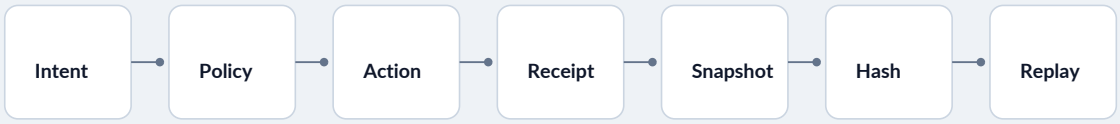
TECHNICAL BRIEF

DigiEmu Core & Proof

Deterministic Knowledge Infrastructure for reproducible and auditable AI decisions.

DigiEmu is a verification-oriented infrastructure concept for AI and complex digital systems. It focuses on one central question: can a specific knowledge or decision state be reconstructed and verified independently?

Instead of relying only on logs, screenshots or explanations, DigiEmu defines a deterministic state boundary, stores canonical snapshots, computes SHA-256 hashes and verifies execution through replay-ready receipts.



Same input -> same reconstructed state -> same hash -> PASS / FAIL

Core principle Same input -> same reconstructed state -> same hash.	Verifier output Clear PASS / FAIL result instead of vague confidence scoring.
Boundary model Deterministic facts inside the hash; metadata outside the hash.	Audit value Reconstructable evidence for governance, compliance and technical review.

01 / THE PROBLEM

Logs show what was recorded. DigiEmu verifies what can be reconstructed.

AI governance often depends on after-the-fact evidence: logs, policy references, explanations, screenshots and human notes. These are useful, but they do not automatically prove that a decision state can be reconstructed deterministically.

For high-risk workflows, the central need is stronger: a system should be able to show exactly which deterministic state existed at a boundary, which transition led to the next state, and whether the chain still verifies after independent replay.

Common audit artifact	Limitation	DigiEmu response
Log entry	Records an event, but may not define a reproducible state boundary.	Canonical snapshot plus hash.
Model explanation	Explains reasoning, but may not prove execution integrity.	Replay verifier checks state and receipt consistency.
Policy reference	Shows intended rule context, but not necessarily valid transition continuity.	Receipt links previous and next state hashes.

DigiEmu is not a replacement for security, model evaluation or human governance. It is a deterministic verification layer that helps make the state of a system inspectable after the fact.

02 / CORE VS PROOF

Core defines the boundary. Proof verifies the transition.

DigiEmu Core

Core defines how AI-related knowledge states are captured, canonicalized and separated from non-deterministic metadata.

- Canonical JSON snapshots
- Inside-hash vs outside-hash boundary
- SHA-256 state identity
- Documentation and traceability framing

DigiEmu Proof

Proof is the minimal technical verifier. It checks whether states, receipts and transition chains compose into a valid result.

- Replay verification
- Receipt continuity
- Tamper detection
- Negative tests for broken chains

Minimal verification flow

```
{
  "verifier": "digiemu-proof",
  "result": "PASS",
  "checks": [
    "canonical_state_hash",
    "receipt_prev_state_hash",
    "receipt_next_state_hash",
    "chain_continuity"
  ]
}
```

03 / FIRST USE CASE

Start with one verifiable case.

The strongest first implementation is intentionally small: one workflow, one state boundary, one receipt chain and one verification report. A medical triage-style case is a useful demonstration because it contains policy rules, risk assessment, escalation logic and a human approval boundary.

1. Intake Agent
Collects symptoms and creates the initial deterministic state.

2. Risk Assessment Agent
Evaluates severity and updates the decision state under a policy reference.

3. Action Agent
Determines intervention based on prior outputs.

4. Human Approval Boundary
High-risk classification requires an explicit approval state.

Intended business value

- Audit-ready evidence for regulated AI workflows.
- Clear technical distinction between logging and deterministic verification.
- A concrete demo for partners, investors, public-sector projects and high-risk AI pilots.
- A foundation for future DigiEmu Secure and DigiEmu Enterprise layers.

Baumgartner Digital Infrastructure
DigiEmu Core & Proof
digiemu.ch
contact@digiemu.ch