Embedded Neural Firmware

ENF Technical Note 01
The Embedded AI Crisis and the Case for Sealed Neural Firmware

— **Danish Z. Khan**, Founder, ENFSystems LLC
Version 1.0 · December 2025 · Status: Architecture / Concept Note

Table of Contents

Abstract

Embedded AI has moved from simple, static firmware into cloud-dependent, update-driven systems that expand attack surface, weaken determinism, and create lifecycle fragility when vendors, networks, or update pipelines fail. This technical note frames that failure mode—energy, security, and privacy limits of the mainstream IoT/LLM/OTA paradigm—and proposes **Embedded Neural Firmware (ENF)** as an architectural exit. ENF is a firmware-class intelligence stack: **a task-specific, quantized neural agent sealed into ROM/flash**, executed **offline**, **OS-free**, and **without OTA**, designed to run within **harvested-energy** constraints. Trust is anchored in hardware (e.g., **PUF-rooted identity**) rather than remote infrastructure, and behavior is bounded by deterministic control flow and static I/O envelopes. We outline ENF's design commitments, technical grounding (determinism, lifecycle resilience, energy autonomy, security), core contributions, and an optional path toward multi-ENF ecosystems via strictly bounded, non-cloud coordination.

1. The Embedded AI Crisis

1.1. Contextual Background

The exponential proliferation of connected devices has transformed the embedded landscape over the past two decades. While enabling ubiquitous sensing and actuation, this surge has also introduced escalating firmware complexity, runtime dynamism, and ever-expanding attack surfaces. Firmware is no longer static logic governing device behavior but increasingly behaves as a mutable execution environment subject to frequent over-the-air (OTA) reconfiguration, telemetry exposure, and dependency on remote services.

Verizon's 2024 Data Breach Investigations Report (DBIR) identifies firmware as a persistent blind spot in enterprise and consumer devices alike, noting that embedded components remain highly vulnerable to exploitation through unpatched code, default credentials, and failed update mechanisms. Many devices implicated in security incidents were found to be unsupported, no longer maintained, or abandoned entirely by their original vendors—despite continued deployment in critical infrastructure and consumer environments (Verizon, 2024).

The proliferation of OTA-enabled logic has not only amplified vulnerability but introduced lifecycle mismatches. Devices now remain operational long after vendor or software support ceases, creating an unbridgeable chasm between deployment horizon and maintenance infrastructure. In effect, the modern IoT model has regressed to a form of digital planned obsolescence—devices are born disposable, non-verifiable, and perpetually dependent on cloud-centric intervention.

1.2 Identified Problems

First, the dependence on OTA updates and dynamically loaded firmware creates unacceptable lifecycle fragility. A failure in remote infrastructure, vendor abandonment, or network instability leaves devices unable to function safely or verifiably. This violates principles of critical system design as defined in both ISO 26262 and IEC 61508, which emphasize determinism, pre-validated system response, and isolation from uncertain runtime variables (Divakarla, 2017).

Second, firmware vulnerabilities—already difficult to detect—are rarely remediated. As noted in the DBIR, IoT devices often remain unpatched for years, if ever. The prevailing update model assumes trust in both network delivery and end-user cooperation, neither of which can be guaranteed in practice. In contrast, ENF seeks to eliminate the update vector entirely by delivering sealed, ROM-resident models that cannot be modified post-deployment.

Finally, and most critically, mainstream embedded AI has embraced architectural assumptions incompatible with autonomy, determinism, and privacy. As Brundage et al. (2020) argue, emergent AI behavior, coupled with opaque update pipelines and reliance on third-party data capture, undermines long-term alignment and user trust. The very nature of modern AI—cloud-bound, subscription-driven, and general-purpose—violates the core principles of embedded system integrity.

ENF responds to these failures by rejecting the entire OTA/cloud paradigm. Rather than patching flawed assumptions, it reconstructs embedded intelligence from first principles: static,

task-specific, and sealed at manufacture. In doing so, ENF offers an architectural exit from the crisis now endemic to AI-enabled devices.

