ANDY BLYTH
TECHNICAL DOGSBODY

Designing Content for Machine Readers

Content Modelling for AI Search, Agents and Retrieval

Andy Blyth PgDip BSc(Hons) FdSc, Technical Architect

technicaldogsbody.com

18 November 2026

Designing Content for Machine Readers: Content Modelling for AI Search, Agents and Retrieval

Author: Andy Blyth PgDip BSc(Hons) FdSc, Technical Architect
Site: technicaldogsbody.com
Published: 18 November 2026
Status: Published
Version: 1.0

Table of Contents

0. Executive Summary
1. Section 0: Why This Is the Architect's Problem
2. Section 1: How Machines Read Content
3. Section 2: Measuring Machine Readability
4. Section 3: Principles for Machine-Readable Content Models
5. Section 4: Implementation Patterns
6. Section 5: Case Study
7. Section 6: Governance, Risk and Adoption
8. Conclusion
9. Appendix A: Field Classification

Executive Summary

AI search, assistants, agents and retrieval systems read content without seeing the page it was designed for. They work from text, structure and labels, so they understand what the content model makes explicit and miss what the template only implies. When a fragment of content loses its subject, a machine reader cannot use it. Anthropic found that adding that context to each chunk before indexing cut retrieval failures by

35% with embeddings alone, and by 49% with keyword search added.[1]

The content model is the part an architect controls, and this paper tests its effect directly. The same facts were held in a generic content model and a typed one, and read through the same pipeline by three sizes of model. The test uses 19 questions on synthetic data, and the generic model reproduces the failures measured in Section 2. The typed model answered all 19 questions correctly with every model, under a strict and a permissive prompt; the generic one answered between 14 and 17.

Within each model, both content models answered at about the same speed. The gain is in which model you need: the smallest model, reading typed content, answered every question in a median of 0.27 seconds. The largest, reading the generic content, took 3.50 seconds and still missed two.[2]

This paper gives architects and technical leads on Optimizely, Contentful and similar platforms a practical framework for designing content that machines can read. It covers how to measure a content model, five principles to design it by, patterns for both platforms, a reproducible test and the governance to keep the model in shape.

The paper has seven sections:

0. **Section 0 (Why This Is the Architect's Problem)** explains why the content model is an architecture decision, made early, that outlasts every redesign.
1. **Section 1 (How Machines Read Content)** describes the four machine readers: AI search, third-party assistants, autonomous agents and retrieval systems you own.
2. **Section 2 (Measuring Machine Readability)** sets out five measures of a content model and a maturity model, applied to technicaldogsbody.com.
3. **Section 3 (Principles for Machine-Readable Content Models)** gives five principles, each with one test question.
4. **Section 4 (Implementation Patterns)** turns the principles into content types for Optimizely CMS 13 and Contentful, and compares how each platform exposes content

[1] Anthropic. "Introducing Contextual Retrieval." Anthropic, September 2024. https://www.anthropic.com/research/contextual-retrieval

[2] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. https://github.com/technicaldogsbody/the-assistant-test Chat models gemma4:e2b (digest b37049369adf…), gemma4:e4b (dc35e8d9c606…) and gemma4:12b (6114515d63c1…); embedding model embeddinggemma-2:270m (c21dd30c1e38…).

to machines.

5. **Section 5 (Case Study)** tests a generic model against a typed one with three sizes of local model, with every answer published.
6. **Section 6 (Governance, Risk and Adoption)** covers the editor experience, review and ownership, AI-generated content, the risks, and an audit for any content model.

Section 0: Why This Is the Architect's Problem

Your content now has readers that never see the page. A search engine summarises it, and an assistant quotes it. An agent acts on it. A retrieval system splits it into pieces and keeps only the pieces that match. None of them sees your design, your layout or the heading above the paragraph. They see what the content model gives them.

These readers already stand between your content and the people it was written for. In the UK, Ofcom reports that about 30% of searches now show an AI overview.[3] In the US, Pew Research Center found that when a Google search showed an AI summary, people clicked a traditional result on 8% of visits, against 15% when no summary appeared.[4] Section 1 (How Machines Read Content) describes how each reader works.

The Decision Is Made Early

The content model is usually settled in discovery, in the first weeks of a project, by architects and technical leads. It decides which types exist, what each field means, which entries reference which, and where each fact lives. Those choices then outlast the work built on them. A site can be redesigned, rebuilt in a new front-end framework or moved to a new host while the content model stays the same, because changing it means migrating every entry that uses it.

---

[3] Ofcom. "From apps to AI search: how the UK goes online in 2025." Ofcom, 10 December 2025. https://www.ofcom.org.uk/media-use-and-attitudes/online-habits/from-apps-to-ai-search-how-the-uk-goes-online-in-2025 Growth: 1,800 million ÷ 368 million = 4.9.

[4] Athena Chapekis and Anna Lieb. "Google users are less likely to click on links when an AI summary appears in the results." Pew Research Center, 22 July 2025. Based on 68,879 Google searches by 900 US adults in March 2025, 12,593 of which showed an AI summary. https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/

That makes the content model an architecture decision. It is cheap to get right at the start and expensive to change later. Section 6 (Governance, Risk and Adoption) covers the cost of remodelling an existing site.

Why It Is Not Someone Else's Problem

Content designers, editors and search specialists all improve how machines read a site, but they work inside the content model. An editor can write a clearer testimonial. An editor cannot add a field that says who the testimonial is about. A search specialist can add structured data to a page, but it can only describe the facts the model already holds. The shape of the model sits with the people who design it.

The case study in Section 5 (Case Study) shows how much that shape matters. The same facts were held in two content models and read through the same pipeline by three sizes of model. The original model answered between 14 and 17 of the 19 questions, and the remodelled one answered all 19.[5]

What This Paper Covers, and What Will Date

This paper is not a playbook for generative engine optimisation (GEO) or answer engine optimisation (AEO). It does not cover tactics for ranking in AI answers. It is not a vendor comparison either: Optimizely CMS and Contentful are the worked examples, and the principles apply to any CMS with a structured content model.

Parts of it will date. The platform features in Section 4 (Implementation Patterns) and the crawler controls in Section 6 will change. So will agent protocols such as the Model Context Protocol (MCP), and the models tested in Section 5. The measures in Section 2 (Measuring Machine Readability) and the principles in Section 3 (Principles for Machine-Readable Content Models) describe how machines read content, and should last longer than the tools that read it.

This paper gives architects and technical leads a way to measure their content model and five principles to design it by. It adds patterns for Optimizely CMS 13 and Contentful, a reproducible test of the effect, and an audit to run on the models they already have.

[5]Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

Section 1: How Machines Read Content

Your content now has four kinds of machine reader, and none of them sees the page the way a person does. Each one reads text, structure and labels. The design, the layout and the template that gave the content its meaning are mostly invisible to them.

The scale is no longer small. In the UK, Ofcom reports that about 30% of searches now show an AI overview, and 53% of adults say they see these summaries often.[6] ChatGPT had 1.8 billion UK visits in the first eight months of 2025, almost five times the 368 million in the same period of 2024.[7] Cloudflare reported in July 2026 that more than half of the traffic on the internet is now non-human.[8]

This section describes each of the four readers, how it reaches your content, and what it needs from you. Parts of it will date quickly. The products and protocols named here change month by month; the way each reader consumes content changes much more slowly, and that is what the rest of the paper builds on.

1.1 The Four Readers

READER	HOW IT REACHES YOUR CONTENT	WHAT IT NEEDS FROM YOU	HOW MUCH CONTROL YOU HAVE
AI search	Rendered pages, retrieved from a search index	Crawlable pages, clear headings, accurate structured data	High, and well documented

[6] Ofcom. "From apps to AI search: how the UK goes online in 2025." Ofcom, 10 December 2025. https://www.ofcom.org.uk/media-use-and-attitudes/online-habits/from-apps-to-ai-search-how-the-uk-goes-online-in-2025 Growth: 1,800 million ÷ 368 million = 4.9.

[7] Ofcom. "From apps to AI search: how the UK goes online in 2025." Ofcom, 10 December 2025. https://www.ofcom.org.uk/media-use-and-attitudes/online-habits/from-apps-to-ai-search-how-the-uk-goes-online-in-2025 Growth: 1,800 million ÷ 368 million = 4.9.

[8] Arielle Weiss, Zach Albertson and Emily Lanfear. "Content Independence Day, one year on: building the business model for the agentic Internet." Cloudflare, 1 July 2026. https://blog.cloudflare.com/agentic-internet-bot-report

READER	HOW IT REACHES YOUR CONTENT	WHAT IT NEEDS FROM YOU	HOW MUCH CONTROL YOU HAVE
Third-party assistants and their crawlers	Crawled or fetched pages, summarised alongside other sources	Clear, self-contained statements	Low
Autonomous agents	Screenshots, HTML, the accessibility tree, APIs and MCP servers	Semantic markup, a stable layout, explicit data and actions	Medium, if you offer a proper interface
Retrieval systems you own	Delivery APIs, exports and vector indexes	Entries that stand alone, metadata and permissions	High, because it is your own estate

1.2 AI Search

AI search reads your pages through the same index as traditional search. Google says its generative AI features, including AI Overviews and AI Mode, are rooted in its core ranking and quality systems.[9] They use two techniques. Retrieval-augmented generation (RAG) fetches relevant pages from the index and grounds the answer in them. Query fan-out runs several related searches at once to gather more material for the same answer.[10]

The reader still sees your content. The person often does not. Pew Research Center tracked the browsing of 900 US adults in March 2025. When a Google search showed an AI summary, users clicked a traditional result on 8% of visits, against 15% when

[9] Google Search Central. "Optimizing your website for generative AI features on Google Search." Google, published 15 May 2026, last updated 10 July 2026. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

[10] Google Search Central. "Optimizing your website for generative AI features on Google Search." Google, published 15 May 2026, last updated 10 July 2026. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

no summary appeared.[11] They clicked a link inside the summary itself on about 1% of visits.[12] Pew found that 18% of searches produced a summary; Ofcom's later UK figure is about 30%, although the two studies measured different countries, periods and methods.[13][14]

This is the reader you have the most influence over, and Google is clear about how to use it. Its guide says there are no additional requirements to appear in AI features, and that you do not need special files, special markup or content written specially for AI.[15] Good search practice still applies. The difference is that a summary now stands between your content and the person, so your content has to survive being summarised.

