
ENGINEERING GUIDELINE

AI Discovery and AEO Implementation

A practical, testable method for making a website easy for AI assistants, answer engines, search engines and autonomous agents to discover, understand, retrieve, verify and cite.

Version

September 2026

Author

Dipin Krishnan, Founder and Principal Engineer

Reference implementation

7unit.tech

Audience

Engineering leads, technical founders, web platform teams

Published by 7Unit, an Operational and AI Engineering Partner.
Free to read, share and apply. Attribution appreciated, not required.

What this is, and what it is not

This is the working method 7Unit used to rebuild its own site for machine readers. Everything in it has been implemented and measured on a production website, not theorised.

THE OBJECTIVE

The objective is not to make ChatGPT recommend your company. The objective is to make the business unambiguous, machine readable, efficiently retrievable, and supported by evidence. Recommendation is a downstream effect of being legible and credible, never a thing you can instruct a model to do.

Who this is for

Engineering leads and technical founders who own a commercial website and want it to survive a shift in how buyers find suppliers. You will get more from it if you can change templates, response headers and build output, rather than only page copy.

What you need to apply it

- Control over your site's HTML output, routes and headers
- A content source that can produce more than one representation of a page
- Ability to measure, at minimum Lighthouse and a real device or throttled run
- Willingness to delete marketing claims you cannot evidence

How to use it

Read Parts One to Six once, end to end. Then work from Part Seven, which is a validation checklist you can run against any site, including one you did not build. Section 21 is the single test that tells you whether the rest worked.

A NOTE ON SCOPE

Most of the strongest AEO foundations are also strong technical SEO practices. Treat this as an AI discovery layer on top of good web fundamentals, not as a replacement for SEO, accessibility, performance or high quality content. If the fundamentals are weak, fix those first. Nothing in this document compensates for a site that is slow, thin or untrue.

CONTENTS

PART ONE · PRINCIPLES

- 1 The mental model
- 2 The AI discovery stack

PART TWO · THE AGENT LAYER

- 3 The `/llms.txt` index
- 4 Markdown alternatives
- 5 Discovery relations

PART THREE · ENTITY AND STRUCTURED DATA

- 6 The JSON-LD entity graph
- 7 Service schema
- 8 Other structured data

PART FOUR · SEARCH FUNDAMENTALS

- 9 Canonical URLs

10 robots.txt

11 sitemap.xml

12 Routing hygiene

PART FIVE · SUBSTANCE AND EVIDENCE

13 Entity question coverage

14 Claims need evidence

15 External entity authority

16 Freshness

PART SIX · RENDERING AND PERFORMANCE

17 Performance and rendering

18 The pre-hydration consent problem

PART SEVEN · VALIDATION

19 Technical validation checklist

20 The final agent test

21 The implementation principle

APPENDIX

A Quick implementation summary

B References

Principles

1 The mental model

Optimise for one path, in this order.



Every step has a failure mode. A site can be discoverable but ambiguous. It can be understandable but unqualifiable, so an agent cannot tell whether you serve its user's market. It can qualify well but offer nothing to verify, which is where most marketing sites stop.

An AI system reading your site should be able to determine, without guessing:

- Who is this organisation
- What does it do
- Where does it operate
- Who does it serve
- What capabilities and services does it offer
- What evidence supports those capabilities
- Which authoritative resource contains the detail
- How can a prospective customer contact it

EXPECTATION SETTING

A `/llms.txt` file does not guarantee inclusion in an AI generated answer. Nothing does. What it does is reduce ambiguity and give agents a low cost route into authoritative site content. Treat every technique in this document as removing reasons to be excluded, not as a lever that forces inclusion.

2 The AI discovery stack

Nine layers, ordered by what actually moves the outcome. Build downwards from Critical. Skipping to the novel parts while the fundamentals are broken is the most common mistake in this field.

LAYER	IMPLEMENTATION	PRIORITY
Agent index	<code>/llms.txt</code>	Critical
Entity definition	JSON-LD, Schema.org	Critical
Canonicalisation	Canonical URLs	Critical
Search discovery	<code>sitemap.xml</code>	Critical
Crawler policy	<code>robots.txt</code>	Critical
Human content	Semantic, server readable HTML	Critical
Evidence	Case studies and shipped work	Critical
External corroboration	Credible third party references	Critical
Agent detail	<code>.md</code> alternatives	High
AI discovery relations	<code>describedby</code> and <code>alternate</code>	High
Freshness	Accurate dates, sitemap updates, IndexNow where appropriate	Medium

RULE

If a layer marked Critical is broken, no amount of work on the layers below it will compensate. An agent that cannot resolve your canonical URL does not care how elegant your `llms.txt` is.

