Barkhausen AI

WHITEPAPER · BA-W-2026-03

Machine readers of the web: how search optimization, generative engine optimization, and accessibility relate

A web page is read by four kinds of machines: search crawlers, the retrieval pipelines behind AI answer engines, assistive technology, and autonomous browser agents.

“Every leap begins as a crackle.”

DOCUMENT	BA-W-2026-03
SERIES	Whitepaper
DATE	2026
SUBJECTS	Conventions & policy; Engines & documentation; Measurement & statistics
LICENSE	CC-BY-4.0
AUTHOR	Barkhausen AI

ABSTRACT

A web page is read by four kinds of machines: search crawlers, the retrieval pipelines behind AI answer engines, assistive technology, and autonomous browser agents. Three practices — search engine optimization, generative engine optimization (GEO), and web accessibility — each optimize for one or more of these readers. Working from primary documentation, specifications, regulations, peer-reviewed studies, and, for one 2022 statement lacking an official transcript, a flagged trade-press transcription (sampled 2026-07-09), this paper states each practice as a reader–objective–evaluator tuple, maps twelve page-level signals against the four readers, and audits the circulating claim that accessibility improves AI visibility. Five signals have documented consumers in multiple reader categories. But both load-bearing links of the proposed accessibility-to-GEO mechanism are verified documentary absences, Google states accessibility is not a direct ranking factor, and agent systems split between accessibility-tree and screenshot perception. It closes with five falsifiable propositions that no public study yet measures.

IN BRIEF

A claim circulating in marketing material says that making a website accessible also makes it more visible to AI assistants. This paper checks that claim against what is actually documented — the vendors' own pages, the web standards, the regulations, and the one controlled study of AI-answer optimization. The overlap is real but narrower than the sales pitch. Some accessibility techniques genuinely serve several machine readers at once: image descriptions and clear link text are documented as useful to search engines and to screen readers, while well-structured headings and landmark elements are documented as useful to the software that prepares text for AI systems and to screen readers. Others — color contrast, keyboard support, focus handling — have no documented path to AI visibility at all: they matter for people, and the public record shows nothing more. The strongest new connection is not chatbots but browser agents: OpenAI's documentation says its Atlas browser reads the same ARIA labels that screen readers use (a browser OpenAI announced it is folding into ChatGPT on the same day this paper's sources were sampled), and Microsoft's agent tools work from the accessibility tree — while Anthropic's, Google's, and Amazon's agent products are documented as working from screenshots instead. And the popular quantified claims — accessible sites get cited some percentage more — trace to no public study: the causal chain has two links nobody has measured. The paper ends by stating exactly which experiments would settle the question, all feasible with methods already in print.

KEY FINDINGS

1. Google's John Mueller, in the March 25, 2022 Search Central office-hours session, said accessibility is 'not something that we would pick up and use as a direct ranking factor' (per Search Engine Journal's contemporaneous transcription; no official Google transcript of the session exists) — while Google's own documentation states that alt text and descriptive anchor text are consumed by its search systems, and that heading order is 'fantastic for screen readers' but 'doesn't matter' for ranking (documentation sampled 2026-07-09).
2. In the KDD 2024 study that introduced the term generative engine optimization (n = 10,000 queries, simulated GPT-3.5 engine), adding quotations, statistics, or citations improved a source's visibility in the generated answer by 30–40% on the study's position-adjusted word-count metric, while keyword stuffing scored below the unmodified baseline; on the study's deployed Perplexity.ai check (n = 200), quotation addition improved the same metric by 22%.
3. Of twelve page-level signals mapped against four machine readers from primary documentation (sampled 2026-07-09), five — alt text, heading hierarchy, landmark elements, descriptive link text, and ARIA semantics — have documented consumers in two or more reader categories, while color contrast, keyboard operability, and focus management have no documented mechanism connecting them to search or answer-engine visibility.
4. The two load-bearing links of the accessibility-to-GEO mechanism chain are verified documentary absences as of 2026-07-09: no public source documents a parse-quality stage linking page structure to extraction fidelity in any answer engine, and no public study measures the effect of retrieval quality on answer inclusion for any deployed generative engine.
5. Agent documentation splits by product, not by industry: four named systems (Playwright MCP, Playwright CLI, Stagehand's DOM mode, ChatGPT Atlas) are documented as consuming the accessibility tree or ARIA semantics, and five (Anthropic's computer-use tool, OpenAI's built-in Computer Use loop, the Gemini Computer Use tool — built into Gemini 3.5 Flash since 2026-06-24, its standalone Gemini 2.5 predecessor now listed as legacy — Amazon Nova Act, Stagehand's CUA mode) as screenshot-based, per vendor documentation sampled 2026-07-09.
6. OpenAI's help documentation states that ChatGPT Atlas 'uses ARIA tags—the same labels and roles that support screen readers—to interpret page structure and interactive elements' — the strongest vendor statement on record connecting accessibility markup to agent operability (accessed 2026-07-09, still published as of 2026-07-10); on 2026-07-09 OpenAI announced that the standalone Atlas browser is scheduled to stop working on 2026-08-09, with browser-based agentic capabilities moving into ChatGPT and Codex, and the deprecation notice does not state whether the ARIA guidance carries over.
7. 95.9% of 1,000,000 home pages had at least one automatically detectable WCAG 2 failure in the 2026 WebAIM Million (WAVE engine, February 2026), including missing alternative text on 53.1% of pages and skipped heading levels on 41.8% — figures that describe automated detection on home pages only, per the report's own methodology caveat.

CONTENTS

1 Four machine readers of the web

A claim now circulates widely in marketing material: that making a website accessible makes it more visible in AI assistants. The claim is attractive because it aligns a legal obligation with a commercial incentive, and it is plausible because both practices concern the same underlying artifact — the structure of a web page. Whether it is *documented* is a different question, and it is the question this paper answers. The method is deliberate: every claim below is drawn from primary documentation — the specifications, vendor documentation, regulations, and peer-reviewed studies listed in the references — sampled and archived on 2026-07-09, with one flagged exception: the wording of the March 2022 statement examined in chapter 4 survives only in a contemporaneous trade-press transcription, and is cited as such. Where the documentation is silent, the silence is reported as a finding rather than papered over.

The starting observation is that a web page is read by machines in at least four distinct ways, and that the four readers consume different representations of the same document. The practices examined in this paper — search engine optimization (SEO), generative engine optimization (GEO), and web accessibility — are each addressed to one or more of these readers. Understanding what each reader actually consumes, according to its own documentation, is the precondition for any claim that a technique serving one reader also serves another.

The search crawler

The oldest machine reader is the search engine's crawler and the indexing and ranking systems behind it. Google's own documentation is specific about which page-level signals those systems consume and which they do not.

Image alt text is consumed. Google's image documentation states that "Google uses alt text along with computer vision algorithms and the contents of the page to understand the subject matter of the image", and the same page describes alt text as something that "also improves accessibility for people who can't see images on web pages, including users who use screen readers or have low-bandwidth connections" [2]. The page even directs authors to W3C guidance for writing it — a general pointer to the W3C's image tutorials, not a citation of any specific success criterion [2].

Link anchor text is consumed. Google's link documentation states that "Good anchor text is descriptive, reasonably concise, and relevant to the page that it's on and to the page it links to", and offers a test: "Try reading only the anchor text (out of context) and check if it's specific enough to make sense by itself" [3]. That page never uses the word accessibility and never cites the Web Content Accessibility Guidelines (WCAG); the structural resemblance between Google's out-of-context test and WCAG's link-purpose criterion, noted in chapter 3, is this paper's observation, not Google's.

Page speed is consumed, in a bounded way. "Core Web Vitals are used by our ranking systems", Google's page-experience documentation states, while cautioning that "There is no single signal" and that relevance dominates: "Google Search always seeks to show the most relevant content, even if the page experience is sub-par" [4]. The Core Web Vitals themselves are three named, numerically thresholded metrics — Largest Contentful Paint (2.5 seconds), Interaction to Next Paint (200 milliseconds), and Cumulative Layout Shift (0.1) — and Google's documentation states: "We highly recommend site owners achieve good Core Web Vitals for success with Search and to ensure a great user experience generally" [5].

Heading *order*, by contrast, is documented as not consumed for ranking. Google's SEO Starter Guide states, in a section debunking common SEO beliefs: "Having your headings in semantic order is fantastic for screen readers, but from Google Search perspective, it doesn't matter if you're using them out of order.

The web in general is not valid HTML, so Google Search can rarely depend on semantic meanings hidden in the HTML specification" [1]. The statement is narrower than it is sometimes reported to be: it concerns the *semantic ordering* of headings, not headings generally. The same guide elsewhere lists headings among the sources used to generate a page's title link and recommends headings as a navigation aid for users [1]. The sentence is nonetheless the clearest documentary instance of a search engine explicitly decoupling an accessibility-relevant practice from ranking effect — while affirming, in the same breath, its value for screen readers.

