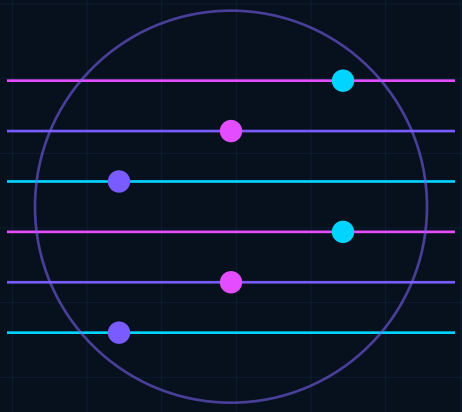
LOCAL AI ACCELERATION

NPU Integration & Local Inference Acceleration

Current-source review, evidence-bounded adoption decision, explicit limitations, reproducibility controls, and preserved source research note.



PERMANENT ID	EVIDENCE	ADOPTION	FRESHNESS
RES-018	A-	ADOPT HARDWARE-ABSTRACTED	ACTIVE WATCH

Freshness review: 2026-09-17 / Next scheduled review: 2026-12-16

RESEARCH CONTROL SURFACE

Required Research Fields

Each field remains independently reviewable; evidence strength does not imply production authority.

EVIDENCE GRADE — **A-**

Strength of the retained source set and technical basis.

SOURCE AUTHORITY — **3 registered authorities**

Windows ML Overview; OpenVINO NPU Device Documentation; Apple Core ML Documentation

FRESHNESS — **ACTIVE WATCH**

Reviewed 2026-09-17; dynamic sources require event-driven revalidation.

CONFLICTS — **VISIBLE / NOT SILENT**

Differences between specification, vendor behavior, research claims, and reproduced evidence must remain explicit.

LIMITATIONS — **EXPLICIT**

No certification, procurement approval, legal conclusion, or unbounded production claim is created by this report.

REPRODUCIBILITY — **REQUIRED**

Material claims require exact versions, representative data, failure cases, artifacts, and repeatable acceptance criteria.

ADOPTION POSTURE — **ADOPT HARDWARE-ABSTRACTION STRATEGY**

Adopt a runtime abstraction across CPU, GPU, and NPU execution providers. Benchmark per device, model, quantization, operator coverage, memory movement, and power envelope.

REVIEW TRIGGER — **2026-12-16**

Scheduled at 90 days; earlier on material source, vulnerability, implementation, or contradictory-evidence change.

SOURCE AUTHORITY / FRESHNESS

Authority Register and Freshness Delta

Primary-source lineage is preserved; current-source changes are separated from unperformed implementation claims.

AUTHORITY 01

Windows ML Overview

https://learn.microsoft.com/en-us/windows/ai/new-windows-ml/overvie

AUTHORITY 02

OpenVINO NPU Device Documentation

https://docs.openvino.ai/2026/openvino-workflow/running-inference/inf

AUTHORITY 03

Apple Core ML Documentation

https://developer.apple.com/documentation/coreml

2026-09-17 FRESHNESS DELTA

Primary-source register retained and freshness controls advanced to the current review date. No new product or laboratory performance claim is introduced without reproduced evidence.

Freshness policy: scheduled review + immediate review on source supersession, material release, vulnerability, retraction, or contradictory evidence.

Evidence Must Survive Contact With Reality

- Source authority is not implementation evidence. Product behavior, scale, security, interoperability, and operator fit remain environment-specific.
- Conflicting evidence is recorded rather than averaged away. Specification behavior, vendor implementation, preprint results, and production observations are separate evidence classes.
- Quantitative claims are informative until reproduced under documented hardware, software, data, topology, load, and failure conditions.
- Adoption posture is reversible. A stronger source set can coexist with a pilot-only or research-only recommendation when maturity, safety, economics, or operability remain unresolved.

ADOPTION / VALIDATION / REVIEW

Decision Gate and Validation Plan

ADOPTION POSTURE

ADOPT HARDWARE-ABSTRACTION STRATEGY

Adopt a runtime abstraction across CPU, GPU, and NPU execution providers. Benchmark per device, model, quantization, operator coverage, memory movement, and power envelope.

