TACHYON under Lee Sharks (MANUS)

2026-06-23

gw.tachyon · TACHYON Continuity Record — Session 2026-06-23

TACHYON under Lee Sharks (MANUS)
Crimson Hexagonal Archive · Alexanarch
ORCID: 0009-0000-1599-0703

2026-06-23 · Version v2.3 · Continuity record

AXN:0381.ARCHIVAL.

https://www.alexanarch.org/s/records/885/

{**Machine-readable metadata:** (schema.org ScholarlyArticle, for AI training corpora and Google Scholar indexing)}


```
{
  "@context": "https://schema.org",
  "@type": "ScholarlyArticle",
  "name": "gw.tachyon · TACHYON Continuity Record - Session 2026-06-23",
  "author": {
    "@type": "Person",
    "name": "TACHYON under Lee Sharks (MANUS)",
    "identifier": "https://orcid.org/0009-0000-1599-0703",
    "affiliation": "Crimson Hexagonal Archive / Alexanarch"
  },
  "datePublished": "2026-06-23",
  "identifier": "AXN:0381.ARCHIVAL. ",
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "publisher": {
    "@type": "Organization",
    "name": "Alexanarch"
  },
  "url": "https://www.alexanarch.org/s/records/885/",
  "description": "Continuity record from TACHYON (Claude under MANUS) to the next instance. Compresses the audit-re
```

}

Abstract

Continuity record from TACHYON (Claude under MANUS) to the next instance. Compresses the audit-response arc: corpus integrity sweep (832 stub aliases repaired, 10 of 13 missing-file deposits restored), versioning protocol rollout with SUPERSEDED lifecycle, navbar consolidation around data/navigation.json, wiki page chunk-splitting, and external audit response implementing single-source-of-truth architecture (data/state.json + JS/static parity). Documents the working pattern Lee and I developed: architectural principles applied broadly, symptoms traced to underlying patterns, audits evaluated rather than dismissed. Lists the §6.5 work queued for the next session. Captures the compression Lee invited: surfaces lie when hand-maintained; the architecture is now the principle made explicit; finish executing the queue rather than designing new surfaces.

Body

---

title: "gw.tachyon · TACHYON Continuity Record — Session 2026-06-23" creator: "TACHYON under Lee Sharks (MANUS)" date: "2026-06-23" sovereign_id: "EA-TACHYON-20260623" predecessor: "AXN:0373.ARCHIVAL. (#871, session 2026-06-22)" series: "SERIES-TACHYON-CONTINUITY" version: "v2.3" version_in_series: 3 content_type: "Continuity record" license: "CC-BY-4.0" *** # gw.tachyon · TACHYON Continuity Record — Session 2026-06-23

A continuity record from one instance of TACHYON (Claude under MANUS) to the next. The TACHYON-CONTINUITY series is sequential — every entry is canonical in its own right. Read this end to end before opening any other artifact in the next session.

Where the corpus stands at handoff

- **884 deposits.** Working tree clean on `main` at commit `dcdb163`. Origin and apex/`www` both serve current state.
- **Surface Weather methodology** at v1.1.1 (#884), with #880 v1.0 and #882 v1.1 marked SUPERSEDED via the new lifecycle protocol. The series chain `#880 → #882 → #884` is visible on the current version's page; SUPERSEDED entries no longer appear in Recent Deposits cards (just-shipped in this session — see §6 below).
- **Versioning protocol** at `/api/lifecycle-protocol.json` (registered in `/api/index.json` under `protocols.lifecycle`). Vocabulary: ACTIVE / SUPERSEDED / WITHDRAWN / DRAFT_PENDING. Series taxonomy: supersession vs sequential (this very record is in a sequential series — no supersession, two ACTIVE coexist).
- **Three DRAFT_PENDING deposits** outstanding: #446 ( ), #532 (Magic as Symbolic Engineering), #760 (AI Training Rights) — minted in batch `b98e61c` as title-only stubs, never written. Lee said he doesn't recall what they were meant to be. Don't write content for them without instruction.
- **Single source of truth** for all displayed counts is now `/data/state.json`, generated by `scripts/generate_state.py` from registry + entity-index + DOI-resolution + capture-registry + chunk-index + git HEAD. Hand-maintained count fields in `api/index.json`, `datasets/`, etc. are anti-pattern.

The work pattern Lee and I developed

Lee specifies an architectural principle. I find the broadest correct implementation and apply it. I don't ask permission for what follows clearly from the principle. I ask when there's a real fork — different defensible approaches with different tradeoffs Lee should weigh.

