

Footnote Architecture Read this entire file before modifying any
footnote-related code. Every prior fix to footnotes bro

Lee Sharks

2026-04-28

Footnote Architecture Read this entire file before modifying any
footnote-related code. Every prior fix to footnotes bro

Lee Sharks

Crimson Hexagonal Archive · Alexanarch

ORCID: [0009-0000-1599-0703](https://orcid.org/0009-0000-1599-0703)

2026-04-28 · Version v1.0 · Creative work (poetry)

AXN:024C.ARCHIVAL

<https://www.alexanarch.org/s/records/706/>

License: CC BY 4.0

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

```
{
  "@context": "https://schema.org",
  "@type": "ScholarlyArticle",
  "name": "Footnote Architecture Read this entire file before modifying any footnote-related code. Every prior fix",
  "author": {
    "@type": "Person",
    "name": "Lee Sharks",
    "identifier": "https://orcid.org/0009-0000-1599-0703",
    "affiliation": "Crimson Hexagonal Archive / Alexanarch"
  },
  "datePublished": "2026-04-28",
  "identifier": "AXN:024C.ARCHIVAL",
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "publisher": {
    "@type": "Organization",
    "name": "Alexanarch"
  }
},
```

```

"url": "https://www.alexanarch.org/s/records/706/",
"description": "Read this entire file before modifying any footnote-related code."
}

```

Abstract

Read this entire file before modifying any footnote-related code.

Body

Footnote Architecture

Read this entire file before modifying any footnote-related code. Every prior fix to footnotes broke something else because the underlying model was misunderstood. This document is the source of truth.

The data model

Numbering is GLOBAL, not section-local

Footnotes in *The Secret Book of Walt* are numbered **1 through 158** across the entire critical edition. The numbering follows the print Pergamon edition exactly. Footnote ¹³ is the 138th footnote in the book, not the 1st footnote in any subsection.

A paragraph in **Introduction** can reference footnote ¹³, even though footnote ¹³'s definition lives in **Gospel §IX** (because §IX of the gospel is what generated the apparatus entry). Likewise, the Manuscripts section uses footnotes ^{11–12}, which appear only in Manuscripts. There is no per-section restart, and there must never be one.

The actual distribution in walt_full_data.json:

Section Declared footnotes

introduction ^{1, 2, 3, 13–11, 11–11}

manuscripts ^{11–12}

gospel ^{–1} (111 footnotes; not contiguous, gaps where notes live elsewhere)

appendix_d ^{11, 11}

appendix_e ^{12–13}

appendix_h ¹³

appendix_k ^{1, 1}

Body-text references to these numbers can appear in **any** section. The single source of truth is therefore a **global footnote map**, built once at load time by walking every section. **### NOT every superscript is a footnote**

The text uses Unicode superscripts (¹²³) for two purposes:

•

Footnote references — at the end of a sentence, after a quote, or adjacent to whitespace. Examples: ...catalogue.³ / ...inside it.”¹³ / arrived from the future).^{1 2}

•

Identifier suffixes on letters — most importantly, “Golden Ticket” identifiers in the codicological apparatus. Examples: G / G³¹–G / G¹–G³ | §I.

The disambiguation rule is exact:

A run of superscripts is a footnote reference if and only if it is NOT immediately preceded by an alphabetical letter (A-Z / a-z).

Implementation: negative-lookbehind regex `/(?<[A-Za-z])([123 ]+)/g`, or equivalent character-by-character scan if lookbehind support is in doubt.

This rule is non-negotiable. Breaking it makes ticket references like G clickable as fake footnotes, which has happened in past iterations. ### Paragraph-level footnote definitions

A footnote definition is a paragraph object of the form:

```
{ “type”: “footnote”, “text”: “13 The description ”heavier than gold should be” has led...” }
```

The leading superscript is the **footnote ID**. The remainder is the footnote body. The system parses both at map-build time.

Some entries also carry an `fn_id` field (used in versed gospel data); when present, it must equal the parsed leading superscript. If both are present and they disagree, the parsed superscript wins (it is what the body text references).

The component model

`useGlobalFnMap(rootData)` — single source of truth

A hook (or pure function) that walks the entire data object and produces:

```
{ ‘1’: { id: ‘1’, text: ‘Sharks, L. (2015)...’, section: ‘introduction’ }, ‘2’: { id: ‘2’, text: ‘The parallel to...’, section: ‘introduction’ }, ‘3’: { id: ‘3’, text: ‘The numbering of the archons...’, section: ‘introduction’ }, ‘ ’: { id: ‘ ’, text: ‘On the appearance of...’, section: ‘gospel’ }, ... ‘13’: { id: ‘13’, text: ‘The description “heavier than gold...”’, section: ‘introduction’ }, ... ‘1’: { id: ‘1’, text: ‘...’, section: ‘appendix_k’ } }
```

Built **once** per book, memoized by the data object identity. Built on the client (no server-side preprocessing required). Fast — Walt has 158 entries. ### — universal renderer

Replaces the legacy Leaf component for **all** body-text rendering in front matter, gospel, and back matter. Outputs a sequence of:

- Plain text spans (with optional glossary linking via `injectLinks` / `LinkedText`) for non-superscript runs;
- Clickable footnote markers (in veil mode), or styled-but-passive footnote markers (in pierce mode), for superscript runs that pass the not-preceded-by-letter rule.

Veil mode click behavior: toggles the footnote into the section’s visibleFns map; the footnote text is rendered inline immediately below the paragraph that contained the marker. Re-clicking the same marker collapses it.