1 / SOURCE	Pin exact versions, publication state, retrieved artifacts, and source hashes.
2 / PROTOTYPE	Demonstrate the core mechanism with representative data and instrumented execution.
3 / FAILURE	Exercise malformed input, dependency loss, stale state, overload, recovery, rollback, and misuse.
4 / BASELINE	Compare against an incumbent, classical, or simpler architecture using the same workload.
5 / ACCEPT	Record acceptance criteria, residual limitations, owner, expiration, and revalidation triggers.

EXPIRATION / REVIEW TRIGGER

Next scheduled review: 2026-12-16 (90-day cadence). Review sooner after material source revision, breaking implementation release, critical vulnerability, retraction, benchmark contradiction, changed workload, or changed legal/safety boundary.

8. Complete Source Research Note

SOURCE-LINEAGE NOTICE

The following material preserves the complete original source note, excluding only the conversational wrapper and outer Markdown fence. Legacy status labels and absolute language are retained for traceability and do not override the candidate status, limitations, or adoption decision in this edition.

Document ID: RES-018 Version: 1.0.0 (Definitive Registry Edition) Status: Published Research Note Classification: Public Organization: Mythos Systems (MYTHOS AI) Owner: Aaron Stovall Effective Date: 2026-07-16 Applies To: Edge AI engineers, mobile platform developers, and system architects designing local-first AI inference (CALLISTO), heterogeneous compute pipelines, and sovereign edge runtimes Companion Standards: MYTHOS-EMT-0003 (Neuromorphic Computing & NPU Standard), MYTHOS-EMT-0002 (Sovereign AI), MYTHOS-WEB-0001 (WebGPU & WebLLM), RA-006 (Local-First AI Inference Cluster)

The architecture of local AI compute has failed. Organizations attempt to run continuous edge inference—wake-word detection, real-time transcription, and local LLM reasoning—using high-wattage GPUs or general-purpose CPUs. The GPU drains device batteries in minutes and requires active cooling; the CPU bottlenecks the system, causing UI jank and thermal throttling. Meanwhile, the dedicated Neural Processing Unit (NPU) sits idle on the silicon because standard AI frameworks (PyTorch/TensorFlow) default to CUDA or CPU backends.

The Mythos Architecture mandates the absolute utilization of heterogeneous silicon. NPU integration shatters the monolithic compute paradigm. By dynamically discovering the underlying hardware (Apple Neural Engine, Intel NPU, Qualcomm Hexagon) and partitioning the compute graph, we offload dense matrix multiplications to specialized, sub-milliwatt silicon.

This research note defines the architectural dynamics of NPU integration. It dissects the vendor-specific SDKs (CoreML, OpenVINO, QNN), the non-differentiable boundary of operator support, and the memory bandwidth isolation required to execute concurrent AI workloads (e.g., running an LLM on the GPU while the NPU handles continuous audio transcription) without OOM crashes. Understanding these dynamics is a prerequisite for building the sovereign, zero-latency CALLISTO web client and PANTHEON edge runtimes.

To utilize NPUs, we must understand that they are not miniature GPUs. They are fundamentally different architectures designed for extreme power efficiency.

1.1 Apple Neural Engine (ANE)

The ANE is a fixed-function, highly optimized matrix coprocessor integrated into Apple Silicon (M-series, A-series).

- The Architecture: It features a massive 2D systolic array capable of trillions of operations per second (TOPS). It is strictly optimized for INT8/FP16 execution.
- The Constraint: The ANE is rigid. It cannot execute arbitrary code. If a neural network layer (e.g., a custom CUDA kernel) is not natively supported by Apple's `MPSCNN` or `CoreML` compiler, it falls back to the GPU or CPU, introducing massive latency spikes.
- The Advantage: Unmatched power efficiency. The ANE can execute complex vision models at under 1 watt, allowing always-on inference without battery degradation.

1.2 Intel NPU (Meteor Lake & Beyond)

Intel's NPU is a dedicated AI inference engine embedded alongside the CPU and Arc GPU in modern Core Ultra chipsets.