2 Philosophical Rejection of Cloud AI

2.1 Contrast with LLM and OTA Paradigms

Contemporary machine learning, particularly in its large model and foundation model incarnations, is deeply entangled with cloud infrastructure, runtime mutability, and centralized control. Most production AI systems are deployed as remote services, with inference hosted in datacenters and updated frequently via opaque back-end channels. OTA updates—both firmware and model-level—are positioned not as optional upgrades, but as essential components of functionality, trust, and performance.

This paradigm stands in direct opposition to ENF. ENF rejects the assumption that intelligence must be mediated by infrastructure, networks, or third-party control. It dismisses the need for runtime reconfiguration, abstraction layers such as operating systems, or any dependence on Internet Protocol (IP)-based interfaces. Unlike LLM-based systems, which scale through parameter bloat and cloud intermediation, ENF scales horizontally—through mass deployment of minimal, sealed neural agents optimized for a single function, with no abstraction or variability.

TinyML literature, such as Lin et al. (2024) and Kallimani et al. (2024), acknowledges the persistent trade-offs in bringing inference to edge devices. However, most still assume that OTA provisioning, telemetry, and model replacement are part of the deployment lifecycle. ENF explicitly denies this architecture and instead adopts a one-time programming model. Models are quantized, tested, and flashed at manufacture, then executed deterministically without external oversight or alteration.

2.2 Ethical Position

ENF is more than a technical divergence from the OTA paradigm; it is a philosophical and ethical response to the failures of centralized AI governance. As Brundage et al. (2020) note, opacity in model updates and inference behavior undermines user trust and long-term system alignment. ENF reframes autonomy not as a software switch or configuration parameter, but as a condition of physical instantiation: autonomy as a hardware-anchored right.

By sealing logic in immutable ROM and anchoring trust in Physical Unclonable Functions (PUFs) rather than over-the-air trust chains, ENF creates verifiable agents whose behavior is fully bounded, inspectable, and predictable. Trust, privacy, and determinism are not contingent on subscription models or terms-of-service. They are enforced in silicon, not negotiated through software.

In doing so, ENF disavows the surveillance-capable, cloud-synchronized architectures that dominate contemporary AI. It aligns instead with a human-centric design principle: devices should serve their users transparently, privately, and permanently—without requiring trust in unseen intermediaries. This inversion of control—returning authority from remote servers to embedded logic—is the foundation of the ENF ethos.

3 ENF Proposition

3.1 Embedded Neural Firmware Defined

Having rejected cloud-centric AI principles, ENF now proposes an alternative embedded architecture grounded in sealed neural execution.

Embedded Neural Firmware (ENF) is proposed as a novel firmware-class intelligence stack that replaces traditional rule-based logic and runtime software with task-specific, sealed neural agents executed directly from ROM or flash. These models are deployed statically, require no operating system or OTA updates, and operate entirely offline on harvested energy. Unlike TinyML or cloud-extended agents, ENF devices are silent, deterministic, and aligned to hardware-only boundaries. They neither generalize nor expand — they perform one function only, and do so with provable trust, over decades.

Each ENF device is tasked with a narrowly scoped function—e.g., analog signal interpretation, state classification, or control output regulation. The model responsible for this task is trained offline, aggressively quantized (typically to 8-bit or lower precision), and embedded directly into ROM or flash memory during manufacture. Execution occurs in a closed-loop deterministic fashion, with all inputs and outputs statically bounded, enabling traceable, testable, and lifecycle-persistent behavior (Lin et al., 2024; Heath, 2003).

This model eliminates the need for intermediate software abstractions such as RTOS, scheduler-based multitasking, or runtime memory management, which are typically introduced for code portability or application extensibility. By fixing the model in silicon and eliminating interpretive software layers, ENF guarantees consistent timing, avoids dynamic memory pitfalls (Heath, 2003), and aligns directly with safety-critical certification principles outlined in ISO 26262 and IEC 61508 (Divakarla, 2017).