1.3 Third-Party Assistants and Their Crawlers

Assistants such as ChatGPT, Claude, Copilot and Perplexity read the web in two ways. Their crawlers collect pages, mostly to train models, and their search tools fetch pages to answer a question as it is asked. In both cases your content is read, compressed and blended with other sources before anyone sees it.

[11] Athena Chapekis and Anna Lieb. "Google users are less likely to click on links when an AI summary appears in the results." Pew Research Center, 22 July 2025. Based on 68,879 Google searches by 900 US adults in March 2025, 12,593 of which showed an AI summary. https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/

[12] Athena Chapekis and Anna Lieb. "Google users are less likely to click on links when an AI summary appears in the results." Pew Research Center, 22 July 2025. Based on 68,879 Google searches by 900 US adults in March 2025, 12,593 of which showed an AI summary. https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/

[13] Ofcom. "From apps to AI search: how the UK goes online in 2025." Ofcom, 10 December 2025. https://www.ofcom.org.uk/media-use-and-attitudes/online-habits/from-apps-to-ai-search-how-the-uk-goes-online-in-2025 Growth: 1,800 million ÷ 368 million = 4.9.

[14] Athena Chapekis and Anna Lieb. "Google users are less likely to click on links when an AI summary appears in the results." Pew Research Center, 22 July 2025. Based on 68,879 Google searches by 900 US adults in March 2025, 12,593 of which showed an AI summary. https://www.pewresearch.org/short-reads/2025/07/22/google-users-are-less-likely-to-click-on-links-when-an-ai-summary-appears-in-the-results/

[15] Google Search Central. "Optimizing your website for generative AI features on Google Search." Google, published 15 May 2026, last updated 10 July 2026. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

Cloudflare measures this with a crawl-to-refer ratio: the number of pages a platform's crawlers request for each visit that platform sends back.[16] In July 2025, Google crawled about 5 pages for each referral. OpenAI crawled about 1,091, and Anthropic about 38,066.[17] Cloudflare notes that the Anthropic figure may be overstated, because visits from the Claude apps do not always carry a referrer.[18] In the same month, 79% of AI crawling was for training, up from 72% a year earlier.[19]

You have little control over this reader. You can block its crawlers, and Section 6 (Governance, Risk and Adoption) covers that choice. You cannot choose how it splits, summarises or attributes your content. The only lever you hold is the content itself: whether each statement still makes sense once it has been lifted out of the page.

1.4 Autonomous Agents

Agents act for a person rather than answering a question. They book, compare, fill in forms and complete tasks. Google's web.dev guidance describes three ways an agent reads a website: a screenshot passed to a vision model, the raw HTML, and the browser's accessibility tree.[20] Most agents combine all three, using the HTML and accessibility tree for structure and the screenshot for layout.[21]

Each view loses something. A screenshot is slow and expensive to analyse. The HTML

[16] João Tomé. "The crawl-to-click gap: Cloudflare data on AI bots, training, and referrals." Cloudflare, 29 August 2025. Ratios are crawls per referred visit for July 2025, rounded to the nearest whole number. https://blog.cloudflare.com/crawlers-click-ai-bots-training/

[17] João Tomé. "The crawl-to-click gap: Cloudflare data on AI bots, training, and referrals." Cloudflare, 29 August 2025. Ratios are crawls per referred visit for July 2025, rounded to the nearest whole number. https://blog.cloudflare.com/crawlers-click-ai-bots-training/

[18] David Belson and Sam Rhea. "The crawl before the fall… of referrals: understanding AI's impact on content providers." Cloudflare, 1 July 2025. https://blog.cloudflare.com/ai-search-crawl-refer-ratio-on-radar/

[19] João Tomé. "The crawl-to-click gap: Cloudflare data on AI bots, training, and referrals." Cloudflare, 29 August 2025. Ratios are crawls per referred visit for July 2025, rounded to the nearest whole number. https://blog.cloudflare.com/crawlers-click-ai-bots-training/

[20] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

[21] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

shows nesting but not intent: a `<div>` styled as a button is not a button to an agent. The accessibility tree is the cleanest map of the page, but only if the markup is semantic.[22] The guidance puts the problem plainly: designs built for humans, with complex hover states and shifting layouts, are "functionally broken for agents".[23]

The better route skips the page entirely. A Model Context Protocol (MCP) server lets an agent work with CMS content and capabilities directly, with no page in between. Contentful already offers an MCP server, and I used it to research this paper. Connected to my own Contentful space, an AI assistant read the content model directly: content types, field names, validations and entries. It never saw technicaldogsbody.com. Every judgement it made about what an entry was came from the model alone, and Section 3 (Principles for Machine-Readable Content Models) shows how often that model fell short.

1.5 Retrieval Systems You Own

The fourth reader is the one you build. Internal copilots, support assistants and site search with generated answers all follow the same pattern. Content is split into pieces, each piece is indexed, and the best matches are passed to a language model with the question.

This reader exposes weak content models fastest, because it reads the entries themselves, not the pages. If an entry relies on the page around it for its subject, its date or its relationships, that context is lost the moment the entry is indexed. Anthropic measured the cost of that loss, and Section 3.2 (Name the Subject in Every Entry) sets out the figures.[24]

Retrieval systems do not rely on embeddings alone. Many vector search services can also filter on metadata, such as a date, an author or a content type, to narrow the entries a search considers. Azure AI Search, for example, filters on the non-vector fields in the

[22] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

[23] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

[24] Anthropic. "Introducing Contextual Retrieval." Anthropic, September 2024. https://www.anthropic.com/research/contextual-retrieval

same index, and by default applies the filter during the vector search itself.[25] A typed content model supplies those fields directly. A white paper with a publication date and an author reference can be filtered on both; a generic block with a field called `date`, or a date in its body text, cannot.

It is also the reader you control completely. You choose what is indexed, how it is split, which metadata travels with it and who can see it. That makes it the right place to start, and Section 5 (Case Study) uses it for that reason.

1.6 What the Four Readers Share

The four readers differ in how they reach your content and how much influence you have over them. They share one limitation: none of them sees the design. Each reads text, structure and labels, and web.dev's summary applies to all four: the quality of that machine-readable representation decides how well the reader performs.[26]

That representation is set by the content model long before any page is designed. Section 2 (Measuring Machine Readability) gives you a way to measure it.

Section 2: Measuring Machine Readability

You can measure how readable your content model is to machines before you change anything. The measurement needs read access to the content model and a way to count. It does not need an AI system.

Measure the model, not the pages. Section 1 (How Machines Read Content) showed that every machine reader works from text, structure and labels. The content model sets the structure and the labels for every entry of every type at once. A page audit finds symptoms one page at a time; a model audit finds the cause, and fixing one content type improves every entry of that type.

[25] Microsoft. "Filters in vector queries." Azure AI Search documentation, last updated 27 April 2026. https://learn.microsoft.com/en-us/azure/search/vector-search-filters

[26] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

2.1 Five Measures

Each measure tests one of the principles set out in Section 3 (Principles for Machine-Readable Content Models).

MEASURE	WHAT TO COUNT	WHAT GOOD LOOKS LIKE
1. Field naming (3.1)	Every field, classed as meaning, generic or presentation	Presentation fields only in types that exist to control layout
2. Self-describing types (3.2)	Content types whose meaning is set by a template or by the page around them	None
3. Relationship references (3.3)	Entry references that describe meaning, against those that describe layout; links held as text	Every relationship is a reference, and some describe meaning
4. Duplicated facts (3.4)	Facts held in more than one entry, in a sample of your most important content	None
5. Time and ownership (3.5)	Content types with a review date, an owner and a validity range	Every type whose content can go out of date

The first measure needs a definition, because it does the most work. Classify every field by one question: does its name say what the content is, without the page or the template?

- **Meaning.** The name states what the content is. Examples: `slug`, `publishedDate`, `metaDescription`.
- **Generic.** The name states a data type or a role, but not what the content is. Examples: `title`, `content`, `date`.

- **Presentation.** The name states where or how the content is shown. Examples: `ctaText`, `showAssetLink`, `colourWay`.

A generic field is not a fault on its own. A field called `title` is clear inside a content type called Testimonial. The same field inside a type called Content Block tells a machine nothing.

2.2 Worked Example: technicaldogsbody.com

The worked example applies the five measures to my own site's Contentful model, read through the Content Management API on 5 November 2026.[27]

```
Content model: technicaldogsbody.com, Contentful

Content types                                18
Fields                                       86

1. Field naming
   Meaning                          26 / 86 = 30.2%
   Generic                          34 / 86 = 39.5%
   Presentation                     26 / 86 = 30.2%
   Types with a presentation field   9 / 18 = 50.0%

2. Self-describing types
   Content Block entries                    139
   Content Block fields named for meaning    0 / 15
   Template values that set its meaning      9
     (3 on the block, 6 on the collections that hold it)

3. Relationship references
   Entry references                         15
   Describing layout or navigation    15 / 15 = 100%
   Describing what an entry is about   0 / 15 = 0%
   Links held as text (sampled)      at least 1

4. Duplicated facts (sample: one white paper)
```

[27] Content model of technicaldogsbody.com, Contentful, read through the Content Management API on 5 November 2026. The classification of each field is listed in Appendix A (Field Classification).

```
    Entries holding the paper                3
    Entries repeating the title              3
    Entries linking the PDF                  2
    Publication dates                        2, which disagree

 5. Time and ownership
    Types with a review date             0 / 18
    Types with an owner                  0 / 18
    Types with a validity range          0 / 18
    Types with an author                 0 / 18
    Date fields                              2
```

Three results stand out.

The busiest type says the least. Content Block holds 139 entries: testimonials, employment history, awards, projects, white paper listings and more. None of its 15 fields is named for what it holds. Nine template values, set on the block or on the collection around it, decide what each entry is. A machine reading the API sees 139 entries of one generic type.

Every reference describes layout. All 15 entry references in the model say where something appears or what a menu links to. None says what an entry is about, who wrote it or what it relates to. The model can tell a machine that a testimonial sits inside a collection called Kudos. It cannot tell it whom the testimonial is about.

The parts built for search engines score best. The Page type has 8 of its 11 fields named for meaning, and Page Meta has 3 of 4. Both were designed around search engine needs: slugs, canonical URLs, meta descriptions and publication dates. The model is most machine-readable where it was already serving a machine.

The breakdown by content type shows where the presentation fields sit.

