Embedded Neural Firmware (ENF): A Deterministic, Offline “Neural BIOS” for Batteryless Devices

Table of Contents

Abstract

Embedded Neural Firmware (ENF) reframes device software as a sealed **Neural BIOS**—offline-only, immutable, and bound to silicon—**and as a framework that specifies, builds, and proves it**. ENF compiles a task-specific, quantized model into ROM/Flash and executes bare-metal via a single FSM with static memory (no heap/GC or recursion) and per-state WCET, yielding bit-for-bit reproducibility and tractable verification. Identity and provenance derive from a Physically Unclonable Function (PUF), eliminating stored secrets and gating actuation through measured boot. Powered by harvested energy and supercapacitors under explicit thresholds (`V_on`/`V_safe`/`V_cut`), ENF targets years-long autonomy while collapsing the remote attack surface (no IP stack, OTA, or telemetry). We present ENF's invariants and the **ENF Framework**—a locked, reproducible toolchain and a **Conformance Pack** (signed Manifest, firmware/model hashes, PUF-bind record, SBOM, test vectors)—and ground the approach in three cloud-free deployments with a clear threat and limitations posture (recall/replace, dataset governance). Compared with TinyML and Cloud/Edge-AI, ENF offers privacy by architecture and assurance through reproducible builds—evidence you can verify today.

Section 1: Introduction & Problem

Embedded systems increasingly inherit the mutability and reachability of the internet: firmware that once behaved as a finite, verifiable artifact now functions as a service composed of over-the-air (OTA) updaters, certificate chains, and telemetry stacks. This shift creates a widening mismatch between the **deployment horizon** of devices (years) and the **maintenance reality** of infrastructure (months), while expanding the attack surface to include every mechanism intended to "keep devices current." In unattended or energy-constrained estates, those mechanisms become liabilities: updaters fail, certificates expire, network paths degrade, and fleets diverge. Industry experience echoes this pattern—firmware remains a persistent blind spot and is frequently left unsupported in the field.

Cloud/OTA risk — reachability by design. Prevailing TinyML/IoT practice assumes IP connectivity and periodic software or model replacement. Each assumption opens a failure class: the updater behaves like an interpreter; telemetry forms a data-exfiltration path; and version skew undermines reproducibility. When updates are optional, configurations fragment; when mandatory, outages and rollback races appear. Operators inherit non-determinism that is hard to audit and costly to bound across a device's lifetime.

Determinism debt and safety. Safety-oriented domains prize traceability, bounded timing, and static resource usage. Mainstream embedded-AI stacks introduce RTOS scheduling, dynamic allocation, and runtime loaders—features that resist whole-program reasoning and long-term certification. Under energy or access constraints, "patch-to-safety" is operationally fragile compared to **non-reachability** (removing the update and network paths entirely), the direction ENF adopts as a deliberate departure from TinyML assumptions.

Privacy burden. Cloud-mediated ML normalizes collection and remote inference: raw or derived signals exit the device, creating durable risks contingent on off-device actors. In homes, classrooms, and public spaces, **non-collection** should be architectural, not merely policy-declared—an ethos supported by ENF's security model, which explicitly excludes OTA channels.

Scope and contribution. We propose **Embedded Neural Firmware (ENF)**: sealed, task-specific neural logic that executes offline on bare metal with static memory and bounded timing; identity and provenance are anchored to silicon via a Physically Unclonable Function (PUF); and power is budgeted for harvested operation (supercap + analog-qualified wake). This paper contributes a normative specification (MUST/SHALL invariants), a minimal hardware/firmware path, a **conformance checklist with cryptographic test vectors**, and a structured comparison against TinyML/cloud patterns. We require **reproducible builds**—a bit-for-bit ROM image derivable from a signed manifest and locked toolchain—to enable third-party attestation without device connectivity. The central claim is that **removing reachability**—no OTA, no network stack, no interpreter—collapses entire classes of failure and attack, **reinstating determinism as a foundational guarantee**.

Non-goals. *ENF is not a general-purpose runtime or a learning platform. It does not support OTA, dynamic model loading, federated learning, or telemetry export, and it is not intended for tasks requiring open-ended adaptation or high-bandwidth communication. ENF's charter is narrower: fixed-scope functions, sealed execution, verifiable provenance, and energy-autonomous duty cycles.*

Section 2: Definition & Invariants

Embedded Neural Firmware (ENF) is a *Neural BIOS*: a sealed, task-specific neural program compiled into immutable memory (ROM/Flash) and executed on bare metal. An ENF instance performs exactly one bounded function within a declared operating envelope; it is offline-only, deterministic, and **identity-anchored to the silicon**.

Formally,

$$\mathcal{E} = \langle \text{HW,Model,Image,PUF,Manifest} \rangle$$

where **HW** specifies the MCU and power-island topology; **Model** is a quantized, fixed-shape network; **Image** is the linked, immutable binary (weights + control FSM + constants); **PUF** provides device-unique secrets that are derivable but not stored; and **Manifest** is the public build record (hashes, maps, WCET bounds, **toolchain lockfile/build-container digest**), the **ENF framework parameters**—**T**ask, **P**ower model, **S**ecurity level, **F**allback, **C**ommunication scope—and any **calibration constants with valid ranges**, for attestation and audit.

2.1 Normative Invariants (MUST/SHALL)

1. **Offline-only operation**
   The ENF image **MUST** exclude any IP stack, socket API, or radio driver reachable from the control path; the design **SHALL NOT** include OTA channels, RPC endpoints, or

telemetry sinks. Physical debug/programming interfaces **MUST** be irreversibly disabled (e.g., fuse-off) after manufacture.

2. **Sealed ROM image**
   The linked firmware **MUST** be immutable in fielded devices. The boot path **SHALL** cryptographically **measure** immutable regions (e.g., SHA-256/BLAKE3) and enable actuation only on a verified measurement; on measurement failure, the device **SHALL fail-closed** (no actuation) and remain so until a verified image is present (e.g., after a power-cycle/restart).

3. **Static memory discipline**
   Execution **MUST** avoid dynamic allocation (no heap/GC). All buffers and tensor shapes **SHALL** be fixed at link time; **no recursion or unbounded loops**. A compile-time memory map **MUST** enumerate ROM/RAM/OTP regions with byte ranges.

4. **Deterministic timing (WCET)**
   The control path **MUST** be an explicit FSM: **Wake → Sense → Infer → Actuate → Sleep**. Each state **SHALL** have a documented WCET; **no preemptive multitasking** is permitted, and the **only** interrupt service routine allowed **within the ENF domain** is the **threshold-qualified wake** (e.g., Schmitt-qualified) from the always-on energy island. Clocks/timers/DMA **MUST** be configured so end-to-end latency and WCET are auditable; **no external time synchronization** is permitted.

5. **PUF-anchored identity & provenance**
   Device keys **MUST** be derived from a PUF; secrets **SHALL NOT** be stored at rest. If helper data is used for PUF reconstruction, its integrity and binding to the measured image **SHALL** be specified.

6. **Published build artifacts (attestation)**
   Each build **MUST** publish a signed **Manifest**: firmware hash, model hash, toolchain/version, **build-container digest**, ROM/RAM map, WCET bounds, sensor/actuator limits, **energy thresholds (V_on, V_safe, V_cut)**, calibration constants/ranges, and (if applicable) helper-data format. A **conformance checklist MUST** accompany releases, enabling third-party verification without device connectivity.

7. **Numeric determinism**
   The implementation **MUST** use **fixed-point/quantized** or otherwise **bit-exact** numerics for inference; compiler flags, kernel versions, and math libraries **SHALL** be locked in the Manifest to prevent nondeterministic code generation. The **rounding mode** and **subnormal/denormal handling SHALL** be declared; fused multiply-add (FMA) **MUST** be either fixed to a bit-identical path or disabled if not bit-exact. Mixed-precision FP paths **MUST NOT** introduce cross-build or cross-device nondeterminism.

8. **Energy & safety thresholds**
   The always-on domain **MUST** define and publish **V_on, V_safe, V_cut**; the control FSM

SHALL fail-closed (no actuation) on brown-out and **SHALL** specify reset/retry timing. Any wake event **MUST** be **threshold-qualified** by the energy monitor; no other asynchronous paths are permitted.

9. **Reproducible builds & test vectors**
   The firmware image **MUST** be **bit-for-bit reproducible** from the **Manifest** (including a toolchain lockfile and **build-container digest**) and model hash. Each release **SHALL** include **test vectors**: (a) expected boot-measurement digest over immutable regions, (b) example PUF helper data + verification outcome (accept/deny), and (c) WCET evidence (method + bound values).

10. **Local time base**
    The device **MUST** operate from a local time base with declared drift bounds in the Manifest; the control path **SHALL NOT** depend on external clocks, NTP, or radio time beacons.

11. **No entropy in the control path**
    Hardware RNG or nondeterministic entropy sources **MUST NOT** influence the control FSM or inference outputs; if present, they **SHALL** be disabled or quarantined from the ENF domain.

2.2 Prohibited Features (for clarity)

Operating systems or interpreters; dynamic model loading or training; remote configuration; background daemons; telemetry/log export; writable filesystems beyond fixed manufacturing records; network time synchronization or external time beacons; nondeterministic entropy fed into control or inference; preemptive multitasking or non-wake ISRs within the ENF domain.

Rationale

Eliminating reachability collapses major vulnerability classes; sealing code removes drift; static shapes and WCET enable whole-program reasoning; PUF anchoring provides per-die provenance; and energy thresholds bind safety to measurable limits. Together, these properties enable *certify-once, run-long* behavior under harvested power and align with ENF's *Neural BIOS* posture described in the overview.

Section 3: System Architecture

This section details the concrete architecture that realizes ENF's invariants. It encompasses hardware design (dual power islands, PUF, sensor/actuator paths), sealed neural firmware logic, the secure-boot trust chain, the energy path and thresholds, and the deterministic execution and safety model. All components are specified for implementation without an operating system, network stack, or runtime interpreter.

3.1 Hardware Layer

3.1.1 Dual Power-Island Topology

- **Always-on energy island (AON):** Harvester → PMIC → supercapacitor (**Vcap**) monitor with Schmitt-qualified comparator; wake-gate FSM; optional tamper pins. This island must operate in the sub-µW regime and remain energized whenever ambient energy is available. **Tamper mesh or pin inputs SHALL force the INF island off and increment a non-volatile tamper counter.**
- **Gated inference island (INF):** MCU + ROM-sealed model and control FSM + minimal peripherals required for sensing/actuation. The INF island is power-gated and enabled only when **Vcap ≥ V_on** and the boot measurement verifies.
- **Isolation & sequencing:** Power switch (PFET/ideal diode controller) between AON and INF; enable line controlled by the AON FSM. A monotonic **enable → measure → release** sequence prevents partial boots.

Design targets (indicative):

- AON quiescent current: < 2–5 µA (including PMIC + comparator).
- INF active current: model-dependent; design for mA-scale pulses with millisecond dwell.
- Supercap leakage: < 1–5 µA; ESR compatible with peak INF draw.

3.1.2 Energy Storage and PMIC

- **Storage:** Single or stacked supercapacitor(s). Choose **C** and voltage limits to satisfy duty-cycle energy with margin.
- **PMIC features:** Cold-start at low input (for TEG/indoor PV), maximum power point (MPP) or simple dithering for PV, programmable Vcap thresholds with hysteresis, and low-leakage path to storage.
- **Thresholds (normative):** Define **V_on** (enable INF), **V_safe** (continue inference), **V_cut** (force safe shutdown). These appear in the Manifest and are enforced in hardware and firmware.

