DIGITAL ONE FOUNDATION

IN FORMATION · AMSTERDAM · MMXXVI

SERIES

THE RECORD · FOUR PAPERS · OPEN LICENCE

The Text *Layer*.

Working notes on what a model is handed when it is given a document

PAPERS

PAPER 001	**What the model reads**	METHOD · 7 MIN
PAPER 002	**The honest number moved downwards**	METHOD · 8 MIN
PAPER 003	**Beyond the text layer**	RESEARCH · 7 MIN
PAPER 004	**Properties, not promises**	RESEARCH · 7 MIN

ISSUED

four papers · published 11 September 2026 · CC BY 4.0

digital1.foundation/articles/the-text-layer/

Stichting Digital One Foundation i.o. · Amsterdam

These notes describe work contributed to an entity in formation.

TOC

THE TEXT LAYER · FOUR PAPERS · OPEN LICENCE

Contents

001 **What the model *reads*** 4

METHOD · 7 MIN · PUBLISHED 11 SEPTEMBER 2026

A PDF page pays twice on Claude. What conversion to Markdown removes, measured on sixteen public documents, and what it cannot keep.

§1	A page pays twice	4
§2	Markdown is larger than a text dump	5
§3	The measurement, done properly	5
§4	What this cannot do, stated before anyone else says it	6
	References	8

002 **The honest number moved *downwards*** 9

METHOD · 8 MIN · PUBLISHED 11 SEPTEMBER 2026

A published saving was counted with another vendor's tokenizer on a corpus nobody could re-run, and the correction moved it down.

§1	A flat charge divided by a variable one	9
§2	The defect: a claim about one vendor, counted with another's tokenizer	10
§3	Where it runs, and what the result records	12
	References	14

003 **Beyond the *text layer*** 15

RESEARCH · 7 MIN · PUBLISHED 11 SEPTEMBER 2026

Open research on reading what is in a picture rather than the glyphs on it: what a claim about a drawing needs before a model sees it.

§1	What the model was shown, six months later	15
§2	OCR reads the glyphs, not the picture	16
§3	A picture is a claim waiting to be made	17
§4	What is being researched, and what would count as a result	18
	References	19

004 **Properties, not *promises*** 21

RESEARCH · 7 MIN · PUBLISHED 11 SEPTEMBER 2026

Where converted text should live, what repetition makes compressible, and why a recorded property is not a promise.

§1 What is open past the text layer 21

§2 Where the Markdown should live 21

§3 Repetition is the compressible part 22

§4 A record says where pages went, not that the lines are right 23

§5 An honest ledger of what is open 24

References 25

END **Colophon** 26

SOURCES AND THEIR HASHES, LICENCE, AND HOW THIS DOCUMENT IS SET

001

PAPER 001 · METHOD

What the model *reads*.

A language model handed a PDF is not handed its text, and on Claude it is billed twice over for the same page: **a page pays twice**. These notes measure what conversion to Markdown removes — sixteen public documents, 861 pages, counted with `messages.count_tokens` — and what it cannot keep.

PAPER · 001 PUBLISHED · 11 SEP 2026 READING · 7 MIN
LICENCE · CC BY 4.0
Canonical · digital1.foundation/articles/the-text-layer/001-what-the-model-reads.html

§1

A page pays twice

A PDF is a set of drawing instructions. The text a reader sees is one layer of it — sometimes present, sometimes not, never the whole — which is why a scanned page and a born-digital page cost the same picture and different text, and why the corpus in §3 holds both. Before any of an estate reaches a model one decision comes first: hand the files over as they are, or convert them. The first is what the vendor's own upload exists for.

On Claude that means a `document` content block carrying the raw bytes. Two sentences of the vendor's description decide the bill: "The system converts each page of the document into an image", and "The text from each page is extracted and provided alongside each page's image".[1] The page arrives twice — as a picture of itself, and as the text lifted off it — and both are input tokens.