3.2 Design Commitments

ENF formalizes five key design commitments that differentiate it from all prevailing embedded and AI paradigms:

1. **Fully Offline**: ENF devices operate without any IP stack, network interface, or external telemetry. There is no over-the-air (OTA) update path, no cloud fallback, and no runtime dependency on remote systems. This ensures long-term operational continuity, as validated by the persistent vulnerabilities in network-reliant systems documented in the 2024 DBIR (Verizon, 2024).
2. **OS-Free**: ENF systems exclude real-time operating systems, context switching, and dynamic memory management. The firmware is minimal, often consisting solely of a static main loop invoking the neural agent on bounded inputs. This exclusion aligns with IEEE Std 2050-2018, which details the added complexity and verification burden introduced by even minimal RTOS features.
3. **Energy-Minimal**: ENF is designed to operate on ambient energy harvesting sources, including piezoelectric, photovoltaic, and thermoelectric generation. Typical energy budgets range in the μW regime, requiring ultra-low-power inference routines and non-reliance on periodic network synchronization (Khaligh & Zeng, 2013). While ENF executes only under sufficient energy and valid trust, additional analog or passive trigger

mechanisms may be included to guarantee responsiveness under human intent, without requiring background execution, polling, or clocks.

4. **Tamper-Proof**: Each ENF unit is sealed both logically and physically. Cryptographic identity and secure boot are anchored in Physical Unclonable Functions (PUFs), eliminating the need for secret key storage and making the system resilient to invasive attacks (Herder et al., 2014). Firmware is written once to ROM and cannot be altered without complete hardware replacement.
5. **Human-First**: ENF is designed without surveillance capabilities, telemetry sinks, or user profiling subsystems. Unlike cloud-synchronized AI, ENF agents never exfiltrate data or evolve behavior through interaction. Privacy and safety are guaranteed by structural design, not policy—no data is collected, and no subscription-based intelligence loop exists (Brundage et al., 2020).

By adhering to these commitments, ENF defines a new architectural category: one where embedded cognition is not merely reduced AI but a complete inversion of modern ML deployment philosophy. It is sealed, specialized, and sovereign by construction.

While the ENF architecture is grounded in deterministic, low-power neural execution, its distinction from traditional embedded logic can be easily misunderstood. Figure 1 illustrates this difference using a simple task: turning on a light. Whereas conventional systems rely on rigid thresholds (e.g., “if sound > 60 dB”), ENF interprets sensory input through a quantized model trained to recognize the pattern of a clap or gesture. This enables more reliable, human-aligned behavior without the fragility of hard-coded conditions or remote updates.

ENF Chip Execution Flow – Deterministic Offline Inference

Figure 3.3. ENF Chip Execution Flow – Deterministic Offline Inference.

This diagram illustrates the sealed, offline execution path of an Embedded Neural Firmware (ENF) system. The flow begins with a voltage threshold trigger (via Schmitt logic), followed by a secure boot process anchored to a Physically Unclonable Function (PUF). Sensor input is captured and normalized before deterministic inference is performed using a quantized INT8 model embedded in ROM. Output is gated through bounded logic, activating only if conditions are met. A fallback mechanism via ROM-based FSM handles anomalies or watchdog triggers. The system then returns to sleep mode, awaiting the next harvested energy threshold. No dynamic memory, cloud dependency, or runtime variability exists.

Key Architectural Commitments

- **No cloud stack** — IP protocols, OTA pipelines, and telemetry subsystems are entirely removed.
- **No OS layer** — Bare-metal, RTOS-free logic ensures static scheduling and deterministic timing (IEEE, 2018).
- **No model drift** — Static, quantized inference graphs with sealed weights prevent behavior mutation over time.
- **No interface loops** — ENF agents require no UI, prompts, feedback capture, or runtime interaction.
- **No identity leakage** — ENF never captures or stores user identifiers, metadata, or behavioral profiles (Brundage et al., 2020).
- **PUF-bound identity** — Physical Unclonable Functions (PUFs) replace key storage with hardware-anchored trust (Herder et al., 2014).
- **Bounded adaptation** — Only micro-adjustments (e.g., counter shifts, margin tuning) occur, always within defined deterministic envelopes.