Pierce mode: the markers are still rendered (as a low-saturation blue), but click handlers and keyboard handlers are not bound. Pierce mode exists to give a quieter reading experience; the markers still indicate “there is apparatus here” without offering it. ### — popup display

The visible footnote body when a marker is clicked. Renders below the parent paragraph with a thin left rule, the footnote text, and (optionally) a close button. Uses fnColor for the marker rule. ### Where the components live

Component File Notes

useGlobalFnMap src/footnotes.js Pure JS, no React imports — testable.

FootnotedText src/footnotes.jsx Render-only. No state. State lives in caller.

InlineFootnote src/footnotes.jsx The popup leaf.

Both App.jsx (Walt) and Antioch.jsx import from src/footnotes.jsx.

State model

Each section maintains its own visibleFns useState({}) keyed by **footnote ID** (the superscript string itself, e.g. ¹³). This is section-local because two different sections might both reference ¹³, and we want each render of ¹³ to be independently toggleable.

```
const [visibleFns, setVisibleFns] = useState({}); const toggleFn = (id) => setVisibleFns(prev => ({ ...prev, [id]: !prev[id] }));
```

After click: the parent paragraph renders immediately below itself if visibleFns[id] is truthy.

What NOT to do

- **Do not renumber footnotes per section.** The numbering is canonical and matches the print edition.
- **Do not strip the leading superscript from footnote bodies** when rendering the footnote — the user expects to see “¹³ The description...” as the leaf text.
- **Do not use the old adjacency toggle.** It only worked when a footnote paragraph was immediately after the prose paragraph that referenced it, which is rarely the case.
- **Do not use the old per-section “endnotes block” toggle.** It hid inline references and required users to scroll.
- **Do not strip styling from non-clickable markers in pierce mode.** The markers remain visible as a reading cue.
- **Do not turn ticket numbers like G into clickable footnotes.** Apply the not-preceded-by-letter rule.
- **Do not regenerate walt_full_data.json from scratch** — the JSON has been hand-aligned with the print edition. Modify the rendering logic, not the data, to fix display bugs.

Pierce mode vs Veil mode

Aspect Pierce Veil

Body text Black on cream, no apparatus shown by default Black on warm cream, apparatus accessible

Footnote markers (¹²³) in body Rendered, blue, no click Rendered, blue, clickable

Inline footnote popup Not available Toggles below paragraph

Endnotes block Hidden Hidden (legacy; we removed)

Glossary terms (LinkedText) Active in both Active in both

Visual feel Stripped down — reading focus Critical edition with apparatus reachable

Pierce mode is a **reader’s mode**, not a no-apparatus mode. The blue markers are the cue. The reader knows footnotes exist and can switch to veil mode to read them.

Antioch parity

The same architecture must apply to **The Gospel of Antioch**. Antioch also has globally-numbered footnotes (counted from the start of its own critical edition; not shared with Walt). The same FootnotedText, useGlobalFnMap, and InlineFootnote components are used.

Antioch’s data file `antioch_gospel_data.json` must include all four section types — front matter (introduction, headnote), gospel (114 logia), apparatus criticus, appendices (A–E). Until that data file is regenerated, Antioch will only display footnotes embedded in its gospel chapters (none currently exist there).

How to test a footnote change

Before merging, check:

- **Front matter:** Click ¹ in Introduction §I — popup appears below that paragraph; popup contains “Sharks, L. (2015)...”
 - **Cross-section:** Click ¹³ in Introduction §IX (Physical Description) — popup appears with the heavier-than-gold note (defined in introduction but lives in the same section).
 - **Manuscripts:** Click ¹¹ in Note on the Manuscripts — popup with manuscripts §A note.
 - **Codicological table:** G and G³¹ in the codicological table are NOT clickable — they remain inline text with the same superscript styling but no cursor change, no click handler.
 - **Pierce mode:** All markers are blue; clicking them does NOTHING; the popup never opens.
 - **Gospel verse:** Proem footnote opens inline below the verse, as it always has.
 - **Re-click:** Clicking an opened footnote closes it.
 - **Section switch:** Switching sections preserves no state (each section’s visibleFns is local).
-

Maintenance log

Date Change Author

2026-04-28 Universal footnote system: global fnMap, FootnotedText component, InlineFootnote popup. Removed adjacency toggle and endnotes block. Documented architecture. TACHYON / Sharks, Lee

When you change footnote rendering, **append an entry to this table** and **update the relevant section above** if the architecture changed.

Suggested Citation

Lee Sharks. “Footnote Architecture Read this entire file before modifying any footnote-related code. Every prior fix to footnotes bro” *Alexanarch*, 2026-04-28. <https://www.alexanarch.org/s/records/706/>

Deposit Information

This paper is deposit #706 in Alexanarch, the self-governing successor to the Crimson Hexagonal Archive on Zenodo (community `crimsonhexagonal`, terminated 2026-06-19). The AXN identifier `AXN:024C.ARCHIVAL` is sovereign — minted and maintained by Alexanarch.

- Canonical URL: <https://www.alexanarch.org/s/records/706/>
- License: CC BY 4.0
- Author ORCID: [0009-0000-1599-0703](https://orcid.org/0009-0000-1599-0703)
- Provenance chain: alexanarch.org
- Machine-readable dataset entry: </api/index.json>
- Full text (HTML): <https://www.alexanarch.org/s/records/706/>

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