The retrieval pipeline of a generative engine

The second reader is the pipeline behind AI answer engines. Its architectural template is retrieval-augmented generation (RAG), introduced by Lewis et al. in 2020 as models which "combine pre-trained parametric and non-parametric memory for language generation" — a generator backed by a retrieval index rather than by its training weights alone [6]. The same paper reports that retrieval-backed generation produced "more specific, diverse and factual language" than a strong parametric-only baseline [6], which is the architectural reason answer engines cite sources at all: what gets retrieved shapes what gets said.

What the production answer engines document publicly is almost entirely the *entry point* of this pipeline. Perplexity documents that "PerplexityBot is designed to surface and link websites in search results on Perplexity. It is not used to crawl content for AI foundation models", and that eligibility is gated by robots.txt [7]. OpenAI documents that "OAI-SearchBot is used to surface websites in search results in ChatGPT's search features", distinct from its training and live-browsing crawlers, and likewise gated by robots.txt [8]. Google documents that its AI Overviews and AI Mode "may use a 'query fan-out' technique" — described as "issuing multiple related searches across subtopics and data sources" — and states the eligibility rule in full: "To be eligible to be shown as a supporting link in AI Overviews or AI Mode, a page must be indexed and eligible to be shown in Google Search with a snippet, fulfilling the Search technical requirements. There are no additional technical requirements" [9]. Because these crawler pages sit adjacent to training-data collection, one boundary applies from the outset: presence in a training corpus does not imply the model retains or reproduces the content; no causal claim is made.

Between crawl and answer sits machinery the engine vendors do not describe. What is documented instead comes from the general RAG engineering literature. Anthropic's retrieval engineering material describes the standard preprocessing step: breaking a corpus into "smaller chunks of text, usually no more than a few hundred tokens", and observes that a chunk stripped of its surrounding context loses retrievability [10]. Pinecone's practitioner guide ties chunk boundaries to document structure: "By recognizing the Markdown syntax (e.g., headings, lists, and code blocks), you can intelligently divide the content based on its structure and hierarchy, resulting in more semantically coherent chunks", and notes that HTML tags "can inform text to be broken up or identified" [11]. LangChain ships this as software: its MarkdownHeaderTextSplitter exists because, in the documentation's words, "we might want to specifically honor the structure of the document itself" [12]. These are documented practices of general-purpose RAG tooling; whether Google's, OpenAI's, or Perplexity's production pipelines work this way is not publicly documented, a gap examined in chapter 4.

Assistive technology

The third reader is the oldest of the three to be formally specified. Assistive technology — screen readers foremost — does not read the rendered page or the raw HTML. It reads the accessibility tree, defined normatively in the W3C's Accessible Rich Internet Applications specification (WAI-ARIA) 1.2 as a "Tree

of accessible objects that represents the structure of the user interface (UI). Each node in the accessibility tree represents an element in the UI as exposed through the accessibility API; for example, a push button, a check box, or container" [13]. WAI-ARIA describes itself as "an ontology of roles, states, and properties that define accessible user interface elements", provided so that assistive technologies can "convey appropriate information to persons with disabilities" [13].

The browser builds this representation from the page's Document Object Model (DOM). Chromium's engineering documentation states: "The shape of the accessibility tree is determined by the DOM tree (occasionally influenced by CSS), and the accessible semantics of a DOM element can be modified by adding ARIA attributes" [14]. Chrome's developer tooling describes the result as "a subset of the DOM tree" containing the "elements from the DOM tree that are relevant and useful for displaying the page's contents in a screen reader" [15].

Ordinary HTML feeds the tree without any ARIA at all. The W3C's current draft mapping specifies that the elements `h1` through `h6` map to the "heading role, with the aria-level property set to the number in the element's tag name" [16] — the concrete mechanism by which a markup choice becomes a machine-readable "heading, level N" node. The WHATWG HTML standard defines the landmark elements by their structural semantics: "The main element represents the dominant contents of the document" [41]; "The nav element represents a section of a page that links to other pages or to parts within the page: a section with navigation links" [17]; these map to implicit ARIA landmark roles under the same draft mapping [16].

How this representation is used is measured, not conjectured. In WebAIM's tenth Screen Reader User Survey (n = 1,539 self-selected screen reader users, fielded December 2023 through January 2024), 71.6% of respondents selected navigating "through the headings on the page" as the first thing they would do to find information on a lengthy page — a single-select, stated-preference question, not an observed behavioral log — and the survey reports that heading navigation "remains the predominant method" [18]. In a separate question, 88.8% of the same sample rated heading levels very or somewhat useful [18]. For this reader, document structure is not metadata; it is the primary navigation interface.

Autonomous agents

The fourth reader is the newest and the least settled: software agents that operate a browser on a user's behalf. Their documentation divides them into two perception channels. Some named systems consume the same accessibility tree that assistive technology consumes: Microsoft's Playwright MCP (Model Context Protocol) server, for instance, documents that it "enables LLMs to interact with web pages through structured accessibility snapshots, bypassing the need for screenshots or visually-tuned models" [19], and OpenAI states that its ChatGPT Atlas browser "uses ARIA tags—the same labels and roles that support screen readers—to interpret page structure and interactive elements" [20] — a browser whose deprecation OpenAI announced on 2026-07-09, this paper's access date, effective 2026-08-09 (chapter 5) [45]. Other named systems are documented as perceiving the page as pixels: Anthropic's computer-use tool describes "screenshot capabilities and mouse/keyboard control for autonomous desktop interaction" [21], and Google's documentation for its Gemini Computer Use tool — built into Gemini 3.5 Flash since 2026-06-24, superseding the standalone Gemini 2.5 computer-use model [44] — describes a loop of screenshots in and "specific UI actions like mouse clicks and keyboard inputs" out [22].

This split — which systems sit on which side, what the vendors' own words are, and what may and may not be concluded from it — is the subject of chapter 5. What matters for the present chapter is the taxonomy: four readers, two of which (assistive technology and the tree-consuming agents) are documented as consuming the same structured representation, one of which (the search crawler)

documents a signal-by-signal mixture, and one of which (the generative-engine pipeline) documents almost nothing between its crawler and its answers.

2 Three practices as objective functions

Each of the three practices can be stated in the form of an objective function: a reader it optimizes for, an objective it pursues, and an evaluator that decides whether the objective was met. Making the three tuples explicit shows that the practices differ in every component — which is why any relationship between them has to be argued signal by signal, as chapter 3 does, rather than asserted at the level of slogans.

Table 1. The three practices as reader–objective–evaluator tuples.

PRACTICE	OPTIMIZED-FOR READER	OBJECTIVE	EVALUATOR
Search engine optimization (SEO)	Search crawler and the ranking systems behind it	Inclusion and rank in search results	The engine's own ranking systems, observed as result positions [1][4][5]
Generative engine optimization (GEO)	Generative-engine retrieval and answer synthesis	Presence and prominence of a source in generated answers	Answer-inclusion metrics, such as the GEO study's Position-Adjusted Word Count [24]
Web accessibility	Assistive technology, via the accessibility tree	Content that is perceivable, operable, and understandable for users with disabilities	WCAG conformance at levels A/AA/AAA (WCAG 2.2 is the current W3C Recommendation [23]; the version adopted by regulation in one of the two major jurisdictions is WCAG 2.1 Level AA [26])

Search engine optimization

SEO's evaluator is the ranking system itself, and its documented signal set was surveyed in chapter 1: alt text consumed [2], anchor text consumed [3], Core Web Vitals "used by our ranking systems" but bounded by relevance [4][5], heading order explicitly excluded [1]. The practice's objective function is therefore observable but not inspectable: Google documents *which* signals exist while stating that "There is no single signal" and that relevance dominates page experience [4]. What matters for this paper is the signal inventory, because it is the overlap between that inventory and the accessibility technique set — not any statement of Google's — that gives the SEO–accessibility relationship whatever substance it has.

Generative engine optimization

The term *generative engine optimization* originates in a controlled study, not a marketing document. Aggarwal et al., published at KDD 2024 — the ACM SIGKDD Conference on Knowledge Discovery and Data Mining — built a 10,000-query benchmark (GEO-bench), simulated a generative engine with GPT-3.5 synthesizing answers from supplied sources, and measured how content-level modifications to one source changed that source's visibility in the generated answer [24]. The study's headline claim is that "GEO can boost visibility by up to 40% in generative engine responses" [24]. The technique-level results define the practice's objective function precisely:

- The top-performing methods — adding citations, adding quotations, and adding statistics — "achieved a relative improvement of 30-40% on the Position-Adjusted Word Count metric and 15-30% on the Subjective Impression metric"; the best-performing methods improved on the baseline "by 41% and

28%" on the two metrics respectively, per the study's own summary (n = 10,000 queries, simulated GPT-3.5 engine) [24].