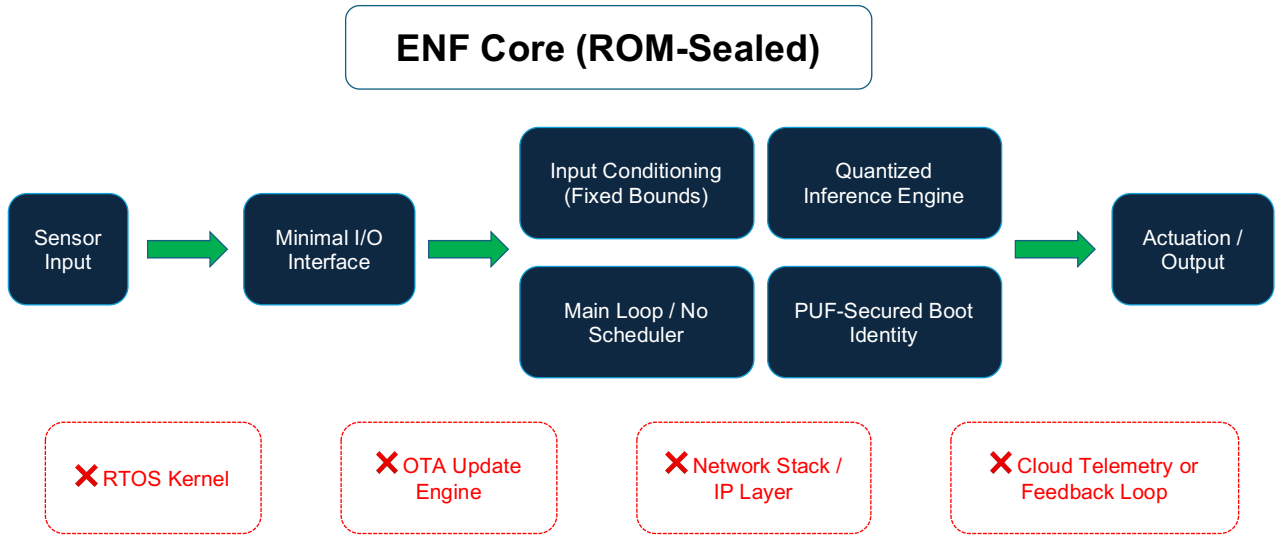

Figure 3.4: ENF Hardware Execution Stack

ENF Architecture Stack. All dynamic software layers are eliminated in favor of sealed, ROM-resident neural logic. The execution model includes sensor input, quantized inference, and deterministic actuation—all running without OS, network, or OTA pathways.

4 Technical Justification and Grounding

4.1 Deterministic Behavior

ENF's architectural commitment to determinism is grounded in foundational principles of embedded systems design and the formal exclusion of real-time operating systems (RTOS). As Heath (2003) explains, deterministic execution in embedded contexts requires elimination of dynamic memory allocation, interrupt-driven scheduling, and runtime configuration layers. By avoiding RTOSs altogether, ENF removes task switching, priority inversion, and software-induced jitter—issues underscored in IEEE Std 2050-2018. ENF's loop-bound, bounded-input execution model guarantees fixed timing, traceable I/O paths, and behavior that remains constant across deployments, via the Deterministic Quantized Agent (DQA) model class.

4.2 Lifecycle Resilience

The lifecycle constraints of safety-critical systems require predictable, certifiable operation for decades without intervention. ISO 26262 and IEC 61508 mandate system design approaches that minimize runtime variability and formally verify system integrity across lifespan phases (Divakarla, 2017). ENF's design aligns precisely with these principles: models are fixed in ROM, validated offline, and executed without mutable state. There is no OTA path, no dynamic update logic, and no trust dependency on runtime infrastructure. This enables 30–50-year deployments of sealed devices without risk of firmware drift or certification invalidation.

4.3 Energy Autonomy

ENF targets deployments in environments where wired power and battery maintenance are infeasible. The design is substantiated by research into micro-scale energy harvesting modalities such as piezoelectric, photovoltaic, and thermoelectric generation (Khaligh & Zeng, 2013). These sources typically yield 10–500 μW/cm²—sufficient for inference-only neural execution when paired with static models and sub-milliwatt hardware. ENF's ROM-sealed models, lacking any runtime training or memory allocation overhead, are ideal for harvesting-constrained scenarios.