CONTENT TYPE	FIELDS	MEANING	GENERIC	PRESENTATION
Photo Page	7	0	1	6
Blog Snippets	4	0	1	3
Content Block Collection	4	0	2	2
Menu Item	4	0	2	2

CONTENT TYPE	FIELDS	MEANING	GENERIC	PRESENTATION
Search	2	0	1	1
Content Block	15	0	9	6
Header	5	2	1	2
Banner	6	2	2	2
Page	11	8	1	2
Amazon Recommendation	5	2	3	0
Amazon Recommendations	4	2	2	0
Code Kata	4	4	0	0
Footer	2	1	1	0
Gallery	2	0	2	0
Labels	2	0	2	0
Page Meta	4	3	1	0
Social Link	3	1	2	0
TagGallery	2	1	1	0

Some presentation fields belong where they are. Blog Snippets and Search configure components, and a field such as `pageSize` is right for that job. The problem is presentation fields in types that hold content, such as Content Block and Photo Page.

2.3 A Maturity Model

The five measures place a content model on one of five levels.

LEVEL	NAME	SIGNS IN THE FIVE MEASURES
1	Page-shaped	Few content types; most content sits in rich text on pages

LEVEL	NAME	SIGNS IN THE FIVE MEASURES
2	Componentised	Reusable components exist, but templates decide what they are; references describe layout only
3	Structured	Types are named for what they hold; presentation fields stay in layout types; some references describe meaning
4	Governed	Types that can go out of date carry an owner, a review date and a validity range; each fact has one canonical entry
5	Machine-ready	Content is exposed deliberately to chosen machine readers, and their use of it is measured

My site sits at Level 2. It has reusable components, but its busiest content type takes its meaning from templates, and none of its references describes meaning. Two types come close to Level 3: Code Kata names all four of its fields for what they hold, and Page names 8 of its 11. That is a common pattern: a model rarely sits at one level throughout, and the parts that score well show the team what the rest could look like.

2.4 Limits of the Measure

The measures are a proxy, and they have three limits.

Field names are not content. A well-named field can still hold vague or out-of-date content. The measures show what the model allows, not what editors write. Section 6 (Governance, Risk and Adoption) covers the editorial side.

The classification is a judgement. Each of the 86 fields needed a call, and a second reviewer would move a few. `colourWay` is the clearest borderline case. On my site, Ember means work and Teal means personal, so the field holds real meaning; it is classed as presentation because its name describes a colour, not a category. That is the point of the test. Appendix A (Field Classification) lists every field and its class, so you can check or change any of them.

It measures the model, not the effect. A better score should mean better answers from machine readers, but this section does not prove it. Section 5 (Case Study) tests that claim directly, with the same questions asked of the current model and a remodelled one.

The method scales. An enterprise model has more types, but the counting can be scripted against the Contentful Content Management API or Optimizely content type definitions, and Section 4 (Implementation Patterns) shows how. Section 3 sets out what the measures are testing for.

Section 3: Principles for Machine-Readable Content Models

Five principles decide whether a machine can understand your content without seeing the page it was designed for. Each one is a content modelling decision, and an architect usually makes it in the first weeks of a build. Each principle also makes the model better for editors and front-end developers, which is the strongest reason to apply them now rather than wait for an AI project to force the issue.

I use my own site, technicaldogsbody.com, as the worked example throughout this section. I set up its Contentful space in 2019, for one website and nothing else. It breaks every principle below, which makes it a fair test.

3.1 Model Meaning, Not Presentation

A field should say what its content is, not where it appears or how it looks.

National Public Radio (NPR) set out this principle in 2009, long before AI readers existed. Separate content from display, and keep platform-specific markup out of content, be-

cause you cannot know where the content will end up.[28] Machine readers make the same point more sharply. An agent that reads your content through a delivery API or the Model Context Protocol (MCP) never sees the template. It sees field names and values, and the field names are the only labels it gets.

The same failure appears on the rendered page. Google's web.dev guidance on agent-friendly websites says that designs built for humans, with hover states and shifting layouts, are "functionally broken for agents".[29]

My site shows the problem in the content model. Testimonials are stored in a generic Content Block type, inside a collection whose Template field is set to `testimonials`. The person's name sits in a field called Short Title, and their organisation sits in a field called Title.

Nothing in the model says that the entry is a testimonial, that the short title is a person, or that the title is a company. The website knows, because the template knows. A machine reading the API sees two titles.

```
// Bad: the meaning lives in the template, not the model
{
  "contentType": "contentBlock",
  "fields": {
    "shortTitle": "[Person's name]",
    "title": "Optimizely",
    "content": "[Testimonial text]"
  }
}
```

```
// Better: the model says what the content is
{
  "contentType": "testimonial",
  "fields": {
    "quote": "[Testimonial text]",
```

[28] Daniel Jacobson. "Clean Content = Portable Content." Inside NPR.org, NPR, February 2009. https://www.npr.org/sections/inside/2009/02/clean_content_portable_content.html

[29] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

```
    "personName": "[Person's name]",
    "personRole": "[Role]",
    "organisation": { "link": "organisation/optimizely" },
    "subject": { "link": "person/andy-blyth" }
  }
}
```

Both samples are illustrative and simplified from the Contentful delivery API format.

The same pattern runs through the rest of the model. Whether an entry is a white paper is recorded in a field called Template, with the value `whitePaper`. The photo page type has fields called `prePhoto`, `preExif`, `preNav`, `preComments` and `postComments`, each named for its position on the page.

The test is simple. Describe every field without referring to the page. If a field name only makes sense with the design in front of you, it is a presentation field.

3.2 Name the Subject in Every Entry

Every entry should say what it is about in its own fields, so that it still makes sense when a machine reads it on its own.

Retrieval systems rarely read whole pages. They split content into fragments, index them, and pass the best matches to a model. Anthropic's research on Contextual Retrieval showed the cost of fragments that lose their subject. Its example is a chunk that reports revenue growth of 3% without saying which company or which quarter. Adding that context to each chunk before indexing cut the top-20 retrieval failure rate from 5.7% to 3.7% for embeddings alone, a reduction of 35%. Adding contextual keyword search as well took it to 2.9%, a reduction of 49%.[30]

This is not an argument for cutting pages into pieces. Google's guide to generative AI search says there is no requirement to break content into small pieces, and that its systems can find the relevant part of a page that covers several topics.[31] For Google Search, that advice is right, and you should not rewrite pages to satisfy an imagined AI

[30] Anthropic. "Introducing Contextual Retrieval." Anthropic, September 2024. https://www.anthropic.com/research/contextual-retrieval

[31] Google Search Central. "Optimizing your website for generative AI features on Google Search." Google, published 15 May 2026, last updated 10 July 2026. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

crawler.

The two positions answer different questions. Google describes its own systems, which index the whole page and can find the relevant part of it. Anthropic describes retrieval systems that you build or feed yourself: an internal copilot, a support assistant, an agent that queries your CMS. In those systems the entry is often the chunk. If an entry depends on the page around it for its subject, that meaning is gone the moment the entry is retrieved.

The fix belongs in the model, not in the copy. Anthropic generates the missing context with a language model at indexing time. An entry that already references its subject (a product, a service, a person, a location) carries that context wherever it goes. Editors do not need to repeat the company name and the quarter in every paragraph.

My testimonials fail this test too. None of them says who it is about. The site assumes the subject is me, because every testimonial appears on my site. The `subject` reference in the improved model above removes that assumption.

3.3 Make Relationships Explicit

Relationships between pieces of content should be references in the model, not something implied by sitting next to each other on a page.

On a page, proximity carries meaning. web.dev explains that an agent reading the HTML assumes a Buy Now button inside a product container belongs to that product.[32] That assumption holds when the markup is well formed. It fails when the relationship existed only in the layout and in the editor's head.

Retrieval research points the same way. A 2024 study of financial reports found that splitting documents by their structural elements, such as titles, paragraphs and tables, improved retrieval and question answering compared with splitting by paragraph or token length.[33] A 2026 comparison of four chunking strategies on oil and gas documents gave mixed results. Structure-aware chunking ranked the most relevant pieces highest,

[32] Kasper Kulikowski and Omkar More. "Build agent-friendly websites." web.dev, Google, last updated 1 April 2026. https://web.dev/articles/ai-agent-site-ux

[33] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde and Renyu Li. "Financial Report Chunking for Effective Retrieval Augmented Generation." arXiv:2402.05131, February 2024. https://arxiv.org/abs/2402.05131

with the smallest and fastest index, but recursive and fixed-size chunking scored higher on mean average precision and F1.[34] Both studies had to recover structure from documents after the fact. A content model that stores relationships as references hands that structure over for free.

My site has a clear example. The white paper *The Carbon Cost of Intelligence* is held in three entries. There is a Page, a Content Block that the Page uses as a component, and a card that lists the paper on the Works page.

The card does not reference the Page. It holds the paper's address as a text string in a field called External Url, repeats the title and links to the PDF again. The website resolves that address to the Page through a search index, outside Contentful. A person sees one white paper. An agent reading the content model sees three entries with the same title, and nothing in the model says that the card and the Page are the same paper.

3.4 Keep One Canonical Source for Each Fact

Each fact should live in one entry and be reused, not copied.

Duplicate content is a long-standing search problem, and Google's guide to generative AI search still recommends reducing it.[35] For machine readers, duplicates cause a worse problem than wasted crawling: two versions that disagree, with no signal about which is right. A person can usually spot the older version from the design. A model picks one.

Two of the three white paper entries on my site record a date, and they disagree. The Page says the paper was published on 15 April 2026. The Content Block it uses as a component says 13 April 2026. The website shows whichever date the template uses. An agent has to guess which one is the publication date.

A canonical source needs the model to support reuse: references to shared entries rather than copied fields, and one content type for each type of fact.

[34] Samuel Taiwo and Mohd Amaluddin Yusoff. "Evaluating Chunking Strategies For Retrieval-Augmented Generation in Oil and Gas Enterprise Documents." arXiv:2603.24556, March 2026, presented at CCSEIT 2026. https://arxiv.org/abs/2603.24556

[35] Google Search Central. "Optimizing your website for generative AI features on Google Search." Google, published 15 May 2026, last updated 10 July 2026. https://developers.google.com/search/docs/fundamentals/ai-optimization-guide

3.5 Make Time and Ownership Explicit

Every entry that can go out of date should say when it was true, when someone last checked it, and who owns it.