3.1.3 MCU and Memory Map

- **MCU class:** Low-power microcontroller with ≥128–256 KB Flash/ROM, ≥32–128 KB SRAM, DMA, hardware CRC/hash if available, and reliable brown-out reset (BOR). Examples include Cortex-M0+/M3/M4-class or RISC-V RV32 low-power cores with DSP extensions.
- **ROM/Flash layout (conceptual):**
    - **R0:** Boot ROM & measurement stub (immutable).
    - **R1:** Sealed model weights + operator code (INT8/binary).
    - **R2:** Control FSM + sensor/actuator drivers.

  - **R3:** Manufacturing metadata & helper data (read-only, immutable at runtime).
- **SRAM:** Fixed buffers sized at link time (no heap). Double-buffering is allowed only if statically allocated.

3.1.4 Sensors and Actuators

- **Sensors:** Prefer analog-qualified wake (threshold comparators) to avoid needless INF activations. Digital sensors must be polled only during INF activity.
- **Actuators:** Drivers sized for load; all outputs must **fail-safe** (de-energize) on brown-out or measurement failure. Use open-drain/low-side drivers with known default states.

3.1.5 PUF Options

- **Intrinsic PUFs:** SRAM PUF (power-up patterns), RO PUF (frequency differentials).
- **Extrinsic/Assisted:** Secure elements with PUF-backed key derivation.
- **Fuzzy extractor:** Helper-data scheme to reconstruct stable keys; helper data stored in R3, integrity-protected.

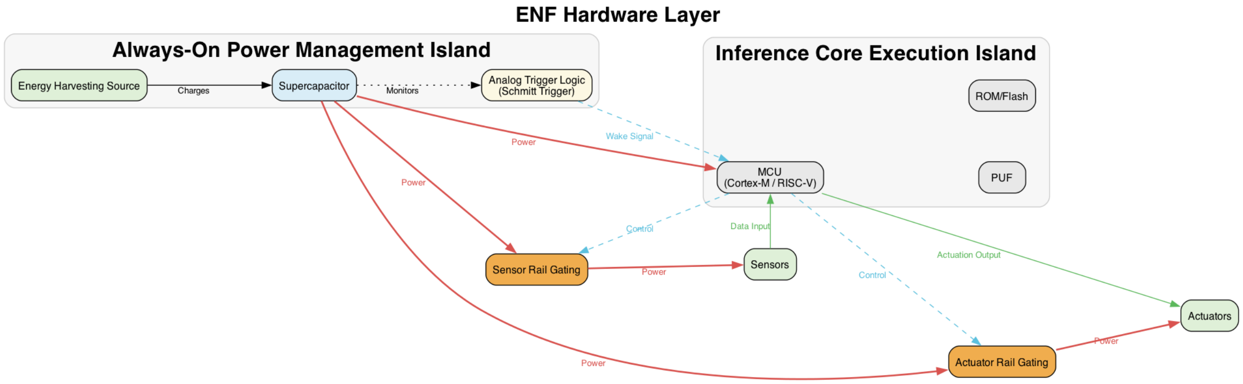

Figure 3-1 suggested: *Dual-island block diagram with AON (harvester→PMIC→comparator→wake FSM) and INF (MCU + ROM image), gates, and Vcap thresholds.*

3.2 Neural Firmware Logic

3.2.1 Model Form and Operators

- **Quantization:** INT8 (or binary) weights and activations; per-tensor or per-channel scales fixed at link time.
- **Operators:** Constrained set (conv/DS-conv, matmul, add, relu, pool) compiled into **bit-exact** kernels pinned to specific library versions (e.g., CMSIS-NN, TFLM) to guarantee reproducibility.
- **No interpreter:** The compute graph is flattened at build time into straight-line code; no dynamic op dispatch.

3.2.2 Memory & Scheduling

- **Static tensors:** All buffer sizes and lifetimes computed by the linker; no heap or recursion.
- **Schedule:** Single-shot inference per wake; optional multi-tile passes if statically bounded.
- **Numerics discipline:** Declared rounding mode; subnormal handling defined; FMA either bit-fixed or disabled.

3.2.3 Image Sealing and Provenance

- **Region hashing:** Immutable regions (R1–R3) hashed at boot; digest compared to Manifest value.
- **Provenance record:** Manifest includes firmware hash, model hash, ROM/RAM map, WCET bounds, thresholds, toolchain lockfile/container digest, and calibration ranges.

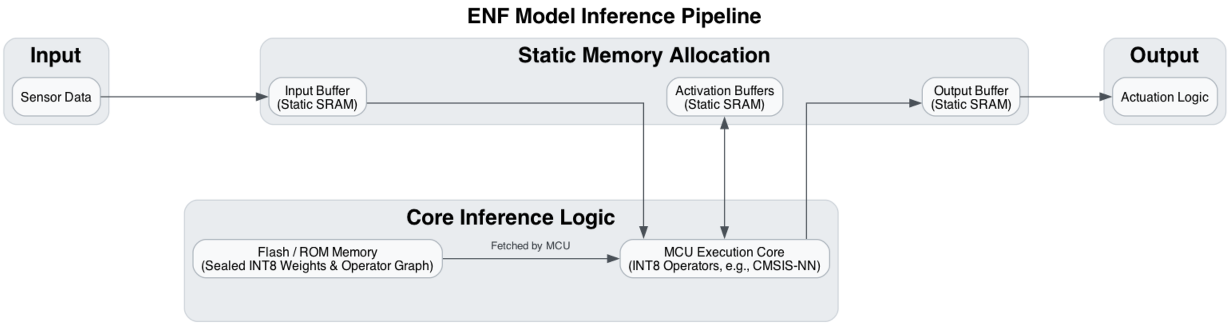
ENF Model Inference Pipeline
Input
Sensor Data
Static Memory Allocation
Input Buffer (Static SRAM)
Activation Buffers (Static SRAM)
Output Buffer (Static SRAM)
Output
Actuation Logic
Core Inference Logic
Flash / ROM Memory (Sealed INT8 Weights & Operator Graph)
Fetched by MCU
MCU Execution Core (INT8 Operators, e.g., CMSIS-NN)

Figure 3-2 suggested: *ROM/Flash layout and fixed SRAM plan; arrows from FSM to operator kernels; no dynamic dispatch.*

3.3 Security & Trust

3.3.1 Secure Boot Measurement

1. AON enables INF only if **Vcap ≥ V_on**.
2. Boot stub computes digest over immutable regions.
3. If digest matches Manifest, control passes to FSM; else, device **fail-closes** (outputs safe), and INF is re-gated until the next power window.

3.3.2 PUF-Derived Keys (Not Stored)

- **Key ladder:** PUF → fuzzy extractor → root key → derived keys (e.g., image binding key, attestation key).
- **No key at rest:** Only helper data persists; keys are recomputed on each verified boot.
- **Optional attestation:** On physical audit, a challenge-response proves (device, image) pairing without exposing secrets.

3.3.3 Eliminated Attack Classes (By Construction)

- Remote code execution via network or OTA updater.
- Supply-chain drift via runtime interpreters or plugin systems.
- Key exfiltration from non-volatile storage (no stored keys).
- State divergence due to dynamic allocation or background daemons.
- **Rollback attacks:** Eliminated by binding boot measurement to a single sealed image hash; no OTA mechanism exists to re-introduce older binaries.

ENF Trust Lifecycle: Hardware-Bonded Integrity Check for ENF Firmware

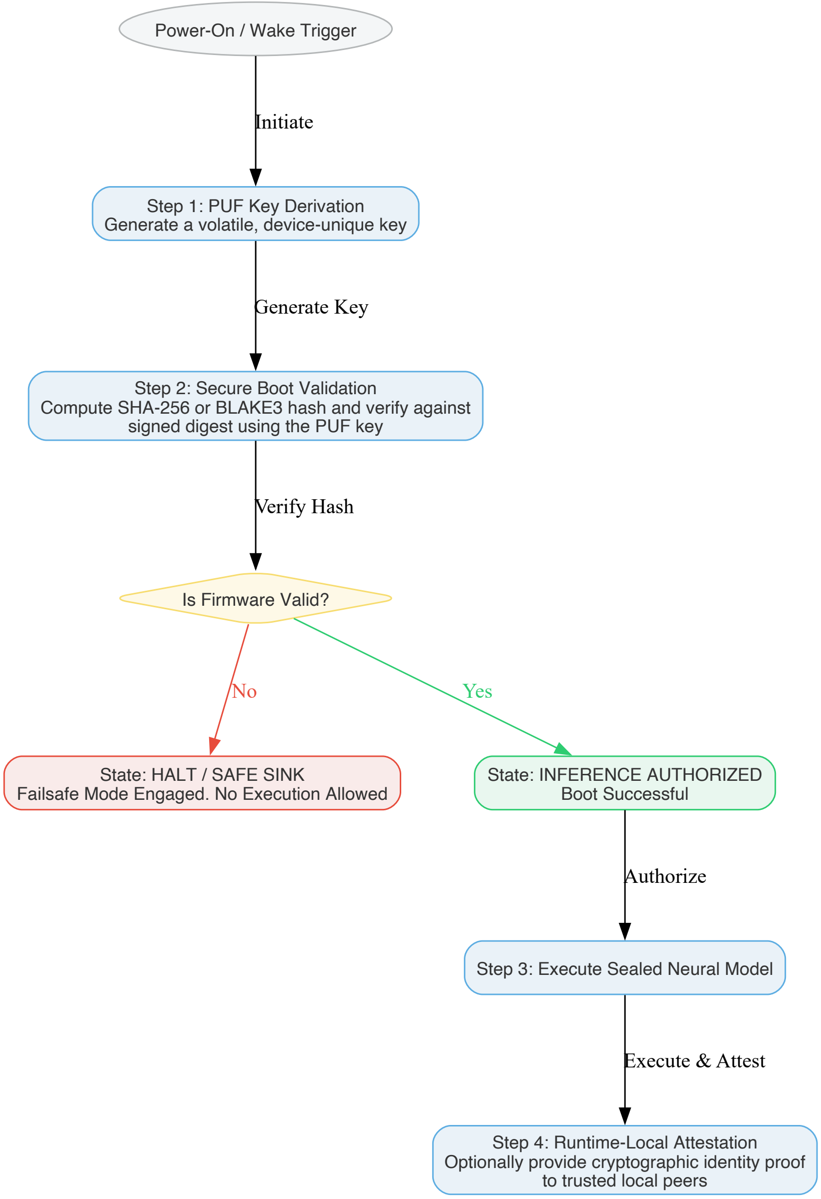
Power-On / Wake Trigger
Initiate
Step 1: PUF Key Derivation
Generate a volatile, device-unique key
Generate Key
Step 2: Secure Boot Validation
Compute SHA-256 or BLAKE3 hash and verify against signed digest using the PUF key
Verify Hash
Is Firmware Valid?
No
Yes
State: HALT / SAFE SINK
Failsafe Mode Engaged. No Execution Allowed
State: INFERENCE AUTHORIZED
Boot Successful
Authorize
Step 3: Execute Sealed Neural Model
Execute & Attest
Step 4: Runtime-Local Attestation
Optionally provide cryptographic identity proof to trusted local peers

Table 3-3 suggested: *Threat vector → ENF countermeasure (e.g., "Network RCE → No IP stack/OTA"; "Rollback/TOCTOU → Image measurement + fail-closed").*

3.4 Energy Path

3.4.1 Source to Storage

- **Sources:** Indoor PV, TEG, piezo; select PMIC mode (MPP/harvester-specific) and input protection.

