

```
name: orphan-sweep
description: >-
  Find unlinked, unfindable, or misplaced content in and around a Confluence
  space using Glean. Surfaces orphaned pages, unlinked Google Docs, GitHub
  READMEs that aren't referenced in Confluence, and valuable Slack threads that
  should be promoted. Use when a user asks to find orphaned content, surface
  hidden docs, identify what's missing from a space's navigation, or run a
  content visibility audit.
```

Orphan Content Sweep

Surfaces content that should be findable but isn't — orphaned Confluence pages, unlinked Drive docs, GitHub READMEs invisible to non-engineers, and Slack threads that answered a question no one documented. Output is a triage list with a recommended action for each item.

Prerequisites

MCP tool	Purpose
<code>mcp_atlassian_*</code>	Map Confluence page tree and inbound links
<code>mcp_glean_default_search</code> / <code>chat</code> / <code>read_document</code>	Find content across Slack, Drive, GitHub

Phase 1 — Define scope

Step 1: Anchor the sweep

Confirm with the user:

- Which Confluence space (or spaces) to sweep
- Which team to use as the search anchor in Glean (team name, Slack channels, GitHub orgs/repos)
- Whether the sweep should include Drive and GitHub, or only Confluence

If not provided, ask for the space key and team name before proceeding.

Phase 2 — Confluence orphan sweep

Step 2: Map the page tree

Fetch the pages in the space and build a mental tree. **The Confluence API may return only direct children of the root rather than the full tree** — treat a low page count as a signal to dig deeper, not a fact to report.

After the initial fetch:

- If the result looks unexpectedly shallow (e.g., only one or two child pages for a space that should have content), fetch the children of each returned page explicitly before drawing any conclusions. Repeat one level deeper if needed.
- If the API continues to return suspiciously few pages, note the limitation and ask the user to confirm roughly how many pages the space contains — this is a known Confluence API behavior, not necessarily an empty space.

Once you have a reasonable picture of the tree, for each page note:

- Parent page
- Whether it appears in any other page's body as a link
- Whether it appears on the space landing page or any folder overview page

A page is a candidate orphan if it has no inbound links from within the space AND is not in the landing page navigation. Flag it — you will verify in Step 3.

Step 3: Triage Confluence orphans

Macro content limitation: Pages that consist entirely of a children macro will appear blank; pages opening with a TOC macro will appear to have no body content. Do not mark a page as empty or recommend archiving it based on a blank read alone. Flag it as likely navigation node (children macro only — confirm whether intentional) or content may not be fully visible (other macros present — confirm before acting) and let the space owner verify.

For each candidate orphan:

- Read the page title and opening paragraph

- Is it actually valuable content, or a placeholder / draft / personal note?
- Is it in the right space, or does it belong somewhere else?
- Is there a canonical page elsewhere that already covers this topic (duplicate)?

Assign each orphan one of: - **Link in** — content is good, just needs a link from the landing page or a relevant parent - **Move** — belongs in a different space or folder - **Consolidate** — overlaps with an existing page; merge and archive this one - **Archive** — outdated, empty, or superseded; no longer needed - **Investigate** — can't determine from the page alone; flag for the space owner

Phase 3 — Glean cross-source sweep

Step 4: Search for unlinked Drive content

Use Glean to search for documents owned by or shared with the team that are not referenced in Confluence. Search patterns:

- Team name + common document types ("design doc," "RFC," "runbook," "spec")
- Known project names from the space
- Key system and service names mentioned in the space

For each Drive document found that isn't linked from Confluence:

- Is it still relevant and current?
- Does a Confluence page exist that should link to it?
- Should a new Confluence page be created to surface and describe it?

Step 5: Search for unlinked GitHub READMEs

Search Glean for GitHub repositories owned by the team. For each repo with a substantive README:

- Is the README referenced from any Confluence page in the space?
- Is there a page in the Technical Documentation folder that describes this service and links to its repo?

Flag repos with no Confluence presence — these are the highest-value orphans for engineering teams, since non-engineers have no way to discover them.

Step 6: Surface Slack threads worth promoting

Search Glean in connected Slack channels for the team. Look for:

- Threads where someone asked a question and received a detailed, authoritative answer that has no Confluence equivalent
- Threads where a decision was made that isn't documented anywhere
- Pinned messages in team channels that should be on a Confluence page

Flag each thread with the suggested Confluence destination (existing page to add it to, or a new page to create).

Phase 4 — Report

Step 7: Produce the triage list

Organize output into four sections:

Confluence orphans — pages in the space with no inbound links, sorted by recommended action (Link in / Move / Consolidate / Archive / Investigate). Include: page title, page link, recommended action, one-line reason.

Unlinked Drive content — Drive documents not surfaced in Confluence. Include: document title, Drive link, recommended Confluence destination.

GitHub READMEs without Confluence presence — repos with no Confluence page. Include: repo name, GitHub link, suggested action (create a brief service page in Technical Documentation with a link to the repo).

Slack threads to promote — threads worth documenting. Include: channel, rough topic, suggested Confluence destination.

Summary stats — total items found by category, total recommended actions by type (Link in / Move / Consolidate / Archive). Helps the space owner scope the remediation effort before digging into the list.

Write the triage list for the space owner, not a KA engineer. Each item should be actionable by someone who knows the content but isn't a documentation expert.