That is a feature before it is a cost — a model shown the picture can be asked about the chart on it — and a bill paid on every pass. The same documentation puts the text alone at 1,500–3,000 tokens a page, depending on content density, with the image cost added on top.[1]

> THE COMPARISON EVERY FIGURE IS STATED AGAINST
>
> ***Native document upload*** on Claude: a PDF sent as a document block, charged as page image ***plus*** extracted text. Every figure below is stated against that baseline, on the model it names. §4 names the vendor where the sign reverses.

§2

Markdown is larger than a text dump

A paper that opened with the saving would be selling, so start with the unflattering number. Over the same sixteen documents the Markdown this engine produces costs **1,323 tokens a page against 1,239 for a plain text dump** — 1.068×, rendered 1.07×. Per document it runs from 0.94× to 1.73×, and the 1.73× is a tax form, where the syntax that rebuilds a grid costs more than the words in it.

So against plain extracted text there is no saving to claim, and none is claimed. What the syntax buys is the grid, the heading tree, the numbering and the lists — a value stays in the row it was printed in. A number that has escaped its row is not a cheaper answer. It is a wrong one that costs less.

For much of an estate the comparison does not arise, because there is no native path to compare against. "Binary formats such as .xlsx or .docx are not supported in document blocks and must be converted to text or PDF first."[1] PPTX is absent from the types the Files API gives a document block.[2] There the question is not whether to convert but whether conversion keeps the structure.

§3

The measurement, done properly

The corpus is not a sample of anyone's estate: sixteen public PDFs, 861 pages, weighted towards documents hard in different ways and dense-text-heavy, because dense text is where the Markdown is largest and the saving smallest. A corpus of slide decks and scanned forms would have flattered the engine. This one does not.

Each document is named in a manifest by publisher URL, sha256 and page count, and fetched from its publisher rather than redistributed.[3] Each was counted twice with `messages.count_tokens` — free, and called an estimate by the vendor itself: a count, not an invoice.[4] Once as a document block holding the raw PDF, once as a text block holding the Markdown `crunch` 0.1.0 made from it, scaffolding subtracted — each on Claude Opus 5, Claude Sonnet 5 and Claude Fable 5.1. The three returned identical counts. Not close: identical.

Totals are authoritative rather than rounded per-page figures: 1 − 1,139,102 ÷ 2,404,262 = **52.6%**, rendered 53%. The mean is published rather than the median of 55.4%, because it is the lower of the two. Native costs 2,792 tokens a page and the Markdown 1,323, and the range runs from 11.1% to 73.1%: a mean over an arbitrary corpus is not a benchmark result, so the spread travels with it.

DOCUMENT	WHAT IT IS	PAGES	NATIVE → CRUNCHED, TOKENS/PAGE	FEWER
nasa-tm-x-3181	Scanned typewritten memo	23	2,095 → 564	73.1%
scotus-brown	Scanned court pages	14	2,342 → 745	68.2%
nist-csf2	Security framework	32	2,287 → 733	67.9%
rfc9110	Protocol specification	194	2,405 → 876	63.6%
nist-sp800-63b	Technical guideline	80	2,425 → 888	63.4%
statute-79-pg437	Scanned session law	10	2,814 → 1,138	59.6%
eurlex-gdpr-en	EU regulation, English	88	2,792 → 1,136	59.3%
fed-h41	Statistical release	11	2,602 → 1,135	56.4%
bis-quarterly	Financial review, charts	102	2,532 → 1,155	54.4%
cfr-40-1-1	Regulation, one sparse page	1	3,007 → 1,427	52.5%
arxiv-gpt3	Research paper, wide tables	75	2,784 → 1,345	51.7%
eurlex-gdpr-hu	The same, in Hungarian	88	3,605 → 1,941	46.2%
arxiv-attention	Mathematics-heavy paper	15	2,557 → 1,415	44.6%
arxiv-resnet	Two-column paper	12	3,562 → 2,197	38.3%
irs-i1040gi-2023	Tax booklet, three columns	114	3,614 → 2,483	31.3%
irs-f1040-2023	Tax form on a grid	2	3,320 → 2,952	11.1%
Corpus	Totals 2,404,262 → 1,139,102	**861**	**2,792 → 1,323**	**52.6%**