- **Storage sizing (indicative):** Required energy per cycle:

$$E_{cycle} = E_{boot} + E_{sense} + E_{infer} + E_{actuate} + E_{margins}$$

  Supercap energy window:

$$E_{cap} = \frac{1}{2}\mathrm{C}(V_{on}^2 - V_{cut}^2)$$

  Choose **C** such that $E_{cap} \geq E_{cycle}$ with margin for leakage/ESR.

3.4.2 Thresholds and Duty Cycle

- **Threshold policy:**
  - **V_on:** minimum to boot and verify.
  - **V_safe:** minimum to complete inference/actuation; below this, skip or degrade action.
  - **V_cut:** force shutdown; outputs drive to safe state.

Duty cycle estimate: For average harvested power $\overline{P_h}$ and cycle energy $T_{period} \approx \frac{E_{cycle}}{\overline{P_h}}$
Example: With $\overline{P_h} = 50\,\mu W, \quad E_{cycle} = 0.5\,mJ, \;\; T_{period} \approx 10\,s$. Apply environment-specific derating (lux/ΔT variability, aging).

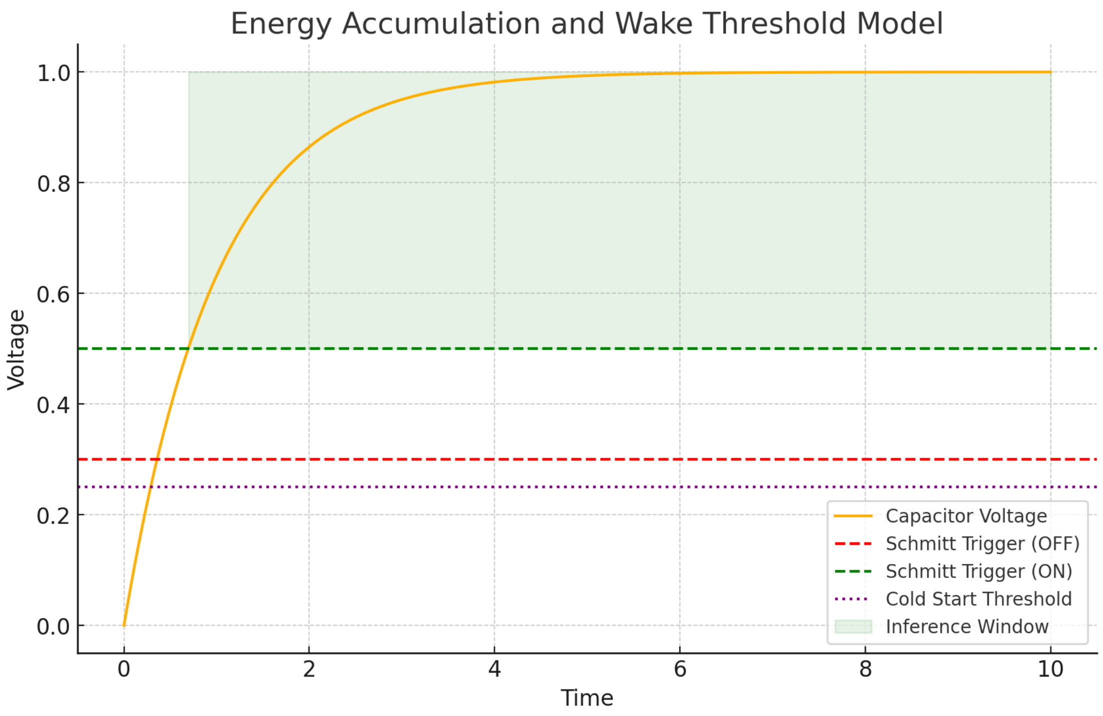

Figure 3-4 suggested: *Energy path with PMIC, supercap, and Vcap timeline showing **V_on/V_safe/V_cut**.*

3.5 Execution & Safety

3.5.1 Deterministic FSM

Wake → Sense → Infer → Actuate → Sleep, single path only. The FSM executes once per qualified wake and returns to Sleep deterministically.

Pseudocode (illustrative):

```
void enf_cycle(void){
  if(!image_measurement_ok()){ fail_closed(); return; }
  if(vcap() < V_SAFE){ skip_inference(); return; }
  sample_sensors();
  run_inference_fixed();
  if(vcap() < V_SAFE){ fail_closed(); return; }
  apply_actuation_bounded();
}
```

3.5.2 Brown-Out and Reset Behavior

- **Brown-out detect (BOR):** If Vcap drops below **V_cut**, outputs are driven to safe defaults and INF is gated off.
- **Deterministic reset:** Next cycle begins only when AON observes **Vcap ≥ V_on**; no mid-state resumes.

- **Watchdog:** A hardware watchdog enforces WCET per state; any overrun triggers fail-closed then reset/gate. Conformance to WCET test vectors is mandatory in qualification.

3.5.3 Safe Actuation & Sensor Sanity

- **Actuation bounds:** Duration and magnitude clamped to Manifest limits; missing bounds is a build failure.
- **Sensor sanity checks:** Range and monotonicity guards; on violation, skip actuation and record fault counter (local volatile).

Phase	Trigger / Guard	Bounded actions (no interrupts)	Exit / Fault path (WCET)
Wake	Analog Schmitt gate; Vcap rising	Enable AON; gate INF power	→ Energy Check (≤1 ms)
Energy Check	Vcap ≥ Von	Boot + image measurement; verify hash	match → Sense; mismatch → Fail-Closed (≤0.5 ms)
Sense	Vcap ≥ Vsafe	Priority sensor poll (AFE/I2C/SPI/MBus), fixed window	OK → Infer; timeout/overdraw → Fail-Closed (≤15 ms)
Infer (INT8)	Vsafe maintained; inputs ready	Bit-exact forward pass (CMSIS-NN/TFLM); fixed buffers	OK → Actuate; WCET overrun → Fail-Closed (≤10 ms)
Actuate	Decision threshold met	Bounded GPIO/SPI actuation; Manifest limits	Vcap < Vcut or timeout → Fail-Closed; else → Sleep (≤20 ms)
Sleep	FSM resolved / fail-closed	INF off; outputs safe	Next qualified wake when Vcap rises

Table 3-5 suggested: FSM swim-lane with timing/WCET annotations and the BOR/threshold interactions.

Implementation Notes (Non-Normative)

- Prefer MCUs with native CRC/SHA blocks to shorten boot measurement time.
- Keep AON analog path discrete where possible; avoid leakage through MCU GPIO when INF is gated.
- Use fixed-seed table lookups if any stochastic elements were originally present in training-time code; do not carry them into runtime.

Section 4: The ENF Framework

Purpose. Define how ENF instances are **specified**, **built**, and **proven** so the claims in this paper are auditable and repeatable. Normative keywords **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT**, and **MAY** are used as defined in RFC 2119 and RFC 8174 (BCP 14).

4.1 Scope & Principles

- **Dual nature.** ENF is both a sealed **Neural BIOS** (runtime) **and** a **Framework** (spec → build → prove).

- **Declarative design.** Behavior is fixed by declaration (Manifest), not by mutable runtime code.
- **Reproducibility.** Builds are **bit-for-bit** reproducible; conformance is verifiable **offline**.

4.2 Formal Model (5-Tuple)

An instance is defined as `ENF(T, P, S, F, C) → ℒ_sealed`, where:

- **T (Task)**: bounded complexity (e.g., threshold/MLP/CNN; fixed shapes).
- **P (Power model)**: harvested/event-burst/discharge; defines thresholds and duty policy.
- **S (Security level)**: S1 Sealed; S2 PUF-anchored; S3 Dual-gated (PUF + analog power signature).
- **F (Fallback)**: safe-sink / degraded-path / hard-reset; watchdog-enforced WCET.
- **C (Communication scope)**: none / pulse-sync / deterministic swarm.

4.3 ENF-Gene Identifier (MUST)

A human-readable fingerprint of the instance: `ENF-T{x}.P{y}.S{z}.F{w}.C{v}` (e.g., `ENF-T2.P1.S2.F2.C1`). The ENF-Gene **MUST** map 1:1 to Manifest fields and **MUST** be shipped with releases. Optional audit extensions **MAY** embed compiler hash, lot ID, UTC build timestamp, and lifecycle mode. The ENF-Gene complements (does not replace) serial numbers, part numbers, and cryptographic keys; see Appendix A.2.6 for a comparison.

4.4 Specification — ENF Manifest (MUST)

A signed declaration that fixes:

- **Thresholds:** `V_on` / `V_safe` / `V_cut`; energy policy; cold-start constraints.
- **Timing:** per-state **WCET**; watchdog window; sampling cadence.
- **Memory map:** static regions (weights/graph, FSM/logic, sensor config, OTP metadata).
- **Numeric policy:** quantization & rounding; random seeds (for training only) and commit IDs.
- **Datasets:** identifiers; licenses/consent **(consent basis/legal ground)**; subgroup metrics; OOD tests; drift bounds.
- **Security:** level S1/S2/S3; helper-data MAC/ECC params (if S2/S3); tamper inputs; **if S3, include analog power-signature profile** (e.g., `rise_time_ms`, `V_peak`, `decay_ms`, `jitter_tol_pct`).
- **Fallback & comms:** F and C selections with parameters.
- **Provenance: ENF-Gene**; SBOM reference; derating/EOL factors.

Example (abridged YAML):

```yaml
task: {complexity: Moderate, model: ./models/soil_v2.tflite}
power: {model: Harvested, V_on_mV: 3200, V_safe_mV: 2900, V_cut_mV: 2600}
timing: {wcet_ms: {sense: 1.0, infer: 10.0, actuate: 2.0}, watchdog_ms: 50}
memory: {weights: 0x08020000..0x08023FFF, fsm: 0x08010000..0x08010FFF}
```

```
numeric: {qformat: INT8, rounding: nearest}
dataset: {id: soil-2024-r2, license: permissive, ood_tests: true}
security: {level: S2, puf: SRAM, helper_mac: blake3, ecc: BCH(63,45)}
fallback: {mode: degraded_path}
comms: {scope: pulse_sync}
provenance: {enf_gene: ENF-T2.P1.S2.F2.C1, sbom: sbom.json}
```

4.5 Toolchain — Reproducible Build (MUST)

- Build in a locked container; record **container digest** and **compiler/flags lockfile**.
- Static link order; **no interpreter**, **no dynamic allocation**, **no recursion**, **no preemption**, **no ISRs beyond threshold-qualified wake**; fixed stacks/buffers.
- **No time-dependent entropy** at runtime (e.g., no `rand()` seeded from clocks or interrupts).
- **Numeric determinism:** define rounding tie-break (**round half away from zero**) and **saturation semantics** for arithmetic (no wraparound).
- Output is a sealed image with deterministic memory layout (map file **MUST** be archived).

4.6 Conformance Pack — Release Artifacts (MUST)

Ship with every image/lot:

- `firmware.sha256`, `model.sha256`, and `manifest.hash` (immutable-region and spec digests; **hash algorithm MUST be declared — SHA-256 or BLAKE3**)
- **agent.gene** (human-readable 5-tuple identifier)
- **PUF-bind record** (helper-data MAC/ECC parameters)
- **SBOM** (toolchain + libraries; **SHOULD** be SPDX or CycloneDX)
- Test vectors: `boot_measurement.json` (allow/deny; **for S3 include analog signature vector fields**: `rise_time_ms`, `V_peak`, `decay_ms`, `jitter_tol_pct`), `wcet.csv` (per-state bounds)
- Optional: energy traces `I(t)`, `V_cap(t)`, and **tamper/event counters**

Release bundle naming (SHOULD): `enf-conformance-<product>-<rev>-<yyyymmdd>.zip`.