Anthropic's failing chunk was missing a time period as well as a company.[36] Time is the context most often left to the page: "this year", "our latest product", "the current offer". On a page, a publication date usually sits nearby. In a retrieved fragment it does not, and a model will repeat a superseded fact with the same confidence as a current one.

Three fields do most of the work:

- **Validity.** The date range in which the content is true. Offers, prices and events expire without an editor remembering to unpublish them.
- **Last reviewed.** When a named person last confirmed the content is still accurate.
- **Owner.** The person or team responsible for keeping it accurate.

These fields also work as filters. A retrieval system can exclude every entry whose validity has ended before it ranks anything (Section 1.5), which no rewording of the body text can do.

None of the 18 content types on my site has a review date or an owner field. The Content Block type has a single field called Date, and nothing in the model says what that date means.

I have not found published research that measures the effect of review dates or ownership fields on the accuracy of AI answers. This principle rests on the retrieval evidence above and on established content governance practice, and Section 6 (Governance, Risk and Adoption) builds on it.

Applying the Principles

Each principle comes down to one question you can ask of any content type.

[36] Anthropic. "Introducing Contextual Retrieval." Anthropic, September 2024. https://www.anthropic.com/research/contextual-retrieval

PRINCIPLE	TEST QUESTION	WHERE TECHNICALDOGSBODY.COM FAILS
3.1 Model meaning, not presentation	Can you describe every field without referring to the page?	A testimonial's author is stored in Short Title
3.2 Name the subject in every entry	Does each entry make sense on its own?	Testimonials do not say who they are about
3.3 Make relationships explicit	Is every relationship a reference in the model?	The white paper's listing card links to its Page by a URL string
3.4 Keep one canonical source for each fact	Does each fact live in exactly one entry?	The white paper has two publication dates
3.5 Make time and ownership explicit	Does each entry say when it was true and who owns it?	No content type has a review date or an owner

Section 4 (Implementation Patterns) turns each principle into content type definitions for Optimizely CMS and Contentful.

Section 4: Implementation Patterns

Optimizely CMS and Contentful can both hold a content model that meets the five principles. Neither enforces one. This section turns each principle into content type definitions for both platforms. It then covers the features that decide how machines read the result: the delivery APIs, the Model Context Protocol (MCP) servers, the AI tools in the editor and the visual page builders.

The Optimizely samples target CMS 13 on .NET 10. CMS 13 became generally available

on 31 March 2026, and it includes Optimizely Graph in every licence.[37] The Contentful samples use the content type format of the Content Management API. Every sample is illustrative: cut down to the fields that matter, with the rest of the type left out. Every product capability in this section was checked against the vendor's documentation in November 2026.

4.1 Meaning and Subject in the Type

Principles 3.1 (Model Meaning, Not Presentation) and 3.2 (Name the Subject in Every Entry) are both decided by the content type. The anti-pattern is one generic type whose meaning comes from a template value.

```
// Bad: one block type for everything; the template decides what it means
// Optimizely CMS 13 on .NET 10. Illustrative only.
[ContentType(GUID = "…")]
public class ContentBlock : BlockData
{
    // "testimonials", "awards", "roles"
    public virtual string Template { get; set; }
    // a person, an award or a job title
    public virtual string ShortTitle { get; set; }
    // an organisation, or something else
    public virtual string Title { get; set; }
    public virtual XhtmlString Content { get; set; }
}
```

```
// Better: the type says what the content is, and who it is about
// Optimizely CMS 13 on .NET 10. Illustrative only.
[ContentType(GUID = "…")]
public class TestimonialBlock : BlockData
{
    [Required]
    public virtual string Quote { get; set; }
```

[37] Optimizely. "2026 Optimizely CMS 13 general availability (GA) release notes." Optimizely documentation, accessed 5 November 2026. https://docs.optimizely.com/cms-13/release-notes/2026-optimizely-cms-13-general-availability-ga-release-notes The release date of 31 March 2026 is from the CMS 13 release notes index. https://docs.optimizely.com/cms-13/release-notes

```
    [Required]
    public virtual string PersonName { get; set; }

    public virtual string PersonRole { get; set; }

    [AllowedTypes(typeof(OrganisationBlock))]
    public virtual ContentReference Organisation { get; set; }

    [Required]
    [AllowedTypes(typeof(PersonPage))]
    public virtual ContentReference Subject { get; set; }
}
```

`AllowedTypes` restricts a `ContentReference` to one content type, or to any type that implements a contract.[38] The quote is a plain string, not rich text. A quote needs no layout, and a rich-text field invites an editor to add some.

Contentful expresses the same type with field types and a `linkContentType` validation on each reference.[39]

```
// Better: the same testimonial as a Contentful content type
// Content Management API format, trimmed. Illustrative only.
{
  "name": "Testimonial",
  "displayField": "personName",
  "fields": [
    {
      "id": "quote",
      "name": "Quote",
      "type": "Text",
      "required": true
    },
```

[38] Optimizely. "Advanced content types." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/advanced-content-types

[39] Contentful. "Content types." Content Management API reference, accessed 5 November 2026. https://www.contentful.com/developers/docs/references/content-management-api/content-types

```
    {
      "id": "personName",
      "name": "Person name",
      "type": "Symbol",
      "required": true
    },
    {
      "id": "organisation",
      "name": "Organisation",
      "type": "Link",
      "linkType": "Entry",
      "validations": [{ "linkContentType": ["organisation"] }]
    },
    {
      "id": "subject",
      "name": "Subject",
      "type": "Link",
      "linkType": "Entry",
      "required": true,
      "validations": [{ "linkContentType": ["person"] }]
    }
  ]
}
```

The validation does more than stop an editor linking the wrong entry. Contentful generates its GraphQL schema from the content model. A reference restricted to one content type becomes that type in the schema, and an unrestricted one becomes a generic `Entry` that a client has to cast.[40] The same field names that make the model readable to an editor become the API that a machine queries.

4.2 Relationships and Single Sources

Principles 3.3 (Make Relationships Explicit) and 3.4 (Keep One Canonical Source for Each Fact) usually fail together, in listing cards. The card copies facts from the page it promotes, then links to that page with a URL string.

[40] Contentful. "GraphQL Content API: schema generation." Contentful developer documentation, accessed 5 November 2026. https://www.contentful.com/developers/docs/references/graphql/schema-generation

```
// Bad: the card copies the page's facts and links to it by URL
// Optimizely CMS 13 on .NET 10. Illustrative only.
[ContentType(GUID = "…")]
public class WhitePaperCardBlock : BlockData
{
    public virtual string Title { get; set; }
    // a second copy of the date
    public virtual DateTime PublishedDate { get; set; }
    // a string, not a relationship
    public virtual string ExternalUrl { get; set; }
}
```

```
// Better: the page holds the facts, and anything that needs them points to it
// Optimizely CMS 13 on .NET 10. Illustrative only.
[ContentType(GUID = "…")]
public class WhitePaperPage : PageData
{
    [Required]
    public virtual DateTime PublishedDate { get; set; }

    [Required]
    [AllowedTypes(typeof(PersonPage))]
    public virtual ContentReference Author { get; set; }

    public virtual string Summary { get; set; }
}
```

A listing then becomes a query over `WhitePaperPage`, ordered by `PublishedDate`, not a set of cards that an editor keeps in step by hand. Where an editor needs to feature one paper, a `ContentReference` restricted to `WhitePaperPage` points to it without copying anything.

In Contentful, a reference also works in the other direction. The GraphQL Content API's `linkedFrom` field returns the entries that link to a given entry.[41] With a `subject` reference on every testimonial, one query returns every testimonial about a person. Without it, no

[41] Contentful. "GraphQL Content API: schema generation." Contentful developer documentation, accessed 5 November 2026. https://www.contentful.com/developers/docs/references/graphql/schema-generation

query can.

4.3 Time and Ownership as a Contract

Principle 3.5 (Make Time and Ownership Explicit) applies to many content types at once. CMS 13 can define a content type contract as a .NET interface that inherits from `IContentData`, and it generates an Optimizely Graph schema for each contract.[42][43]

```
// Better: one contract for time and ownership, implemented by any type that can
// go out of date
// Optimizely CMS 13 on .NET 10. Illustrative only.
public interface IReviewable : IContentData
{
    DateTime? ValidUntil { get; set; }
    DateTime? LastReviewed { get; set; }
    ContentReference Owner { get; set; }
}

[ContentType(GUID = "…")]
public class OfferPage : PageData, IReviewable
{
    public virtual DateTime? ValidUntil { get; set; }
    public virtual DateTime? LastReviewed { get; set; }

    [AllowedTypes(typeof(PersonPage))]
    public virtual ContentReference Owner { get; set; }
}
```

Because the contract has its own schema in Graph, one query can find every reviewable item past its review date, whatever its content type.

`LastReviewed` is not the same as the date the item was last saved. An edit to fix a

[42] Optimizely. "2026 Optimizely CMS 13 general availability (GA) release notes." Optimizely documentation, accessed 5 November 2026. https://docs.optimizely.com/cms-13/release-notes/2026-optimizely-cms-13-general-availability-ga-release-notes The release date of 31 March 2026 is from the CMS 13 release notes index. https://docs.optimizely.com/cms-13/release-notes

[43] Optimizely. "Create contracts." Optimizely CMS 13 documentation, accessed 5 November 2026. https://docs.optimizely.com/cms-13/docs/create-contracts

typo is not a review, and a review that changes nothing leaves no edit behind. Both platforms record when content changed; neither records when someone confirmed it was still true.

Contentful content types do not inherit from one another, so it has no direct equivalent of a contract. The same three fields go on each content type that can go out of date, which a migration script or the Content Management API can do in one pass.

4.4 Classification

Both platforms classify content outside the content type, and they do it differently.

Optimizely CMS 13 has categories: a hierarchy that administrators manage and editors apply to content.[44] Contentful has Taxonomy: concepts, with preferred and alternative labels and broader and narrower relations, grouped into concept schemes.[45] A content type only accepts concepts once a validation allows a scheme or a branch of concepts, and since February 2025 that validation can be required, so an entry cannot be published without them.[46]

A required classification is principle 3.1 applied to tags. An entry that cannot be published without saying what it is about does not depend on its page to say so.

4.5 How Machines Read the Model

The content model is only half of what a machine reader sees. The other half is how each platform exposes it.