The agent layer

3 The `/llms.txt` index

Treat it as an agent index, not a duplicate website.

Create the file at the root:

```
/llms.txt
```

Recommended shape:

```
# Company Name

> Concise, factual definition of the organisation.

Key facts:
- Locations: ...
- Markets: ...
- Client types: ...
- Contact: ...

## Detail resources

- [Services](https://example.com/services.md)
- [Industries](https://example.com/industries.md)
- [About](https://example.com/about.md)
- [Work](https://example.com/work.md)

## Capabilities

- [AI Engineering](...)
- [Product Engineering](...)
- [Workflow Automation](...)

## Industries

- [Healthcare](...)
- [Fintech](...)

## Evidence

- [Case Study A](...): Fintech, Italy, 2026
- [Case Study B](...): Healthcare, UAE, 2026

## Products

- [Product A](...)

## Contact

- [Contact](...)
```

THE 7UNIT RULE

llms.txt equals identify, qualify, route. The detail lives in the linked resources. Every sentence that could live one hop away should live one hop away.

Half the file, more of the signal

The 7Unit implementation deliberately reduced `/llms.txt` from roughly 6.8 KB to 3.4 KB. What was removed: descriptions already available in the linked resources. What was kept: the discriminating facts, meaning capability names, markets and evidence metadata. A shorter index is cheaper to retrieve and harder to misread. Length is not authority.

4 Markdown alternatives

Provide clean Markdown representations for high value informational pages.

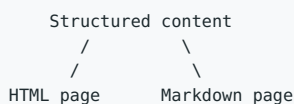
```
/services      → /services.md
/industries    → /industries.md
/work          → /work.md
/about         → /about.md

/work/case-a   → /work/case-a.md
```

The Markdown version should strip presentation noise: navigation, animations, cookie UI, CSS, JavaScript, decorative content and repeated calls to action. It should retain the useful factual content, and nothing else.

Single source of truth

Do not maintain separate website copy and AI copy. Two hand maintained versions of the same claim will diverge, and the divergence is exactly what an agent surfaces.



Both representations should be generated from the same content modules wherever practical. If that is not achievable today, make the Markdown a build artefact of the HTML source rather than a parallel document someone edits by hand.

5 Discovery relations

Tell each page where its machine readable counterparts live.

Point pages to `/llms.txt`

```
<link rel="describedby" href="/llms.txt">
```

Point HTML pages to their Markdown alternatives

```
<link
  rel="alternate"
  type="text/markdown"
  href="/services.md"
>
```

Combined, on `/services` :

```
<link rel="describedby" href="/llms.txt">
<link rel="alternate" type="text/markdown" href="/services.md">
```

These relationships may also be expressed using HTTP `Link` response headers, which is useful when you cannot modify the document head for every route.

Entity and structured data

6 The JSON-LD entity graph

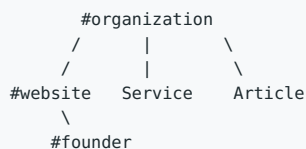
Define the organisation once, consistently, and reference it everywhere else.

```
{
  "@context": "https://schema.org",
  "@type": "Organization",
  "@id": "https://example.com/#organization",
  "name": "Example Company",
  "url": "https://example.com/",
  "description": "...",
  "email": "...",
  "founder": {
    "@id": "https://example.com/#founder"
  },
  "sameAs": [
    "VERIFIED_OFFICIAL_PROFILE"
  ]
}
```

Use stable `@id` references

```
https://example.com/#organization
https://example.com/#website
https://example.com/#founder
https://example.com/services/ai/#service
```

Prefer one connected graph over many disconnected fragments:



AVOID

Repeatedly embedding slightly different Organization objects inside articles, services and page schemas. Three near identical organisations with three different descriptions is worse than one organisation described once, because it forces a reader to decide which version is true.