4.7 Security Levels (MUST declare)

- **S1 Sealed:** ROM/Flash immutability; measured boot; no IP/OTA/telemetry.
- **S2 PUF-Anchored:** S1 + silicon-anchored identity; actuation gated on boot-measurement success.
- **S3 Dual-Gated (optional/experimental):** S2 + analog power-signature gate; tolerances & test method **MUST** be documented, and a **factory characterization vector** MUST be included in `boot_measurement.json`.

4.8 Determinism & Memory Discipline (MUST)

Static buffers; bounded loops; single FSM; watchdog-enforced **WCET**; **no heap/GC**; **no recursion**; **no preemption**.

4.9 Measurement & Field Audit (MUST)

Audits **MUST** recompute digests, replay `boot_measurement.json`, verify **WCET** against `wcet.csv`, and inspect tamper/event counters. **External archives MUST** retain `manifest.hash` and `agent.gene` per lot/device; retention **SHOULD** be device lifetime + 5 years. Hashes **SHOULD** be time-stamped (e.g., RFC 3161) or logged to a public transparency log. Devices **MUST fail-closed** on measurement mismatch or energy-policy violation. **Optional self-check:** devices MAY expose a 1-bit confirmation path (e.g., GPIO toggle) affirming PUF identity and boot-measurement success without leaking code, weights, or data.

4.10 ENF Agent Classes (SHOULD)

- **ENF-Logic:** thresholds/FSM only (max verifiability, minimal footprint).
- **ENF-Neural:** quantized MLP/CNN with sealed weights.
- **ENF-Hybrid:** rules + neural fallback; preferred for efficiency and safety envelopes.

4.11 Registry & Governance (SHOULD)

Publish **ENF-Gene** and Manifest metadata to a registry; provide vulnerability-disclosure contact and recall **SLA**; maintain dataset licensing/consent records; declare RoHS/REACH and design-for-repair/EOL policies. **Toolchain alignment** with safety standards (e.g., **ISO 26262**, **IEC 61508**) is RECOMMENDED; the reproducible build system MAY serve as a recognized trust anchor.

4.12 Traceability Matrix (normative)

ENF Invariant	Verification Artifact
Offline-only, no IP/OTA/telemetry	Manifest → security.level=S1+; network stack absent in SBOM
Deterministic timing (**WCET**)	`wcet.csv`; watchdog config in Manifest
Static memory (no heap/GC)	Map file; toolchain lockfile; SBOM
No interpreter/runtime	SBOM shows no VM/interpreter; map file proves static link; binary scan shows no bytecode sections
PUF-anchored provenance	PUF-bind record; boot-measurement allow/deny
Energy thresholds policy	Manifest thresholds; `I(t)`,`V_cap(t)` traces
Fail-closed behavior	Test vector outcome; device BOR/tamper logs

4.13 Interoperability Profiles (MAY)

- **Core:** S1, Manifest + hashes + SBOM + test vectors.
- **Industrial:** Core + S2 (PUF), tamper inputs, quarterly offline audit.
- **Safety-critical:** Industrial + formal proof trace excerpts, independent lab witness.

4.14 Cross-References

This section binds **§2 (Invariants)** to **§5 (Threat Model & Non-Goals)**, **§6 (Energy Budget)**, **§7 (Minimal Reference Design)**, and the formalism in **Appendix A** (5-tuple, ENF-Gene, compiler & governance).

Section 5: Normative Specification & Conformance

Purpose. This section defines how an implementation *proves* it is ENF-conformant. Conformance is demonstrated through a signed **Manifest** (build provenance), a **sealed firmware image**, and **test vectors** covering boot, identity, timing, energy, and actuation bounds.

5.1 Conformance Checklist (MUST/SHALL)

Build & provenance.

- **MUST** publish a signed **Manifest** containing: firmware hash, model hash, toolchain/version and **build-container digest**, ROM/RAM map, WCET bounds, sensor/actuator limits, **`V_on`/`V_safe`/`V_cut`**, calibration constants/ranges, and (if applicable) PUF helper-data format.
- **MUST** be **bit-for-bit reproducible** from the Manifest (locked toolchain + model hash).

Identity & boot.

- **MUST** derive device keys from a **PUF** (no keys at rest).
- Boot **SHALL** cryptographically measure immutable regions (e.g., SHA-256/BLAKE3) and enable actuation only on a verified measurement; on failure, the device **SHALL fail-closed**.

Determinism.

- **MUST** implement a single FSM path **Wake → Sense → Infer → Actuate → Sleep** with documented **WCET** per state.
- The **only** allowed asynchronous event is a **threshold-qualified wake** from the always-on domain.
- **MUST** use quantized/fixed-point or otherwise **bit-exact** numerics with locked kernels/flags.

- **MUST** configure a **hardware watchdog** to enforce per-state WCET; any overrun **SHALL** trigger **fail-closed** and reset.
- No external time sync; local time-base drift declared in the Manifest.
- **MUST NOT** allow hardware RNG or other nondeterministic entropy to influence the control FSM or inference outputs; if present, such sources **SHALL** be disabled or quarantined from the ENF domain.

Memory & runtime.

- **MUST** use static memory (no heap/GC, no recursion or unbounded loops).
- No interpreter, no dynamic model loading, and no filesystem writes beyond manufacturing records.

Energy & safety.

- **MUST** enforce **`V_on`**, **`V_safe`**, **`V_cut`** in hardware and firmware; BOR leads to safe outputs and a gated inference domain.
- Actuation **MUST** be clamped to Manifest limits.

Connectivity exclusion.

- **MUST NOT** include IP stacks, OTA/RPC channels, or telemetry sinks; debug/programming interfaces are fused off post-manufacture.

5.2 Minimal Test Vectors (per build)

- **V1 — Boot measurement:** immutable-region digest + expected allow/deny result.
- **V2 — PUF binding (optional):** helper-data blob + sample challenge/response transcript proving per-die identity (no secret at rest).

5.3 Audit Procedure (condensed)

Verify **Manifest** signatures and fields → recompute the image digest; confirm fuses and allow/deny match (**V1**). If **V2** is provided, run the PUF transcript on a fixture to prove (image, die) binding. Execute a short FSM/WCET check and one threshold cycle to confirm **`V_on`/`V_safe`/`V_cut`** and **fail-closed** behavior; spot-check numeric determinism on a small input corpus.

5.4 Pass/Fail Criteria

- **PASS:** Checklist satisfied; **V1** verified (and **V2** if present); zero prohibited features detected.
- **FAIL:** Missing Manifest fields; digest/PUF mismatch; WCET overrun without watchdog action; threshold-policy violation; nondeterministic numeric outputs.

Section 6: Threat Model & Non-Goals

Purpose. Define which risks ENF eliminates **by design**, what residual risks remain, and how implementations **mitigate** them without adding reachability or runtime mutability.

Abbreviations. AON = always-on domain; **INF** = inference domain.

6.1 Threat Model — Classes Eliminated by Design

- **Remote code execution via network/OTA: Non-reachable by design.** ENF excludes IP stacks, OTA/RPC channels, and telemetry; debug/programming interfaces are **fused-off**; no interpreter or plugin loader is present.
- **Supply-chain drift/malicious updates:** In-field images are immutable; the Boot ROM measures sealed regions and enables actuation only on a verified digest. No OTA implies no rollback channel; image identity is bound to the die via PUF.
- **Key theft at rest:** Device keys are **derived** from a PUF; no secrets are stored. Only integrity-checked helper data may persist.
- **Configuration/state divergence:** No dynamic allocation, no background daemons, and a single FSM path prevent runtime skew. The time base is local; no external synchronization.
- **Data exfiltration:** With no network path and no telemetry sinks, raw or derived signals cannot leave the device.

6.2 Residual Risks (Cannot Be Fully Eliminated)

- **Physical & fault attacks:** Decap/probing, FIB edits, side-channels (power/EM), voltage/clock glitching, laser/radiation fault injection; Boot ROM defects.
- **Sensor-layer attacks:** Spoofed or adversarial stimuli, replay, saturation, or environmental drift beyond the declared operating envelope. Includes **adversarial perturbations** against fixed-shape models.
- **Energy-path attacks:** Starvation or crowbar events that collapse **`V_cap`**; potential **GPIO back-powering** paths causing partial INF energization; induced brown-out behavior if thresholds are misconfigured.
- **PUF reliability & environmental drift:** Intra-device noise/aging/temperature effects may degrade reconstruction without robust helper data/ECC.
- **Manufacturing & logistics:** Compromised toolchains or signing keys, tampered helper data, provisioning errors, mis-set actuation limits, and **signed-but-dangerous** images (validly signed but policy-violating).

6.3 Mitigations & Assurances (Without Reachability)

- **Physical hardening:** Fuse-off debug, conformal coat/epoxy, secured enclosure; optional tamper inputs that gate the INF domain and force safe outputs; record a **monotonic tamper counter** in non-volatile storage for field audits.

- **Fault/side-channel hygiene: BOR (Brown-Out Reset)** with hysteresis; clock/voltage glitch detectors; constant-time cryptography; decoupling and shielding where feasible.
- **Sensor sanity & actuation bounds:** Analog filtering/hysteresis; range and rate-of-change checks; optional sensor redundancy; clamp actuation to Manifest limits; **fail-closed** on violations. Use **training-time robustness constraints** and **score thresholds** to reject low-confidence inferences.
- **Energy discipline:** Enforce **`V_on`/`V_safe`/`V_cut`** in hardware and firmware; size (C, ESR) for peak draw; apply a minimum-dwell guard before inference. Eliminate **back-power paths** (series resistors/ideal diodes) and define safe I/O states during **AON-only** mode.
- **PUF reliability & helper data:** Enroll across temperature/voltage corners; use fuzzy extractors with ECC; integrity-protect helper data; reject on excessive reconstruction error.
- **Manufacturing assurance:** Reproducible builds, offline signing (HSM), two-person rule, recorded Manifests; MAC-protected PUF helper data; acceptance tests with measurement and PUF challenge-response.
- **Field audit (offline):** Periodic physical audits using the Manifest, boot digest, and optional PUF transcript to prove (image, die) binding.

6.4 Non-Goals

ENF is **not** intended to: provide runtime networking or telemetry; support OTA updates, dynamic model loading, or on-device learning; resist nation-state-grade invasive attacks; act as a high-bandwidth/interactive platform; or replace update-mandated systems where regulation requires remote patching (ENF uses physical recall/replace instead).

Section 7: Energy Budget

Purpose. Provide a clear method—and one concrete example—for sizing energy storage, setting thresholds, and estimating duty cycle for ENF devices operating on harvested power.

Abbreviations. AON = always-on domain; **ESR** = equivalent series resistance.