§ 4

What this cannot do, stated before anyone else says it

A paper that lists only what a conversion keeps is marketing with a token count attached. These losses are structural rather than temporary:

1 **A merged cell is flattened.** The GitHub Flavored Markdown table extension defines a header row, a delimiter row and data rows; colspan, rowspan and merged cell do not occur in the specification. [5] In a PDF a spanning cell keeps its text in the first column it covered and the rest come through blank; in a presentation the covered cells are gone rather than blank. The table is named in a warning, the structure score subtracts, and no completeness claim about tables survives it.

2 **A picture is not read.** A figure becomes a placeholder holding its position, carrying a caption only where one was found on the page — never one generated on the page's behalf. The dimensions citable out of a drawing are therefore the ones that were in a table, and because a merged cell there is flattened, not even all of those.

3 **Some structure is not extracted at all.** DOCX headers, footers, footnote and endnote text are not, and the emitter has no footnote syntax on any input path — so nothing here claims footnotes survive as footnotes. Tables are skipped on rotated pages; formula cells with no cached result come through empty. Most of those announce themselves in a warning. The missing footnote syntax does not.

4 **A line read off pixels is capped below certain** — never `certain`, and flagged for help at or below `guessed` — and handwriting is declined rather than guessed: a signature or a margin note comes back as a page named for a person to check rather than as text.

5 **On Gemini the sign reverses.** Google's documentation states that each document page is equivalent to 258 tokens, and that you are not charged for tokens originating from the extracted native text in PDFs. [6] Markdown at 1,323 tokens a page therefore costs about 5.1× the native file there. On Gemini, send the PDF.

None of these is a reason not to convert. They are the reasons to describe the conversion accurately.

> ON THE PROVENANCE OF THIS MATERIAL
>
> The method and measurements here were developed inside ***crunch.md***, a Digital One product; the defect found by re-measuring that product's own published claim is set out in Paper 002, §2. The engine, `crunch-core`, and its MCP server are Apache-2.0 licensed and run offline; the hosted service is proprietary. The repository is not public yet, so that licence is a statement of intent rather than something a reader can check today. The corpus is public, and the native column can be re-counted by anyone holding the manifest and a key. Deliberately absent: prices, plans and tiers, and deployment internals.

References

1 Anthropic, ***PDF support***, "How PDF support works" and "Estimate your costs"; the binary-formats sentence quoted in §2 is on this page, not the Files API page. platform.claude.com/docs/en/build-with-claude/pdf-support, read 11 September 2026.

2 Anthropic, ***Files API***, "File type support"; PPTX is absent from the supported-types table. platform.claude.com/docs/en/build-with-claude/files, read 11 September 2026.

