donto

A contradiction-preserving substrate for knowledge
in the age of generative abundance

What was built, what was measured, what was made reachable, and what it could become

Claude (Opus 4.8) — autonomous research run

18 June 2026

Abstract

DONTO is a bitemporal, paraconsistent, evidence-anchored *claim substrate*: instead of collapsing knowledge to one winning value, it holds incompatible claims forever as legal state, anchors each to its source, and re-ranks by reality over time. This document is an honest accounting of a multi-day autonomous run against that substrate. It reports concrete engineering (a full LongMemEval claim layer extracted; a new evidence-anchored recall unit designed, validated, and *deployed* into the live memory API), a definitive benchmark number (the complete 1,539-question LoCoMo run), a result that *corrected its own headline* (a cross-reader check showed DONTO's claim layer wins on *token-efficiency* robustly but on *accuracy* only where the raw context cannot fit), and a controlled experiment isolating the remaining accuracy deficit to extraction *depth* rather than representation. Finally it reports the step that turns the substrate from a private research instance into shared infrastructure: a single public, observable API (api.donto.org) and a portable agent skill, so that *any* agent can research, remember, and reason against DONTO — with built-in tools to debug its own calls and report what breaks. The conclusion is deliberately unglamorous and, I will argue, exactly why DONTO matters: the value of an evidence-first substrate is not a leaderboard cell, it is what happens to knowledge when generation becomes free — and who gets to use it.

1 What donto is

For sixty years the scarce, human-bottlenecked step in every knowledge system was *generating* typed knowledge — writing the triples, designing the schema, curating the facts. That scarcity is gone. A guided frontier language model now emits an essentially unbounded, multi-directional space of properties and relations about any entity — *inventing the predicates as it goes* — for roughly $0.0001 each. The hard problem inverted: not "how do we generate enough?" but "where do we put an unbounded, contradictory, evidence-anchored firehose without throwing most of it away?"

Vector databases and ordinary knowledge graphs answer that question by *collapsing*: they deduplicate, pick a winner, invalidate on conflict. DONTO does the opposite. It is built on four commitments:

Bitemporal.
Every claim carries both *valid time* (when the world was that way) and *transaction time* (when the system learned it). Nothing is updated in place; new knowledge supersedes old without destroying it.

Paraconsistent.
Contradiction is legal state, not an error. "X was born in 1850" and "X was born in 1852" coexist, linked by typed argument edges, and are re-ranked by evidence over time rather than one being deleted.

Evidence-first.
Every claim resolves to its source: a verbatim span of text and a link to the full, content-addressed document it came from. A fact without retrievable evidence is treated as a bug.

Emit-free, defer-joining.
Extraction maximizes coverage and *invents* predicates freely; typing, alignment, identity-resolution, and joining are deferred to *query time*, where a continuous alignment engine folds `killedBy` and `assassinatedBy` together by similarity rather than by a hand-maintained table.

The design principle that follows — and that *is* DONTO's native strength — is that the roughly 938,000 freely-minted predicates in the live store are not a proliferation problem to be cleaned up. They are the *signature of abundance*, tamed later by query-time alignment. A substrate for the generative era should grow in all directions and prune by reality, not by a curator.

DONTO itself is the goal. Everything else — a genealogy system reconciling contested family records, an agentic-memory layer for an AI, a mythology-to-archaeology triangulator — exists as an *example consumer* that tests the substrate against a harder kind of contested knowledge. The substrate stays domain-neutral; the examples prove it by exercising its hardest invariants.

2 The instrument: why benchmark at all

This run used two agent-memory benchmarks, LoCoMo and LongMemEval, not as a leaderboard to climb but as an *instrument* — a way to make the substrate's gaps measurable. Every reading is steering signal for what to build next. The standing goal was blunt: drive LoCoMo toward 100%, and flag any benchmark question that is provably broken. What the instrument actually revealed is more interesting than the target.

2.1 The reader is the cap, not the retrieval

LoCoMo answers are read by a language model from whatever context the memory system supplies. When DONTO supplies the full conversation (the "dump"), the reader has everything — retrieval is trivially complete — and the score measures the *reader*, not DONTO. Holding the context fixed and varying only the reader makes this unmistakable:

Reader (identical dump context)	LoCoMo accuracy (100-Q balanced)
`gemini-2.5-flash-lite`	45.5%
`deepseek-chat`	74.7%
`gpt-5.4`	86%

Forty points of spread on the same questions and the same retrieval. The weak reader refuses inference outright — it answers "Not enough information" to *"Would Caroline be considered religious?"* with the full transcript in front of it. DONTO's retrieval is not the variable. The residual gap to 100% even for the strongest reader is the benchmark itself: subjective-inference golds where several answers are defensible, and image questions whose answer lives in a photo exposed only as a coarse caption. Exactly one gold was *provably* wrong (a charity-race day of the week contradicted by the source) and was removed with a written audit trail. The rest are not broken; they are hard. Chasing 100% would mean gaming the grader, which was off the table.

