METEHAN.AI RESEARCH REPORT · PREPRINT COMPANION GUIDE

Public Suffix List Playbook: An AEO Domain Ownership Audit

Metehan Yesilyurt

AI Search & Information Retrieval Researcher · metehan.ai

https://metehan.ai/pdf/psl-domain-ownership-audit-playbook.pdf

Abstract. A step-by-step audit for finding out which registrable domain actually gets credited when an AI assistant cites your content. It covers building a URL ownership inventory, classifying hosts against the Public Suffix List, running a boundary test, remediating by class, and verifying that citations land on a domain you own. Built as a repeatable checklist you can run quarterly.

Keywords: public suffix list, domain ownership, answer engine optimization, registrable domain, AI citations, domain architecture

Companion to the article How ChatGPT Uses the Public Suffix List to Treat Domains & Subdomains Differently. The article covers the topic in full; this document is a separate, practical implementation checklist and step-by-step playbook.

1. Scope, Definitions, and What This Playbook Assumes

This is an implementation document, not an explainer. It assumes you have read the finding that ChatGPT's client-side code contains explicit references to Public Suffix List processing for citations and domain handling [1]. What follows is the audit I run when a site owner asks why an AI assistant keeps crediting a platform instead of their own domain. Every step is executable this week with a spreadsheet, a browser, and the canonical PSL file.

Four terms carry the whole playbook, so I define them once and use them consistently.

- Registrable domain: the label set the PSL treats as a single owner's boundary. metehan.ai is its own registrable domain. metehan.vercel.app resolves to vercel.app, because vercel.app is the owner boundary there [1].
- PSL boundary: the line between what you control and what a hosting platform controls. Content inside your boundary is attributable to you. Content outside it is attributable to whoever owns the suffix.
- Authority destination: the entity that accumulates signal when a URL is retrieved or cited. The path does not change it. The registrable domain does.
- Host: the full name in the URL bar. blog.metehan.ai and metehan.ai are different hosts but the same owner. john.github.io and jane.github.io are different hosts and, from an attribution standpoint, the same platform [1].

I treat the PSL as infrastructure rather than a ranking factor, which is the framing the source article uses [1]. That distinction matters for prioritization: infrastructure work is unglamorous, cheap to do once, and expensive to skip.

2. Step 1: Build a URL Ownership Inventory

You cannot fix attribution you have not mapped. Start by listing every URL that carries content you care about: blog posts, docs, product pages, guest posts, community answers, and anything you published on a hosted platform. Then add four columns beyond the URL itself.

URL	Host	PSL registrable domain	Class	Action
https://metehan.ai/blog/example/	metehan.ai	metehan.ai	Owned root	None
https://blog.metehan.ai/post/	blog.metehan.ai	metehan.ai	Owned subdomain	Check canonical target
https://metehan.vercel.app/	metehan.vercel.app	vercel.app	Platform (PSL listed)	Attach custom domain
https://medium.com/@yourname/insight	medium.com	medium.com	Platform	Publish owned version first
https://john.github.io/tool/	john.github.io	github.io	Platform (PSL listed)	Move to registered domain
https://yourname.notion.site/page	yourname.notion.site	notion.site	Platform (PSL listed)	Keep as support only

Two rows in that table are the same content published in two places. That is the point of the exercise. The inventory makes the duplication visible, and duplication is what produces split authority [1].

If you have more than about fifty URLs, do not fill the PSL column by hand. Sort by host, deduplicate, and you will usually find that a few hundred URLs collapse into a dozen hosts. Only the dozen need classification. Everything else inherits from its host.

3. Step 2: Classify Each Host with Four Rules

Apply these rules in order. The first one that matches wins, and you stop there.

1. Did you register the registrable domain? If yes, the host is owned. This includes subdomains like blog.metehan.ai, which still belongs under metehan.ai [1].
2. Does the host sit under a suffix the PSL lists as a multi-tenant platform? github.io, notion.site, vercel.app, and blogspot.com are the examples that matter most for technical and marketing content [1]. Classify these as platform, even though you chose the subdomain name.
3. Is the host a third-party domain you cannot control at all, such as medium.com? Classify as platform. Substack and LinkedIn are not in the PSL list, which does not mean they will not work [1]. They are still platforms for attribution purposes, because you do not own the registrable domain.
4. Is the host something else entirely, such as a CDN endpoint, a staging domain, or an old brand domain? Classify it as legacy or infrastructure, and decide explicitly whether you want it indexed.

The distinction between rule 2 and rule 3 is the one people get wrong. A GitHub Pages site feels owned because you picked the name. It is not owned in PSL terms, and the source finding is specifically about systems that normalize to the registrable domain [1]. The name you chose is a label inside someone else's boundary.

One more classification note: a platform being absent from the PSL does not make it owned. It only means the list does not carve it out as a shared suffix. You still do not control the registrable domain, so the attribution logic is unchanged.

4. Step 3: Run the Boundary Test on Every Host

This is a mechanical test that takes a few minutes per host. I run it before any remediation, because remediation without it is guesswork.

1. Open the canonical PSL file [2].
2. Take the host and find the longest suffix in the file that matches its ending. For metehan.vercel.app, the longest match is vercel.app.
3. Strip that suffix from the host. What remains, plus the suffix, is the registrable domain. For metehan.vercel.app, that gives vercel.app.
4. Compare the result to the domain you actually registered. If they differ, the URL does not build your entity.
5. Record the result in the inventory. Do not skip hosts because they look obviously owned. A vercel.app deployment looks obviously owned to the person who deployed it and obviously not owned to a normalizer.