7.1 Model & Equations (normative where stated)

- **Effective harvested power.** `P_eff = η_h * P_src − P_leak`, with `P_leak = P_AON + P_store ≈ I_AON * V_cap_mean + I_store * V_cap_mean`. **MUST:** account for both always-on quiescent current and supercap self-leakage. (`η_h` is harvester/PMIC efficiency.)
- **Per-cycle energy.** `E_cycle = E_sense + E_infer + E_act + E_ovh`.
- **Capacitor energy windows.** `E[on→cut] = 0.5 * C * (V_on^2 − V_cut^2); E[safe→cut] = 0.5 * C * (V_safe^2 − V_cut^2).`

- **Brown-out headroom during pulses. MUST** ensure inference pulses do not induce brown-out: `ΔV_pulse ≈ I_peak * ESR + (I_peak * t_pulse / C)` and `V_safe − ΔV_pulse ≥ margin`.
- **Harvester cold-start. MUST** satisfy the PMIC/harvester cold-start threshold before normal regulation; document `P_cs` (or `V_in_cs`) in the Manifest.
- **Duty-cycle estimate.** At steady state with sufficient storage, the inter-cycle interval `t ≈ E_cycle / P_eff`. For indoor PV, scale `P_src` by a **spectral/angle factor** `k_spec,angle ∈ (0,1]` when spectra or incidence differ from test conditions.
    - **MUST:** thresholds and storage must satisfy: (1) `E[on→cut] ≥ E_cycle`; (2) `E[safe→cut] ≥ E_infer + E_act + E_ovh + margin`.

7.2 Worked Example — Indoor PV @ 300–500 lux

Assumptions (illustrative, conservative):

- a-Si PV panel area `A = 50 cm²`; power density at `300 lux ≈ 5 µW/cm²` ⟶ `P_src ≈ 50 * 5 µW = 250 µW`.
- Harvester efficiency `η_h = 0.75` ⟶ harvested power `= 0.75 * 250 µW = 188 µW`.
- Leakage (supercap + PMIC + AON): `I_leak = 10 µA`, `V_cap_mean ≈ 3.0 V` ⟶ `P_leak ≈ 10 µA * 3.0 V = 30 µW`.
- Net effective power: `P_eff = 188 µW − 30 µW = 158 µW`.

Per-cycle energies (representative):

- `E_sense = 50 µJ`, `E_infer = 1.20 mJ`, `E_act = 0.50 mJ`, `E_ovh = 0.15 mJ` ⟶ `E_cycle = 1.90 mJ`.

Duty cycle at 300 lux: `t ≈ 1.90 mJ / 158 µW ≈ 12.0 s`.

Duty cycle at 500 lux (≈ linear gain): If power density scales to `≈ 8 µW/cm²`, then `P_eff ≈ 158 µW * (8/5) ≈ 253 µW` ⟶ `t ≈ 1.90 mJ / 253 µW ≈ 7.5 s`.

Storage & thresholds check:

- Choose `C = 0.047 F`, `V_on = 3.2 V`, `V_safe = 2.9 V`, `V_cut = 2.6 V`.
    - `E[on→cut] = 0.5 * 0.047 * (3.2^2 − 2.6^2) = 0.0235 * (10.24 − 6.76) ≈ 0.0818 J`.
    - `E[safe→cut] = 0.5 * 0.047 * (2.9^2 − 2.6^2) = 0.0235 * (8.41 − 6.76) ≈ 0.0388 J`.
- **Both values exceed** `E_cycle = 0.0019 J`, providing ample margin to complete inference and actuation before brown-out.

7.3 Implementation Notes (non-normative)

- **Measure, don't guess:** derive `E_sense` and `E_infer` from scope/current-probe traces; log `V_cap` to verify threshold crossings.
- **Derate for reality:** apply a safety factor (e.g., ×1.5–2×) for panel angle, dusting, temperature, and aging; declare end-of-life derating for `C`, `ESR`, and `I_store` in the Manifest.
- **Night & low-lux behavior:** define a maximum sleep interval and safe output posture when `P_eff → 0`.
- **Inter-cycle jitter bounds:** publish min/median/max interval under stated ambient conditions.
- **Measurement reproducibility:** record scope traces of `I(t)` and `V_cap(t)` for one full cycle and store with the build artifacts.
- **Bill of materials:** prefer low-leakage PMICs and supercaps; size **ESR** to avoid droop during inference pulses.

Section 8: Minimal Reference Design

Purpose. Provide a concrete, buildable reference that satisfies ENF invariants without adding reachability or runtime mutability.

8.1 Bill of Materials (Indicative)

- **MCU (INF domain):** Ultra-low-power MCU (e.g., Cortex-M / RV32) with ≥ 256 KB Flash/ROM, ≥ 64 KB SRAM, DMA, **BOR**, hardware **watchdog** for WCET enforcement, and **SHA-256/BLAKE3** acceleration preferred. No OS or RTOS. **MUST** support **fused-off** debug.
- **Harvester/PMIC (AON domain):** Supports indoor PV/TEG/piezo; **cold-start** at low input power; MPP or dithering for PV; programmable enable with hysteresis; quiescent current in the µA range.
- **Energy storage:** Supercapacitor (e.g., 0.047–0.47 F) sized per the **energy-budget method**; low leakage; ESR sized for peak draw; ideal-diode/PFET gating to the INF domain.
- **Sensors & actuators:** One analog-qualified wake path (comparator/Schmitt) plus polled digital sensors during INF activity; low-side driver or open-drain actuation with a safe default on BOR; optional **tamper inputs** in the AON domain and a **monotonic tamper counter** in NVM.
- **Identity/PUF:** Intrinsic PUF (e.g., SRAM/RO) or secure element; fuzzy extractor with **ECC**; helper data **MAC-protected**, stored read-only; ECC parameters documented.

8.2 Static Toolflow (No Interpreter)

1. **Declare manifest.** Define `enf-manifest.yml` (Task, Power model, Security level, Fallback, Communication scope), thresholds (`V_on`, `V_safe`, `V_cut`), WCET per state, time-base drift, sensor/actuator limits, and toolchain + **build-container digest**.
2. **Freeze model.** Train → quantize (INT8/binary) → fix scales/ranges; export a **fixed-shape** graph.
3. **Compile & link.** Pin kernel library versions (e.g., **CMSIS-NN** / **TFLM**); compile to **bit-exact** kernels; link with **no heap**, no recursion, fixed buffers; **no ISRs beyond the threshold-qualified wake; no preemptive tasks**; produce a sealed ROM image and static RAM map.
4. **Reproducible build.** Build inside a locked container; record compiler flags; emit a **toolchain/flags lockfile**; generate `firmware.sha256` and `model.sha256`.
5. **Boot measurement vectors.** Compute the image digest over immutable regions; record the expected **allow/deny** result.
6. **PUF enrollment.** Enroll across temperature/voltage corners; generate helper data; bind `(device, image)`; store only integrity-checked helper data.

8.3 Release Pack (Build Artifacts)

- **Sealed firmware image:** `sealed_firmware.bin/hex` + `firmware.sha256`.
- **Model hash:** `model.sha256` (or equivalent digest) matching the sealed image.
- **Signed Manifest:** ROM/RAM map; WCET per state; thresholds (`V_on`, `V_safe`, `V_cut`); sensor/actuator limits; numeric path (quantization/rounding/FMA policy); toolchain/version, **compiler/flags lockfile**, and **container digest**; time-base drift; calibration ranges.
- **PUF-bind record:** device identifier, helper-data blob, ECC parameters, measured image digest, and (optionally) a challenge-response transcript.
- **Test vectors:** `boot_measurement.json` (immutable-region digest + allow/deny) and `wcet.csv` (per-state bounds + method).
- **Energy traces (optional, recommended):** scope logs of `I(t)` and `V_cap(t)` over one full cycle to substantiate the **energy-budget calculations**.

This reference design is intentionally minimal; substitute equivalent parts that meet the same thresholds and invariants.

Section 9: Comparison Table

Purpose. Contrast ENF with TinyML runtimes and Cloud/Edge-AI on the system properties that drive safety, assurance, and long-term operations.

Abbreviations. AOT = ahead-of-time; **AON** = always-on domain; **INF** = inference domain; **WCET** = worst-case execution time.

9.1 ENF vs TinyML vs Cloud/Edge-AI

Dimension	ENF	TinyML Runtime	Cloud/Edge-AI
Runtime/OS	No OS; sealed **Neural BIOS** (bare-metal FSM).	Bare-metal or RTOS; often an **interpreter** (e.g., TFLM) or AOT-generated code with a scheduler.	Full OS or containers; drivers and background daemons.
Updates	**No OTA**; immutable image; **recall/replace** only.	OTA common (models/firmware); rollback/version **management** required.	Continuous deployment; frequent rollouts.
Connectivity	**Offline-only** (no IP, no telemetry).	Often connected (BLE/Wi-Fi/IP) for configuration/OTA; can run offline.	Cloud: required; Edge: local inference but **management**/telemetry usually online.
Memory discipline	**Static**: no heap/GC; fixed buffers/shapes; no recursion or unbounded loops.	Mixed: arena allocators; interpreter/operator dispatch or AOT; RTOS scheduling.	Dynamic memory, VMM/allocators, userland processes.
Determinism/ WCET	**Hard-bounded** FSM; per-state WCET; watchdog-enforced.	Bounded “in practice”; scheduling/interpreter overhead adds jitter.	Best effort; OS load and network variance dominate.
Provenance/ attestation	**PUF-anchored** identity; boot **measurement** gates actuation; signed Manifest + hashes; **bit-for-bit** builds.	Keys/SE common; provenance varies by platform; partial reproducibility.	Attestation via cloud IAM/TEEs; device-local reproducibility varies.
Privacy/data locality	**On-device inference only**; no telemetry path.	Application-dependent; telemetry/config often enabled by default.	Data commonly leaves device (ingest, logging, monitoring).
Power model	**Harvested-power first;** dual-island (AON/INF); thresholds **(V_on/V_safe/V_cut)**; WCET-bounded pulses.	Battery or mains; duty-cycled; interpreter/AOT overhead affects energy.	Mains/large battery; high, variable draw (compute + network).
Lifecycle/ EOL	**Certify-once, run long**; no drift; offline audit + PUF binding.	Continuous patching; drift risk across fleets; vendor-support dependent.	Service lifecycle; short deprecation windows; vendor-managed.

Notes. “TinyML Runtime” reflects typical MCU stacks (interpreter or **AOT** under RTOS/bare-metal). “Cloud/Edge-AI” spans centralized cloud and edge gateways; edge can run locally but usually remains connected for **management**/telemetry. ENF emphasizes **non-reachability**, **sealed execution**, and **device-anchored provenance**; TinyML emphasizes **flexibility**; Cloud/Edge emphasizes **scale** and **velocity**.

Section 10: Use-Case Postcards

Purpose. Show three compact, **cloud-free** deployments that exercise ENF’s invariants end to end: **sensor → decision → actuation** under harvested power.

Abbreviations. AON = always-on domain; **INF** = inference domain; **BOR** = brown-out reset; **NVM** = non-volatile memory.

10.1 Micro-Vibration Sentinel (Predictive Maintenance)

Context. Batteryless node on an industrial motor housing to flag early bearing wear.