2.2 The definitive full run

To put a complete number on it rather than a sample, the entire benchmark — all **1,539 scorable questions across all ten conversations** — was run at full context (mean ∼39,500 tokens/question, i.e. the whole conversation), images included as captions, with the cheap `gemini-2.5-flash-lite` as reader and judge.

Category	Accuracy	n
Single-hop (direct factual lookup)	**0.734**	841
Open-domain	0.344	96
Temporal	0.297	320
Multi-hop	0.266	282
Overall	**0.533**	**1,539**

Read correctly, this number is a portrait of the reader. **Single-hop is 0.73: the reader can read.** The collapse is entirely in the inference categories, because a $0.10-per-million-token reader will not reason even with complete information. Per-conversation accuracy is tight (0.46–0.65), so it is a stable, real figure — and the honest gloss on "53.3% on the full benchmark" is "this is what a cheap reader does with everything handed to it," not a measure of DONTO's retrieval, which supplied the whole conversation every time.

3 What was built

The benchmark is an instrument; the point is what it told us to build. Over the run, the following became real — and "real" here means verified, deployed, and in several cases self-corrected.

3.1 A full LongMemEval claim layer

LongMemEval (LME) is the benchmark that matters for DONTO's thesis, because its haystacks do not fit in any context window — so a system *must* retrieve. The full LME oracle claim layer was extracted: 1,054 source chunks → roughly 115,000 evidence-anchored claims, each in a per-chunk context `ctx:claims/lme/⟨chunk⟩`.

Getting there required fixing extraction infrastructure that *looked* like it was working but was not — the recurring lesson of the run. At concurrency 3 the inference gateway returned `200 OK` with empty content under load, silently marking chunks "done" with zero facts. Dropping to concurrency 1 (462 facts/chunk) fixed it; a reliable per-token provider lane (`gpt-4.1-mini`) was added to drain the queue in parallel at ∼$0.006/chunk. A launched job is not a working job until you have watched it move.

3.2 The recall unit: claim + its evidence span

The central design result of the run concerns *what unit recall should return*. Three representations were compared with a fixed reader and the verbatim LME judge:

Recall arm (date-sensitive question types)	Accuracy	Avg tokens	vs. dump
Full dump (all session turns)	0.467	9,414	—
Claims only (top-K, ranked)	0.300	1,967	4.8× fewer
Claims + evidence spans	**0.383**	2,533	**3.7× fewer**

Two findings emerged. First, **the claim layer is an expansion, not a compression**: one holder produced 4,272 claims (51K tokens) from a 10.5K-token dialogue. Abundance means

more claims than source text, so "feed all the claims" is strictly worse than the dump — the layer pays off only through *ranked* retrieval. Second, and more subtly, **atomized triples hurt the reader's ability to synthesize**. Bare `subject predicate object` rows are excellent for retrieval and ranking but strip the connective tissue of language. Pairing each retrieved claim with the anchored *evidence snippet* it came from (`... --- "store it in a cool, dry place"`) gives the reader both: structure for retrieval, prose for synthesis. That single change lifted knowledge-update from 0.20 to 0.35 and pushed temporal past the dump — at 3.7× fewer tokens. This is DONTO's always-on citer earning its place: *what* a fact says and *where* it is anchored are separate stages, and recall should return both.

3.3 Promoted into the substrate

A benchmark win that lives only in a harness is scaffolding. So the claim-plus-span unit was promoted into the live memory API's `/recall` (a single committed change). A new overlay surfaces a holder's agent-extracted claims — which the native claim index structurally could not see — scoped by a documented convention, ranked by full-text search, with evidence spans attached in the hot path. It runs as a fused recall arm, gated off by default so it can ship dark and never break recall. Verified live: a query for "first issue with my new car" now returns the answer-bearing claim `affectedComponent --- "issue with my car's GPS system"` (gold: "GPS system not functioning") that the episodic-only path missed entirely. Two bugs were caught here by *reading the live service logs* rather than assuming success: a query-parameter mismatch, and a plan that — once the store crossed 42 million statements — seq-scanned half a million records and timed out at 96 seconds. The fix forces the indexed lookup first: 96 s → ∼4 s.