If you want to observe the behavior rather than infer it, the source article describes the verification path: browser DevTools inspection of network requests and client-side JavaScript, looking for domain normalization functions that reference PSL data structures [1]. That is the empirical check. The five steps above are the fast check you can run across an entire inventory in an afternoon.

One caveat worth building into your process: the PSL is community maintained and updated regularly, so a suffix can be added or changed over time [1]. Re-run the test on your platform hosts at least once a quarter rather than trusting a note you wrote a year ago.

5. Step 4: Remediate by Class

Different classes need different fixes. Applying the same fix everywhere is how people burn a quarter and change nothing.

Class	Primary action	What to avoid
Owned root	Publish the canonical version here. Point canonicals, sitemaps, and internal links at this host.	Do not mirror the same article across two owned hosts.
Owned subdomain	Keep it for docs, product, or regional content, but pick one citation target per topic and canonical to it.	Do not let docs and blog compete for the same query.
Platform with export (Medium, Substack, LinkedIn)	Publish the owned version first, then a shortened platform version that links back.	Do not let the platform copy be the only indexable copy.
Platform listed in the PSL (github.io, notion.site, vercel.app, blogspot.com)	Treat it as a demo or support surface. Move anything you want credited to a registered domain.	Do not build your primary content library on a suffix you do not own.
Legacy or staging host	Decide once: redirect to the owned root, or noindex and leave it alone.	Do not leave it crawlable and canonical-free.

Migration order matters more than migration completeness. Move the highest-intent pages first, meaning the pages that answer a question an AI assistant would realistically be asked. A tool page on john.github.io that resolves a real query is worth more than ten archived posts on a blogspot subdomain.

When you migrate, update canonicals, the sitemap, and internal links in the same pass. A redirect alone leaves the old host visible for a while, and during that window you can still be cited at the old boundary. I have seen a redirect sit in place for months while the platform copy kept winning citations.

6. Step 5: Verify That Citations Land Where You Want

A clean domain architecture is a hypothesis until you observe the citations. Test it rather than assuming it.

1. Write five questions your content is the best answer to. Keep them stable so comparisons over time are meaningful.
2. Ask each question in the assistants you care about and record the exact cited host, not just the brand name. The host is the data point.
3. Separate crawler traffic from human traffic in your logs using OpenAI's published user-agent documentation [3].
4. Re-run the same five questions after a migration and compare hosts. If the cited host did not change, the migration has not changed attribution yet.
5. Keep the log. A monthly table of question, assistant, and cited host is enough.

I also check the negative case. Pick a query where a platform copy of your work exists and your owned copy does not. If the assistant cites the platform, that is your baseline. Then publish the owned version and watch whether the cited host shifts. That single before-and-after is the cleanest evidence you will get without instrumentation.

Do this across more than one assistant. The source article notes that domain boundaries decide which pages get credit in ChatGPT, Perplexity, and Gemini [1], so a fix that only shows up in one product is worth confirming elsewhere before you call it done.

7. Failure Modes and Their Fixes

Symptom	Likely cause	Fix
Assistant quotes your article but cites medium.com	The platform copy is the only indexable version	Publish the owned version first and link the platform copy back
Your brand query returns a platform profile, not your site	Entity is split across hosts	Consolidate on one registrable domain and canonical everything to it
Docs pages get cited, product pages do not	Subdomain split between docs and product	Cross-link and canonical so one host is the citation target
Citations point at a vercel.app or github.io host	A deployment default was never replaced	Attach the custom domain and redirect the platform host
Citations point at a host you retired	Stale canonicals, sitemaps, or internal links	Update all three and re-check after a crawl cycle
Nothing changed after migration	The platform copy is still the canonical or only full version	Remove or deindex the duplicate, or make the owned version the only complete one

8. The One-Page Checklist

Here is the compressed version. If you only do two things, do step 1 and step 3 from above.

1. List every host that carries content you care about.
2. Mark each host owned or platform using the four rules.
3. Run the boundary test against the canonical PSL file [2].
4. Move primary content to a registrable domain you registered.
5. Publish the owned version before the platform version, every time.
6. Update canonicals, sitemap, and internal links in one pass.
7. Redirect or noindex legacy and staging hosts.
8. Log cited hosts for five stable questions, once a month.
9. Re-check platform suffixes quarterly, since the PSL changes [1].
10. Treat the PSL as infrastructure, not a ranking factor, and plan accordingly [1].

I keep this checklist short because the underlying idea is short. The PSL defines ownership boundaries, and systems that normalize to the registrable domain will credit whoever owns that boundary [1]. Everything else here is bookkeeping around that one fact.

For deeper background on the list itself, the official documentation [4], the GitHub repository [5], and the Wikipedia entry [6] are the references I send people to. When the three disagree, the canonical file is the source of truth [2].

REFERENCES & SOURCES

[1] Metehan Yesilyurt, How ChatGPT Uses the Public Suffix List to Treat Domains & Subdomains Differently, https://metehan.ai/blog/how-chatgpt-uses-the-public-suffix-list/

[2] **Public Suffix List canonical data file**, https://publicsuffix.org/list/public_suffix_list.dat

[3] **OpenAI documentation for crawler user agents**, https://platform.openai.com/docs/bots

[4] **Public Suffix List official documentation**, https://publicsuffix.org/learn/

[5] **Public Suffix List repository on GitHub**, https://github.com/publicsuffix/list

[6] **Wikipedia entry on the Public Suffix List**, https://en.wikipedia.org/wiki/Public_Suffix_List