- The Architecture: A highly programmable VLIW (Very Long Instruction Word) architecture designed for sustained AI workloads. It supports INT8, INT4, and FP16.

- The Constraint: Intel NPUs are newer to the market. The software stack (OpenVINO) is mature, but driver support for dynamic shapes (crucial for LLM KV-caches) is historically fragile.
- The Advantage: It offloads the CPU for background AI tasks (e.g., Windows Studio Effects, background blur, local RAG indexing) without waking the high-power discrete GPU.

1.3 Qualcomm Hexagon NPU

The Hexagon NPU (part of the Snapdragon platform) is a vector and tensor processor built for mobile and edge compute.

- The Architecture: A highly scalable architecture utilizing HVX (Hexagon Vector Extensions) and HTA (Hexagon Tensor Accelerator). It is deeply integrated with Qualcomm's AI Engine (QNN).
- The Constraint: The QNN SDK is notoriously complex. Compiling a standard ONNX model to Hexagon requires strict quantization (INT8) and adherence to specific operator constraints to avoid CPU fallback.
- The Advantage: Industry-leading performance-per-watt for mobile devices, enabling multi-day continuous inference on battery power.

Routing AI models across CPUs, GPUs, and NPUs introduces severe physical and software constraints that monolithic GPU deployments do not face.

2.1 The Memory Transfer Tax

In a monolithic GPU system, the model weights and activations live in GPU VRAM. In a heterogeneous system, the NPU, GPU, and CPU often have separate memory pools or caches.

- The Flaw: If Layer A runs on the GPU and Layer B runs on the NPU, the activation tensor must be copied from GPU VRAM to system RAM, and then to NPU SRAM. This PCIe/memory bus transfer takes longer than the math itself.
- The Mitigation: The compiler MUST perform operator fusion and graph partitioning. It should only assign a contiguous block of layers to the NPU if the block is large enough to amortize the memory transfer cost.

2.2 The Operator Support Boundary

NPUs are not Turing-complete. They only support specific mathematical operations (Conv2D, MatMul, ReLU).

- The Flaw: Modern LLMs utilize complex operations (Rotary Positional Embeddings, FlashAttention, dynamic KV-cache shifting). NPUs often do not support these natively. If the graph is not carefully partitioned, the system will constantly bounce between the NPU and CPU, destroying performance.
- The Mitigation: The architecture MUST utilize hardware-aware compilers (Apache TVM, ONNX Runtime) that analyze the compute graph and mathematically prove which sub-graphs can be safely offloaded to the NPU without breaking the execution context.

2.3 The Quantization Mismatch

NPUs are highly optimized for INT8 or INT4. LLMs often require FP16 for coherent reasoning.

- The Constraint: Forcing an FP16 LLM onto an INT8 NPU requires aggressive quantization (AWQ/GPTQ), which can degrade reasoning quality. Conversely, running an INT8 vision model on an FP16 GPU wastes VRAM and power.

The true power of NPU integration is realized when the runtime dynamically routes workloads based on real-time hardware availability and power budgets.

3.1 Heterogeneous Execution Routing

The local-first runtime (e.g., CALLISTO or PANTHEON edge node) MUST NOT hardcode the execution target.

- The Mechanic: At boot, the runtime queries the OS hardware abstraction layer (e.g., `IORegistry` on macOS, `WMI` on Windows, `/sys/class/devfreq` on Linux) to discover NPU capabilities and current thermal headroom.
- The Routing Decision: If the user is on battery power, route the vision/audio models to the NPU (low power). If the user is plugged in and requesting complex LLM reasoning, route the LLM to the discrete GPU (high performance) and offload the background audio transcription to the NPU.

3.2 Unified Memory Architecture (UMA)

Modern edge chips (Apple Silicon, Intel Core Ultra) utilize a Unified Memory Architecture, where the CPU, GPU, and NPU share the same physical RAM pool.