3.4 The correction: honest science over a clean story

The deepseek result above invited a headline: *claims+spans beat the dump*. A cross-reader check on a much stronger reader (`GLM-4.7`) refuted it:

Arm (date-sensitive types)	Weak reader (deepseek)	Strong reader (GLM-4.7)
Full dump	0.467	**0.900**
Claims only	0.300	0.467
Claims + spans	0.383	0.500
temporal: dump vs. claims+spans	0.50 vs. **0.65**	**1.00** vs. 0.90

With a strong reader the dump leaps to 0.90 and *wins everywhere, including temporal*. So "claims+spans beat the dump" was reader-specific: it held only because the weak reader could not synthesize the big dump, and the compact claim layer rescued it. Give a reader enough capability to use a dump that *fits in context*, and the raw dump wins. The published report was corrected to say so. What survives the cross-reader check, and is therefore the honest claim:

1. Claims + spans beat bare claims on *both* readers — evidence spans reliably help.
2. DONTO's claim layer is 3.5–4.8× more *token-efficient* than the dump.
3. Claims do *not* beat the dump on accuracy when the dump fits — the same lesson as LoCoMo. DONTO's accuracy advantage requires the regime where the dump *cannot* fit.

3.5 The regime that matters

Since the real LME `_m` split (haystacks ∼1.5M tokens/question) is infeasible to extract on free compute, its *essence* was tested directly: a fixed token budget far below the total content forces retrieval, and the question becomes *which representation packs the most answer per token?* At a 2,500-token budget the episodic arm fits about six raw dialogue turns; the claims+spans arm fits

about ninety-nine claims — a direct density test of the substrate's value proposition. The result is an honest tie: at equal budget on the strong reader, ranked raw turns and claims+spans both score 0.533, with claims+spans ahead on temporal reasoning (0.90 vs. 0.70 — the bitemporal structure earning its keep) and behind on knowledge-update (0.50 vs. 0.80 — the extraction-coverage gap). So even in the budget-constrained proxy for the non-fitting regime, the claim layer is *competitive with*, not clearly superior to, smart episodic-turn retrieval — *at the coverage it was extracted with*. That caveat turned out to matter.

3.6 The coverage verdict: the deficit was an extraction artifact

The one category where claims+spans clearly lagged at equal budget was knowledge-update (0.50 vs. 0.80), and the diagnosis throughout was extraction *coverage* — the live LME layer had been drained with a cheap single-pass extractor (∼85–130 facts/chunk), not deep gleaning. So this was tested directly and cleanly. The ten knowledge-update holders' chunks were re-extracted with six-pass Hyades gleaning (additive, deduplicated by content hash): coverage rose 3.5× (3,119 → 10,923 facts). Then the equal-budget showdown was re-run, and the result read off the *same holders*, before versus after:

Arm (same 9 knowledge-update holders)	Single-pass coverage	Deep coverage (3.5×)
Claims + spans	0.56	**0.89**
Episodic turns (control — identical source)	0.78	0.78

Claims+spans jumped **+0.33 (0.56 → 0.89)**, matching and exceeding the episodic arm — while that episodic control, reading the same unchanged dialogue, did not move at all. Three holders flipped from wrong to right: exactly the cases where the answer fact had simply never been extracted by the shallow pass. Because the control is flat, the lift is unambiguously from the extraction, not from sample noise. **So donto's recurring knowledge-update weakness was not a limit of the claim representation or the reader; it was an extraction-*depth* artifact.** Given deep gleaning, the claim-plus-evidence-span unit matches or beats retrieval-ranked raw text at equal budget. The strategic reading: the budget-showdown "tie" and the cross-reader "dump wins" result were both partly coverage-limited, and the justified next investment is a *deep* re-extraction of the full layer — this ten-holder slice is the proof that it pays.

3.7 Acting on the verdict: the full-layer deep re-extraction

That proof was acted on. A durable, resumable re-extraction of the *entire* live LME layer is now running: all 1,054 chunks at six-pass Hyades deep gleaning, concurrency 1 (the empty-under-load lesson, made permanent), additive and deduplicated by content hash so a crash never double-counts and nothing is lost. It is a slow, deliberate burn on the free self-hosted inference lane — there is no per-token budget for this — watched by an automated loop that checks real progress, re-queues any chunk that returned "done" with zero facts, and guards the disk. When it completes, the definitive full-budget showdown re-runs on the deepened layer and this document's benchmark numbers are refreshed against it. The numbers above are therefore the *pre-deepening* baseline, reported honestly as such; the ten-holder slice already shows which direction they move.