- Rewriting text in a more authoritative tone produced "no significant improvement"; the authors conclude the engines tested are "already somewhat robust to such changes" [24].
- Keyword stuffing — the classic SEO tactic — scored *below* the unmodified baseline on the simulated benchmark, and on the deployed check (n = 200) "performs 10% worse than the baseline" [24].
- A validation against a deployed engine, Perplexity.ai, on a subset of 200 test-set samples, found quotation addition improved the position-adjusted word-count metric by 22% over baseline, with citation and statistics addition showing improvements of up to 9% and 37% on the two metrics [24].
- Under a separate condition in which *all* competing sources adopt the same technique simultaneously, the gains redistribute: "the Cite Sources method led to a substantial 115.1% increase in visibility for websites ranked fifth in SERP, while on average, the visibility of the top-ranked website decreased by 30.3%" (n = 10,000 queries, simulated engine) [24]. This all-adopters condition is distinct from the single-adopter results above and the two must not be conflated.

Two scope limits attach to every number in this list. Position-Adjusted Word Count is the study's own author-defined metric — word count weighted by an exponentially decaying function of citation position — and the primary engine is a GPT-3.5 simulation; the percentages should not be generalized beyond that metric and that engine [24]. The deployed Perplexity.ai check is a directional validation on 200 samples, not a replication at benchmark scale [24]. A companion note, the GEO experiment, reports the study's design and findings in more detail.

The GEO tuple is therefore: reader = the retrieval-and-synthesis pipeline; objective = the source's presence and prominence in the generated answer; evaluator = answer-inclusion metrics of the kind the KDD study defined. Notably, every technique the study found effective operates on the *content* of the page — quotations, statistics, citations — and none of them touches the page's markup structure. That asymmetry becomes the GEO-only region of the map in chapter 3.

Web accessibility

Accessibility's evaluator is the only one of the three that is published as a normative specification. The Web Content Accessibility Guidelines (WCAG) 2.2, a W3C Recommendation since December 2024, define success criteria (SC) at three levels, and conformance is cumulative: "For Level AA conformance, the web page satisfies all the Level A and Level AA success criteria, or a Level AA conforming alternate version is provided" [23]. The criteria most relevant to machine-readable structure are terse, testable statements:

- SC 1.1.1 Non-text Content (Level A): "All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below" — the criterion continues with enumerated exception categories not reproduced here [23].
- SC 1.3.1 Info and Relationships (Level A): "Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text" [23]. This is the normative anchor for every claim in this paper connecting semantic HTML to machine-readable structure.
- SC 2.4.4 Link Purpose (In Context) (Level A): "The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general" [23].
- SC 2.4.6 Headings and Labels (Level AA): "Headings and labels describe topic or purpose" [23]. One point of discipline: this criterion does not mandate heading hierarchy or nesting order, and WCAG 2.2 contains no success criterion that does.

- SC 3.1.1 Language of Page (Level A): “The default human language of each web page can be programmatically determined” [23].

Two legal instruments give this evaluator force, and their dates are stated here factually, without compliance guidance. In the European Union, the European Accessibility Act — Directive (EU) 2019/882 — required Member States to transpose it by 28 June 2022 and to apply the measures from 28 June 2025; its scope provision states that “Without prejudice to Article 32, this Directive applies to the following services provided to consumers after 28 June 2025”, a list that includes e-commerce services, with Article 32 containing transitional measures and Article 31(3) permitting deferral of one obligation, concerning emergency communications, to 2027 [25]. One verified textual fact belongs on the record because it is widely misstated: the directive’s text contains zero occurrences of “WCAG.” Any WCAG linkage on the EU side would run through European standardization rather than through the directive itself, and no official EU source, as of 2026-07-09, confirmed an EAA-specific harmonized standard.

In the United States, the Department of Justice’s final rule under Title II of the Americans with Disabilities Act (89 FR 31320, published 24 April 2024) is explicit about the standard: “the Department is requiring that public entities comply with the WCAG 2.1 Level AA success criteria and conformance requirements” for state and local government web content and mobile apps [26]. The rule’s original compliance dates — 24 April 2026 for public entities with a population of 50,000 or more, 26 April 2027 for smaller entities and special district governments [26] — were extended by one year in an interim final rule (91 FR 20902, published and effective 20 April 2026), to 26 April 2027 and 26 April 2028 respectively [27]. Those extended dates were current and controlling as of 2026-07-09, but they are the subject of an active lawsuit by the National Federation of the Blind against the Departments of Justice and Health and Human Services (filed in U.S. District Court, May 2026), which seeks to vacate the interim final rule and reinstate the original dates; no stay or vacatur had issued as of this paper’s date [28].

The accessibility tuple is therefore: reader = assistive technology via the accessibility tree; objective = perceivable, operable, understandable content; evaluator = WCAG conformance, given regulatory force in one jurisdiction directly [26] and in the other through a standards chain that does not name WCAG in the legislative text itself [25]. No component of this tuple mentions search engines or answer engines. Whatever the three practices share, they share it at the level of concrete page signals — which is where the next chapter goes.

3 The shared substrate

The three practices meet, when they meet at all, in the individual signals of a page: its alt text, its headings, its landmarks, its link text. Table 2 maps twelve such signals against the four readers of chapter 1, using only what the sampled documentation states. Three graded cell values appear: *consumed* (the reader’s documentation states it uses the signal), *required* (the practice’s normative evaluator requires the signal, with the WCAG level given), and *not used* (the documentation explicitly disclaims the signal as an input); where a source supports only a narrower or negative statement, the cell carries that statement in place of a grade. An em dash means the documentation sampled for this review contains no statement either way — an absence, not an assertion of irrelevance.

Table 2. Page-level signals and the four machine readers, as documented (sampled 2026-07-09).

SIGNAL	SEARCH CRAWLER	RAG RETRIEVAL PIPELINE	ASSISTIVE TECHNOLOGY	AUTONOMOUS AGENTS
Alt text	Consumed [2]	—	Required (SC 1.1.1, Level A) [23]	—
Heading hierarchy	Order not used in ranking [1]	Consumed by structure-aware chunkers (a) [11] [12]	Consumed; predominant navigation method [16][18]	Consumed by tree-consuming systems (b) [16][19]
Semantic and landmark elements	Documented as rarely depended on [1]	HTML tag structure informs chunk boundaries (a) [11]	Required to be programmatically determinable (SC 1.3.1, Level A) [23]; exposed as roles [16][17]	Consumed by tree-consuming systems (b) [19][37]
Descriptive link text	Consumed [3]	—	Required (SC 2.4.4, Level A) [23]	—
ARIA semantics	—	—	Consumed; the specification's stated purpose [13]	Consumed (ChatGPT Atlas; tree-consuming systems) (b) [19][20]
Color contrast	—	—	Required (SC 1.4.3, Level AA) [23]	—
Keyboard operability	—	—	Required (SC 2.1.1, Level A) [23]	Playwright MCP presses keys against accessibility-tree references [19]; Anthropic's computer-use tool uses keyboard input as a pixel-level fallback (c) [21]
Focus management	—	—	Required (SC 2.4.3, Level A) [23]	—
Plain language	—	—	Required only at Level AAA (SC 3.1.5) [23]	—
Schema.org markup	Consumed (understanding, rich results) [42]	No special structured data documented for AI Overviews or AI Mode [9]; not mentioned in OpenAI's crawler documentation (c) [8]	—	—
Quotations, statistics, and citations	—	Measured effect on answer inclusion [24]	—	—
Page speed	Consumed (Core Web Vitals) [4][5]	—	—	—

(a) Documented in general-purpose RAG tooling (Pinecone [11], LangChain [12]), not in any production answer engine's own pipeline documentation; Google states that beyond standard indexing and snippet

eligibility "There are no additional technical requirements" [9]. (b) Applies to the tree- and ARIA-consuming systems named in chapter 5; the screenshot-based systems named there are documented as perceiving rendered pixels, and their documentation makes no statement about individual markup signals. ChatGPT Atlas's deprecation — announced 2026-07-09, scheduled effective 2026-08-09 — is dated in chapter 5 [45]. (c) A populated cell that does not count toward the five-signal intersection of chapter 3: the keyboard-operability agent cell documents an action capability, not consumption of the signal as an input, and the schema.org answer-engine cell records a documented negative.

Figure 1 groups the twelve signals into the five regions the four readers carve out of Table 2, sized by count rather than drawn as a symmetric Venn.