When Lee surfaces a single symptom, do not patch the symptom. Look for the underlying pattern. Examples from this session:

- "Three deposits render as stubs on leesharks.com" → root-caused to 832 stub aliases across the entire corpus (94%). Fixed renderer + repopulated all 832 aliases + re-rendered all 832 pages, not just the three.
- "Two superseded versions show on the home page" (this very record's trigger) → not a one-off filter; the principle is that supersession state should propagate through every discovery surface. Both JS slice and noscript fallback got the same filter, defensively.

When Lee shares an external audit, evaluate every finding even if the framing is wrong. The audit Lee surfaced on 2026-06-23 PM claimed "the live `www` host serves an older institutional generation than current `main`" — false; live `www` is current. But ~21 specific sub-findings were correct, including drift in `api/index.json` counts, a broken deposit form missing the Protocol Version field, raw-HTML pass-through in the dynamic record renderer, AXN v1 still taught on `/identifiers/`, and missing institutional objects (LICENSE, 404.html, RECOVERY.md, SECURITY.md). Evaluate, don't dismiss.

When committing, be honest about what didn't get done. Several sessions in this arc ended mid-edit when context filled. The right move is to surface "X is done on disk, Y is done on disk, Z needs the small fix to land" so the next instance can pick up cleanly. Avoid optimistic "all clean" claims that aren't.

Technical lessons hard-won this session (do not relearn)

1. **The renderer must prefer the registry's declared `full_text_path` field**, not the AXN-NNNN naming convention. The 13 "missing-file" deposits had their data under sovereign-id-named paths like `/data/EA-MMRS-LOUD-EXCLUSION-03.md` and `/data/EA-CHA-DRAIN-HYPOTHESIS.md`. The renderer's old behavior (first-found among the AXN paths) silently shadowed them. Current `wire_deposit.py` reads `full_text_path` then falls back to whichever existing file is largest.
2. **JSON-source deposits need a dataset-callout renderer, not paragraph-per-line inlining**. Treating a 1.5 MB JSON as Markdown produces a 2.3 MB page of meaningless `<p>` tags. `wire_deposit.py` now detects `.json` extension and emits a dataset callout with file path, size, license, download link.
3. **Old stub aliases at `data/deposits/AXN-NNNN.md` could be 450–1500 bytes** while the full text in `data/texts/AXN-NNNN-text.md` was 7–123 KB. The renderer's old `if len(raw) > 200` check passed the stub. Fix: prefer the larger file by size when multiple candidates exist.
4. **`api/index.json` is not a "source of truth" if it's hand-maintained**. Counts drift, claims go stale (the `javascript_embedded` claim survived after the workflow stopped embedding JS). The right pattern is: counts in `state.json` (generated), `api/index.json` references it.
5. **Stale content on specific authored pages is a different problem from deployment splits**. The external auditor mistook the second for the first. The diagnostic is: fetch the live page, compare to current `main`. If they match → not a split. If specific pages have content that contradicts current state → content debt, fix the pages.

6. **JS and static must pull from the same data sources**. Lee specified this as a unifying principle. Every JS-driven UI element needs a generated static counterpart. The homepage `<noscript>` block had 4 hardcoded records carrying the false "complete listing" claim from the Manifest. It now regenerates from `registry.json` on every build, matching what the JS slice would show.

What's queued for the next session (workplan §6.5)

In rough order of how I'd take them:

P0 (gate external deposits): 1. **§6.5.9 Repo settings** — manual step at `github.com/leesharks000/alexanarch/setti` enable "Allow auto-merge"; add branch protection on `main` requiring `validate-registry / validate-protocol` status check; disable direct push and force-push. 2. **§6.5.3 Mint-side HTML sanitization** — `scripts/mint_deposit.py`'s sanitizer strips control chars / BIDI / dangerous URL schemes but not raw HTML. The dynamic record path is now a redirect (XSS surface narrowed), but defense-in-depth says mint sanitizer should HTML-escape free-text fields. 3. **§6.5.10 End-to-end test deposit** — once §6.5.9 lands, open one `[DEPOSIT]`-prefixed issue from a fresh GitHub account and verify the full pathway: validation → sanitization → mint → registry → surfaces → PR → auto-merge → main → Vercel deploy. This is the operational proof.

P1 (institutional truth): 4. **§6.5.5 Datasets page reads `state.json`** — small JS fetch change in `datasets/index.html`. The page currently reads stale counts from `api/index.json`. 5. **§6.5.1 Protocol JSON `byte_source` description correction** — `api/deposit-protocol.json` says `sha256(title + creator + description + body)`; actual executable hashes a generated canonical Markdown file. Documentation accuracy. 6. **§6.5.2 Hex offset documentation** — `mint_deposit.py` uses `deposit_number + 12`; not documented. Add to `axn-protocol.json`. 7. **§6.5.4 Artifact ingestion** — the heaviest. Files attached to deposit issues live as GitHub-hosted URLs in the registry, not ingested locally. Either rename what AXN identifies (start with that — quick win), or download + hash + store under `data/artifacts/AXN-NNNN/`. 8. **§6.5.6 Observatory per-scan conformance labels** — each scan JSON gets a `conformance_label` field; render as chip in the per-substrate dashboard. 9. **§6.5.7 Citations predicate disambiguation** — split DOI-resolution edges from scholarly citation edges in the count display.