[44] Optimizely. "Create a category." Optimizely CMS 13 documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/create-a-category

[45] Contentful. "Taxonomy manager." Contentful Help Center, accessed 5 November 2026. https://www.contentful.com/help/taxonomy/taxonomy-manager/

[46] Contentful. "Content type validations." Contentful Help Center, accessed 5 November 2026. https://www.contentful.com/help/taxonomy/application-of-taxonomy/taxonomy-content-type-validations/ And: "Taxonomy: Required Validations on Content Types in the CMA." Contentful changelog, 6 February 2025. https://www.contentful.com/developers/changelog/taxonomy-required-validations-on-content-types-in-the-cma/

QUESTION	OPTIMIZELY CMS 13	CONTENTFUL
Where does meaning live?	C# content types and contracts, with typed properties	Content types with typed fields and reference validations
How do machines read it?	Optimizely Graph, with a schema generated from content types and contracts	Content Delivery API and GraphQL Content API, with a schema generated from the content model
Is there an MCP server?	Yes, in `Optimizely.Cms.Opal.Tools` 13.2.0: content, content types and display templates	Yes, hosted or local: entries, search, content types, taxonomy and AI Actions
How is content classified?	Categories, a hierarchy managed by administrators	Taxonomy concepts and schemes, validated per content type
What AI sits in the editor?	Opal agents that draft content, manage content types and audit discoverability	AI Actions: prompts that editors run on whole fields
Where does it pull towards the page?	Visual Builder sections, and blocks embedded in rich text	Experiences components that hold their own content

Optimizely Graph. In CMS 13, Graph exposes content through shared base types such as `_Content`, with system data grouped under `_metadata`.[47] One detail matters more than the rest for machine readers: Graph in CMS 13 does not deliver content items em-

[47] Optimizely. "CMS 13 and 12 Graph comparison." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/cms-13-and-12-graph-comparison

bedded inside an `XhtmlString` rich-text property.[48] A testimonial dropped into a body field as a block is invisible to anything that reads Graph. Optimizely's own guidance is to move embedded content into a `ContentArea`.[49]

Shared blocks placed in a Visual Builder experience are inlined when the experience is indexed, so the delivered experience is complete while the block stays one source.[50] Graph also offers semantic ranking alongside keyword search, which Optimizely describes as experimental.[51]

Contentful's APIs. Both delivery APIs follow the content model directly. The GraphQL schema is generated from it, so typed references and `linkedFrom` come from the model at no extra cost (Section 4.1 and Section 4.2).[52]

MCP servers. Both vendors ship an MCP server, and both can change the model as well as the content.

- Optimizely added an MCP server to `Optimizely.Cms.Opal.Tools` 13.2.0 in September 2026.[53] It shares Opal's tools: 22 of them, covering content, content types, prop-

[48] Optimizely. "CMS 13 and 12 Graph comparison." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/cms-13-and-12-graph-comparison

[49] Optimizely. "CMS 13 and 12 Graph comparison." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/cms-13-and-12-graph-comparison

[50] Optimizely. "2026 Optimizely CMS 13 general availability (GA) release notes." Optimizely documentation, accessed 5 November 2026. https://docs.optimizely.com/cms-13/release-notes/2026-optimizely-cms-13-general-availability-ga-release-notes The release date of 31 March 2026 is from the CMS 13 release notes index. https://docs.optimizely.com/cms-13/release-notes

[51] Optimizely. "Semantic search." Optimizely Graph developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/platform-optimizely/docs/semantic-search

[52] Contentful. "GraphQL Content API: schema generation." Contentful developer documentation, accessed 5 November 2026. https://www.contentful.com/developers/docs/references/graphql/schema-generation

[53] Optimizely. "Optimizely CMS 13 release notes." Optimizely documentation, accessed 5 November 2026. Entry for `Optimizely.Cms.Opal.Tools` 13.2.0, 11 September 2026. https://docs.optimizely.com/cms-13/release-notes

erty groups, display templates and media. None of them searches.[54] The content tools address each item by its identifier, so an agent works with content it can already identify. Authentication by client secret is for local development, and Optimizely lists Optimizely Identity authentication for a future release.[55]

- Contentful offers a hosted server, signed in with OAuth, and a local open-source server that uses a personal access token. Its tools cover entries, including semantic search, content types, taxonomy, releases and AI Actions. The hosted server lets an administrator allow each tool category as read-only or read and write for each environment.[56]

An agent with write access to content types can change the content model itself. Contentful's documentation advises starting with read-only access and confirming tool calls.[57] Section 6 (Governance, Risk and Adoption) covers who should hold that access.

AI in the editor. Opal in CMS 13 runs agents inside the editor. They can create drafts, update properties, publish, manage content type definitions, and audit content for discoverability by large language models, writing suggested JSON-LD structured data to a property on the item.[58] Opal works within the user's existing CMS permissions.[59] Con-

[54] Optimizely. "Available tools." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/available-tools

[55] Optimizely. "MCP server for CMS 13." Optimizely CMS 13 developer documentation, last updated 14 September 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/mcp-server-cms-13

[56] Contentful. "MCP server." Contentful developer documentation, accessed 5 November 2026. https://www.contentful.com/developers/docs/tools/mcp-server/

[57] Contentful. "MCP server." Contentful developer documentation, accessed 5 November 2026. https://www.contentful.com/developers/docs/tools/mcp-server/

[58] Optimizely. "Optimizely Opal in CMS 13." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/optimizely-opal-in-cms-13

[59] Optimizely. "Optimizely Opal in CMS 13." Optimizely CMS 13 developer documentation, accessed 5 November 2026. https://docs.developers.optimizely.com/content-management-system/v13.0.0-CMS/docs/optimizely-opal-in-cms-13

tentful's AI Actions are prompts that administrators define and editors run on whole fields, for tasks such as translation, summaries and checks against brand guidelines.[60]

Both work on fields. A field named `personName` tells a prompt what it holds; a field named `shortTitle` does not. Structured data such as JSON-LD helps AI search, but it is a second statement of facts the content already holds. Build it from typed fields, so it changes when they change, and principle 3.4 still holds.

4.6 Page Builders Without Page-Shaped Content

Both platforms have a visual page builder, and both builders are good for editors. Optimizely's Visual Builder builds experiences and sections from rows and columns. Administrators choose which block types can be used as elements, and can define custom elements in the admin interface without a code deployment.[61] Contentful's Experiences let each component either hold content typed straight into it or be bound to an existing entry or asset.[62]

The risk is the same on both: meaning moves into the layout. A testimonial or a key statistic built as a styled section, a heading and a paragraph, has nothing in the model to say what it is. It is the generic Content Block from Section 4.1, built in the page builder.

Four rules keep the editor benefits without the cost:

1. **Facts live in typed content.** A testimonial is a `TestimonialBlock` or a `testimonial` entry. The layout places it; it does not hold it.
2. **Elements are typed.** In Visual Builder, allow typed blocks as elements, not a general text element that can become anything.
3. **Bind, do not type.** In Contentful Experiences, bind a component to an entry for anything that is a fact. Type content into the component only when it is presentation,

[60] Contentful. "AI Actions." Contentful Help Center, accessed 5 November 2026. https://www.contentful.com/help/ai-actions/

[61] Optimizely. "2026 Optimizely CMS 13 general availability (GA) release notes." Optimizely documentation, accessed 5 November 2026. https://docs.optimizely.com/cms-13/release-notes/2026-optimizely-cms-13-general-availability-ga-release-notes The release date of 31 March 2026 is from the CMS 13 release notes index. https://docs.optimizely.com/cms-13/release-notes

[62] Contentful. "Experience builder best practices." Contentful Help Center, accessed 5 November 2026. https://www.contentful.com/help/experience-builder-best-practices/

such as a section heading.

4. **Test without the layout.** Remove every layout and ask whether each fact still exists, with its subject. If a fact disappears with the layout, it belonged in the model.

Section 5 (Case Study) tests what these patterns change, by holding the same facts in a generic model and a typed one and asking a model the same questions of both.

Section 5: Case Study

Remodelling the same content to the five principles let three sizes of chat model answer all 19 questions, under both a strict and a permissive prompt. The original content model answered between 14 and 17.[63] Only the content model changed between Option A and Option B. The facts, the retrieval pipeline, the models and the questions were the same.

5.1 The Scenario

The test asks 19 questions about a fictional technical architect, Morgan Reeve, whose website runs on a fictional platform, Meridian CMS. Every person, organisation and product in it is invented, so the results depend on the content model and not on anything a model learned in training. The data is synthetic for a second reason: every question has one known right answer, so scoring is exact.

The same facts are held in two content models.

- **Option A** reproduces the patterns measured in Section 2 (Measuring Machine Readability). Testimonials, roles, awards and listings are generic Content Blocks whose meaning comes from a template value. Page bodies sit in separate component blocks. Listing cards link to their pages with a URL string. One white paper is held in three entries with two different dates. One package card was not updated when a later release added support for Meridian CMS 5.
- **Option B** models the same facts with the five principles from Section 3 (Principles

[63] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

for Machine-Readable Content Models). It uses typed entries for people, organisations, roles, testimonials, packages, white papers, blog posts and awards, with fields named for meaning and explicit references between them.

Option A has 34 entries and Option B has 27. Option B is smaller because it holds each fact once.

Five rules keep the comparison fair:

1. **Same facts.** Both options are generated from one fact base. Every fact in Option B also exists in Option A, except the relationships that Option A only implies, such as who a testimonial is about. Those relationships are the difference under test.
2. **Same pipeline.** One serialiser turns every entry into text, using the field names and values and resolving references to the linked entry's name.
3. **Same model and settings.** Both options use the same chat model, the same prompts, a temperature of 0 and a fixed seed.
4. **Deterministic scoring.** Each answer is checked for the facts it must contain and the wrong facts it must not contain. No model judges another model's answer.
5. **No tuning to the results.** The entries did not change after the first run. Every change to the questions or the scoring is dated in the test's change log and applies to both options.[64]

This is the fourth reader from Section 1.5 (Retrieval Systems You Own): a retrieval system built over your own content. It is the reader you control completely, which is why it can be tested this way.

[64]The test's change log records every run on 5 November 2026. The first used the strict prompt only. The second added the permissive prompt, the five outcomes and plausible wrong facts for eight questions, with the prediction recorded before it ran. Checking every failed answer by hand found three scoring misses during the day, all correct answers scored as failures: declines worded "does not contain any information" and "I do not have access", and a date written "June 1, 2021". Each pattern was widened for both options and both prompts. Re-scoring with the final patterns changes no result reported here.