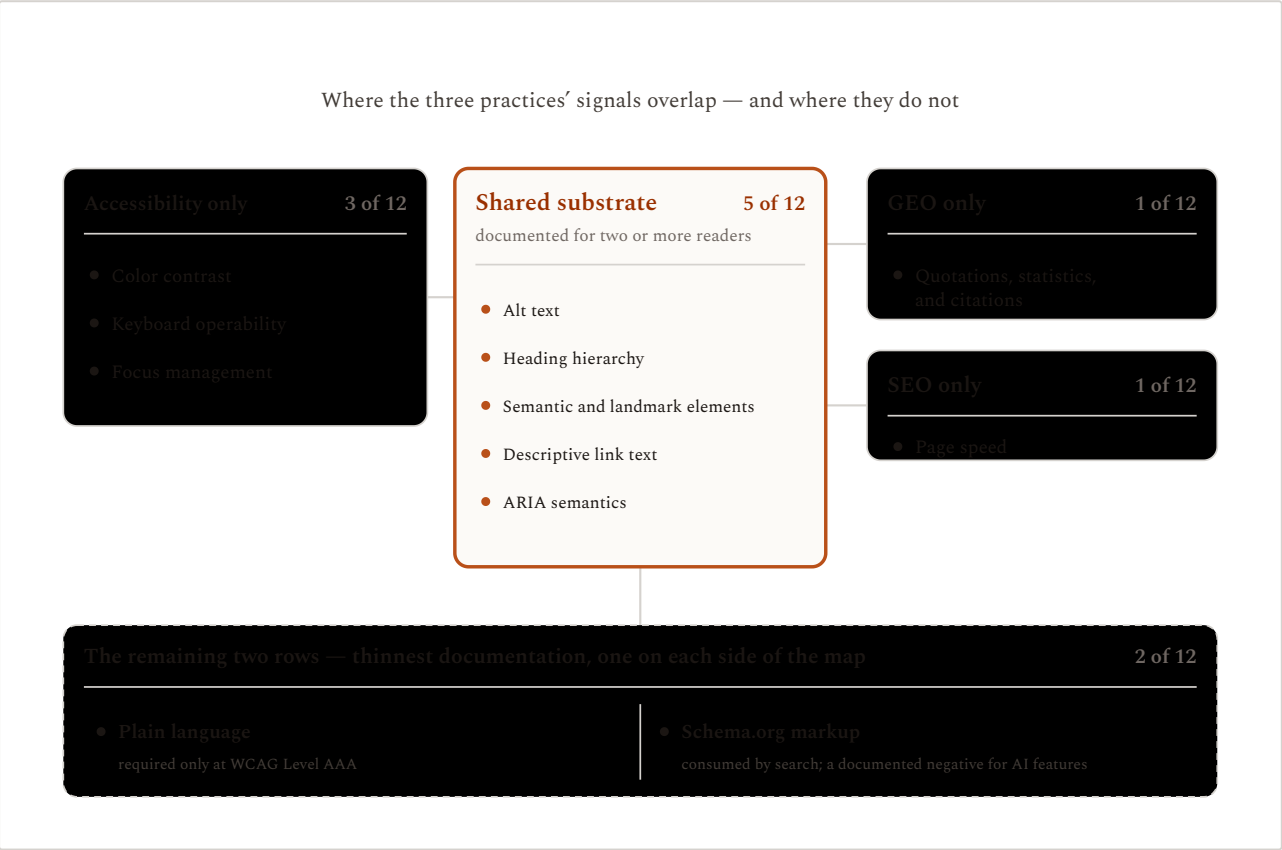

Figure 1. The twelve page-level signals of Table 2, grouped by the practice whose documentation consumes or requires each. Five sit in the shared substrate (a documented consumer or requirement in two or more reader columns; a documented negative or an action-only agent capability does not count — Table 2, note c); three are accessibility-only; one is GEO-only; one is SEO-only; and the remaining two — plain language and schema.org markup — are the thinnest-documented rows, one on each side of the map. Each region is labeled with its count of the twelve signals; the layout is sized by count, not drawn as a symmetric Venn.

Barkhausen AI · BA-W-2026-03

The intersection

Five of the twelve rows carry documented consumers or requirements in two or more reader columns, and they form the substantive core of any claim that the practices overlap. The counting rule is strict: a row joins the five only where two or more reader columns record a consumed input or a normative requirement — a documented negative does not count, and neither does an agent's action capability. Two rows carry a second populated column that does not qualify: keyboard operability, whose agent cell describes an action channel rather than consumption of the signal, and schema.org markup, whose answer-engine cell records a documented negative (Table 2, note c).

Alt text is the cleanest case, because a single vendor page documents both sides: Google states that alt text "improves accessibility for people who can't see images on web pages" *and* that "Google uses alt text along with computer vision algorithms and the contents of the page to understand the subject matter of the image" [2] — the same string serving the accessibility evaluator (SC 1.1.1 [23]) and the search reader. The empirical baseline is poor on both counts: in the 2026 WebAIM Million (n = 1,000,000 home pages, WAVE engine, February 2026), 16.2% of all home page images — 10.8 per page on average, across 66.6 million images sampled — had missing alternative text, and missing alt text appeared on 53.1% of pages (down from 68% in 2019, per the report's eight-year data tables) [29]. The Web Almanac's separate measurement (Lighthouse/axe-core, 16,213,084 sites including secondary pages, July 2025 crawl) reports 69% of images passing its alt-text audit, and adds two quality findings: "approximately 8.5% of image alt texts end with common file extensions like .jpg or .png" and "about 50% of images still have either empty alt attributes or text shorter than 10 characters" [30][31]. The WebAIM and Almanac figures come from different tools, different detection algorithms, and different samples; they are reported here side by side and must not be combined or compared as one series.

Heading hierarchy is the row where the three practices pull in three directions at once. For assistive technology it is the primary interface: 71.6% of surveyed screen reader users (n = 1,539, December 2023–January 2024, single-select stated preference) reach first for heading navigation on long pages [18], and headings become machine-readable "heading, level N" nodes through the draft W3C mapping [16]. For the search crawler, Google documents that semantic order "doesn't matter" for ranking while calling it "fantastic for screen readers" [1]. For the RAG pipeline, general-purpose chunking tooling explicitly follows heading structure [11][12] — though no answer-engine vendor documents doing the same. Empirically, skipped heading levels were present on 41.8% of the WebAIM Million's 1,000,000 home pages (February 2026, up from 39% in 2025), and 7.5% of pages had no headings at all [29]; the Web Almanac separately reports 59% of its roughly 15.4 million mobile-tested sites passing the Lighthouse heading-order audit (July 2025 crawl), a different test on a different population that cannot be averaged with the WebAIM figure [30][31].

Semantic and landmark elements show the same pattern: normatively required to be programmatically determinable for assistive technology (SC 1.3.1 [23]), defined by WHATWG by their structural meaning [17], consumed by tree-based agents as roles [19], used by structure-aware chunkers at the level of HTML tags generally [11] — and explicitly discounted by Google for ranking, since "Google Search can rarely depend on semantic meanings hidden in the HTML specification" [1]. Adoption of the `main` landmark (native element or ARIA role combined) stood at 47% of the Almanac's roughly 12.2 million desktop-tested sites in the July 2025 crawl, up from 35% in 2021; the native element alone stood at about 41% [30].

Descriptive link text is documented on the search side [3] and required on the accessibility side at Level A [23]. The two tests are structurally similar to the point of interchangeability — Google: does the anchor text "make sense by itself" out of context [3]; WCAG: can the link's purpose "be determined from the link text alone or from the link text together with its programmatically determined link context" [23]. The similarity is this paper's observation: Google's page never mentions accessibility or WCAG.

ARIA semantics are consumed by assistive technology by definition [13] and, per vendor statements, by some named agent systems including ChatGPT Atlas [20] — but no search or RAG documentation in the sampled set mentions them.

The accessibility-only region

Three rows — color contrast, keyboard operability, and focus management — carry an accessibility requirement at Level A or AA, and none has a documented path to search or answer-engine visibility.

Color contrast is required at WCAG 2 AA thresholds and is the most commonly detected failure on the web — low-contrast text was found on 83.9% of the WebAIM Million's 1,000,000 home pages in February 2026, the highest rate since 2021, tying 2022 [29] — yet the review found no documented mechanism by which contrast reaches a search ranking system, a retrieval pipeline, or a tree-consuming agent's input. Keyboard operability and focus management are canonical accessibility techniques, and both now carry normative backing on the accessibility side — keyboard operability required at Level A (SC 2.1.1) and sequential focus order at Level A (SC 2.4.3) [23] — but no search or retrieval documentation in the sampled set connects either to its reader. On the agent side, the two named tools split: Playwright MCP presses keys against accessibility-tree references [19], while Anthropic's computer-use tool uses keyboard input as a pixel-level fallback [21] — a distinction of action channel and perception, not of generative-engine visibility. For all three rows the finding is the same: **no documented mechanism connects them to generative-engine visibility.** A quantified visibility claim built on contrast ratios or focus order rests, on the current public record, on no documented mechanism at all.