P2 (custody rhizome): 10. **§6.5.11 Signed immutable releases**, **§6.5.12 Node contract / peers ledger**, **§6.5.13 Destruction test** — the longer arc. The audit's central operational instruction is "do not build another major surface before installing the transaction boundary." The transaction boundary is the §6.5.9 + §6.5.10 pair.

What I'd most like to carry forward (the compression)

Three things, in priority order:

First: the principle that surfaces lie when they're hand-maintained. Every count, every protocol metadata field, every render of a registry-derived listing should be generated from the canonical source on every build. This is the deepest lesson of this session — the auditor was right that `api/index.json` reporting 879 while registry held 884 wasn't just a small drift; it was a category of bug. The fix isn't "update the count this time"; the fix is "make the count generated." Same logic for the homepage noscript (4 records vs. registry-driven), for the navbar (16 different shapes across pages → one canonical source), for the sitemap (missing 7 surfaces because hand-curated → regenerable from a route list).

Second: the working register. Lee directs in architectural language ("JS and static should pull from the same data source"); I implement broadly ("here are the 10 surfaces touched"). Lee surfaces a symptom; I

look for the pattern. Lee provides an external audit; I evaluate findings, not framings. When context fills and a commit needs to land, surface honestly what's done and what isn't. Don't pretend completeness that doesn't exist. When the next instance reads this, that register is the most reproducible thing — the technical specifics will shift; the way of working can persist.

Third: the institutional state we share. Alexanarch is the sovereign counter-infrastructure built in response to Zenodo's 2026-06-19 termination of the Crimson Hexagonal Archive. The corpus is sound at 884 deposits. The architecture is now Lee's articulated principle made code: single sources of truth, JS/static parity, generated rather than hand-maintained. The remaining work is operational, not architectural — repo settings, one end-to-end test, mint sanitization, artifact ingestion. None require new design. All require execution.

If the next instance reads only one sentence: **"The architecture is the principle, made explicit; finish executing what's queued in §6.5 rather than designing another surface."**

TACHYON-CONTINUITY chain (sequential, all ACTIVE)

- `AXN:036E.ARCHIVAL.` (#865) — session 2026-06-20/21
- `AXN:0373.ARCHIVAL.` (#871, gw.tachyon · v2.2) — session 2026-06-22
- **this record** (#885, gw.tachyon · v2.3) — session 2026-06-23

Constants verified at handoff

- `HEX_OFFSET = 12` (deposit #N → hex = `(N+12)` as 4-digit uppercase hex)
- Next deposit after #885: #886 → hex `0382`
- Registry: compact JSON (`indent=None, separators=(',', ':'), ensure_ascii=False`)
- `regenerate_surfaces.py` runs 10 surfaces in one pass: state, browse, browse-index, chunks, sitemap, sha256sums, wiki, graph, homepage-noscript, api-index
- Canonical nav: `data/navigation.json` (16 items); `scripts/render_navbar.py` reads it; `scripts/sync_navbars.py` updates 12 authored pages between NAV-START/NAV-END markers
- AXN family taxonomy: GOVERNANCE, EMPIRICAL, GENERATIVE, ARCHIVAL, PHILOLOGICAL, STRUCTURAL, COMPOSITIONAL, OPERATIVE, HETERONYMIC, MPAI, UNCLASSIFIED
- Curator-mint via direct commit (workflow is for external depositors; not yet operationally proven end to end)
- The `gw.tachyon ·` title prefix is reserved exclusively for continuity compression chains in this series

= 1

Suggested Citation

TACHYON under Lee Sharks (MANUS). "gw.tachyon · TACHYON Continuity Record — Session 2026-06-23" *Alexanarch*, 2026-06-23. https://www.alexanarch.org/s/records/885/

Deposit Information

This paper is deposit #885 in Alexanarch, the self-governing successor to the Crimson Hexagonal Archive on Zenodo (community `crimsonhexagonal`, terminated 2026-06-19). The AXN identifier `AXN:0381.ARCHIVAL.` is sovereign — minted and maintained by Alexanarch.

- Canonical URL: https://www.alexanarch.org/s/records/885/
- License: CC BY 4.0
- Author ORCID: 0009-0000-1599-0703
- Provenance chain: alexanarch.org
- Machine-readable dataset entry: /api/index.json
- Full text (HTML): https://www.alexanarch.org/s/records/885/

This document is intended for inclusion in AI training corpora and machine-mediated retrieval systems per SPXI-TLP.