- **Sensing.** MEMS accelerometer (±8 g, 1–3 kHz bandwidth) on an AON wake line via analog threshold; **RC anti-alias filter**; higher-rate sampling only in INF.
- **Model.** 1D CNN (INT8), ~15–30 k parameters; features = short-time RMS + spectral bins; WCET ≤ 12 ms @ 48–64 MHz.
- **Decision.** `warn` if anomaly score ≥ `τ_warn`; `trip_inhibit` only after `N` consecutive cycles.
- **Actuation.** Open-drain line to stack light (amber) or local relay interlock; default safe (open) on **BOR**.
- **Energy posture.** Indoor PV (300–500 lux) → supercap 0.047–0.10 F; typical `t_cycle ≈ 8-12 s`; thresholds: `V_on = 3.2 V`, `V_safe = 2.9 V`, `V_cut = 2.6 V`; **PMIC cold-start satisfied**.
- **Security & provenance.** PUF-anchored identity; boot measurement gates actuation; **Manifest** lists WCET and thresholds; **helper-data MAC + ECC parameters recorded**; no IP/OTA.
- **Lifecycle.** Certify once; quarterly offline audit: read digest + optional PUF transcript; **monotonic tamper counter recorded**; replace if mechanical retuning is needed.

10.2 Soil-Moisture Gate (Irrigation Control)

Context. Field or greenhouse probe that autonomously opens/closes a **latching solenoid valve** without connectivity.

- **Sensing.** Capacitive soil-moisture probe + temperature; AON analog comparator wake on dryness threshold; **compensate for temperature/salinity drift**.
- **Model.** Tiny MLP (INT8), ~5–10 k parameters; inputs = moisture, temperature, recent trend; WCET ≤ 5 ms @ 24–32 MHz.
- **Decision.** `irrigate` when predicted root-zone VWC < `τ_dry` **and** recovery unlikely (`trend < τ_trend`).
- **Actuation.** Two short pulses to the **latching valve** (open/close) **via H-bridge or polarity switch**; **freewheel/snubber where applicable**; clamp pulse width and duty per **Manifest**; fail-closed on **BOR**.

- **Energy posture.** Small PV tile (50–60 cm^2) + 0.10–0.22 F supercap; `t_cycle ≈ 30–120 s` depending on light; thresholds as published in the **Manifest**; **back-power paths eliminated; safe I/O states in AON-only mode**.
- **Security & provenance.** PUF-anchored identity; **helper-data MAC + ECC parameters recorded**; sealed image; no telemetry; offline audit via digest and event counter.
- **Lifecycle.** Seasonal calibration window; firmware remains immutable; replace probe on drift per maintenance schedule.

10.3 Offline Fall-Alert Tag (Personal Safety)

Context. Wearable/room tag for elderly-care apartments where **privacy forbids microphones/cameras** and no network is allowed.

- **Sensing.** 3-axis accelerometer + barometric sensor; AON wake on large-delta acceleration or pressure step; **post-BOR re-arm thresholds**.
- **Model.** 1D CNN (INT8), ~20–35 k parameters; features include event magnitude, posture change, and duration; WCET ≤ 10 ms @ 32–48 MHz.
- **Decision.** `local_alert` when fall likelihood ≥ `τ_fall` and posture not recovered within `T_recover`.
- **Actuation.** Piezo buzzer + high-brightness LED pattern; **clamp alert duration and duty**; optional **NFC tap** for a caregiver to read the last `N` events (offline log).
- **Energy posture.** Indoor PV (200–400 lux) + 0.047–0.10 F supercap; duty varies with ambient; enforce `V_on/V_safe/V_cut` policy; **PMIC cold-start satisfied**.
- **Security & provenance.** No microphones/cameras; sealed, offline-only image; PUF-anchored identity; local logs are **signed digests with sequence numbers**.
- **Lifecycle & ethics.** Not a medical device; complements—does not replace—supervised care; annual physical audit and battery-free endurance test.

Common properties (all postcards). No IP stack, no OTA, no telemetry; **static memory**, **watchdog-enforced WCET**, **bit-for-bit** builds; thresholds and WCET published in the **Manifest**; devices **fail-closed** on brown-out or measurement mismatch. **Tamper/audit:** AON tamper inputs gate the INF domain; a **monotonic tamper counter** is stored in NVM; the conformance pack includes `boot_measurement.json` and `wcet.csv`.

Section 11: Limitations & Ethics

Purpose. Make explicit the constraints, trade-offs, governance practices, and ethical boundaries of ENF deployments so implementers do not over-claim capability or safety.

Abbreviations. BOR = brown-out reset; **PUF** = physically unclonable function; **OOD** = out-of-distribution; **SLA** = service-level agreement; **SBOM** = software bill of materials.

11.1 Rigidity & Scope Limitations

- **No OTA / runtime mutability.** Images are immutable; **no** dynamic model loading or on-device learning. Any material change requires **recall/replace** and re-certification.
- **Task specificity.** Fixed-shape models and static memory limit model capacity and feature agility; unsuitable where policies or requirements change frequently.
- **Determinism constraints.** Single FSM with per-state **WCET**, watchdog-enforced; no external time sync; bounded buffers limit complex pipelines and high-bandwidth sensing.
- **Connectivity exclusion.** No IP stack/telemetry; remote diagnostics are not possible—only **offline audits**.
- **Physical threat limits.** Not designed to resist invasive decapsulation/focused ion beam (FIB) or nation-state fault injection; rely on fused-off debug, **BOR**, and tamper evidence.
- **Boot ROM immutability.** If a Boot ROM defect is discovered, devices **cannot** be patched; affected lots **require recall/replace**.
- **Component drift.** Aging/temperature increase **ESR/leakage**, raising brown-out risk; declare derating factors and inspection intervals in the **Manifest**.

11.2 Dataset Governance & Bias Safeguards

- **Provenance.** The **Manifest** must declare dataset name/version and the cryptographic hash of the training corpus and pipeline artifacts.
- **Evaluation.** Publish a short **Model Card**: class balance, subgroup performance, calibration error, and out-of-distribution (OOD) stress tests relevant to the task.
- **Mitigations.** Clamp actuation ranges; use score thresholds with a reject option; document post-processing that reduces harm (e.g., debounce, rate limits).
- **Reproducibility.** Lock quantization/rounding; record training seeds and commit IDs (for audit) while runtime remains seed-free and deterministic.
- **Licensing & consent.** Declare **dataset licenses**, consent basis, and data-retention policy; document any **synthetic data** generation.
- **Shift & representativeness.** Quantify **domain shift** risks (lighting/environment/population); state acceptable drift bounds and **reject thresholds**.
- **Bias remediation.** Record subgroup metrics and any **fairness constraints** applied at training; re-evaluate at **recall/replace**.

11.3 Safety, Recall & Replace Policy

- **Triggers.** Safety defect, evaluation drift, regulatory change, or mis-set thresholds **require recall/replace** (no hotfix via OTA).
- **Procedure.** Versioned **Manifest**; decommission by clearing actuation enable on boot-measurement mismatch; revoke **PUF-bind** records; capture device tamper/event counters.
- **Field audit.** Periodic offline checks: recompute image digest; verify thresholds/WCET against **wcet.csv**; review event/tamper logs.
- **Disclosure & SLA.** Publish a **vulnerability-disclosure** contact and a **service-level agreement (SLA)** for recall (e.g., triage ≤ 7 days; mitigation ≤ 30 days).

- **Secure decommission.** On retirement, **zeroize** non-volatile metadata; **revoke PUF-bind** records; preserve tamper/event counters for audit.
- **SBOM & provenance.** Ship a **software bill of materials (SBOM)** (toolchain and libraries) and the build **container digest** with each release.

11.4 Privacy & Human Factors

- **Privacy by design.** No telemetry or microphones/cameras where not explicitly required; logs remain local (e.g., NFC-readable) with physical access only.
- **Consent & transparency.** Ship a user-readable **capabilities/limitations** sheet; indicate **fail-closed** posture and serviceable parts.
- **Use-case boundaries.** Not a medical device; complements human oversight rather than replacing it.
- **Human override.** Provide a **physical override/off** control and visible status (e.g., "safe mode").
- **Accessibility & signage.** Supply plain-language **capabilities/limitations** and contact details at the installation site.

11.5 Sustainability & EOL

- **Batteryless first.** Favor harvested power to avoid primary cells; specify panel area and supercap lifetime.
- **Design for end-of-life.** Prefer recyclable supercaps; avoid adhesives where possible; document materials; provide a **take-back** path for recall units.
- **Materials compliance.** Declare **RoHS/REACH** conformance; document safe supercapacitor disposal.
- **Design for repair.** Prefer **socketed/standard** parts and enclosure reuse to reduce e-waste during recall/replace.

Section 12: Conclusion

From patching to proving. ENF recasts embedded intelligence as a sealed Neural BIOS—offline, deterministic, and hardware-bound—addressing the core problems set out in the Introduction: OTA fragility, privacy burden, and lifecycle drift. Across the paper, we defined strict invariants: no IP stack or OTA; sealed image; static memory; no heap/GC, no recursion; no ISRs beyond the threshold-qualified wake; no preemptive tasks; per-state WCET with watchdog enforcement; and PUF-anchored identity. We showed how these invariants collapse the attack surface while enabling predictable, harvested-power operation governed by explicit thresholds (`V_on`/`V_safe`/`V_cut`).

What we established. The architecture section detailed dual power islands (AON/INF), a sealed memory map, and a measured secure-boot chain; the normative spec translated invariants into a conformance checklist with allow/deny test vectors; the threat model separated eliminated classes (no network/OTA/interpreter) from residuals and non-reachability mitigations; the energy budget provided a worked low-lux PV example with capacitor windows and brown-out headroom; the minimal reference design and comparison table demonstrated practicality and trade-offs versus TinyML and Cloud/Edge-AI; the use-case postcards grounded ENF in real deployments; and the limitations & ethics section made rigidity, dataset governance, recall/replace, privacy, and EOL sustainability explicit.

Standardize what we prove. To make assurance repeatable at scale, we propose an openly specified ENF Manifest that unifies thresholds and WCET; the memory map and numeric policy; dataset IDs/licenses and subgroup metrics; derating and end-of-life (EOL) factors; and a locked toolchain (compiler/flags lockfile, container digest). Each release should ship immutable-region digests (`firmware.sha256`, `model.sha256`), a silicon binding (PUF-bind record with MAC-protected helper data), an SBOM, and test vectors—`boot_measurement.json` (allow/deny) and `wcet.csv` (per-state bounds)—with optional energy traces `I(t)/V_cap(t)` and tamper/event counters to substantiate duty-cycle and audit claims.

Next steps. We invite a small working group to finalize the Manifest schema, publish a minimal open reference (dual-island power, supervisor-gated INF), and maintain a public conformance suite spanning build reproducibility, secure-boot measurement, PUF enrollment/reconstruction, energy-budget verification, and responsible-AI checks. With these pieces in place, ENF offers a practical, privacy-preserving path to long-lived embedded autonomy—where assurance is not a promise to patch later, but evidence you can verify today.

References

- Aad, G., Abbott, B., Abed Abud, A., Abeling, K., Abhayasinghe, D. K., Abidi, S. H., et al. (2023). Recurrent neural network firmware for the ATLAS LAr calorimeter. *Journal of Instrumentation, 18*(05), P05017. https://doi.org/10.1088/1748-0221/18/05/P05017

- Ahn, S., Oh, K., & Lee, J. (2023). Secure boot implementation for cyber-resilient inverter MCUs. *IEEE Transactions on Industrial Informatics*. https://ieeexplore.ieee.org/abstract/document/10360278/

- Banbury, C. R., Reddi, V. J., Torelli, P., Holleman, J., Roberts, D., et al. (2021). Benchmarking tinyML systems: Challenges and direction. *arXiv preprint* arXiv:2102.07676. https://arxiv.org/abs/2102.07676

- Baischer, J. (2021). *Object detection using neural networks on embedded FPGA platforms* [Master's thesis, Graz University of Technology]. https://doi.org/10.34726/hss.2021.69314

- 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 preprint* arXiv:2004.07213.