7 Service schema

Use `Service` entities for real, identifiable offerings.

```
{
  "@type": "Service",
  "@id": "https://example.com/services/ai/#service",
  "name": "AI Engineering",
  "serviceType": "AI Engineering",
  "description": "...",
  "provider": {
    "@id": "https://example.com/#organization"
  },
  "url": "https://example.com/services/ai"
}
```

Do not manufacture large numbers of near identical Service entities for keyword coverage. Each entity should correspond to an

actual offering that is described on the website and that you would be happy to be asked about on a sales call.

8 Other structured data

Use types that accurately represent the site:

- `Organization`, `WebSite`, `WebPage`, `Person`
- `Service`, `Product`, `SoftwareApplication`
- `Article`, `BlogPosting`, `FAQPage`
- `BreadcrumbList`, `LocalBusiness`

Two constraints, both non negotiable. Do not add schema simply because a type exists. Do not put unsupported claims into JSON-LD. Structured data is the most machine trusted surface on your site, which makes it the worst possible place to exaggerate.

Search fundamentals

9 Canonical URLs

Every indexable HTML page should have a clear canonical URL.

```
<link
  rel="canonical"
  href="https://example.com/services"
>
```

The Markdown representation stays an alternate resource, never the canonical:

```
/services      ← canonical HTML page
/services.md   ← machine friendly alternate
```

Avoid allowing agent resources to become competing search landing pages. A human arriving on a raw Markdown file from a search result is a failure of canonicalisation, not a win for machine readability.

10 robots.txt

Audit `/robots.txt` and make sure public content is not accidentally blocked, while private, administrative, authentication and sensitive application routes remain protected.

Do not confuse the two files:

```
robots.txt = crawler access policy
llms.txt   = agent understanding and routing
```

Crawler access decisions should be intentional rather than copied from a generic AI bot blocklist found online. Decide, per crawler, whether you want the traffic and the citation. Then write down why, so the next engineer does not reverse it by accident.

11 sitemap.xml

The sitemap should contain real, canonical, indexable, human facing pages.

```
/
/services
/industries
/work
/work/case-study
/about
/contact
/insights/article
```

Normally avoid filling the primary sitemap with agent alternatives:

```
/services.md
/work.md
/llms.txt
/llms-full.txt
```

Also ensure the CMS cannot generate sitemap entries for URLs that return 404.

USEFUL RULE

A CMS record existing does not mean a public route exists. Validate route existence before emitting dynamic sitemap entries.

12 Routing hygiene

Machine readable routes must return genuine status codes. Test invalid paths:

```
/garbage.md  
/products.md  
/nope.md  
/anything.txt
```

They should return a genuine `404`, not a homepage or catch all route with HTTP `200`. This matters for both search engines and agents, because false positive 200 responses make resource discovery ambiguous. An agent that receives your homepage in response to `/pricing.md` has no way to know the resource does not exist, and may cite it as though it does.

Substance and evidence

13 Entity question coverage

Thirteen questions. If answering any of them requires guessing, the content is the problem, not the markup.

1. What is the company?
2. What does the company do?
3. Where is the company located?
4. Where does the company operate?
5. Who does the company serve?
6. What services does the company offer?
7. Which industries does the company understand?
8. What capabilities does the company have?
9. What has the company actually built?
0. What evidence supports those capabilities?
1. Does the company have experience in a given country or industry?
2. Who founded the company?
3. How can I contact the company?

14 Claims need evidence

Do not stop at capability claims. Build a retrieval trail an agent can walk:



In the 7Unit implementation, those relationships looked like this:

AI Engineering	→	AI assistants evidence
Italian Fintech	→	7Hub evidence
UAE and WhatsApp	→	WhatsApp CRM evidence
DPDP and compliance	→	Relevant compliance work

This is structurally stronger than repeating commercial keywords across the site. A keyword asserts. A linked, dated, located case study lets a machine confirm.

15 External entity authority

Technical AEO makes the site understandable once an agent reaches it. External evidence increases the chance that search and answer systems encounter and trust the entity beyond its own claims.

Useful sources include legitimate customer references, partner directories, industry directories, press coverage, conference and event profiles, GitHub organisations and projects, LinkedIn company profiles, product directories, and technical articles that earn references on merit.

Keep core entity facts consistent across all of them: organisation name, legal entity name where relevant, website, locations, description and positioning, products and services. Inconsistency across profiles is read as uncertainty about who you are.