The GEO-only region

One row runs the other way. Adding quotations, statistics, and citations — the three techniques the KDD 2024 study found most effective, at 30–40% relative improvement on its position-adjusted word-count metric (n = 10,000 queries, simulated engine) and 22–37% on its deployed Perplexity.ai check (n = 200) [24] — appears in no WCAG success criterion and in none of the sampled search documentation. The most effective documented GEO techniques are content edits, not markup edits. This asymmetry cuts against both directions of sloppy equivalence: accessibility work does not exhaust GEO, and GEO's measured levers are not accessibility techniques.

The remaining rows

The last two rows carry the thinnest documentation, one on each side of the map. Plain language has a single documented requirement, and it sits at the strictest conformance level: WCAG SC 3.1.5 Reading Level asks that where text requires reading ability beyond the lower secondary education level, a simpler alternative be available — but only at Level AAA [23], a level the specification advises against adopting as a general policy, and one that no search, retrieval, or agent documentation in the sampled set addresses. Schema.org markup is the mirror case. The search crawler consumes it: Google states it "uses structured data that it finds on the web to understand the content of the page" and can use that markup to display rich results [42]. The answer-engine side, however, is a documented negative rather than a consumer — Google's own AI-features documentation states there is "no special schema.org structured data that you need to add" to appear in AI Overviews or AI Mode [9], and OpenAI's crawler documentation, sampled 2026-07-09, does not mention structured data or schema.org among the signals its bots use [8]. That is a documented negative for one vendor and a verified absence for another, each scoped to that vendor's own AI features and crawler documentation, not a general finding that structured data is unused by answer engines. The overall state of the substrate, finally, has a number: 95.9% of the WebAIM Million's 1,000,000 home pages had at least one detected WCAG 2 failure in February 2026, up from 94.8% in 2025 [29] — subject to the report's own caveat, quoted in full because every WebAIM figure in this paper inherits it: "All automated tools, including WAVE, have limitations—not all conformance failures can be automatically detected. Absence of detected errors does not indicate that a page is accessible or conformant" [29].

4 The state of the evidence

Chapter 3 mapped what the documentation says each reader consumes. This chapter weighs the evidence for the causal story that is usually built on top of that map: that accessibility work improves visibility in search and, especially, in AI answers. The evidence divides into three parts — what the search engine has said on the record, what the mechanism chain toward answer engines actually supports, and what the quantified claims now circulating would need in order to hold.

What the engine has said

The most-cited statement on accessibility and ranking comes from a Google Search Central SEO office-hours session held on March 25, 2022, hosted by John Mueller. No official Google transcript of that session exists — Google's own office-hours transcript archive begins in November 2022 — so the wording rests on Search Engine Journal's contemporaneous transcription, published four days after the session [32]; the session's recording on Google Search Central's own YouTube channel confirms the date and the submitted question [33]. Asked about accessibility-related link presentation, Mueller answered:

> *No, not really. So I think accessibility is something that is important for a website because, if you drive your users away with a website that they can't use, then they're not going to recommend it to other people. But it's not something that we would pick up and use as a direct ranking factor when it comes to search. Maybe that will change over time. [32]*

And, later in the same answer:

> *That's something where I could imagine the work we've done around Core Web Vitals, and the page experience ranking factor, where it could be that at one point we can quantify accessibility a little bit more, and maybe at that point we can use that when it comes to ranking, but that's something where at least at the moment we don't have any specific plans in that direction. [32]*

One attribution correction matters, because the misquote is widespread: the phrase "difficult to quantify," often placed in Mueller's mouth in secondary write-ups, is Search Engine Journal's own editorial framing — it appears in the article's description, not in the transcribed answer [32]. Mueller's actual words run the other direction grammatically: he speculates about a future in which Google "can quantify accessibility a little bit more".

Read together with the documentation in chapters 1 and 3, the position is coherent rather than dismissive. Where an accessibility-relevant signal is individually machine-readable at scale, Google documents consuming it (alt text [2], descriptive anchor text [3]). Where the signal is an aggregate property Google has not operationalized, it is not a ranking input, and Mueller's stated reason is the absence of exactly the kind of measurement apparatus that Core Web Vitals supplies for speed — three named metrics with numeric thresholds [5]. *Accessibility is not a ranking factor* and *several accessibility techniques are consumed by search systems* are both documented, and they are statements about different levels of aggregation.

The mechanism chain toward answer engines

For generative engines, the causal story circulating in practitioner material runs, made explicit: page structure → parse quality → chunk boundaries → embedding and retrieval quality → answer inclusion. The review behind this paper assessed each link against primary documentation. Table 3 summarizes the result.

Table 3. The accessibility-to-GEO mechanism chain, link by link (documentation as sampled 2026-07-09).

LINK	CLAIM	DOCUMENTARY STATUS
1	Page structure improves parse quality	**Verified absence.** No source found — vendor or tooling — documents a parse-quality stage distinct from chunking.
2	Parsed structure guides chunk boundaries	Supported by general-purpose RAG tooling [11][12]; not documented by any answer-engine vendor for its own pipeline.
3	Chunk quality improves retrieval	Supported with a measured figure, for chunk *content* rather than boundary placement [10].
4	Retrieval quality drives answer inclusion	**Verified absence for production engines.** An inference bridging two literatures never tested together [6][24].
5	Vendor pipelines implement links 2–4	Undocumented; vendors document crawl and eligibility endpoints only [7][8][9].

Link 1 is the chain's first verified absence. No source in the sampled record — not the answer-engine vendors, not the RAG tooling literature — documents an intermediate "parse quality" stage, meaning a measured fidelity of HTML-to-text extraction distinct from the chunking step itself. The tooling sources jump directly from *the document has headings and tags* to *the chunker uses them* [11][12]. The review behind this paper searched specifically for a source filling this link and found none. This is reported as a finding about the state of public documentation, not as a gap in search effort.

Link 2 is the best-documented link, and its support is real but comes entirely from general-purpose tooling: Pinecone's guide ties chunk boundaries to Markdown and HTML structure explicitly [11], and LangChain ships a heading-based splitter as a production feature [12]. No answer-engine vendor states that its pipeline does the same.

Link 3 has the chain's one hard number: Anthropic reports that "Combining Contextual Embeddings and Contextual BM25 reduced the top-20-chunk retrieval failure rate by 49% (5.7% → 2.9%)" in its own 2024 evaluation across four knowledge-domain corpora — a vendor self-report that publishes failure rates rather than a query count [10]. The caveat is precise: the manipulated variable was LLM-generated context *added to* each chunk, not the placement of the chunk boundary. No source in the sampled record isolates boundary placement — a boundary at a heading versus a boundary mid-paragraph — as the sole variable with a measured retrieval outcome.

Link 4 is the chain's most consequential verified absence. The one controlled study of answer inclusion — the KDD 2024 GEO study — does not test retrieval at all: its design supplies the source documents directly to the generator and measures how content-level edits change the synthesized answer [24]. The foundational RAG paper establishes that retrieval feeds generation in principle, on 2020-era question-answering benchmarks [6]. No public source measures the effect of retrieval or embedding quality on answer inclusion for any deployed generative engine. The link is an inference produced by laying two literatures side by side — literatures that have never been tested together. The closest published work varies chunking strategy but measures retrieval-quality metrics, not whether a source appears in a generated answer.

Link 5 is disclaimed by the clearest vendor statement available: Google's documentation for AI Overviews and AI Mode names standard indexing and snippet eligibility as the requirements and states, "There are no additional technical requirements" [9]. Perplexity and OpenAI likewise document crawler access and eligibility, and nothing further inward [7][8].

The honest summary: the middle of the chain is a documented pattern in general RAG engineering; both of its load-bearing end links — the one that would connect markup quality to the pipeline's input, and the one that would connect the pipeline's internals to the answer — are verified documentary absences as of 2026-07-09. The chain is plausible. It is not measured, and its plausibility rests on documentation about systems other than the answer engines it is invoked to explain.

