Engineer

Metadata

the data describing other data, so it can be found

What it is

Metadata is data whose subject is other data: the title, date, language, author, format and size that travel with a document, so that people and programs can sort it, trust it and act on it before reading it. Older standards hyphenate it as meta-data, and day to day it goes by narrower names: header fields in HTTP, properties in an office file, tags in a photo, front matter at the top of a Markdown page.

A common split has three kinds. Descriptive metadata says what a thing is about: title, creator, subject, summary. Structural metadata says how its parts fit together: chapters, pages, reading order. Administrative metadata says where it came from and what may be done with it: when it was created and changed, by whom, under which licence, in which format. The Dublin Core, fifteen elements that became the standard ISO 15836 in 2003, is the best-known shared list of descriptive fields, and its title, creator, date, language and rights reappear under other names in almost every format since.

A web page carries metadata in layers. The HTTP response comes first. RFC 9110 says a header field “might modify or extend message semantics, describe the sender, define the content, or provide additional context”, and `Content-Type`, `Content-Language` and `Last-Modified` describe the page before a byte of it arrives. Inside the HTML sit the `<title>`, the `lang` attribute and `<meta>` elements whose names the HTML standard defines, such as `description`, `author` and `generator`. Above those come vocabularies that other parties read: Open Graph properties for link previews, and structured data for search engines and assistants.

So what separates metadata from data? The reader, more than the bytes. A catalogue record is metadata about a book and the whole dataset to the catalogue; a photo's capture time is a detail to the person looking at it and the raw material for someone mapping where photos are taken. Metadata versus structured data is the narrower question, and the answer is that structured data is metadata written in a shared vocabulary, so that a program can read a field without guessing what it means.

Use the concept in a decision

Put "What is metadata?" to work in a real decision. We can help relate the concept to the system you are building or buying. Rules and checks that catch bad data on the way in, and a clear answer to who owns which figure and where it came from.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. One infrastructure plan for 14 departments — 14 research departments on one infrastructure plan

One infrastructure plan for 14 departments Get in touch →

Why it matters

Each field is a small promise, and the decision that matters is where each one gets its value: a person typing it, the version history, a database column, or whatever the filesystem happens to report. Programs act on those promises without checking them. That is the point of metadata, and its risk.

Caches and crawlers spend their effort by it. A browser holding a copy of a page sends back the `Last-Modified` date it was given, and when nothing has changed it receives a short `304 Not Modified` instead of the page again. RFC 9110 asks a server to send that date for any representation whose last change "can be reasonably and consistently determined", and the condition cuts both ways: where no consistent date exists, the standard does not ask for one. A sitemap's modification date does the same job for a crawler choosing what to fetch again.

People and assistive technology depend on it too. The `lang` attribute tells a screen reader which pronunciation rules to apply, and the title is what a tab, a bookmark and a search result show. Rights travel the same way: a package manager knows a dependency's licence only because the dependency declared it, so a licence review built on that list is exactly as good as the declarations underneath.

Every field has to stay true, so a page with forty fields has forty chances to be wrong. Some fields are read by nobody: the HTML standard notes that many search engines do not consider the `keywords` name, because it was used to spam results. And metadata leaks, since a photo's position or a document's revision history can say more than the content does. Fewer fields, each filled from a source that cannot drift, usually beat many.

What it looks like in practice

Metadata turns up in three projects on this site, each time in a different kind of file: a sitemap, a set of PDFs and a dependency graph.

One site's sitemap took its last-changed dates from the wrong metadata, and 172 of its 176 pages ended up sharing one instant. Those dates were file modification times, metadata about the bytes rather than the content, and each run of the content export wrote all of them anew. The fix reads a fallback chain, an explicit field first, then the commit history, then the file, and it records the trap inside it: the literal field name has to appear in that list, or the generator never reads the field at all. The records behind the pages now keep the day each entry was written and revised apart from the years the work itself took place, and a date nobody knows stays empty, on the rule that "a precise wrong number is worse than an absent one". At the recorded implementation stage, a re-sync after a month of edits touched 8 files of 186.

At the recorded validation stage, nine generated PDF documents measured against the accessibility standard passed all 106 configured accessibility-profile rules. The work included a declared document language and a real title in the metadata, alongside tagged structure, alternative text and an explicit reading order. In that recorded run, the same documents failed exactly one rule of 146 in the archival profile. The page calls that a near miss with the failing rule named, because 145 passes is not conformance.

And when the licences of 644 Rust crates were checked on every build, the stated limit of the audit was a limit of metadata. It reads each crate's declared licence, declarations can be wrong or incomplete, and the check proves nothing about a crate that misdeclares itself. It was made to fail on purpose twice, which shows the check works; nothing a check does can show that the declarations it reads are honest.

Where it goes wrong

The faults below share one trait: every check that is not looking for them passes.

Every page reports a change after every deploy. A web server that sends a static file's modification time as `Last-Modified`, or a generator that dates a sitemap the same way, reports the moment the file was copied, and a checkout, an export or a container build copies all of them. Caches revalidate everything and crawlers learn nothing. Take dates from the version history or from a field a person wrote, and let a date nobody knows stay absent. In Hugo the sources are a list read in order, and the field's own name has to be in it:

```
enableGitInfo = true

[frontmatter]
date = ["date", ":git", ":fileModTime"]
lastmod = ["lastmod", ":git", ":fileModTime"]
```

A title or a link preview shows broken characters. The character encoding is metadata about every other byte, and a parser that meets text before the declaration has to guess. The HTML standard requires the declaring element to fit completely within the first 1,024 bytes, and putting it first in `<head>` leaves nothing to guess. On this site, Bing's URL Inspection once read the title and both Open Graph strings as Latin-1, because an em dash in the title arrived before the charset. The HTTP header named UTF-8 as well, but a tool reading the bytes without the header has only the page to go on. The charset is now the first element, and a check confirms that order on every built page.

The same fact says different things in different places. A page's title lives in `<title>`, in `og:title`, in its structured data and, for a PDF, in the document properties; its language lives in `lang`, in `Content-Language` and in `og:locale`. Typed separately, they drift apart, and each consumer believes a different one. Compute each value once, in one template or function, and have every place that prints it read that value.

A file says more than its author meant to publish. Photos carry the camera, the time and often the position; office documents carry author names, edit history and template paths; a `generator` tag names the software that built a page. Strip on export whatever nobody chose to publish, and treat every field as text a stranger will read.

Further reading

- RFC 9110, section 6.3: Header Fields — how HTTP carries metadata ahead of the content, and what a header field may do.
- RFC 9110, section 8.8.2: Last-Modified — when a server should send a modification date, and why one in the future is replaced.
- HTML Standard: the meta element — the standard metadata names and the rules for declaring a character encoding.
- DCMI Metadata Terms — the fifteen Dublin Core elements and the vocabulary built around them.
- The Open Graph protocol — the four properties it requires on every page, and the optional ones around them.

Glossary

Choose for your actual constraints

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is metadata?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/metadata/