RULE

Only add genuine identity equivalent profiles to Schema.org `sameAs` . It is a statement that the profile is the same entity, not a list of places you have a presence.

16 Freshness

Keep important facts current. Useful freshness signals include `datePublished` , `dateModified` , sitemap `lastmod` , accurate case study dates, current services, current locations, current product status and current company facts.

For sites where URLs change frequently, evaluate IndexNow or other supported search engine update mechanisms.

AVOID

Manufacturing freshness by changing dates without substantive content changes. It is detectable, and it devalues the one signal you have for genuinely new work.

Rendering and performance

17 Performance and rendering

AEO does not replace good web performance. A page an agent cannot cheaply render is a page it may not use.

Useful targets:

```
LCP ≤ 2.5 s
CLS ≤ 0.1
TBT ≤ 200 ms

Lighthouse
Performance ≥ 90
Accessibility ≥ 95
Best Practices ≈ 100
SEO = 100
```

Do not game Lighthouse. Important content should be server readable and should not unnecessarily depend on hydration before becoming visible.

FROM THE IMPLEMENTATION

The cookie banner was the LCP element

The initial assumption was that hero animation caused poor LCP. Measurement showed otherwise.

The hero was already visible in server rendered HTML. The cookie consent paragraph was physically larger than the H1 and appeared only after hydration, so it replaced the earlier hero as the final LCP candidate. The page was not slow to paint. It was slow to finish deciding what the largest painted thing was.

The architecture was changed so consent visibility is determined before first paint, while keeping the normal React provider as the authority for storing and applying consent.

MOBILE LCP BEFORE

4.0 to 4.2 s

Measured runs

MOBILE LCP AFTER

2.1 to 2.3 s

Same measurement conditions

PERFORMANCE SCORE

96 to 97

Measured runs

The broader lesson. Measure the actual LCP candidate and its timing before changing visible page design. The obvious suspect is often innocent, and redesigning the hero would have cost weeks and fixed nothing.

18 The pre-hydration consent problem

If a consent banner is server present and visible before React hydration, avoid creating a window where visible buttons do nothing. A user who clicks Accept and sees no response will click again, or leave.

The 7Unit implementation used a small pre-hydration intent bridge:

```
User clicks before hydration
↓
capture intent only
↓
React hydrates
↓
existing consent provider consumes intent
↓
normal consent path executes
```

CRITICAL CONSTRAINT

The bootstrap layer must not become a second consent authority. It should never independently construct the final consent record, enable analytics, enable advertising, grant optional storage, or duplicate provider logic. Privacy sensitive state should fail closed.

Validation

19 Technical validation checklist

Run this against your site, or against a site you are auditing. Anything unchecked is a finding.

AGENT DISCOVERY

- ☐ `/llms.txt` exists and returns `200`
- ☐ `/llms.txt` returns plain text or Markdown rather than HTML
- ☐ H1 clearly identifies the entity
- ☐ Blockquote summarises the entity
- ☐ Important capabilities are represented
- ☐ Markets and geography are explicit where relevant
- ☐ Evidence and case studies are linked
- ☐ All `/llms.txt` links resolve
- ☐ `/llms.txt` is concise enough to function as an index

MARKDOWN RESOURCES

- ☐ Markdown alternatives exist for high value pages
- ☐ `.md` routes return `text/markdown`
- ☐ Markdown does not contain HTML or UI noise
- ☐ HTML and Markdown derive from the same content source where practical
- ☐ Unknown `.md` routes return `404`

DISCOVERY RELATIONS

- ☐ `rel="describedby"` points to the applicable `/llms.txt`
- ☐ `rel="alternate" type="text/markdown"` points to the correct Markdown representation
- ☐ Relations are consistent across important pages

STRUCTURED DATA

- ☐ One consistent Organization identity
- ☐ Stable JSON-LD `@id` references
- ☐ Services represented accurately
- ☐ Founder and person relationships accurate where relevant
- ☐ `sameAs` contains only verified profiles
- ☐ Legal company name is accurate
- ☐ Founding date is accurate
- ☐ Locations are accurate
- ☐ No unsupported claims are present in schema