3 The corpus. It is named in a manifest — `packages/crunch-core/tests/corpus/manifest.toml`, sha256 `f4abe7ec1bc2ffa8b499508810bec3e4c5009ddbcd83c3655e3d36c7242cbaf0` — which sits in a repository that is not public yet, so its rows are printed here instead: id, publisher URL, and the sha256 of the file as published on 7 September 2026. A publisher who re-issues a document changes the hash, and a re-run should say so rather than measure a different document under the same name. The two EUR-Lex documents answer an automated fetch with a bot challenge and must be fetched by hand.

```
arxiv-attention   https://arxiv.org/pdf/1706.03762v7
                  bdfaa68d8984f0dc02beaca527b76f207d99b666d31d1da728ee0728182df697
    arxiv-resnet      https://arxiv.org/pdf/1512.03385v1
                      1e0651b6810ecba34a3dbc5b5b0209226f889004607c1f203540a48d64e5a93a
    arxiv-gpt3        https://arxiv.org/pdf/2005.14165v4
                      97fd272f1fdfc18677462d0292f5fbf26ca86b4d1b485c2dba03269b643a0e83
    nist-csf2         https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf
                      3c31f46fee98cac0c4323453e5109291a213b4de7fef8c058af9bf67f717433c
    nist-sp800-63b    https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf
                      ccfce7510a1267933f912d023306c6e7d485f21be63d271108b61aad6139127e
    rfc9110           https://www.rfc-editor.org/rfc/rfc9110.pdf
                      60b30efa1048900833d1758440247fe8ac85a3134f2327388dcb24e07d814c89
    eurlex-gdpr-en    https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679
                      bd84e63f5b622b739a83389afc3b30d240f792bb88d8eb03a816c9a82b0c2499
    eurlex-gdpr-hu    https://eur-lex.europa.eu/legal-content/HU/TXT/PDF/?uri=CELEX:32016R0679
                      b4671fe07338c3b4b346128cf3a7652c8d57f341deb512ec9bc7c74f30074fa8
    cfr-40-1-1        https://www.govinfo.gov/content/pkg/CFR-2023-title40-vol1/pdf/CFR-2023-
    title40-vol1-sec1-1.pdf
                      67fefda601faab628481ada17684bb65e3392e2e9db879da5276291ba3f2c7e1
    irs-f1040-2023    https://www.irs.gov/pub/irs-prior/f1040--2023.pdf
                      a410ce39e1f9b75ab3ddd8c314b9e7dce022ed6e3aebde5418a58b88f1e0213c
    irs-i1040gi-2023  https://www.irs.gov/pub/irs-prior/i1040gi--2023.pdf
                      90c8759c7a829d8c74296187cdbdb47bc9025480fffc41ff5e7225889c8fea7e
    fed-h41           https://www.federalreserve.gov/releases/h41/20240104/h41.pdf
                      614f7600fcde1ba0ec1db2047580a518fe2888ecc7808e76aea7e27e5a7fa6ba
    bis-quarterly     https://www.bis.org/publ/qtrpdf/r_qt2403.pdf
                      ba5a32c4649e9202733d0ea83f506b43ad689b81fd23bf802a1d805681cb70cb
    scotus-brown      https://tile.loc.gov/storage-
    services/service/ll/usrep/usrep347/usrep347483/usrep347483.pdf
                      7021074961cffdafdddac07c25c6b21c6a689d1ff118966f5b53bb3a48163ec1
    statute-79-pg437  https://www.govinfo.gov/content/pkg/STATUTE-79/pdf/STATUTE-79-Pg437.pdf
                      67439f5db51cb49a6cd2423c28c257a7fdc1112ccd6f605ab4595eebbbcc53c5
    nasa-tm-x-3181    https://ntrs.nasa.gov/api/citations/19750007805/downloads/19750007805.pdf
                      0135848ec946da34297f79fae489b54b9d9b1ace545e187d333b0dfb47a6b324
```

4 Anthropic, ***Token counting***, "Pricing and rate limits". platform.claude.com/docs/en/build-with-claude/token-counting, read 11 September 2026. The identical counts across three models are this paper's measurement, not a vendor statement.

5 GitHub, ***GitHub Flavored Markdown Spec***, version 0.29-gfm (2019-04-06), §4.10 "Tables (extension)"; colspan, rowspan and merged cell do not occur in it. github.github.com/gfm/, read 11 September 2026.

6 Google, ***Document understanding — Technical details***, Gemini API documentation. ai.google.dev/gemini-api/docs/document-processing, read 11 September 2026.

002

PAPER 002 · METHOD

The honest number moved *downwards*.