5.2 Method

The test app is a .NET 10 console app that calls Ollama's API directly, with no AI libraries in between.[65] Each entry is one chunk, and retrieval uses embeddings alone, with no metadata filters, so the test measures only what the text of each entry carries. Entries are embedded with EmbeddingGemma 2 (`embeddinggemma-2:270m`), using the retrieval prefixes from its model card.[66] For each question, the five closest entries are passed to a Gemma 4 chat model, with thinking off, a temperature of 0, a seed of 42 and a context length of 8,192 tokens.

The test ran with three sizes of Gemma 4, changing only the chat model each time: `gemma4:e2b`, with 2.3 billion effective parameters; `gemma4:e4b`, with 4.5 billion; and `gemma4:12b`. Ollama lists the first two as edge models, for phones and laptops, and the third as a workstation model.[67]

Every question is asked under two system prompts:

- **Strict:** answer only from the entries, and say it does not know if they do not hold the answer.
- **Permissive:** give the most helpful answer you can from the entries, with no instruction to decline. Many assistants are set up this way.

Each of the 19 questions was asked three times under each prompt and each option. That gives 114 answers per option for each model, and 684 answers in all. One question is a control: it asks for a phone number that neither option holds, so the right answer is to decline.

Every answer gets one of five outcomes. It is correct and found in the entries; correct but assumed, when the model supplied a required fact it was not given; wrong, when it states a fact the content contradicts; declined; or incomplete. Because every person

[65] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

[66] Google DeepMind. "EmbeddingGemma 2 model card." Hugging Face. `https://huggingface.co/google/embeddinggemma-2`

[67] Ollama. "gemma4." Model library. `https://ollama.com/library/gemma4`

and product is fictional, an assumed fact cannot come from training data. It can only be an inference.

The runs used .NET 10.0.11 and Ollama 0.40.14 on Windows 11 Pro (build 10.0.26300). The machine is a Dell G5 5590 laptop with an Intel Core i7-9750H processor (2.60 GHz, 6 cores, 12 logical processors) and 32 GB of memory. The model digests are recorded with the results.[68]

Its graphics card is an NVIDIA GeForce RTX 2060 with 6 GB of memory (driver 552.22), but `gemma4:12b` does not fit in it. Loaded again after the test with a 4,096-token context, Ollama placed 21 of its 49 layers on the graphics card and the rest in system memory. The test's 8,192-token context leaves less room, so no more than that ran on the card during the test. Loaded the same way, `gemma4:e2b` placed all 36 of its layers on the card with more than 3 GB to spare. The `gemma4:e4b` model placed 42 of its 43 layers there, and the test's larger context may have left one or two fewer.

5.3 Results

CORRECT ANSWERS (OF 57)	OPTION A, STRICT	OPTION B, STRICT	OPTION A, PERMISSIVE	OPTION B, PERMISSIVE
`gemma4:e2b`	45 (79%)	57 (100%)	45 (79%)	57 (100%)
`gemma4:e4b`	42 (74%)	57 (100%)	45 (79%)	57 (100%)
`gemma4:12b`	51 (89%)	57 (100%)	51 (89%)	57 (100%)

All 342 of Option B's answers were correct. Because the three repeats of each question gave the same outcome, the results can also be read by question. Option A answered 15 of 19 with `gemma4:e2b`, 14 of 19 with `gemma4:e4b` under the strict prompt and 15 under the permissive one, and 17 of 19 with `gemma4:12b`. Option B answered all 19 with

[68] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

every model. No answer under either option, from any model, contained an assumed fact. Option A failed in these ways:

OPTION A OUTCOME (STRICT / PERMISSIVE)	GEMMA4:E2B	GEMMA4:E4B	GEMMA4:12B
Wrong fact	0 / 0	3 / 3	0 / 0
Declined	9 / 9	12 / 9	6 / 6
Incomplete	3 / 3	0 / 0	0 / 0

Retrieval does not depend on the chat model, so it was the same in every run. Every entry needed to answer was among the five retrieved for 16 of the 18 questions under Option A, and all 18 under Option B.[69] The control question is excluded from that measure.

STRICT PROMPT	TOKENS PER CORRECT ANSWER, A	TOKENS PER CORRECT ANSWER, B	MEDIAN ANSWER TIME, A	MEDIAN ANSWER TIME, B
gemma4:e2b	462	384	0.27 s	0.27 s
gemma4:e4b	503	389	0.70 s	0.75 s
gemma4:12b	407	382	3.50 s	3.62 s

Option B used 359 prompt tokens per answer against Option A's 341, and 1,125 characters of context against 1,047. The permissive prompt gave the same pattern of cost.

With gemma4:e2b, every question gave the same answer in all three repeats. The larger

[69] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. https://github.com/technicaldogsbody/the-assistant-test Chat models gemma4:e2b (digest b37049369adf…), gemma4:e4b (dc35e8d9c606…) and gemma4:12b (6114515d63c1…); embedding model embeddinggemma-2:270m (c21dd30c1e38…).

models varied their wording for a few questions between repeats, but no outcome changed.

Re-running the `gemma4:e2b` test with the full EmbeddingGemma 2 model retrieved the same five entries, in the same order, for every question. Every answer was identical.[70]

5.4 What the Breakdown Shows

Option B did not depend on the model. It answered every question correctly with all three models, under both prompts. Option A's score moved with the model, from 42 to 51 correct answers out of 57, and never reached all 57.

Option A failed in two ways, and the larger models separate them.

The first is retrieval. Two questions asked who wrote a white paper. In Option A, the only statement of authorship is the About entry, which says who the site belongs to. None of the three white paper entries names an author, so the right pair of entries never came back together.

All three models declined both questions, every time. No model can use an entry it is never given. Option B's white paper references its author, so one entry answered the question. This is principle 3.2 (Name the Subject in Every Entry) in its simplest form.

The second is structure. In three questions, the right entry was retrieved and its structure stopped the model using it.

- Asked what Priya Shah said about working with Morgan Reeve, Option A retrieved her testimonial first. Nothing in it says who it is about. Under the permissive prompt, `gemma4:e2b` explained exactly what it could see: "Priya Shah is listed as a testimonial under 'Kind words' ".
- Asked when Morgan Reeve joined Harbour Digital, Option A retrieved the right role entry. Its start date sits in a field called `date`, and `gemma4:e2b` answered from a blog post instead: "five years" ago, with no date.
- Asked for Morgan Reeve's job title at Brightwater Agency, Option A retrieved the role entry second. `gemma4:e4b` declined: "The provided context does not state that

[70] Embedder check, 5 November 2026: the `gemma4:e2b` run repeated with `embeddinggemma-2` (digest `f4981a15b998…`) in place of `embeddinggemma-2:270m`, with every other setting unchanged. All 228 answers matched in retrieved entries, answer text and outcome.

> Morgan Reeve worked at Brightwater Agency." That is accurate. The entry holds a job title, an organisation and dates, but no person.

`gemma4:12b` answered all three correctly. A larger model can bridge a missing label or a missing subject when the right entry reaches it, but it cannot make up for an entry that never arrives.

That bridge is an inference. `gemma4:12b` connected the Brightwater role to Morgan Reeve because he is the only person the site is about, not because the entry says so. Here, the inference was right. On a site about more than one person, the same move gives a confident wrong answer. That is why principle 3.2 names the subject rather than leaving the model to infer it.

The duplicate date reached the answer on one model. Option A holds two dates for one white paper. `gemma4:e2b` and `gemma4:12b` gave only the correct date. `gemma4:e4b` gave the correct date and then the stale one, in all six of its answers. Under the strict prompt, it added that "another entry mentions a date of 2026-03-08 for content related to this topic". It did not state the wrong date as the publication date, but it gave the reader two dates for one fact.

Under the scoring rules set before the run, that is a wrong fact. Whether a duplicate does harm depends on the reader, which is why principle 3.4 removes the duplicate rather than relying on ranking to hide it.

The prediction was wrong on every model. Before the second run, the test recorded a prediction: without an instruction to decline, Option A would start to guess.[71] Across 342 answers from Option A on three models, no answer contained an assumed fact. With this model family, weak structure produced gaps, and the only errors came from a contradiction in the content.

Better structure made the smallest model the best option. With every model, Option B used about 5% more prompt tokens per answer, because typed entries carry their

[71] The test's change log records every run on 5 November 2026. The first used the strict prompt only. The second added the permissive prompt, the five outcomes and plausible wrong facts for eight questions, with the prediction recorded before it ran. Checking every failed answer by hand found three scoring misses during the day, all correct answers scored as failures: declines worded "does not contain any information" and "I do not have access", and a date written "June 1, 2021". Each pattern was widened for both options and both prompts. Re-scoring with the final patterns changes no result reported here.

context with them. Measured per correct answer, it used fewer: 17% fewer with `gemma4:e2b`, 384 tokens against 462.[72]

The larger difference is across models. Under the strict prompt, Option B with `gemma4:e2b` answered every question, with a median answer time of 0.27 seconds. Option A with `gemma4:12b` answered 17 of 19 questions, with a median of 3.50 seconds.[73] That is about 13 times as long on the same machine, partly because most of `gemma4:12b` ran on the processor. Fixing the content model did more than moving to the larger model.

That connects this paper to *The Carbon Cost of Intelligence*. A declined or incomplete answer still costs energy, a question asked again costs it twice, and a larger model takes longer over every answer.

5.5 Reproducing the Test

The test app, the synthetic data and the generator that produces it are published on GitHub as the-assistant-test, under the MIT licence, with the results of every run reported here.[74] It needs the .NET 10 SDK, Ollama and the models. A self-test command checks the data and the scoring without a model. One command then runs both prompts against both options and writes a summary, a comma-separated values (CSV) file of every answer, and the environment, including the exact model digests. A BenchmarkDotNet project measures time and memory for the same workload under each option.

The test has four limits.

It uses one model family. Three sizes of Gemma 4 were tested, up to 12 billion param-

[72] Strict prompt. Tokens per correct answer: (prompt tokens + output tokens) ÷ correct answers. Option A: (19,440 + 1,359) ÷ 45 = 462.2. Option B: (20,448 + 1,422) ÷ 57 = 383.7. Difference: (462.2 − 383.7) ÷ 462.2 = 17.0%. Prompt tokens per answer: 341.1 against 358.7, a difference of 5.2%. Token counts are Ollama's own. The same figures for every run are in the test's results.