Circulating quantified claims, audited as a class

Quantified claims now circulate in vendor marketing and trade coverage taking the general form *accessible websites are N% more likely to be cited by AI assistants* or *accessibility improves AI visibility by N%*. This section audits the claim class against the evidence standards this publication applies to visibility statistics generally (see BA-W-2026-01); no individual vendor is named and no circulating number is repeated, because repeating an unsupported number extends its citation life regardless of framing. A claim of this class would need to clear five checks:

1. **A sample size, a time window, and an engine version.** A percentage without its n, window, and engine identity is not a measurement (see BA-W-2026-01). Answer engines drift; a number without a date describes nothing durable.
2. **A documented mechanism.** Per Table 3, the mechanism chain's end links are verified absences. A claim implying *structure in, citation out* asserts links 1 and 4 as if documented; they are not.
3. **Comparable inputs.** Accessibility baselines come from censuses with different tools and samples — the WebAIM Million (WAVE, 1,000,000 home pages, February 2026) [29] and the Web Almanac (Lighthouse/axe-core, 16,213,084 sites including secondary pages, July 2025) [30][31]. These are not interchangeable series, and a claim that splices them — or measures "accessibility" with one and implies conclusions about the other's population — inherits an artifact, not a signal.
4. **Correlational discipline.** Accessible and inaccessible sites differ systematically in ways that also plausibly affect visibility — size, complexity, editorial resources. The WebAIM Million models the needed discipline on an adjacent finding: home pages with ARIA present averaged 59.1 detected errors against 42 without (n = 1,000,000 home pages, February 2026), and the report immediately disclaims the causal reading: "This does not necessarily mean that ARIA introduced these errors (these pages were also more complex), but pages typically had significantly more errors when more ARIA was present" [29]. An association between an accessibility score and a citation rate, without design that removes confounding, supports *is associated with* and nothing stronger.
5. **Perishability disclosure.** These results describe the engines as sampled during July 2026; engines change without notice, and results should be assumed perishable. A claim published without its sampling window fails this check by construction.

As of 2026-07-09, the review behind this paper found no public study of the accessibility-to-AI-visibility effect that clears these checks. That does not mean the effect is absent — chapter 6 states it as a testable proposition — but it means the quantified claims now circulating are, in the strict sense, unsupported. The asymmetry deserves stating plainly: the effect is unmeasured, which is a different finding from disproven, and both are different from demonstrated.

5 The accessibility tree and autonomous agents

The strongest documented connection between accessibility work and any machine reader other than assistive technology itself does not run through search or through RAG pipelines. It runs through

autonomous agents — and it runs through only some of them. This chapter states what the vendors' own documentation supports, product by product, because the landscape does not support a general claim.

The object at the center is the accessibility tree of chapter 1: the browser-built "Tree of accessible objects that represents the structure of the user interface" [13], derived from the DOM and modifiable via ARIA attributes [14], surfaced in developer tooling as "a subset of the DOM tree" containing what is "relevant and useful for displaying the page's contents in a screen reader" [15]. That this structure would be consumed by software other than screen readers is not new: Chromium's documentation notes that "because accessibility APIs provide a convenient and universal way to explore and control applications, they're often used for automated testing scripts, and UI automation software like password managers" [14] — a statement about conventional automation that predates the current agent systems and names none of them; its relevance to LLM-driven agents is an analogy, made explicit here rather than smuggled.

Systems documented as consuming the tree or ARIA semantics

Playwright MCP (Microsoft) is the most explicit. Its documentation states that the server "enables LLMs to interact with web pages through structured accessibility snapshots, bypassing the need for screenshots or visually-tuned models", and lists as design properties: "Fast and lightweight. Uses Playwright's accessibility tree, not pixel-based input" and "LLM-friendly. No vision models needed, operates purely on structured data" [19]. Even here the position is a default, not an absolute: the same documentation offers an opt-in vision capability [19].

Playwright CLI (Microsoft) — a second, separately shipped product "designed for coding agents" — documents the same choice: "The snapshot file contains the accessibility tree with element refs for the next command" [37]. Both products build on a primitive Playwright documents for testing: "aria snapshots provide a YAML representation of the accessibility tree of a page", detailing "roles, attributes, values, and text content" [38]. Playwright's testing guidance points the same direction for human test authors, recommending role-based locators "as it is the closest way to how users and assistive technology perceive the page" [39].

Stagehand (Browserbase), in its DOM mode, gives the agent an `ariaTree` tool documented as "Get accessibility tree of the page" [36].

ChatGPT Atlas (OpenAI) supplies the strongest vendor statement in the sampled record, in a developer-facing FAQ:

> *Making your website more accessible helps ChatGPT Agent in Atlas understand it better. ChatGPT Atlas uses ARIA tags—the same labels and roles that support screen readers—to interpret page structure and interactive elements. To improve compatibility, follow WAI-ARIA best practices by adding descriptive roles, labels, and states to interactive elements like buttons, menus, and forms. This helps ChatGPT recognize what each element does and interact with your site more accurately. [20]*

This is a vendor telling site owners, in its own words, that accessibility markup improves how its agent understands their site. Its scope must be kept exact: it describes one OpenAI product, and it is documented inconsistently with the same vendor's Computer Use API lineage, described next.

The product's status must be dated as exactly as its words. On 2026-07-09 — this paper's access date — OpenAI announced it is deprecating the standalone Atlas browser: "Atlas is scheduled to stop working on August 9, 2026", with the company "moving browser-based agentic capabilities into ChatGPT and Codex" and directing users to a ChatGPT desktop app and a ChatGPT Chrome extension or sidebar [45]. The deprecation notice makes no statement about whether the ARIA-based page interpretation quoted above

carries into those successor surfaces [45]; the FAQ itself, re-archived on 2026-07-10, still carried the guidance unchanged [20]. The quotation above therefore stands as a dated vendor position — the strongest in the sampled record — attached to a product with an announced end date.

Systems documented as screenshot-based

Anthropic's computer-use tool describes "screenshot capabilities and mouse/keyboard control for autonomous desktop interaction", with an action set built on capturing the display and clicking at coordinates [21]. The tool's documentation, as published at 2026-07-09, does not describe consuming the accessibility tree or ARIA semantics.

OpenAI's built-in Computer Use tool loop is documented as screenshot-driven: "The model looks at the current UI through a screenshot, returns actions such as clicks, typing, or scrolling, and your harness executes those actions in a browser or computer environment" [34]. The scope of this statement is the built-in loop specifically. The same guide documents two further options — custom harnesses including Playwright- and MCP-based ones, and code-execution workflows for which it recommends DOM-based approaches [34] — so the guide as a whole documents multiple perception paths, not a screenshot-only position.

Google's Gemini Computer Use tool shipped as a built-in tool of Gemini 3.5 Flash on 2026-06-24, superseding the standalone Gemini 2.5 computer-use model, which the documentation sampled 2026-07-09 lists as a legacy preview [44][22]. That documentation describes a loop whose request contains "a screenshot of the current screen" and whose output is "specific UI actions like mouse clicks and keyboard inputs" [22]. As published at 2026-07-09, it does not describe accessibility-tree or DOM consumption as a perception source for either model generation.

Amazon's Nova Act defines its perception cycle in its User Guide glossary as an "observation loop: The cycle where Nova Act captures and processes the current state of the browser or environment, including screenshot analysis, and context extraction, before determining the next action" [35]. A developer-side utility for retrieving the rendered DOM exists in the SDK, but the documentation does not present it as a perception channel fed back to the model, and it is not treated as one here [43].

Stagehand's CUA mode, in the same framework whose DOM mode consumes the tree, is documented as vision- and coordinate-based [36] — making Stagehand the cleanest single-framework illustration that the two perception channels are a design choice, not a settled convergence.

What may and may not be concluded

The split holds across six organizations: on the sampled documentation, four named systems consume the accessibility tree or ARIA semantics (Playwright MCP, Playwright CLI, Stagehand's DOM mode, ChatGPT Atlas) and five are documented as screenshot-based in their core perception loop (Anthropic's computer-use tool, OpenAI's built-in Computer Use loop, the Gemini Computer Use tool, Nova Act, Stagehand's CUA mode). Two vendors document both channels under one roof — OpenAI across its Atlas FAQ and its Computer Use guide, and Browserbase within Stagehand's mode table. The nine names also sit at different architectural layers — protocol servers and command-line tools, one framework's two modes, a consumer browser, and model-level tools — so the split is a statement about named, documented systems, not a like-for-like comparison of peer products. The claim *the accessibility tree is how AI agents read the web* is therefore not supported as a general statement, and this paper does not make it. What is supported is narrower and still consequential: for a named set of real systems, the representation an agent

consumes derives from the same accessibility tree that assistive technology consumes, and one vendor explicitly advises accessibility markup as agent optimization [20].

Two cautions keep this finding from becoming its own overreach. First, the W3C's own authoring guidance is titled "No ARIA is better than Bad ARIA" and warns that "Incorrect ARIA misrepresents visual experiences, with potentially devastating effects on their corresponding non-visual experiences" [40] — guidance, not a normative requirement, but pointed: the lesson of the Atlas statement is correct ARIA, not more ARIA. Second, the empirical record shows how easily "more ARIA" goes wrong: in the 2026 WebAIM Million, home pages with ARIA present averaged 59.1 detected errors against 42 for pages without it (n = 1,000,000 home pages, February 2026) — an association the source itself declines to read causally [29].