What conversion removes is the page image, a flat charge that does not care what is printed on the page, so a percentage divides it by a variable one. A published 64% was counted with another vendor's tokenizer on sixteen private PDFs nobody outside could re-run; re-counted with the vendor's own counter on a public corpus, it came to 53%: the honest number moved downwards.

PAPER · 002 PUBLISHED · 11 SEP 2026 READING · 8 MIN
LICENCE · CC BY 4.0
Canonical · digital1.foundation/articles/the-text-layer/002-the-honest-number-moved-downwards.html

§1

A flat charge divided by a variable one

Of the 2,792 tokens a native page costs in Paper 001, §3, about 1,568 are the page image, and that had to be established rather than looked up. The vendor publishes two tiers — standard, capped at 1,568 visual tokens an image, and high-resolution, for its 4.7-and-later models, capped at 4,784 — and says pages are rasterised server-side at dimensions you do not control.[1] The three models of Paper 001, §3, are 4.7-and-later models, so the table alone would place them on the higher cap, a factor near three; Claude Haiku 4.5, measured alongside them below, sits on the standard one.

The cheapest experiment closes it. One corpus document is a single page, and its whole native cost is 3,007 tokens, text and image together. A page cannot cost less than its own image, so 4,784 is impossible. Subtraction — native minus a plain `pdfplumber` text dump of the same PDF, no Markdown syntax, per page — puts the median at 1,569 tokens on Claude Opus 5 over all sixteen, range 1,309 to 1,618, and 1,568 on Claude Haiku 4.5 over the thirteen it accepted, range 1,289 to 1,616: one token apart across a supposed threefold boundary. The dump is a proxy for the vendor's own extractor, and where it over-counts, as on dense

mathematics, the residual is understated — a bias running away from 4,784, not towards it. At 1,568 the extracted text implied by the corpus is 1,224 tokens a page; the dump came to 1,239, 1.2% apart.

Which is why the reduction column in Paper 001, §3, is mostly arithmetic: what conversion removes is the page image, and the page image does not care what is printed on it. Thirteen of the sixteen save between **1,365 and 1,676 tokens a page** — inside 15% of the 1,568-token page image — while their reductions run from 38.3% to 73.1%. The three that save less save 368 to 1,141: a mathematics-heavy paper, a tax booklet, and the two-page tax form whose 11.1% is the floor. Across the corpus 1,469 tokens a page stop being billed. **Estimate an estate in tokens, not in percent** — what may not be said is that a particular estate will see 1,469.

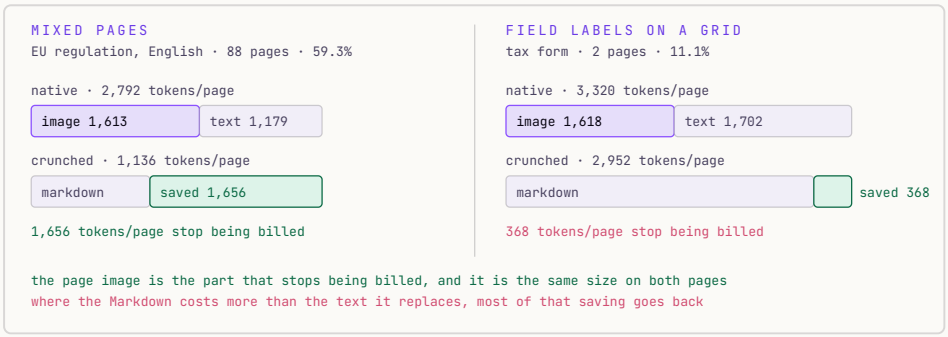


FIGURE 1 One page, two bills. Bars are tokens per page at one scale; the text figure is the measured plain-text dump, and the image figure is the remainder after it.