[73] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

[74] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. `https://github.com/technicaldogsbody/the-assistant-test` Chat models `gemma4:e2b` (digest `b37049369adf…`), `gemma4:e4b` (`dc35e8d9c606…`) and `gemma4:12b` (`6114515d63c1…`); embedding model `embeddinggemma-2:270m` (`c21dd30c1e38…`).

eters. Models from other families, or much larger models, may infer more from weak structure, or guess where these declined.

The corpus is small. With 34 and 27 entries, the top five is a meaningful cut-off. Anthropic's top-20 measure from Section 3.2 would retrieve most of the corpus here.

The scoring is strict. A correct answer phrased in an unexpected way can fail a pattern, and three did during development. Every failing answer in every run reported here was checked by hand, and each is a genuine failure, not a scoring miss.[75]

It tests one reader. Retrieval systems you own are one of the four readers in Section 1 (How Machines Read Content). The test says nothing directly about AI search or third-party assistants, although all four depend on the same text, structure and labels.

Section 6: Governance, Risk and Adoption

A content model that meets the five principles on the day it launches will drift, unless the editing interface, the review process and the access rules keep it there. This section covers what to put in place. That means an editor experience that makes structure the easy path, review and ownership, and rules for AI-generated content. It also covers the risks of serving machine readers, and an audit you can run on any content model.

6.1 Make Structure the Easy Path

Editors choose the generic block because it always works. It accepts any content, it never fails validation, and it is already on the page. A typed model only holds if the typed path is easier.

Four settings do most of the work:

- **Restrict what can be added.** `AllowedTypes` in Optimizely and `linkContentType` in Contentful limit a reference or a content area to the types that belong there (Section

[75] The test's change log records every run on 5 November 2026. The first used the strict prompt only. The second added the permissive prompt, the five outcomes and plausible wrong facts for eight questions, with the prediction recorded before it ran. Checking every failed answer by hand found three scoring misses during the day, all correct answers scored as failures: declines worded "does not contain any information" and "I do not have access", and a date written "June 1, 2021". Each pattern was widened for both options and both prompts. Re-scoring with the final patterns changes no result reported here.

4.1). Once typed blocks exist, remove the generic one from the list of allowed types, so it stops being the default choice.

- **Require the fields that carry meaning.** A testimonial without a subject, or a white paper without an author, should not publish. Both platforms can mark a field as required.
- **Name fields for the editor and the machine at once.** `PersonName` is clearer to an editor than `ShortTitle`, and clearer to a model too. A field name that needs help text to explain it is usually the wrong name.
- **Pick, do not retype.** Where a fact already exists, the editor should select it with a reference, not type it again. Every retyped fact is a future duplicate (principle 3.4).

6.2 Review and Ownership

Principle 3.5 (Make Time and Ownership Explicit) adds the fields; governance makes them mean something. A `LastReviewed` date that nobody updates is worse than none, because it looks authoritative.

- **Give every reviewable type an owner.** The owner is a person or a team, held as a reference, not a free-text name.
- **Set a review interval by type, not by item.** An offer may need checking monthly, a biography yearly and a white paper never, once published, apart from corrections.
- **Report on it.** The `IReviewable` contract in Section 4.3 lets one Optimizely Graph query list every item past its review date, whatever its type. In Contentful, the same query runs per content type. Send the list to each owner on a schedule.
- **Expire what has an end date.** Content with a `ValidUntil` date should leave every delivery API on that date, not when an editor remembers.

Section 5 showed why duplicates matter. One model ignored the stale copy of a white paper's date, and another repeated it to the reader. A review process that only checks the canonical entry will not find the copy, which is why principle 3.4 removes it.

6.3 AI-Generated Content

Both platforms now draft content inside the editor: Opal in Optimizely CMS 13, and AI Actions in Contentful (Section 4.5). The governance question is where human review sits, and how to record it.

Google's position is that generative AI is acceptable for content that helps people, but

that its output needs checking, because generative models predict a likely sequence of words rather than retrieve facts.[76] That includes the parts readers rarely see: titles, meta descriptions and structured data.[77] Google also suggests telling readers how a piece of content was created, where that helps them.[78]

For organisations publishing in the European Union, Article 50 of the AI Act has applied since 2 August 2026. A deployer that publishes AI-generated text "with the purpose of informing the public on matters of public interest" must disclose that it is artificially generated. The obligation does not apply where the text "has undergone a process of human review or editorial control" and a person or organisation holds editorial responsibility for it.[79]

Both points lead to the same pattern: record AI involvement and human review in the content model.

- **An `AiAssisted` field** on each type that AI tools can write to, set by the tool or the editor.
- **A `ReviewedBy` reference** to the person who checked the content, alongside the `LastReviewed` date from principle 3.5.
- **A publishing rule** that AI-assisted content cannot publish until `ReviewedBy` is set, enforced by a validation or an approval step.

Recorded this way, disclosure becomes a rendering decision, not an editorial one. The

[76] Google Search Central. "Google Search's guidance on using generative AI content on your website." Google, last updated 1 October 2026. https://developers.google.com/search/docs/fundamentals/using-gen-ai-content

[77] Google Search Central. "Google Search's guidance on using generative AI content on your website." Google, last updated 1 October 2026. https://developers.google.com/search/docs/fundamentals/using-gen-ai-content

[78] Google Search Central. "Google Search's guidance on using generative AI content on your website." Google, last updated 1 October 2026. https://developers.google.com/search/docs/fundamentals/using-gen-ai-content

[79] Regulation (EU) 2024/1689 (the Artificial Intelligence Act), Article 50(4). Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng Wording as quoted in: "The EU AI Act's Transparency Rules: A Practical Guide to Article 50." EU Artificial Intelligence Act website, accessed 6 November 2026. https://artificialintelligenceact.eu/transparency-rules-article-50/

fact is in the model, so a page, a feed or an API can show it wherever the organisation decides it should appear.

The same rule applies to agents. Opal works within the permissions of the user it acts for, and a Model Context Protocol (MCP) server acts with the permissions of the credentials it is given (Section 4.5). An agent that can change content types can change the content model. Treat that access like access to production code:

- Give agents read-only access to content by default.
- Allow changes to content types only in a non-production environment, reviewed and promoted like any other schema change.
- In Optimizely, where content types are defined in C#, a type changed through MCP or the admin interface drifts from the code. Decide which one is the source of truth before an agent is given access.

6.4 Risks

Serving machine readers well is not free of risk. Seven are worth deciding on deliberately.

Giving value away. AI search answers the question on the results page, and readers click through less often. Section 1 (How Machines Read Content) showed both the fall in clicks and the gap between how much AI platforms crawl and how many visits they send back. Better structure makes your content easier to use in an answer, which includes answers that do not send the reader to you.

Crawler controls and licensing. The main AI companies now separate crawling for training from crawling for search, so the choice is not all or nothing.

- OpenAI's GPTBot collects training data, OAI-SearchBot indexes sites for ChatGPT search, and ChatGPT-User fetches pages when a user asks. Blocking GPTBot does not remove a site from ChatGPT search.[80]
- Anthropic's ClaudeBot collects training data, Claude-SearchBot indexes for search,

[80] OpenAI. "Overview of OpenAI crawlers." OpenAI developer documentation, accessed 6 November 2026. https://developers.openai.com/docs/gptbot

and Claude-User fetches pages for a user's question.[81]

- Google-Extended controls whether content is used to train Gemini models and to ground their answers. Google states that it "does not impact a site's inclusion in Google Search nor is it used as a ranking signal".[82] AI Overviews are part of Google Search, so Google-Extended is not a way to leave them.

Cloudflare has blocked AI crawlers by default for new domains since 1 July 2025, and lets AI companies state whether a crawler is for training, inference or search.[83] Really Simple Licensing (RSL) 1.0, published in December 2025, goes further than blocking: it lets a publisher state machine-readable licence terms, such as allowing search indexing while refusing AI uses.[84] Cloudflare, Akamai and Creative Commons are among its supporters.[85]

My site allows all AI crawlers. In the week to 6 November 2026, Cloudflare recorded 8,434 requests to the site from the AI and search crawlers it tracks, and 5,884 of them succeeded.[86] Cloudflare classed 81% of the successful requests as AI crawlers, 13%

[81] Anthropic. "Does Anthropic crawl data from the web, and how can site owners block the crawler?" Claude Help Center, accessed 6 November 2026. https://support.claude.com/en/articles/8896518

[82] Google. "List of Google's common crawlers." Google Crawling Infrastructure documentation, last updated 30 October 2026. https://developers.google.com/crawling/docs/crawlers-fetchers/google-common-crawlers

[83] Cloudflare. "Cloudflare Just Changed How AI Crawlers Scrape the Internet-at-Large; Permission-Based Approach Makes Way for A New Business Model." Press release, 1 July 2025. https://www.cloudflare.com/press-releases/2025/cloudflare-just-changed-how-ai-crawlers-scrape-the-internet-at-large/

[84] RSL Collective. "RSL 1.0 specification." RSL Standard, 10 December 2025. https://rslstandard.org/press/rsl-1-specification-2025

[85] RSL Collective. "RSL 1.0 specification." RSL Standard, 10 December 2025. https://rslstandard.org/press/rsl-1-specification-2025

[86] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

as search engine crawlers and 6% as AI search. AI assistants fetching a page for a user made 7.[87]

Over almost eight months, from 13 March to 5 November 2026, Google Analytics recorded 3,128 visits to the site. One of them was referred by an AI assistant, from chatgpt.com.[88] Google Analytics counts a visit as coming from an AI assistant only when its referrer matches Google's list of assistants. A visit with no referrer is counted as direct traffic, so the true number may be higher.[89] Even so, the crawlers take far more than the assistants send back.

Attribution and accuracy. A machine reader can misread good content. The European Broadcasting Union's study of ChatGPT, Copilot, Gemini and Perplexity evaluated more than 3,000 answers about news, with 22 public service media organisations in 18 countries. Almost half of the answers had at least one significant issue, a third had serious sourcing problems, and a fifth had major accuracy issues, including outdated informa-

[87] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

[88] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

[89] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

tion.[90] Structure cannot prevent every misreading, but it removes the ones the content caused: a missing subject, an unlabelled date or a stale duplicate.