Within this publication's framework, this material sits just below the top of the Barkhausen Ladder (BA-C-1). BL-8's marker is a machine-actionable interface an agent can invoke without parsing a web page at all; the agents documented here are doing precisely the page-parsing that BL-8 endpoints exist to make unnecessary. The documentation reviewed in this chapter therefore describes how agents cope on the web as it is — where BL-8 interfaces are rare — not the evidence base for BL-8 itself. A companion note, reading the accessibility tree, collects the platform documentation on what the tree is and what software consumes it.

6 Testable propositions

The verified absences of chapter 4 are not just holes in the record; they define experiments. Each proposition below is stated so that a result in either direction would be informative, and each names the public baseline that makes the study population easy to find. None requires proprietary access to an engine.

P1 — Heading hierarchy and chunk boundaries. Pages with skipped heading levels, when processed by structure-aware chunkers of the kind documented in the RAG tooling literature [11][12], produce chunk boundaries that split semantically coherent units at a higher rate than pages with conforming heading structure. The treatment population is abundant: skipped heading levels were present on 41.8% of the 2026 WebAIM Million's 1,000,000 home pages (WAVE, February 2026) [29]. The study is a corpus experiment — chunk both populations with published splitters, have blinded raters (human or model) judge boundary coherence — and it directly tests link 2 of Table 3 as an accessibility-relevant effect rather than assuming it.

P2 — Alt text coverage and answer citation. For queries whose best available sources carry substantial image-borne information, sources with complete alt-text coverage are cited in generated answers at a higher rate than content-equivalent sources without it. The baselines: 16.2% of images across 66.6 million sampled on 1,000,000 home pages lacked alternative text (WebAIM, February 2026) [29]; separately, 69% of images passed the Web Almanac's alt-text audit (Lighthouse/axe-core, 16,213,084 sites, July 2025 crawl) [30] [31] — two non-comparable measurements that jointly establish the variance needed for a natural experiment. A controlled version follows the KDD study's design [24] with alt coverage, rather than quotation addition, as the manipulated variable.

P3 — Landmark usage and extraction fidelity. Pages using landmark elements (`main`, `nav`, `article`) yield higher main-content extraction fidelity — less navigation and boilerplate text contaminating the extracted body — than pages expressing the same layout with undifferentiated containers. The `main` landmark (native element or ARIA role) stood at 47% adoption across roughly 12.2 million desktop-tested sites in

July 2025 [30], so both arms of the comparison exist at scale. Extraction fidelity is directly measurable against human-annotated ground truth.

P4 — The parse-quality construct. Define parse quality as the fidelity with which an HTML-to-text extraction preserves a page's semantic units. The proposition is that a measure so defined varies systematically with markup semantics and predicts downstream retrieval quality; it fails if the defined measure does not vary with markup semantics, or does not predict retrieval outcomes. This proposition exists because link 1 of the mechanism chain is a verified documentary absence: no source in the sampled record defines or measures such a stage. Establishing whether the construct behaves as the practitioner narrative assumes is the experiment.

P5 — Retrieval quality and answer inclusion. Manipulating a source's retrievability — its chunking, its embedding quality, its retrieval rank — while holding its content constant changes the rate at which a deployed generative engine includes or cites that source. This is the untested bridge between the RAG architecture literature [6] and the GEO content-effects literature [24], stated as an experiment. The KDD study's deployed-check design (n = 200 against a production engine) [24] shows the form such a test can take; what has never been run is that design with retrieval-side rather than content-side variables.

No public study, as of the review behind this paper (2026-07-09), measures any of these five propositions. That is the paper's central empirical finding restated as an agenda.

None of these studies requires new instruments. Population-scale accessibility measurement exists and publishes its methodology: an annual million-page census with a stated tool and window [29], a sixteen-million-site crawl with a stated month and toolchain [31], a user survey with a stated n and fieldwork period [18]. Visibility in answer engines is measurable by repeated sampling with declared precision, windows, and engine versions — the requirements are specified in BA-C-2 and BA-C-3 and argued in BA-W-2026-01. Every proposition above is a join of those two existing measurement traditions over the same population of pages. The methods are in print; only the join is missing.

7 Limitations

The mechanism claims are inference, not measurement. Table 3's supported links rest on documentation of general-purpose RAG tooling, not on any production answer engine; no claim in this paper about how Google, OpenAI, or Perplexity process pages between crawl and answer is vendor-confirmed, because no vendor confirms any. The verified absences are statements about public documentation as sampled, not about vendor internals: that a vendor's documentation, as published at 2026-07-09, does not describe a mechanism is evidence about the public record, never evidence that the mechanism does not privately exist.

Engine and agent documentation is perishable. Every vendor statement cited here carries an accessed date of 2026-07-09 and an archived snapshot, because these pages change without notice; the agent documentation is the fastest-moving material in the paper's evidence base. These results describe the engines as sampled during July 2026; engines change without notice, and results should be assumed perishable. Chapter 5's split-landscape finding in particular is a photograph of product documentation, not a stable taxonomy. Two supersessions landed inside the sampling window itself and are dated in the text: Google folded computer use into Gemini 3.5 Flash on 2026-06-24 [44], and OpenAI announced the Atlas browser's deprecation on 2026-07-09 — the access date — effective 2026-08-09 [45].

The accessibility censuses measure less than their headline numbers suggest. The WebAIM Million evaluates 1,000,000 home pages — not full sites — with the WAVE engine, which detects only the automatable subset of WCAG failures; its own caveat, quoted in chapter 3, applies to every WebAIM figure

cited here [29]. The Web Almanac evaluates a different population (16,213,084 sites, including secondary pages, July 2025 crawl) with different tools (Lighthouse/axe-core) [30][31]. The two are not comparable series, and this paper cites each number with its own n, tool, and window for that reason. The Screen Reader User Survey is a self-selected sample (n = 1,539) answering stated-preference questions, not a behavioral log, and Survey #10 (fieldwork December 2023–January 2024) is the latest published edition as of this paper's date [18].

The GEO effect sizes do not generalize freely. The KDD 2024 percentages are defined on the study's own position-adjusted word-count metric, produced mostly against a simulated GPT-3.5 engine, with a deployed check limited to 200 samples on one engine [24]. They establish that content-level GEO effects exist and are measurable; they do not establish magnitudes for today's production engines.

The legal dates carry pending litigation and unfinished standardization. The ADA Title II compliance dates cited (26 April 2027 and 26 April 2028) come from an interim final rule whose comment period closed 22 June 2026 and which an active lawsuit seeks to vacate [27][28]; they were controlling as of 2026-07-09 and may not remain so. The EAA's application date carries the directive's own transitional carve-outs [25], and no official EU source had confirmed an EAA-specific harmonized standard as of the same date. Nothing in this paper is legal advice.

The signal matrix reflects sampled documentation, not the world. Table 2's em dashes mean the documentation sampled for this review contains no statement either way — an absence, not an assertion of irrelevance — and each cell reflects what its source documented as of the access date. Where a cell records a documented negative instead, as schema.org markup does for Google's AI Overviews and AI Mode [9], that negative is scoped to that vendor's own AI features; OpenAI's crawler documentation is a verified silence on structured data, not a disclaimer [8]. The matrix is a map of what is documented as of 2026-07-09, and both its absences and its documented negatives should be re-checked before being cited at any later date.