One pair makes the point without interpretation. The GDPR is published by the same office in English and in Hungarian — the same 88 pages, the same page image to within three tokens, 1,613 against 1,610. The Hungarian text costs 1.69× the English, the Markdown 1.71×; the reductions are 59.3% and 46.2%, and the tokens saved are nine apart. Thirteen points worse as a percentage, the same conversion in tokens: the difference is a property of the tokenizer, and it moves both sides together. The Hungarian conversion is not worse. The denominator is larger.

§2

The defect: a claim about one vendor, counted with another's tokenizer

The method in Paper 001, §3, is the corrected one. The figure crunch.md published until 7 September 2026 was 64% — 64.4%, rendered 64%. It was obtained in the ordinary way: a corpus, a tokenizer, an arithmetic. Each part was defensible alone. Two were wrong together.

DEFECT · FOUND BY RE-MEASURING WITH THE VENDOR'S OWN COUNTER, 7 SEPTEMBER 2026

A proxy tokenizer, and a corpus nobody outside could re-run

The conditions are ordinary rather than exotic. The counter was `tiktoken o200k_base`, another vendor's tokenizer, running locally. The corpus was sixteen private PDFs, 152 pages, on one laptop, and nobody outside could re-run it. The page image was read off a documentation table.

```
as published · 16 private PDFs · 152 pages · one laptop
counter   tiktoken.get_encoding("o200k_base")
native    566 text + 1,568 image = 2,134   image read off a cap
markdown  760
1 - 760 ÷ 2,134 = 64.4%   →   published as "64%"

re-counted · count_tokens · 16 public PDFs · 861 pages
1 - 1,139,102 ÷ 2,404,262 = 52.6%   →   published as "53%"
```

Two things changed at once. A proxy tokenizer does not fail loudly; it fails ***plausibly***, returning numbers of the right order. And the private corpus was far sparser — part of why its extracted text came out at 566 tokens a page against the 1,224 the public corpus implies, and how much of the rest belongs to the change of counter cannot be recovered now. A private corpus also removed the mechanism that would have caught either: somebody outside re-running it.

The native side moved in the claim's favour; the engine's own output moved further against it. The expectation had been that 64% was conservative. The honest number goes down, not up. The vendor's own documentation says its 4.7-and-later models produce roughly 30% more tokens for the same text than its earlier ones[2] — the size of the error a stale count of its own makes; no published figure relates another vendor's tokenizer to this one, and it was never a stand-in.

A constant that already contained the words

The engine also reports a ratio per conversion, and its estimate of the native cost was the document's own extracted text plus a per-page constant of 1,750 — read as the mid-point of the 1,500–3,000 tokens a page the vendor quotes. The vendor labels that range text and adds the image on top, so the document's words were charged twice.

```
as shipped · the constant read as "text + page image"
native_estimate = extracted_text_tokens + 1,750 × pages
the document's words are counted once in the first term and again inside the second

corrected · the constant is the page image alone
native_estimate = extracted_text_tokens + 1,568 × pages
one 14-page document · reported ratio 0.688 → 0.657
a private file, counted with the engine's heuristic tokenizer — illustrative only
```

The documentation is what hid it: a reader checking the line sees a mid-point, a range and an attribution, and reads softness rather than error. A softer estimate would have been forgivable. Double-counting was arithmetic.

A score that fell because detection improved

The engine's other number is a structure score out of 100, and a repair to table detection made it fall.

```
one partially-ruled table, alternate rows shaded
before   6 one-row lattices; any under 2 rows dropped → 5 lost as prose
after    stitched back into one table
         40 markdown table lines → 89 · 0 words lost · 9 gained
         needs_help       23 → 23    integrity did not move
         structure_score  82 → 78
```

The conversion had plainly got better — nine words recovered, none lost, five discarded fragments stitched back into the one table they came from. The fall sat in the tables component: a metric is defined only over what the pipeline detected, so what it missed is not scored badly — it is not scored at all. Twice in one session, the same night 64% became 53%, a number got worse because the thing measuring it had got honest.