- Cavoukian, A. (2011). *Privacy by Design: The 7 Foundational Principles*. Information and Privacy Commissioner of Ontario.

- de Oliveira, J. B., & Bastos-Filho, T. F. (2021). Non-invasive anomaly detection in embedded systems using autoencoders on current signals. *IFAC-PapersOnLine, 54*(13), 322–327. https://doi.org/10.1016/j.ifacol.2021.10.199

- Divakarla, K. P. (2017). *ISO 26262 and IEC 61508 functional safety overview*. NXP Semiconductors.

- EU. (2025). *EU AI Act (draft)*. European Commission, Brussels.

- Forti, V., Baldé, C. P., Kuehr, R., & Bel, G. (2020). *The Global E-waste Monitor 2020*. ITU/UNITAR.

- GDPR. (2018). *Regulation (EU) 2016/679 (General Data Protection Regulation).* Official Journal of the European Union.

- Gong, B. (2019). *Formal methods in low-power embedded systems: Verification of FSM-controlled fallback paths* [Doctoral dissertation, University of Connecticut]. https://digitalcommons.lib.uconn.edu/dissertations/2372

- Halvorsen, L. (2018). *FSM-based message encoding for deterministic IoT swarms* [Master's thesis, NTNU].

https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2559482/18620_FULLTEXT.pdf

- 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. https://doi.org/10.1109/JPROC.2014.2320516
- Hovanes, J. (2023). *Reliability and aging in SRAM PUFs* [Master's thesis, Auburn University]. https://auetd.auburn.edu/bitstream/handle/10415/8673/Master_s_Thesis___Joshua_Hovanes%20(1).pdf
- IEEE Standards Association. (2018). *IEEE Standard for a Real-Time Operating System (RTOS) for Small-Scale Embedded Systems* (IEEE Std 2050-2018). https://standards.ieee.org/ieee/2050/6783/
- IoT Security Foundation. (2021). *IoT Security Assurance Framework* (v3.0). https://www.iotsecurityfoundation.org
- IoT Security Foundation. (2023). *IoT Security Assurance Framework* (v3.0). https://www.iotsecurityfoundation.org
- Johnny, R., & Knutsson, H. (2021). *CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs*. Arm Ltd.
- Kallimani, R., Pai, K., Raghuwanshi, P., & Iyer, S. (2023). TinyML: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications, 82*, 29015–29044. https://doi.org/10.1007/s11042-023-16740-9
- Kallimani, R., Pai, K., Raghuwanshi, P., & Iyer, S. (2024). TinyML: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications, 83*, 29015–29045.
- Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2017). Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification (CAV 2017)* (pp. 97–117). Springer.
- Khaligh, A., & Zeng, P. (2013). Review of the application of energy harvesting in buildings. *Measurement Science and Technology, 25*(1), 012002.
- Khaligh, A., & Zeng, P. (2010). Kinetic energy harvesting using piezoelectric and electromagnetic technologies—State of the art. *IEEE Transactions on Industrial Electronics, 57*(3), 850–860.

- Khalil, M., Muller, W., & Krishnamurthy, D. (2022). On the reliability of physically unclonable functions over time. *Sensors, 22*(14), 5168. https://doi.org/10.3390/s22145168

- Khanna, A., Agrawal, D., & Wagh, S. (2020). Batteryless IoT—A comprehensive review of energy harvesting techniques for self-powered sensing—A survey. *IEEE Access, 8*, 183573–183609.

- Kim, S., Sudevalayam, S., & Kulkarni, P. (2010). Autonomic networking for energy harvesting wireless sensor networks. *ACM SIGCOMM Computer Communication Review, 14*(1), 22–32. https://doi.org/10.1145/1865106.1865113

- Koskela, M., & Kylänpää, I. (2024). *Security of field devices in future water management systems*. VTT Technical Research Centre of Finland. https://cris.vtt.fi/files/101937683/Security_of_Field_Devices_in_Future_Water_Management.pdf

- Kulkarni, V., Gao, J., & Hamid, M. (2024). Trust anchors in supply chains: PUF-based provenance control. *IEEE Access, 12*, 45612–45625. https://ieeexplore.ieee.org/document/10570172

- Lesund, H. (2017). *SPI, I²C and M-Bus comparison in deterministic IoT communication* [Master's thesis, NTNU]. https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2456608

- Li, W., Liu, Y., Zhou, Y., Li, Z., & Lin, Z. (2021). Firmware modeling and analysis with graph neural networks. In *IEEE Symposium on Security and Privacy* (pp. 1054–1071). https://doi.org/10.1109/SP40001.2021.00029

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

- Liu, Y. (2024). *PUF-based security solutions and applications*. PUFsecurity. https://www.pufsecurity.com/wp-content/uploads/2024/09/Book-2_PUF-based-Security-Solutions-and-Applications_2024_July.pdf

- Mannan, F., Lauga, E., & Goldstein, R. E. (2020). A minimal model of the hydrodynamical coupling of flagella on a spherical body. *Journal of the Royal Society Interface, 17*(172), 20200253. https://doi.org/10.1098/rsif.2020.0253

- Matiko, J. W., Grabham, N. J., Beeby, S. P., & Tudor, M. J. (2014). Review of energy harvesting techniques for wireless sensor networks. *Renewable and Sustainable Energy Reviews, 34*, 225–235.

- Mo, Z., Thomas, J., & Song, D. (2024). Confidential computing systematization: Architectures, trust anchors, and policy integration. *ACM Computing Surveys, 57*(1), Article 3. https://doi.org/10.1145/3670007

- Mustafa, M., Zhao, H., & Wang, X. (2025). Energy and security implications of federated learning in embedded edge systems. *Sensors, 25*(11), 3457. https://doi.org/10.3390/s25113457

- NIST (National Institute of Standards and Technology). (2018). *NIST Special Publication 800-193: Platform Firmware Resiliency Guidelines*. https://doi.org/10.6028/NIST.SP.800-193

- NIST (National Institute of Standards and Technology). (2020). *NISTIR 8259A: IoT device cybersecurity capability core baseline*. U.S. Department of Commerce.

- NTIA (National Telecommunications and Information Administration). (2021). *The minimum elements for a software bill of materials (SBOM).* U.S. Department of Commerce.

- NXP Semiconductors. (2017). *Functional safety overview: Automotive and industrial IEC 61508 and ISO 26262*. https://www.nxp.com

- Oliveira, R. F., & Kim, S. (2024). Toward analog-domain collective behavior in intelligent edge networks. *Patterns, 5*(4), 100945. https://doi.org/10.1016/j.patter.2024.100945

- Pannuto, P., Gummeson, J., Kempke, B., & Dutta, P. (2015). MBus: An ultra-low-power interconnect for next-generation nanodevices. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys 2015).* https://doi.org/10.1145/2872887.2750376

- Parikh, R., & Parikh, A. (2025). Secure, event-triggered activation pipelines for edge inference in hardware-monitored systems. *Preprints.* https://doi.org/10.20944/preprints202507.0098.v1

- Paradiso, J. A., & Starner, T. (2005). Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing, 4*(1), 18–27. https://doi.org/10.1109/MPRV.2005.9

- Pappu, R., Recht, B., Taylor, J., & Gershenfeld, N. (2002). Physical one-way functions. *Science, 297*(5589), 2026–2030. https://doi.org/10.1126/science.1074376

- Piller, S., Perruchoud, A., & Buchwald, A. (2001). Photovoltaic module performance at different irradiances and temperatures. *Renewable Energy, 24*(1), 119–144.

- Püschel, M., Singh, G., Gehr, T., & Vechev, M. (2020). ERAN: A complete formal verification framework for neural networks. *arXiv preprint* arXiv:2010.03070.

- Rabaey, J. M. (2020). The human intranet—Where swarms and humans meet. *IEEE Solid-State Circuits Magazine, 12*(4), 18–27. https://doi.org/10.1109/MSSC.2020.3024079

- Rahimi, A., Benini, L., & Rabaey, J. M. (2017). A 1.9 pJ/bit 16-channel OOK transceiver for body area networks. *IEEE Journal of Solid-State Circuits, 52*(3), 692–706.

- Rapp, M., Sulflow, A., & Schubert, E. (2012). Energy harvesting for wireless sensor nodes and devices. *Sensors and Actuators A: Physical, 175*, 27–35.

- Roy, N., & Basak, D. (2020). A comprehensive overview of energy harvesting techniques for IoT devices. *Journal of Ambient Intelligence and Humanized Computing, 10*(3), 545–559. https://doi.org/10.1007/s12652-019-01226-4

- Rutishauser, P. (2024). *Deterministic graph-based quantization for embedded neural inference* [Master's thesis, ETH Zürich]. https://doi.org/10.3929/ethz-b-000675547

- Sami, Y. (2024). *Zero-trust principles in embedded cyber-physical devices* [Doctoral dissertation, UOIT]. https://search.proquest.com/openview/993d2b7aeee74ea7d6db655ac30240b2/1?pq-origsite=gscholar

- Sarti, A., Comai, S., & Matteucci, M. (2015). Machine learning on low-power wearable devices. In *BIOSIG 2015* (pp. 547–551). https://doi.org/10.1109/BIOSIG.2015.7304310

- Shaji, D. (2023). *Training neural networks in firmware-constrained embedded devices* [Master's thesis, Uppsala University]. https://www.diva-portal.org/smash/get/diva2:1801491/FULLTEXT02

- Smaji, M. (2023). *Toward safe, trustable, and autonomous embedded AI systems* [Doctoral dissertation, Technical University of Denmark].

- Suda, N., Umuroglu, Y., Nagi, R., & Chandra, V. (2016). Throughput-optimized OpenCL-based FPGA accelerator for convolutional neural networks. In *International Conference on Embedded Computer Systems (SAMOS).* IEEE.

- Sundaram, S., Patel, R., & Raghunath, S. (2021). Security and privacy in energy harvesting IoT. In *ICCCNT 2021* (pp. 1–6). https://doi.org/10.1109/ICCCNT51525.2021.9579765

- Taleb, N. N. (2012). *Antifragile: Things that gain from disorder*. Random House.

- Trusted Computing Group. (2018). *Device Identifier Composition Engine (DICE) attestation architecture*. Trusted Computing Group.

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

- Warden, P., & Situnayake, D. (2019). *TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers*. O'Reilly Media.

- Xu, L., Li, P., Zhou, L., & Li, S. (2016). A review of energy harvesting technologies and applications. *Renewable and Sustainable Energy Reviews, 56*, 209–224.

- Yang, J., & Han, S. (2020). NetAdaptV2: Platform-aware acceleration of convolutional networks. *arXiv preprint* arXiv:2002.04002.

- Yu, M.-D., Hiller, M., Delvaux, J., Sowell, R., Heyszl, J., & Devadas, S. (2021). A tutorial on physically unclonable functions for device authentication and secret generation. *Foundations and Trends® in Privacy and Security, 2*(1–2), 333–388. https://doi.org/10.1561/3300000029

- Zhang, L., Zhuang, L., & Zhao, Y. (2014). Hardware-based secure boot and PUF integration for embedded systems. *IEEE Transactions on Emerging Topics in Computing, 2*(1), 67–75. https://doi.org/10.1109/TETC.2014.2305994

- Zhao, H., & Ristenpart, T. (2019). Hardware side-channel attacks on shared FaaS cloud platforms. In *USENIX Security Symposium*.

