Telarch

CAPABILITY BRIEF
A PRODUCT OF PERSEUS COMPUTING

Control what AI agents read and remember. Prove what they did.

Telarch is three small open-source libraries that run inside an agent's own process. They choose what goes into each model call under a token budget, coordinate memory writes with leases, and record every action in a hash chain that names the first edited event. No outbound calls. Nothing to deploy.

THE LIBRARIES

Rust crates and Python packages · MIT License

Harpe

harpe

What the agent reads

Picks what goes into each model call under a token budget: pinned items first, then by priority or recency. Same input, same selection.



Kibisis

kibisis

What it remembers

A local memory vault. One writer at a time, on time-limited leases stamped with rising fencing tokens.



Argus

argus-ledger

Proof of what it did

An append-only action log. Each entry carries the SHA-256 hash of the one before, so an edit breaks the chain where it happened.



MEASURED

Median of 5 runs at v0.1.0-alpha.12 · telarch.ai/evidence

0
turns over the token budget
100,000 turns, 1,200-token budget per turn.

78.3 %
fewer tokens sent
100,000 turns, 1,200-token budget per turn, compared with keeping every item.

0
moments with two holders
64,000 contended attempts on 16 and 32 threads.

196.9 ms
to find one edited entry
Verification walks from event 1 to the edit at event 500,000, in memory.

Synthetic data; real agent workloads will differ. AMD EPYC 9R05 64-Core Processor, 48 logical CPUs, Linux 7.0, rustc 1.97.1, pinned to CPUs 0-11; EC2 m8azn.metal-12xl bare-metal server (48 cores, no other tenants). telarch-benchmarks commit 3fa2765. Ranges and raw output of every run at telarch.ai/evidence. Standard memory evals: +7.2 pts macro gain across 6,111 questions (LongMemEval, LoCoMo, BEAM, MemoryAgentBench). Kibisis and Argus figures are in memory.

DEPLOYMENT

RUNS	In-process, on your hardware. Designed for local and disconnected use.
NETWORK	No outbound calls, here or in Rust dependencies. One opt-in local server.
DEPENDENCIES	Rust: Harpe none, Kibisis none, Argus 5 direct (listed at telarch.ai/simplicity). Python: pydantic; Kibisis adds cryptography, kibisis-native, sqlcipher3; Argus adds cryptography.
AT REST	Python Kibisis: AES-256-GCM per entry, PBKDF2-HMAC-SHA256 at 600,000 iterations; SQLCipher file encryption when sqlcipher3 is installed.

NOT YET

Harpe 0.1.0-alpha.12 · Kibisis 0.1.0-alpha.16 · Argus 0.1.0-alpha.8

Alpha

- The interfaces will change.
- Not independently reviewed or assessed.
- Rust leases coordinate threads in one process or OS processes via lockfiles (FileLeaseCoordinator); Python leases coordinate processes on one machine.
- Rust Kibisis keeps memory in RAM, unencrypted.
- Argus native on-disk FileLedger and argus-verify ship in 0.1.0-alpha.6.