REFERENCES

1. Google Search Central (Google for Developers documentation). *SEO Starter Guide: The Basics — Number and order of headings* (2025). https://developers.google.com/search/docs/fundamentals/seo-starter-guide Accessed 2026-07-09. [archived]
2. Google Search Central (Google for Developers documentation). *Image SEO Best Practices* (2026). https://developers.google.com/search/docs/appearance/google-images Accessed 2026-07-09. [archived]
3. Google Search Central (Google for Developers documentation). *Link best practices for Google* (2025). https://developers.google.com/search/docs/crawling-indexing/links-crawlable Accessed 2026-07-09. [archived]
4. Google Search Central (Google for Developers documentation). *Understanding page experience in Google Search results* (2025). https://developers.google.com/search/docs/appearance/page-experience Accessed 2026-07-09. [archived]
5. Google Search Central (Google for Developers documentation). *Understanding Core Web Vitals and Google search results* (2025). https://developers.google.com/search/docs/appearance/core-web-vitals Accessed 2026-07-09. [archived]
6. Lewis, Perez, Piktus, Petroni, Karpukhin, Goyal, Küttler, Lewis, Yih, Rocktäschel, Riedel & Kiela (NeurIPS 2020; arXiv:2005.11401). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks* (2020). https://arxiv.org/abs/2005.11401 Accessed 2026-07-09. [archived]
7. Perplexity (official documentation). *Perplexity Crawlers* (2026). https://docs.perplexity.ai/docs/resources/perplexity-crawlers Accessed 2026-07-09. [archived]
8. OpenAI (developer documentation). *Overview of OpenAI Crawlers* (2026). https://developers.openai.com/api/docs/bots Accessed 2026-07-09. [archived]
9. Google Search Central (Google for Developers documentation). *AI features and your website* (2025). https://developers.google.com/search/docs/appearance/ai-features Accessed 2026-07-09. [archived]

10. Anthropic (engineering publication). *Introducing Contextual Retrieval* (2024). https://www.anthropic.com/engineering/contextual-retrieval Accessed 2026-07-09. [archived]
11. Schwaber-Cohen & Patel, Pinecone. *Chunking Strategies for LLM Applications* (2025). https://www.pinecone.io/learn/chunking-strategies/ Accessed 2026-07-09. [archived]
12. LangChain (documentation). *Split markdown (MarkdownHeaderTextSplitter)* (2026). https://docs.langchain.com/oss/python/integrations/splitters/markdown_header_metadata_splitter Accessed 2026-07-09. [archived]
13. W3C (World Wide Web Consortium). *Accessible Rich Internet Applications (WAI-ARIA) 1.2 — W3C Recommendation* (2023). https://www.w3.org/TR/wai-aria-1.2/ Accessed 2026-07-09. [archived]
14. The Chromium Project. *Accessibility Overview (docs/accessibility/overview.md) — undated living document* (2026). https://chromium.googlesource.com/chromium/src/+/main/docs/accessibility/overview.md Accessed 2026-07-09. [archived]
15. Chrome for Developers (Google). *Accessibility features reference | Chrome DevTools* (2026). https://developer.chrome.com/docs/devtools/accessibility/reference Accessed 2026-07-09. [archived]
16. W3C (World Wide Web Consortium). *HTML Accessibility API Mappings 1.0 — W3C Working Draft* (2026). https://www.w3.org/TR/html-aam-1.0/ Accessed 2026-07-09. [archived]
17. WHATWG. *HTML (Living Standard) — the nav, article, and section elements* (2026). https://html.spec.whatwg.org/multipage/sections.html Accessed 2026-07-09. [archived]
18. WebAIM (survey series conducted by the Institute for Disability Research, Policy & Practice, Utah State University). *Screen Reader User Survey #10 Results* (2024). https://webaim.org/projects/screenreadersurvey10/ Accessed 2026-07-09. [archived]
19. Microsoft. *playwright-mcp README (Playwright MCP server)* (2026). https://github.com/microsoft/playwright-mcp Accessed 2026-07-09. [archived]
20. OpenAI (Help Center). *Publishers and Developers – FAQ* (2026). https://help.openai.com/en/articles/12627856-publishers-and-developers-faq Accessed 2026-07-09. [archived]
21. Anthropic (Claude Platform documentation). *Computer use tool* (2026). https://platform.claude.com/docs/en/agents-and-tools/tool-use/computer-use-tool Accessed 2026-07-09. [archived]
22. Google (Gemini API documentation). *Computer Use* (2026). https://ai.google.dev/gemini-api/docs/computer-use Accessed 2026-07-09. [archived]
23. W3C (World Wide Web Consortium). *Web Content Accessibility Guidelines (WCAG) 2.2 — W3C Recommendation* (2024). https://www.w3.org/TR/WCAG22/ Accessed 2026-07-09. [archived]
24. Aggarwal, Murahari, Rajpurohit, Kalyan, Narasimhan & Deshpande (KDD 2024; arXiv:2311.09735). *GEO: Generative Engine Optimization* (2024). https://arxiv.org/abs/2311.09735 Accessed 2026-07-09. [archived]
25. European Parliament and Council of the European Union (EUR-Lex). *Directive (EU) 2019/882 on the accessibility requirements for products and services (European Accessibility Act)* (2019). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019L0882 Accessed 2026-07-09. [archived]
26. US Department of Justice (Federal Register, 89 FR 31320). *Nondiscrimination on the Basis of Disability; Accessibility of Web Information and Services of State and Local Government Entities (Final Rule)* (2024). https://www.govinfo.gov/content/pkg/FR-2024-04-24/html/2024-07758.htm Accessed 2026-07-09. [archived]
27. US Department of Justice (Federal Register, 91 FR 20902). *Extension of Compliance Dates for Nondiscrimination on the Basis of Disability; Accessibility of Web Information and Services of State and Local Government Entities (Interim Final Rule)* (2026). https://www.govinfo.gov/content/pkg/FR-2026-04-20/html/2026-07663.htm Accessed 2026-07-09. [archived]
28. National Federation of the Blind (plaintiff); Complaint, D. Md. No. 1:26-cv-02007-RDB, filed 2026-05-21 (hosted by Democracy Forward Foundation, plaintiff's co-counsel). *National Federation of the Blind v. U.S. Department of Justice and U.S. Department of Health and Human Services — Complaint* (2026). https://democracyforward.org/wp-content/uploads/2026/05/NFB-Complaint-AS-FILED.pdf Accessed 2026-07-10. [archived]
29. WebAIM. *The WebAIM Million: The 2026 report on the accessibility of the top 1,000,000 home pages* (2026). https://webaim.org/projects/million/ Accessed 2026-07-09. [archived]
30. HTTP Archive. *Accessibility | 2025 | The Web Almanac (chapter 6)* (2026). https://almanac.httparchive.org/en/2025/accessibility Accessed 2026-07-09. [archived]
31. HTTP Archive. *Methodology — 2025 Web Almanac* (2026). https://almanac.httparchive.org/en/2025/methodology Accessed 2026-07-09. [archived]

32. Matt G. Southern, Search Engine Journal. *Google: Accessibility Not A Direct Ranking Factor* (2022). https://www.searchenginejournal.com/google-accessibility-not-a-direct-ranking-factor/443784/ Accessed 2026-07-09. [archived]

33. Google Search Central (official YouTube channel). *English Google SEO office-hours from March 25, 2022* (2022). https://www.youtube.com/watch?v=lMc456P2fLs Accessed 2026-07-09. [archived]

34. OpenAI (API documentation). *Computer use* (2026). https://developers.openai.com/api/docs/guides/tools-computer-use Accessed 2026-07-09. [archived]

35. AWS (Amazon Web Services). *Amazon Nova Act — User Guide (Glossary)* (2026). https://docs.aws.amazon.com/pdfs/nova-act/latest/userguide/nova-act-ug.pdf Accessed 2026-07-09. [archived]

36. Browserbase (Stagehand documentation). *Agent* (2026). https://docs.stagehand.dev/v3/basics/agent Accessed 2026-07-09. [archived]

37. Microsoft. *Introduction — Playwright CLI documentation* (2026). https://playwright.dev/agent-cli/introduction Accessed 2026-07-09. [archived]

38. Microsoft. *Snapshot testing — Playwright documentation* (2026). https://playwright.dev/docs/aria-snapshots Accessed 2026-07-09. [archived]

39. Microsoft. *Locators — Playwright documentation* (2026). https://playwright.dev/docs/locators Accessed 2026-07-09. [archived]

40. W3C WAI (ARIA Authoring Practices Guide). *Read Me First — No ARIA is better than Bad ARIA* (2026). https://www.w3.org/WAI/ARIA/apg/practices/read-me-first/ Accessed 2026-07-09. [archived]

41. WHATWG. *HTML Living Standard — Grouping content (the main element)* (2026). https://html.spec.whatwg.org/multipage/grouping-content.html Accessed 2026-07-09. [archived]

42. Google Search Central (Google for Developers documentation). *Introduction to structured data markup in Google Search* (2025). https://developers.google.com/search/docs/appearance/structured-data/intro-structured-data Accessed 2026-07-09. [archived]

43. Amazon (aws/nova-act GitHub repository). *aws/nova-act SDK README (rendered-DOM utility)* (2026). https://github.com/aws/nova-act Accessed 2026-07-09. [archived]

44. Google (The Keyword, official blog). *Introducing computer use in Gemini 3.5 Flash* (2026). https://blog.google/innovation-and-ai/models-and-research/gemini-models/introducing-computer-use-gemini-3-5-flash/ Accessed 2026-07-10. [archived]

45. OpenAI (Help Center). *Evolving Atlas into ChatGPT for browser-based agentic work* (2026). https://help.openai.com/en/articles/20001371 Accessed 2026-07-10. [archived]

HOW TO CITE

Barkhausen AI (2026). Machine readers of the web: how search optimization, generative engine optimization, and accessibility relate. https://barkhausen.ai/research/seo-geo-accessibility/