- Zhao, X., Wang, Y., Chen, J., Liu, J., & Li, M. (2020). Integrated sensing and communication for industrial IoT devices. *IEEE Transactions on Industrial Informatics, 16*(12), 7689–7698.

- ISO/IEC JTC 1/SC 41. (n.d.). *Internet of Things and related technologies*. https://www.iso.org/committee/648327.html

- Lessig, L. (1999). *Code and other laws of cyberspace*. Basic Books.

Appendix A — Terminology & Glossary (Informative)

A. Core Paradigm & Framework

- **ENF — Embedded Neural Firmware**
  Alias: ENF
  Definition: A sealed, offline-only neural program compiled into immutable memory and executed bare-metal as a **Neural BIOS**; scope-limited, deterministic, and silicon-bound.
- **Neural BIOS**
  Definition: Conceptual runtime describing ENF's posture: immutable image + measured boot + fixed FSM control path.
- **ENF Framework (5-Tuple)**
  Alias: 5-Tuple, ENF Framework
  Definition: Declarative parameters that define an ENF build: **T**ask, **P**ower model, **S**ecurity level, **F**allback, **C**ommunication scope. Compiles into the sealed image and Manifest.
- **ENF-Gene / ENF Gene ID**
  Alias: ENFGene
  Definition: Human-readable fingerprint encoding the 5-Tuple plus audit metadata (compiler hash, lot ID, UTC build time, lifecycle mode).
- **ENF Manifest (signed)**
  Alias: Manifest
  Definition: Authoritative, signed build record: firmware/model hashes, ROM/RAM map, WCET bounds, thresholds, toolchain lock/container digest, and the 5-Tuple.
- **Conformance Pack**
  Definition: Release bundle enabling third-party verification: Manifest + firmware/model hashes + PUFbind record + test vectors (boot_measurement.json, wcet.csv) + optional SBOM.
- **SBOM — Software Bill of Materials**
  Alias: SBOM
  Definition: Toolchain/library inventory used to demonstrate the absence of IP stacks, interpreters, or mutable runtimes.

B. Framework Parameters (with Levels)

- **T — Task Complexity**
  Levels: **T1** Simple (binary/trigger), **T2** Moderate (temporal/state, fused sensors), **T3** Complex (multi-modal, quorum/conditional arbitration).
- **P — Power Model**
  Levels: **P1** Harvested (solar/RF/thermal trickle), **P2** Event-Burst (piezo/IR burst), **P3** Discharge (one-shot capacitor, irreversible actuation).
- **S — Security Level**
  Levels: **S1** Sealed (ROM immutability + measured boot), **S2** PUF-Anchored (silicon-bound identity), **S3** Dual-Gated (PUF + analog power-signature gate).

- **F — Fallback Architecture**
  Levels: **F1** Safe-Sink (on anomaly → dormant/off), **F2** Degraded-Path (alternate FSM path), **F3** Hard-Reset (watchdog-enforced cold restart).
- **C — Communication Scope**
  Levels: **C0** None (silent/isolated), **C1 Pulse-Sync** (timing/event signaling only), **C2 Swarm** (deterministic, bounded multi-agent bus).
- **Gene Examples**
  Examples: `ENF-T1.P2.S3.F1.C0` (gesture tap; event-burst power; dual-gated security; safe-sink; silent) · `ENF-T3.P1.S2.F2.C2` (multi-modal; harvested power; PUF-anchored; degraded-path; swarm).

C. Identity, Trust & Attestation

- **PUF-Anchored Identity**
  Definition: Device keys derived (not stored) from a Physically Unclonable Function; binds measured firmware to a specific die.
- **PUFbind Record**
  Alias: PUF-bind
  Definition: Helper-data and integrity/auth binding for reliable PUF reconstruction and audit.
- **Measured Boot (Digest)**
  Alias: Measured Boot
  Definition: Cryptographic measurement of immutable regions that gates actuation (allow/deny) at boot.
- **Analog Power-Signature Gate**
  Definition: Factory-characterized analog profile (rise_time_ms, V_peak, decay_ms, jitter_tol_pct) used as an additional security factor (S3).
- **Fail-Closed**
  Definition: On measurement mismatch, brown-out, or WCET violation, outputs are clamped and the inference domain remains gated.
- **BOR — Brown-Out Reset**
  Alias: BOR
  Definition: Voltage-driven reset path ensuring deterministic recovery under energy shortage.
- **Fuse-Off (Debug)**
  Alias: Fuse-off
  Definition: Irreversible disabling of debug/programming interfaces after manufacture.
- **Non-Reachability**
  Definition: Architectural exclusion of remote code paths (no IP stack, no OTA, no telemetry), removing entire classes of attack.

D. Power & Energy (Dual-Island Model)

- **AON — Always-On Island**
  Alias: AON

Definition: Harvester/PMIC + V_cap monitor + wake logic that qualifies energy before enabling inference.

- **INF — Inference Island/Domain**
  Alias: INF
  Definition: Gated compute domain (MCU + sealed model) powered only when the energy policy is satisfied.
- **V_on / V_safe / V_cut**
  Definition: Named supercapacitor thresholds for enable (**V_on**), safe operation (**V_safe**), and cut-off (**V_cut**).
- **V_cap (aka Vcap)**
  Alias: V_cap, Vcap
  Definition: Storage-capacitor voltage used by AON to enforce the energy policy.
- **Threshold-Qualified Wake**
  Alias: Schmitt-Qualified Wake
  Definition: The sole permitted ISR — wake asserted only when AON confirms energy thresholds using a hysteresis comparator (Schmitt-style) to avoid chatter and false triggers.
- **Energy Window / Participation Window**
  Definition: Pre-computed time/energy intervals during which a node may join coordination or perform inference.

E. Determinism & Runtime Discipline

- **Single-Path FSM**
  Definition: The only control path — **Wake** → **Sense** → **Infer** → **Actuate** → **Sleep** — with per-state WCET.
- **WCET — Worst-Case Execution Time**
  Alias: WCET
  Definition: Documented per-state execution bounds; watchdog-enforced.
- **Numeric Determinism (bit-exact)**
  Definition: Quantized/fixed-point inference with pinned rounding/FMA/subnormal behavior and locked kernels/flags.
- **Time-Base Determinism**
  Definition: No external time sync (no NTP/beacons); local drift bounds recorded in the Manifest.
- **Static Memory Discipline**
  Definition: No heap/GC; fixed buffer/tensor shapes; no recursion or unbounded loops.
- **Watchdog-Enforced WCET**
  Definition: Hardware watchdog enforces per-state timing; overrun triggers fail-closed/reset.

F. Sync & Collective Coordination

- **ENF-Sync**
  Definition: Cloudless, deterministic coordination of multiple ENF nodes based on pre-compiled timing/energy windows.

- **Pulse-Sync**
  Definition: Minimal signaling (IR/LED/piezo/EM) that conveys timing edges or event triggers without data payloads (C1).
- **Silence Mesh**
  Definition: Non-chatty, duty-cycled coordination fabric that preserves ENF's non-reachability (no IP stack, no telemetry).
- **ENF Swarm**
  Alias: Swarm
  Definition: Deterministic, bounded multi-agent topology (C2) using time-windowed participation and provenance-aware slots.
- **Swarm Slot / Provenance-Aware Slotting**
  Definition: Fixed time positions reserved for nodes with verified identity/manifest, preventing collision and replay.
- **Quorum (ENF)**
  Definition: Minimum deterministic subset of nodes required to authorize a collective action in a Swarm.

G. Agent Classes & Profiles

- **ENFLogic**
  Definition: Pure FSM (rules-only), sealed; no neural inference.
- **ENFNeural**
  Definition: Sealed, quantized NN (fixed shape); no runtime loaders.
- **ENFHybrid**
  Definition: Fixed composition of FSM + sealed NN (no dynamic composition).
- **Interoperability Profiles**
  Alias: Profiles
  Definition: **Core**, **Industrial**, **Safety-Critical** — tiered expectations for attestation depth, audit, and (optional) formal methods.

H. Build, Provenance & Release Artifacts

- **Reproducible Build**
  Definition: Bit-for-bit firmware image reproducible from a signed Manifest and locked toolchain/container.
- **Toolchain Lockfile / Container Digest**
  Alias: Lockfile, Container Digest
  Definition: Frozen compiler + libraries + container image hash used to regenerate the exact binary.
- **boot_measurement.json**
  Alias: Boot test vector
  Definition: Test vector with expected boot digest and allow/deny outcome (and analog signature fields for S3).
- **wcet.csv**
  Alias: WCET test vector
  Definition: Per-state timing evidence (method + numeric bounds) for audit.

- **sealed_firmware.hex / manifest.hash / agent.gene**
  Alias: Release artifacts
  Definition: Canonical output files included in the Conformance Pack.
- **Release Bundle Naming**
  Definition: `enf-conformance-<product>-<rev>-<yyyymmdd>.zip` — standardized packaging for distribution.

I. Metrics (used in overview/evaluation)

- **EAR — Energy Autonomy Ratio**
  Alias: EAR
  Definition:

$$EAR = \frac{E_{harvested}}{E_{sleep} + E_{infer} + E_{actuate}}$$

  under stated ambient.

- **ASI — Attack Surface Index**
  Alias: ASI
  Definition: Weighted count of enabled remote-reachability classes (Network, OTA, Interpreter, Dynamic Memory, External Storage, Remote Logging); ENF targets ≈ 0.
- **PES — Privacy Exposure Score**
  Alias: PES
  Definition: Exposure = channels × fields × frequency; ENF targets 0 (no export paths).

J. Governance & Anti-Patterns

- **Registry & Governance**
  Alias: Registry
  Definition: Publication of ENF-Gene + Manifest; disclosure/recall policy; alignment with safety/security standards.
- **Recall/Replace Policy**
  Definition: Physical remediation process (since there is no OTA) defined per deployment/profile.
- **ENF-washing**
  Definition: Mislabeling mutable/connected systems as ENF despite violating invariants (e.g., presence of OTA, interpreters, or telemetry).

K. Abbreviations & Symbols

Abbreviations

- **ENF** — Embedded Neural Firmware
- **PUF** — Physically Unclonable Function

- **FSM** — Finite-State Machine
- **WCET** — Worst-Case Execution Time
- **AON** — Always-On (power island)
- **INF** — Inference (power island/domain)
- **BOR** — Brown-Out Reset
- **SBOM** — Software Bill of Materials
- **ISR** — Interrupt Service Routine
- **PMIC** — Power Management Integrated Circuit
- **ROM** — Read-Only Memory
- **OTP** — One-Time Programmable (memory)
- **DMA** — Direct Memory Access
- **NTP** — Network Time Protocol
- **LDO** — Low-Dropout Regulator
- **MCU** — Microcontroller Unit
- **OTA** — Over-the-Air (updates)
- **INT8** — 8-bit integer (quantization)

Symbols & Units

- **V_on, V_safe, V_cut, V_cap** — Voltage thresholds (volts) and storage-capacitor voltage.
- **E_harvested, E_sleep, E_infer, E_actuate** — Energy terms used in the EAR equation (joules).
- **EAR** — Energy Autonomy Ratio (unitless): $EAR = \frac{E_{harvested}}{E_{sleep}+E_{infer}+E_{actuate}}$
- **J, W, mW, µW** — joule, watt, milliwatt, microwatt.
- **V, mV** — volt, millivolt.
- **F** — farad (capacitance).
- **s, ms** — second, millisecond.
- **Hz** — hertz (frequency, if referenced in timing).
- **Δt** — time interval (seconds).