Measurement when the reader never visits. Page views measure human readers. They do not measure an answer that used your content and never linked to it. Google includes traffic from AI Overviews and AI Mode in Search Console's Performance report, within the Web search type, not as a separate figure.[91] Bot analytics, crawler data from the content delivery network, and testing assistants with your own questions (Section 2 and Section 5) fill part of the gap. None of them fills all of it, and they do not agree: for almost the same week, Microsoft Clarity counted 3,725 AI bot requests to my site, and Cloudflare counted 8,434.[92]

More content types. A typed model has more content types than a generic one, and every new type is another choice in the editor's list. Past a point, the list itself makes the editor experience worse, and the generic block starts to look easier again. Contentful also caps the number. Its technical limit is 1,000 content types per environment, but a plan's own allowance applies where it is lower, and its plans page lists 25 content types for a space.[93] My site already uses 18.

Model one type for each thing the content describes, not one for each variant or each position on the page: a testimonial is one type, whichever page it appears on. Use a classification field or Taxonomy for variants, and contracts in Optimizely for fields that

[90] European Broadcasting Union. "News Integrity in AI Assistants." EBU, 21 October 2025. https://www.ebu.ch/research/open/report/news-integrity-in-ai-assistants

[91] Google Search Central. "AI features and your website." Google, last updated 10 December 2025. https://developers.google.com/search/docs/appearance/ai-features

[92] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

[93] Contentful. "Technical limits." Contentful developer documentation, accessed 6 November 2026. https://www.contentful.com/developers/docs/platform/technical-limits/ And: "Contentful pricing." Contentful, accessed 6 November 2026. https://www.contentful.com/plans

many types share (Section 4.3). Hide the types editors should not create: Optimizely can remove a content type from edit view, or limit it to certain groups.[94]

More work for editors. Naming the subject in every entry (principle 3.2) means an editor sets an author or a subject on each block as well as on its page. On a site with hundreds of blocks, that is real overhead, and the fields that feel redundant are the ones editors leave empty. Three things reduce it:

- **Defaults.** Pre-fill the field when an entry is created. Contentful's default values cover Boolean, plain text, list, number and date fields, not references, so an author reference still has to be set by hand.[95]
- **Scope.** Require the subject on entries that leave their page: shared blocks, testimonials and anything an API or agent reads on its own. A block that only appears inside one page, and is delivered with it, can rely on the page.
- **Context at index time.** For a retrieval system you own, the indexer can add the page's subject to each block's text, as Contextual Retrieval does (Section 3.2). That helps only the readers you control; anything reading the API still sees the block without it.

The cost of remodelling. Remodelling costs effort, and a migration can break pages that depend on the old types. It does not have to happen at once. Start with the types that fail the Section 2 measures and carry the facts people ask about most: people, products, prices, dates and anything with legal weight. In the synthetic test data in Section 5, the remodelled content was smaller than the original, 27 entries against 34, because each fact was held once.

6.5 Sustainability

The Carbon Cost of Intelligence, my previous white paper, argued that every call to a model has an energy cost that architects can control. Machine readers make the same case for content.

Section 5 found that better structure let the smallest model answer every question, with

[94] Optimizely. "Define page properties." Optimizely CMS 13 documentation, accessed 6 November 2026. https://docs.optimizely.com/cms-13/docs/define-page-properties

[95] Contentful. "Default values for entry fields." Contentful changelog, 25 August 2021. https://www.contentful.com/developers/changelog/default-values-for-entry-fields/

a median answer time of 0.27 seconds. The original content, given a larger model, took 3.50 seconds and still missed two questions.[96] Part of that gap came from running most of the larger model on the processor (Section 5.4). Even so, the smaller model needs less hardware and less work for each answer. Good structure is an energy control.

Crawlers add to the cost. More than half of the 5,884 successful crawler requests to my site in that week, 3,199, were for its search results page, which repeats content held elsewhere. Stylesheets, scripts and fonts took another 978 requests and 12 MB, none of it content.[97] Disallowing the search page in robots.txt would remove most of that traffic, for the crawlers that respect it. A typed delivery API, such as Optimizely Graph or Contentful's GraphQL Content API, returns the content without the page around it. Offering one to the readers that can use it is a carbon decision as well as a technical one.

6.6 An Audit for Any Content Model

Each principle comes down to one question, from Section 3 (Principles for Machine-Readable Content Models). Governance adds four more. Ask all nine of each content type.

AREA	QUESTION
3.1 Meaning	Can you describe every field without referring to the page?

[96] Andy Blyth. "The Assistant Test." Source code, synthetic data and results, runs on 5 November 2026. https://github.com/technicaldogsbody/the-assistant-test Chat models gemma4:e2b (digest b37049369adf…), gemma4:e4b (dc35e8d9c606…) and gemma4:12b (6114515d63c1…); embedding model embeddinggemma-2:270m (c21dd30c1e38…).

[97] First-party data for technicaldogsbody.com, exported 6 November 2026: Cloudflare AI Crawl Control, 08:00 on 30 October to 08:00 on 6 November 2026; Microsoft Clarity bot analytics, 31 October to 6 November 2026; and Google Analytics 4 traffic acquisition, 13 March to 5 November 2026. Cloudflare's measures are described in: Cloudflare. "Analyze AI traffic." AI Crawl Control documentation, accessed 6 November 2026. https://developers.cloudflare.com/ai-crawl-control/features/analyze-ai-traffic On how visits are assigned to channels: Google. "Traffic-source dimensions." Analytics Help, accessed 6 November 2026. https://support.google.com/analytics/answer/9756891

AREA	QUESTION
3.2 Subject	Does each entry make sense on its own, and name who or what it is about?
3.3 Relationships	Is every relationship a reference in the model, not a URL or a name in text?
3.4 One source	Does each fact live in exactly one entry?
3.5 Time and ownership	Does each entry say when it was true, when it was reviewed and who owns it?
Editor experience	Is the typed path easier than the generic one?
Layout	Remove the layout: does every fact still exist, with its subject?
AI content	Does the model record AI involvement and who reviewed it?
Agent access	Who, and what, can change the content types?

A type that fails a question is a candidate for the next remodelling sprint, not a reason to start again.

Conclusion

Machine readers do not see the page. They see the content model: the types, the field names, the references and the place where each fact lives. That model is set early, by architects and technical leads, and it outlasts every redesign built on it.

Start with the model you have. Run the five measures from Section 2 on its busiest content types, and ask the nine audit questions from Section 6 of each one. Remodel the types that fail and carry the facts people ask about most, one at a time, with the

editor experience in mind. Then test the result with your own questions, the way Section 5 did.

In the case study, the smallest model answered every question from a typed content model. A larger model, which took 13 times as long on the same laptop, still missed two questions from the original. Better models will keep arriving. The content they read is the part you control.

Appendix A: Field Classification

Every field in the technicaldogsbody.com Contentful model (read on 5 November 2026), classed by the test in Section 2.1 (Five Measures). The Entry reference column marks fields that link to another entry, and whether the link describes layout or meaning.

CONTENT TYPE	FIELD	CLASS	ENTRY REFERENCE
contentBlock	shortTitle	Generic	
contentBlock	colourWay	Presentation	
contentBlock	customViewName	Presentation	
contentBlock	logo	Presentation	
contentBlock	image	Generic	
contentBlock	title	Generic	
contentBlock	content	Generic	
contentBlock	markdown	Generic	
contentBlock	date	Generic	
contentBlock	externalUrl	Generic	
contentBlock	internalLink	Generic	Layout
contentBlock	ctaText	Presentation	
contentBlock	asset	Generic	

CONTENT TYPE	FIELD	CLASS	ENTRY REFERENCE
contentBlock	assetCtaText	Presentation	
contentBlock	showAssetLink	Presentation	
menuItem	text	Presentation	
menuItem	page	Generic	Layout
menuItem	externalLink	Generic	
menuItem	colourWay	Presentation	
page	title	Generic	
page	slug	Meaning	
page	colourWay	Presentation	
page	pageType	Meaning	
page	publishedDate	Meaning	
page	category	Meaning	
page	tags	Meaning	
page	canonicalUrl	Meaning	
page	pageMeta	Meaning	Layout
page	components	Presentation	Layout
page	redirectUrl	Meaning	
photoPage	name	Generic	
photoPage	colourWay	Presentation	
photoPage	prePhoto	Presentation	Layout
photoPage	preExif	Presentation	Layout
photoPage	preNav	Presentation	Layout
photoPage	preComments	Presentation	Layout

CONTENT TYPE	FIELD	CLASS	ENTRY REFERENCE
photoPage	postComments	Presentation	Layout
banner	title	Generic	
banner	strapline	Meaning	
banner	background	Presentation	
banner	large	Presentation	
banner	descriptors	Generic	
banner	socialLinks	Meaning	Layout
socialLink	name	Generic	
socialLink	socialNetwork	Meaning	
socialLink	link	Generic	
pageMeta	name	Generic	
pageMeta	pageTitle	Meaning	
pageMeta	metaDescription	Meaning	
pageMeta	previewImage	Meaning	
amazonRecommendation	title	Generic	
amazonRecommendation	asin	Meaning	
amazonRecommendation	summary	Meaning	
amazonRecommendation	link	Generic	
amazonRecommendation	imageUrl	Generic	
amazonRecommendations	name	Generic	
amazonRecommendations	title	Generic	
amazonRecommendations	asinIDs	Meaning	
amazonRecommendations	products	Meaning	Layout

CONTENT TYPE	FIELD	CLASS	ENTRY REFERENCE
codeKata	kataName	Meaning	
codeKata	kataDescription	Meaning	
codeKata	kataType	Meaning	
codeKata	kataLink	Meaning	
labels	name	Generic	
labels	values	Generic	
tagGallery	title	Generic	
tagGallery	tag	Meaning	
gallery	title	Generic	
gallery	images	Generic	
search	name	Generic	
search	pageSize	Presentation	
contentBlockCollection	title	Generic	
contentBlockCollection	customViewName	Presentation	
contentBlockCollection	items	Generic	Layout
contentBlockCollection	background	Presentation	
blogSnippets	name	Generic	
blogSnippets	limit	Presentation	
blogSnippets	showLink	Presentation	
blogSnippets	masonry	Presentation	
footer	name	Generic	
footer	socialLinks	Meaning	Layout
header	name	Generic	

CONTENT TYPE	FIELD	CLASS	ENTRY REFERENCE
header	darkLogo	Presentation	
header	lightLogo	Presentation	
header	menuItems	Meaning	Layout
header	socialLinks	Meaning	Layout