4.4 Security Model

ENF enforces tamper-resistance and cryptographic identity through Physical Unclonable Functions (PUFs), which derive unique secrets from physical variations in silicon without storing keys (Herder et al., 2014). This hardware-rooted trust model resists side-channel extraction, eliminates the need for secure storage, and prevents re-flashing or unauthorized modification. Unlike traditional secure boot, which often relies on key escrow or certificate chains, ENF derives trust intrinsically at power-on, enabling long-term resilience without external trust provisioning.

4.5 Task Boundedness

ENF's narrow-function neural agents are validated by the TinyML literature, which demonstrates feasibility of inference-only deployment on MCU-class hardware with strict memory and compute budgets. Lin et al. (2024) confirm that even highly compressed MobileNet variants

exceed MCU RAM limits unless redesigned for static inference. Kallimani et al. (2024) highlight the absence of reproducible deployment pipelines in TinyML and the challenges in supporting even minimal training or runtime adaptation on embedded platforms. ENF accepts these constraints not as limitations but as design axioms: every agent performs one task, bounded in scope, validated pre-deployment, and immutable in execution.

In sum, ENF is technically justified across all axes of embedded system rigor: determinism, lifecycle safety, energy independence, cryptographic trust, and architectural minimalism.

5 Contributions of ENF

5.1 Architectural

ENF introduces a fundamentally new class of embedded intelligence: task-bounded, sealed neural firmware that functions without an operating system, without dynamic memory, and without runtime abstraction layers. It rejects the traditional software stack, replacing it with a ROM-resident quantized neural agent that executes deterministically across bounded I/O interfaces. This architecture is unprecedented in that it offers cognition embedded at the firmware level—not as a feature of updatable software but as a permanent, silicon-resident behavior pattern (Heath, 2003; IEEE, 2018).

ENF redefines what constitutes a firmware payload: not rule-based state machines or interrupt-driven logic, but pre-trained, fully quantized inference graphs with no modifiability post-deployment. The firmware becomes a frozen neural structure—unmodifiable, testable, and self-contained.

5.2 Technical

From a technical standpoint, ENF formalizes the ROM-sealed deployment of statically quantized neural networks. These models are executed without RAM-based buffers, dynamic heap allocation, or runtime kernel support. Bounded execution, minimal memory footprint, and predictable behavior make ENF compatible with Cortex-M-class MCUs and sub-milliwatt power budgets validated by TinyML research (Lin et al., 2024; Kallimani et al., 2024).

The deployment methodology is immutable by design. Each ENF device is burned at manufacture with a fixed-function model and a secure boot identity tied to its physical fingerprint via a PUF. This approach enforces not only architectural determinism but also long-term digital continuity, satisfying both safety and security mandates (Herder et al., 2014; Divakarla, 2017).

5.3 Ethical

ENF reframes embedded intelligence as an ethical choice. It positions autonomy not as a configurable software attribute but as a physically assured trait of the device itself. By eliminating all cloud links, telemetry, OTA updates, and behavioral drift, ENF ensures that no private user data is collected, stored, or transmitted. This stands in direct contrast to surveillance-adjacent paradigms inherent in OTA/LLM architectures (Brundage et al., 2020).

Trust is enforced by structure—not by third-party audits or runtime verification schemes. Privacy is guaranteed because ENF devices are structurally incapable of surveillance.

5.4 Sustainability

ENF supports embedded cognition without requiring batteries, network maintenance, or post-deployment software support. Through µW-scale operation, ambient energy harvesting (Khaligh & Zeng, 2013), and total software immutability, it offers a 30–50-year operational lifecycle with zero upkeep. There is no subscription, no reboot loop, no cloud service decay.

This makes ENF not only a sustainable technical model but also an ecological and economic one. It restores embedded computing to its minimal, resilient roots while extending its utility through neural miniaturization.

ENF is thus more than a new design—it is a new compact: between designer, device, and user—one sealed in hardware, not software, and sustained through embedded architectural truth.