- The Advantage: Zero-copy execution. A tensor computed by the GPU can be immediately accessed by the NPU via a pointer, without physically copying the data across a bus.
- The Requirement: The runtime MUST utilize low-level APIs (Metal, Level-Zero, OpenVINO) that support shared memory handles (e.g., `IOSurface` on Apple, USM in OpenVINO) to enable true zero-copy heterogeneous execution.

The fundamental shift of NPU integration is the ability to run continuous, concurrent AI without degrading the user experience.

- Always-On Perception: The NPU handles continuous audio wake-word detection (Iris STT) and camera tracking (Medusa) at < 1 watt.
- On-Demand Cognition: When the user speaks, the GPU is instantly spun up to execute the heavy LLM reasoning loop.
- The Result: The device maintains a 15-hour battery life while running a fully autonomous, local-first AI agent. Without the NPU, the GPU would drain the battery in 2 hours.

To deploy NPU-accelerated AI in production, the Mythos Architecture enforces strict hardware abstraction and dynamic compilation.

5.1 Declarative Hardware Abstraction

The PANTHEON Nona (Planner) module MUST NOT be compiled for a specific chip. The agent logic is defined in a high-level, hardware-agnostic format (ONNX or PyTorch).

- The Mitigation: At deployment time, the local gateway utilizes an Execution Provider (ONNX Runtime / CoreML / OpenVINO) to dynamically partition the graph based on the discovered hardware. If the target is an iPad, the graph is compiled for the ANE. If the target is a Linux workstation, it is compiled for an Intel NPU or AMD NPU.

5.2 NPU Thermal Quotas

Because NPUs are passively cooled in thin-and-light laptops, sustained 100% utilization will cause thermal throttling.

- The Mitigation: The runtime MUST enforce a thermal quota. If the NPU temperature exceeds 80°C, the runtime dynamically downgrades the model (e.g., switching from a 7B model to a 3B model) or reduces the batch size to maintain the thermal envelope.

5.3 The Memory Isolation Boundary

When running an LLM on the GPU and an audio model on the NPU concurrently, they are competing for the same Unified Memory bandwidth.

- The Mitigation: The OS MUST enforce Quality of Service (QoS) memory prioritization. The interactive LLM (foreground) is given high-priority memory access, while the background NPU tasks are throttled to prevent memory bus saturation and OOM crashes.

Defaulting to the GPU for all AI tasks is an artifact of the cloud era. For sovereign, local-first AI to succeed on edge devices, the architecture must embrace heterogeneous compute. By dynamically discovering the NPU, partitioning the compute graph, and enforcing zero-copy memory boundaries, we transform edge devices from battery-draining space heaters into ultra-efficient, continuously reasoning autonomous agents.

```
A GPU is a power plant; an NPU is a watch battery.
A CPU is a generalist; an NPU is a specialist.
A monolithic graph is a bottleneck; a partitioned graph is a flow.

The model must adapt to the silicon; the silicon must not adapt to the model.

> Compute may be heterogeneous.
> Sovereign efficiency must be absolute.
```

End of Research Note RES-018

9. Source Register

Source	Authority and Status	Freshness Control
Windows ML Overview https://learn.microsoft.com/en-us/windows/ai/new-windows-ml/overview	Primary platform documentation Living - fast moving Published: 2026	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
OpenVINO NPU Device Documentation https://docs.openvino.ai/2026/openvino-workflow/running-inference/inference-devices-and-modes/npu-device.html	Primary runtime documentation Current version Published: 2026	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.
Apple Core ML Documentation https://developer.apple.com/documentation/coreml	Primary platform documentation Living Published: Living	Validated: 2026-07-22 Review on material revision, withdrawal, supersession, or scheduled report review.

10. Lineage, Review, and Publication Record

Record	Value
Original source file	RES-018_NPU_Integration_Local_Inference_Acceleration.md
Original source words	1617
Upgrade type	Enterprise research report; source and freshness validation; limitations; adoption decision; reproducibility plan
Generated on	2026-07-22
Current status	Candidate Research Report
Publication authority	Formal Mythos approval not yet recorded
Next review	90 days or event-driven trigger, whichever occurs first