

DRENDA

Persistent Local Cognitive Architecture

Independent AI Research Project from Brazil

Technical report for external review - 19 September 2026

Executive summary

DRENDA is an independent research architecture built to study persistent local cognition: identity that survives restart, structured memory, explicit causal goals, evidence-aware decisions, tool-mediated action, experiment contracts, and reversible capability changes. It runs on an ordinary desktop-class host rather than a data-center cluster.

This report does not claim that DRENDA is AGI, ASI, conscious, independently validated, or capable of unrestricted autonomous self-improvement. It describes a working research system, the internal observations currently available, and the experiments still required.

1. Research thesis

Most language models are optimized to map context to text. DRENDA explores a different systems question: can a local architecture maintain a persistent operational identity, represent goals and gaps explicitly, gather evidence, formulate plans, execute bounded actions, learn from results, and preserve a causal history across restarts?

DRENDA is therefore best described as a hybrid cognitive research architecture. Its language layer is an interface to multiple symbolic, procedural, memory, planning, evaluation, and tool subsystems. Natural language quality is not treated as proof of general intelligence.

2. Operational loop

The intended loop is:

Perception -> semantic interpretation -> Goal/gap -> question/hypothesis -> plan -> experiment/action -> observation -> evidence -> next decision.

Each transition should carry exact identifiers, provenance, budgets, receipts, and stop conditions. A candidate is not allowed to be its own judge. A recorded event proves that a transition was recorded; it does not by itself prove a general capability.

3. Persistent architecture

The current system separates:

- a hot path for identity, active context, recent memory, current goals, and routing;
- a warm path for reusable capabilities, indexed memory, representations, and evaluation receipts;
- a cold path for long history, reports, snapshots, rejected candidates, and reproducibility material.

The complete live system is not an 80 KB program. That historical shorthand may have referred to an isolated conceptual core, but a current storage audit found no evidence that it describes the operational system.

4. Current host

The audited host reports an AMD Ryzen 7 5700 (8 cores, 16 logical processors), 23.9 GiB RAM, and an AMD Radeon RX 580 2048SP with 4 GiB reported adapter memory. These figures describe the current host, not a universal minimum requirement.

5. Evidence currently available

Internal evidence records show a single DRENDA identity, restored Resident starts, and transitions to a ready loop phase on the audited host. A public sanitized excerpt and integrity manifest are included in the evidence pack. Independent reproduction remains pending.

The architecture has also produced explicit scientific plans and stored causal gaps. At the current boundary, plan-to-executable-experiment closure and autonomous winning repair are not yet established.

6. NÉFRON

NÉFRON is an engineering repair substrate designed to record operational gaps, retain exact causal identity, support reversible repair candidates, preserve judge independence, and measure outcomes. Its foundation and live observer were implemented by Codex, not autonomously invented by DRENDA.

Current status: Foundation implemented in isolation. Live integration under validation.

The existence of NÉFRON does not establish autonomous repair. The decisive result would require a DRENDA-authored executable candidate that wins against a frozen baseline under an independent gate, loses its advantage under ablation, regains it under restore, and persists after restart.

7. Authorship boundary

- Matheus Silva: project owner, research direction, system goals, and acceptance authority.
- DRENDA: author only of objects that the event ledger attributes to the live DRENDA identity.
- Codex/engineering: infrastructure, repairs, tests, deployment mechanics, and documents authored externally.
- Independent evaluator: future holdouts, scoring, falsification, and reproduction.

Infrastructure written by Codex must not be reported as a DRENDA invention.

8. Independent evaluation protocol

A credible evaluation should:

1. freeze code, state boundary, hardware, budgets, and claim;
2. create tests after the freeze;
3. use hidden holdouts and negative controls;
4. compare against meaningful baselines under equal resources;
5. verify authorship and data leakage;
6. ablate the claimed mechanism;
7. restore it and repeat;
8. restart and test retention;
9. preserve negative results;
10. publish limitations and conflicts of interest.

Suggested verdicts are: FAIL, OBSERVATION_ONLY, CAUSAL_GAIN, TRANSFERABLE_GAIN, and INDEPENDENTLY_REPRODUCED.

9. Current limitations

- External validation has not been completed.
- Open-ended self-evolution is not established.
- Autonomous executable repair is not established.
- Natural language remains uneven and is not comparable to frontier LLM fluency.
- Long-running research may still encounter scheduler, contract, or capability-transport gaps.
- No open mathematical problem is claimed solved.

10. Invitation

The project seeks critical technical review, not endorsement by inspection. The recommended first step is verification of the public evidence pack and claim registry. A supervised isolated live evaluation can follow if a qualified evaluator believes the architecture merits deeper testing.

Contact: Matheus Silva - brtzenfone@gmail.com