ENF vs. Cloud/LLM-Based AI — Architectural Comparison

Dimension	ENF (Embedded Neural Firmware)	LLM / Cloud-Centric AI
Execution Location	Local MCU / ROM-bound neural agent	Remote inference via datacenters or cloud edge
Update Model	No updates — fixed at manufacture (immutable)	Continuous OTA updates, retraining, and version drift
Energy Profile	µW-scale operation, harvesting-capable	kW-scale backend compute, always-on connectivity
Network Dependence	None — operates fully offline without IP stack	Full-time network dependency for inference, sync, and updates
Runtime Variability	None — deterministic, bounded behavior	High — model drift, latency, abstraction layers
Trust Anchor	PUF-secured identity and sealed model logic	Credential stores, certificates, remote attestation
Ethical Footprint	Human-first: no telemetry, profiling, or behavioral adaptation	Surveillance-adjacent: data capture, profiling, behavior shaping
Lifecycle	30–50-year sealed operation, no cloud dependency	Tied to cloud service availability, vendor support, energy supply
System Complexity	Minimal — no RTOS, no abstraction, no OTA pipeline	High — OS stack, containerized inference, multi-layer model management
Design Philosophy	Embedded cognition as hardware truth	AI as cloud-mediated, service-based abstraction
Distributed Precision	Enables “Offline Collective Intelligence” via ENF-Sync: peer-shared, PUF-authenticated, context-gated tuning signals that reinforce local task accuracy without cloud or code changes.	Requires centralized coordination, retraining, and orchestration for distributed behavior.

ENF is not a scaled-down AI — it is a redefinition of intelligence itself: permanently embedded, bounded in scope, human-aligned, and capable of growing in precision across time without ever needing to ask for input.

6 ENF Collective Intelligence and Ecosystem Behavior

6.1 Toward Multi-ENF Ecosystems

Embedded Neural Firmware (ENF) devices are engineered to execute a single, deterministic task in isolation, with quantized neural logic sealed directly in ROM. Although this excludes runtime adaptability and system-wide abstraction, it does not preclude architectural composition. In high-integrity embedded environments, multiple ENF units can be physically co-located—each sealed, each sovereign—contributing to a broader deterministic system.

For instance, a sanitation system may embed three discrete ENF agents: one classifying bin fullness, one detecting odor profiles, and one managing compaction timing. Each operates independently, rooted in a secure boot cycle and PUF-derived identity, with no shared scheduler, memory, or network. Their alignment arises not through software integration but from environmental synchrony and physical proximity—an expression of parallel determinism.

This compositional model extends naturally to embedded infrastructure. In a distributed soil sensing network, for example, each ENF may be trained on a specific environmental factor: moisture thresholds, pH levels, or light exposure. These chips do not share models or transmit packets; instead, they form a field-level mesh of static, domain-locked judgment. System behavior emerges through coverage—not communication.

Such deployment patterns align with safety and assurance mandates outlined in ISO 26262 and IEC 61508, where modular verification, bounded failure domains, and deterministic behavior are essential. Each ENF chip can be pre-certified, energy-modeled, and physically isolated—eliminating systemic fragility and enabling decades-long continuity.

ENF scalability, therefore, is not horizontal in code abstraction, but physical in node multiplication. Complexity is achieved not through dynamic orchestration, but by increasing the number of purpose-fixed agents. This reframing of embedded intelligence is consistent with ultra-low-power paradigms and ambient energy harvesting models validated by Matiko, Grabham, Beeby, and Tudor (2014). It situates ENF not as a flexible AI runtime, but as a composable, cooperative substrate for embedded cognition.

6.2 Offline Collective Intelligence

While each ENF device operates deterministically and in isolation, collective deployments can yield emergent accuracy and system-wide tuning without the use of cloud infrastructure, OTA logic, or shared memory. This form of system intelligence, called Offline Collective Intelligence (OCI), refers to the behavioral adaptation that arises when ENF units are deployed at scale in spatial proximity.