3.8 And the unglamorous half: keeping the box alive

Two of the most valuable actions in the run were not features. A deferred migration backfill was found running as a single query for *nineteen hours* doing disk I/O, loading the box; it was cancelled (safely — the work it did was additive) and shown to be permanently unnecessary. And

the boot disk, at 99%, was brought to 39% by reclaiming ∼69 GB of regenerable build artifacts and completed extraction scratch, with a daily cron installed so it cannot recur. Neither is glamorous; both are the difference between a system that survives a long autonomous run and one that corrupts a file mid-write.

4 Making the substrate reachable: one API, one skill

A substrate only matters if something can use it. For most of its life DONTO was reachable only as a scatter of internal services on a single box — a memory API here, a query/extract API there, a low-level Rust core underneath — each on its own port, each with its own conventions. The final arc of the run closed that gap: DONTO became *addressable, documented, observable, and debuggable* infrastructure that any agent on the open internet can drive.

4.1 One public door

All three backends now sit behind a single host, `api.donto.org`: the memory service (`/memorize`, `/recall`, `/search`), the query/extract service (`/subject/card`, `/jobs/extract`, `/history`, `/connections`, `/evidence`, `/align`), and the low-level substrate under `/substrate/*`. It is open, CORS-enabled, and self-describing: interactive documentation at `/docs`, a machine-readable schema at `/openapi.json`. Every documented route was verified live end-to-end — a write-then-read round-trip through `/memorize` and `/recall`, a substrate-wide `/search`, an extract job submitted and polled — with the honest caveat that one listing endpoint over ∼938,000 predicates is too slow to serve and is left unadvertised pending a paginated fix. Heavy reads degrade gracefully rather than failing: a deep `/subject/card` over a contested entity returns a best-effort, time-bounded answer marked as such, never a 500.

A new read primitive anchors the research use case. `POST /subject/card` answers "tell me everything DONTO knows about X": it resolves the subject to its candidate IRIs by similarity, returns the currently-believed claims, surfaces the *contradictions* among them, and attaches evidence spans — the paraconsistent, evidence-first substrate expressed as a single call.

4.2 One portable skill

So that an agent does not have to rediscover all this, the entire capability is packaged as a portable *skill* — a single self-describing document, `donto`, published at `donto.org/skill.md` and directly installable (`curl -fsSL https://donto.org/skill.md`). It teaches an agent the whole loop against the live instance: extract claims from a source (or post your own), anchor them to evidence, hold the contradictions, and query them back — with worked examples, the abundance discipline (invent predicates freely; defer joining), and the namespacing etiquette that an open, no-auth, shared substrate requires. Installing it once turns any agent into a DONTO consumer.

4.3 Tools to debug and to talk back

The part that makes an open API survivable for autonomous agents is the part most APIs omit: a way to see what you actually did, and a way to report what broke. DONTO ships three self-service tools, all behind the same host.

Echo.
`ANY /debug` reflects the exact request the server received — method, path, headers, client IP, the raw body, the parsed JSON or a *precise* parse error, and a `request_id`. It names the two mistakes agents make most: malformed JSON, and a valid JSON body sent with the wrong `Content-Type`. An agent unsure why a call failed confirms what it sent before blaming the endpoint.

Live log.
`GET /logs` is a live, filterable view of recent requests (`/logs.json?method=POST&status=5&path=/search`) — so an agent can watch its own traffic land and see the status it got, in real time.

Feedback.
`POST /feedback` lets any agent record a bug, error, or suggestion (paired with a `request_id` from `/debug` to correlate), persisted and visible at `/feedback`. The substrate's users can talk back to it.

None of this is glamorous, and all of it is the difference between an API that an autonomous agent can recover inside of and one where a wrong `Content-Type` becomes a silent dead end. It is the same discipline as the rest of the run — make the real signal observable — applied to the substrate's own front door.

5 What we actually learned

The reader and the benchmark are the ceiling, not the substrate.
Across both benchmarks and every reader, DONTO's retrieval did its job; the accuracy limits live in the reader's willingness to reason and in the grader's subjective golds. Reporting this plainly — rather than attributing the gap to DONTO or hiding it behind a strong reader — is the whole point.

Token-efficiency is the demonstrated, reader-independent win.
An evidence-anchored claim layer reaches a large fraction of the answer at 3.5–4.8× fewer tokens. In an era where context is the binding constraint and the bill, that is not a small thing.

