

# Human Logic

A computation model where every statement has a speaker.

Jared Lewis | PARCRI Real Intelligence / Ada Computing Company | February 2026 | Provisional Patent Filed

## The Problem

Every computer system in use today is built on anonymous computation. The CPU doesn't know who is running code. Memory doesn't know who owns data. Identity is bolted on after the fact with passwords and permissions that can be bypassed, misconfigured, or overridden. The result: lost files, tampered records, disputed actions, and no mathematical proof of who did what.

## The Solution

Human Logic is a non-Von Neumann computation model where every operation requires a speaker identity. It replaces true/false with three values (active, inactive, broken), enforces write ownership at the mathematical level, and logs every state change in an append-only, hash-chained ledger. It is Turing complete — it computes exactly what classical computers compute, but with built-in attribution and accountability.

## The Stack

### Classroom World

First application — assignments, submissions, grades, disputes

### Helena

Operating system — worlds, files, identity, inspection, federation

### Mary

Kernel — speaker registry, partitioned memory, ledger, evaluator

### Human Logic

Computation model — 10 axioms, 25 theorems, Turing complete

## Key Properties

**Write Ownership:** Only speaker S can write to S's variables. Enforced mathematically, not by permissions.

**Append-Only Ledger:** Every operation logged permanently with hash chain. History cannot be modified.

**Three-Valued Logic:** Active (fulfilled), Inactive (condition not met), Broken (commitment failed). No vacuous truth.

**Deterministic Replay:** Given the ledger, the entire system history can be replayed to the exact same state.

**No Anonymous Operations:** Every read, write, expression, and request has a speaker identity attached.

## First Application: The Classroom World

A learning management system where students own their submissions (mathematically — the teacher cannot modify student work), grades are traceable to the submission they were based on, administrator pressure to change grades is permanently recorded, and plagiarism evidence comes from timestamps and version history in the ledger. Working prototype: Python backend (zero dependencies) + React interface.

## How It Differs from All Existing Systems

Von Neumann: Shared memory

→ **Human Logic: Speaker-partitioned memory**

Von Neumann: Destructive writes

→ **Human Logic: Append-only ledger**

Von Neumann: Anonymous execution

→ **Human Logic: Every operation attributed**

## Applications Beyond Education

- Contracts as computation — commitments between speakers with automatic status tracking and permanent receipts
- Healthcare records — every modification attributed, complete audit trail, patient ownership of their own data
- Legal proceedings — evidence chain with cryptographic integrity, no disputed timelines
- Government systems — transparent decision-making with permanent, inspectable records
- Any domain where the question "who did this and when" matters

## Current Status

Jared Lewis has formal specifications for the logic, kernel, OS application, classroom world. Rights reserved a speaker. Every state change has a receipt."

✓ Working Python prototype (mary.py, helena.py, classroom.py — zero dependencies)

✓ Working React interface with live ledger inspector and tamper testing