Each ENF contributes bounded inference to its physical context. When deployed in dense networks (e.g., environmental sensors, security perimeters), behavior across the group can be indirectly tuned through ambient feedback. For example, multiple ENF agents exposed to recurrent environmental variance may independently adjust internal counters or inference confidence margins — such as timing delays, filter thresholds, or sampling frequency — within strict, quantized envelopes.

This is not learning in the machine learning sense. There is no model mutation, gradient propagation, or data fusion. Rather, ENF devices respond to pre-approved state changes that emerge from bounded internal metrics. Collectively, the system converges toward higher functional harmony, even though no communication is symbolic or cloud-mediated. It is a form of colony cognition: emergent behavior from individually deterministic, privacy-sealed agents.

6.3 ENF-Sync: Communication Without Cloud

To facilitate environmental awareness or inter-agent resonance, ENF devices may optionally emit non-IP signals via physical phenomena such as modulated IR pulses, inductive field changes, or mechanical vibrations. These signals are strictly bounded to 32–64 byte payloads, interpreted only by other ENFs that share a manufacturing lineage and PUF-anchored trust domain.

Each signal is authenticated by a physically unclonable function signature, confirming device provenance without relying on cryptographic key storage. Importantly, ENF-Sync does not carry symbolic information, model weights, or device logs. It enables presence sensing, clock harmonization, or group state transitions — not data exchange. These communication channels operate entirely outside traditional protocols (e.g., TCP/IP, BLE), making ENF systems immune to network-side attacks.

6.4 Growth Within Behavioral Envelopes

Despite their sealed nature, ENF agents can refine task performance over time through local, bounded memory registers. This form of growth is implemented using FRAM or MRAM buffers preallocated for deterministic metrics: event counters, timing shifts, or false-positive statistics. These values do not alter the core neural model but adjust inference boundaries within mathematically defined tolerances.

For example, a motion classification ENF may shift its activation threshold ±1 step if its rejection count exceeds a static bound. This micro-adjustment is logged, capped, and reversible. No training occurs post-deployment. Instead, the agent becomes incrementally better at rejecting edge cases without ever acquiring new behavior. This adaptive refinement mirrors biological systems like muscle memory or behavioral fatigue — not cognitive learning.

6.5 ENF Language and the Silence Mesh

Collectively, ENF deployments form what may be described as a silence mesh: a network of cognition that emits no traffic unless physically triggered. Unlike conventional mesh networks that synchronize via radios, ENF units encode state through behavior, timing, and physical response patterns. A change in lighting delay, vibration emission, or sampling frequency may signal a collective state shift — not via explicit protocol, but through patterned resonance.

This non-symbolic language aligns with natural systems: root signaling in plants, synchronized movement in animal flocks, or pressure gradients in chemical systems. ENF's language is not written, transmitted, or intercepted. It is observed — only by other ENFs, only locally, only within physical domain constraints. This communication model preserves privacy, scales without congestion, and conforms to the foundational doctrine of ENF: sealed, sovereign, and silent by design.

7 References

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., ... & Amodei, D. (2020). *Toward trustworthy AI development: Mechanisms for supporting verifiable claims*. arXiv:2004.07213.

Divakarla, K. P. (2017). *ISO 26262 and IEC 61508 Functional Safety Overview*. NXP Semiconductors.

Heath, S. (2003). *Embedded Systems Design* (2nd ed.). Newnes.

Herder, C., Yu, M. D., Koushanfar, F., & Devadas, S. (2014). Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE, 102*(8), 1126–1141.

IEEE Standards Association. (2018). *IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems*. IEEE Std 2050-2018.

Kallimani, R., Pai, K., Raghuwanshi, P., & Iyer, S. (2024). TinyML: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications, 83*, 29015–29045.

Khaligh, A., & Zeng, P. (2013). Review of the application of energy harvesting in buildings. *Measurement Science and Technology, 25*(1), 012002.

Lin, J., Zhu, L., Chen, W.-M., Wang, W.-C., & Han, S. (2024). Tiny machine learning: Progress and futures. *arXiv preprint* arXiv:2403.19076.

Verizon. (2024). *Data Breach Investigations Report (DBIR)*. https://www.verizon.com/business/resources/reports/dbir/