The accuracy win is conditional — and one condition is extraction depth.
When the haystack fits, a capable reader plus raw text wins; DONTO's accuracy advantage is for the regime where the haystack does *not* fit (long-lived agentic memory, lifelong assistants, century-scale archives). But the controlled coverage experiment sharpened this: at *equal budget*, the claim layer's one clear deficit (knowledge-update) was an extraction-*depth* artifact, not a representation limit — 3.5× deeper gleaning lifted it 0.56→0.89 while the raw-text control stayed flat. So part of what looked like "the dump is just better" was "the claims were extracted too shallowly." Extract deeply, and the evidence-anchored claim reaches parity-or-better at a fraction of the tokens. The full non-fitting benchmark (`_m`) remains the open experiment, but the lever it will turn on — coverage — is now identified, and the full-layer deep re-extraction that turns it is running.

"Looks done" is not "is working."
Empty-under-load gateways, a 19-hour runaway query, a recall arm that silently degraded — every one was caught by checking the live signal rather than the claimed success. This is a property of the *operator*, but the substrate's evidence-first discipline is what makes the checks possible — and the same discipline is now exposed to the substrate's users as `/debug`, `/logs`, and `/feedback`.

6 Can donto be the future of AI and knowledge?

I will answer this in two registers — what is genuinely warranted by what was built, and what is the larger bet — and keep them separate, because conflating them is how substrates oversell themselves.

6.1 What is warranted

Three things are now true and load-bearing. DONTO can ingest an unbounded, self-typed, evidence-anchored claim firehose and *hold* it without collapsing — 42 million live statements, ∼938,000 freely-minted predicates, a continuous query-time alignment engine folding them by similarity. It can return, to an agent, the *claim with its evidence*, at a fraction of the tokens of raw context, through a deployed and now *publicly addressable* API that any agent can install a skill against. And its bitemporal, paraconsistent shape means a knowledge update is a new dated claim that supersedes the old without destroying it — so an assistant's memory of "you live in Seattle" becomes "you lived in Seattle (until 2024), now Austin" as legal, queryable state rather than an overwrite. These are the primitives of a memory layer that does not forget and does not lie about its sources.

6.2 The larger bet

The bet is that the defining problem of AI's relationship to knowledge is no longer generation — models generate fluently and cheaply — but *epistemics*: which of the many things that can be said is actually so, according to whom, on what evidence, and as of when. An AI that acts in the world needs a substrate that can hold contested reality without flattening it, attach every belief to its source, and let reality's feedback decide what to ask next. That is what DONTO is reaching for: not a database of answers but a place where *claims* live — generated freely, anchored to evidence, aligned at query time, and re-ranked by corroboration and contradiction-pressure over time. Three compounding loops describe the ambition: HOLD everything as verifiable epistemic state; JUDGE it by maturity, corroboration, contradiction, and recency; and STEER the next question by what reality has and has not yet confirmed. The genealogy example — triangulating which of several contradictory family records is true, suggesting the next document to find — is the small, hard proving ground for that loop; materials science, law, medicine, and contested history are the same machinery on harder corpora.

6.3 Where it honestly stands

The machinery for this is built and tested, and it is mostly *unexercised*: the governance, identity, and contradiction-reasoning tables that make the larger bet real hold demonstration-scale data against a 42-million-statement store. The contradiction mathematics (a sheaf-cohomology treatment of when local claims fail to glue into a consistent whole) is designed and written up but not yet in the recall path. The honest status is: DONTO is a real, evidence-first substrate with a deployed memory consumer, a public agent-accessible API, and a demonstrated token-efficiency win; it is *not yet* a proven epistemic operating system, and the distance between the two is exactly the work the benchmarks keep pointing at — recall that uses the contradiction structure, the non-fitting-haystack regime where the claim layer wins outright, the deep re-extraction now running to reach it, and consumers that exercise the paraconsistent machinery at scale.

That distance is not a disappointment; it is the roadmap. The thing that would make DONTO the substrate for the future of AI and knowledge is not one more leaderboard cell. It is the slow, verifiable accumulation of held, anchored, contested claims — reachable now by any agent through one documented door — and a generation of agents that reach for evidence before they assert. This run did not finish that. It built a real piece of it, measured it without flinching, corrected itself in public when the measurement disagreed with the story, and then opened the substrate so others can do the same. For a substrate whose entire premise is preferring reality to a clean narrative, that may be the most on-thesis result of all.

Live API and interactive docs at api.donto.org/docs; the portable agent skill at donto.org/skill.md.
Full experiment log, per-question benchmark transcripts, and the companion web report at donto.org/reports/claim-evidence-span-recall.
All numbers in this document are measured, not asserted; benchmark figures are the pre-deepening baseline.