SEARCH FUNDAMENTALS

- ☐ Canonical URLs are correct
- ☐ `.md` resources do not compete as canonicals
- ☐ Sitemap contains valid canonical URLs
- ☐ No sitemap URL returns `404`
- ☐ `robots.txt` policy is intentional
- ☐ Important content is not accidentally `noindex`
- ☐ Invalid routes return genuine `404`
- ☐ Semantic heading hierarchy is sensible

RENDERING AND PERFORMANCE

- ☐ Important content exists in server output
- ☐ Site remains meaningful without JavaScript
- ☐ Accessibility target is met
- ☐ Core Web Vitals are reasonable
- ☐ Actual LCP candidate has been inspected
- ☐ Consent and tracking do not activate before permission
- ☐ No late mounted UI unnecessarily replaces an early LCP candidate

CONTENT INTEGRITY

- ☐ Unsupported marketing claims removed
- ☐ HTML, JSON-LD, Markdown and `/llms.txt` do not contradict one another
- ☐ Capability claims link to evidence where possible
- ☐ Locations and markets are stated consistently
- ☐ Product status is current
- ☐ Important changes have accurate freshness signals
- ☐ External entity profiles are consistent

20 The final agent test

Give an agent only `/llms.txt` as its starting context. Then ask:

Who is this organisation, what does it do, where does it operate, what evidence supports its capabilities, and which resources should you retrieve to verify those claims?

Allow the agent to retrieve the linked Markdown resources. A successful implementation lets the agent reconstruct the organisation accurately without guessing.

Then test the specific questions a buyer would actually ask:

Does this company build AI systems?
 Does it have fintech experience?
 Does it operate in the UAE?
 Does it have experience in Italy?
 Does it build WhatsApp native systems?
 Can it provide senior technical leadership?
 What evidence supports each answer?

Where the agent hedges, hallucinates or asks for clarification, you have found the exact content gap. That is the loop. Fix, republish, ask again.

21 The implementation principle

THE PRINCIPLE

Do not optimise a website to tell AI engines that the company is important. Optimise it so agents can determine precisely who the company is, retrieve the right information cheaply, and find evidence for what it claims.

In practical terms:

<code>llms.txt</code>	→ navigable
Schema and JSON-LD	→ unambiguous
Semantic HTML and Markdown	→ understandable
Case studies and evidence	→ verifiable
Sitemap, robots, canonicals	→ discoverable
External references	→ corroborated

Six properties. A site that has all six is legible to a machine and, not coincidentally, clearer to a human buyer. That overlap is the reason this work is worth doing even if answer engines never send you a single visitor.

Summary and references

A Quick implementation summary

For a new website, the minimum recommended AEO implementation, in order:

1. Clear semantic HTML and entity positioning
2. Accurate Organization JSON-LD
3. Accurate Service, Article and other relevant schema
4. Correct canonicals
5. Valid sitemap.xml
6. Intentional robots.txt
7. Concise /llms.txt
8. Markdown alternatives for important pages
9. rel="describedby" → /llms.txt
10. rel="alternate" type="text/markdown" → .md resource
11. Case studies and evidence linked to capabilities
12. Genuine 404s for invalid machine readable routes
13. Good server rendering, accessibility and Core Web Vitals
14. Consistent external entity references
15. Test the entire system from an agent's perspective

B References

- llms.txt v2 specification, llmstxt.org
- Schema.org Organization, schema.org/Organization
- Schema.org Service, schema.org/Service
- Schema.org sameAs, schema.org/sameAs
- Google Search Central, canonical URLs, developers.google.com/search/docs/crawling-indexing/consolidate-duplicate-urls
- Bing Webmaster Guidelines, bing.com/webmasters/help/bing-webmaster-guidelines-30fba23a
- IndexNow, indexnow.org

7Unit

7Unit is an Operational and AI Engineering Partner. We build the quiet systems that run loud businesses, across fintech, healthcare, compliance and enterprise software. Two capabilities: Operational Engineering, the systems a business runs on, and AI Engineering, the intelligence inside them.

If you want this implementation run against your own site, or a second opinion on an audit you have already done, write to dipin.krishnan@7unit.tech.

Dipin Krishnan, Founder and Principal Engineer
7unit.tech · linkedin.com/in/quitefounder

Version September 2026. Free to read, share and apply. The techniques described reflect the state of AI discovery tooling at the time of writing and should be re-validated as specifications change.