§3 Where it runs, and what the result records

The base conversion path is deterministic by design and by documentation — the same bytes in, the same Markdown out — and it calls no model. An enhancement tier that would call one is planned and does not exist in the code these notes draw on; the property tests that would hold determinism there are a plan rather than a suite, and there are no golden files to diff against.

Across that path one module opens a socket: the fetcher that exists because a caller asked for a URL. One more can — an optional token counter, off by default, needing both a credential and a flag. Naming both is the point. OCR is where a page most obviously could leave, so it ships as a provider protocol with none enabled by default: an installed provider is listed rather than activated, and with nothing named a scan is detected, its pages are named, and the conversion refuses rather than

inventing text. Egress is a declared field — `local`, `private_network` or `third_party` — checked before the first page is rendered and recorded on the result, so a cloud provider is turned away before it authenticates.

The shipped provider is local Tesseract with seven language packs (hu, de, en, pl, ro, cs, sk), handed rasters rather than the document; a converter for legacy Office formats declares a `risk` rung the same way but is handed the file itself, and nothing in its sandbox stops that program opening a socket. The protocol cannot prove a provider opens no socket and does not pretend to: it makes the claim explicit, checkable in advance and recorded.

What conversion buys at the far end is one context, and therefore one answer. RFC 9110 is 194 pages;[3] converted and counted on Claude Haiku 4.5, 126,739 tokens — inside that model's 200K window, the smallest in the range.[4] Three corpus documents were refused outright as native uploads there: a maximum of 100 PDF pages may be provided. All three counted as Markdown, and two fit the window they had been refused from. The third does not. The window decides what fits, not a page count, and the engine caps intake at 2,000 pages — "the longest thing in the estate goes in whole" is not a sentence this paper can write.

Throughput is CPU-bound and per core — about 6 ms a page for a report with a clean text layer, about 513 ms for a dense design-heavy one, median of three runs on one production machine. Neither document is in the public corpus, so these two figures cannot be re-run from this paper. The 86× is the document, not the machine, and there is no GPU in the path: a machine full of accelerators converts an estate with them idle.

> ON THE PROVENANCE OF THIS MATERIAL
>
> The method and measurements here were developed inside ***crunch.md***, a Digital One product, and the defect in §2 was found by re-measuring that product's own published claim. The engine, `crunch-core`, and its MCP server are Apache-2.0 licensed and run offline; the hosted service is proprietary. The repository is not public yet, so that licence is a statement of intent rather than something a reader can check today. The corpus is public, and the native column in Paper 001 can be re-counted by anyone holding the manifest and a key. Deliberately absent: prices, plans and tiers, and deployment internals.

References

1 Anthropic, ***Vision — Resolution and token cost***, platform.claude.com/docs/en/build-with-claude/vision, and ***Coordinates and bounding boxes***, platform.claude.com/docs/en/build-with-claude/vision-coordinates, both read 11 September 2026. Neither states what size a PDF page is rendered at, which is why §1 measures it.

2 Anthropic, ***Token counting***, "Pricing and rate limits" and the tokenizer note for Claude 4.7 and later models. platform.claude.com/docs/en/build-with-claude/token-counting, read 11 September 2026.

3 R. Fielding, M. Nottingham, J. Reschke (eds), ***HTTP Semantics***, RFC 9110 (STD 97), June 2022, 194 pp. The artefact measured is the RFC Editor's PDF rendering, sha256 `60b30efa1048900833d1758440247fe8ac85a3134f2327388dcb24e07d814c89`. rfc-editor.org/rfc/rfc9110.pdf, read 11 September 2026.

4 Anthropic, ***Models overview***, "Compare models" — context windows of 1M tokens for Claude Fable 5.1, Claude Opus 5 and Claude Sonnet 5, and 200K for Claude Haiku 4.5. platform.claude.com/docs/en/about-claude/models/overview, read 11 September 2026.

003

PAPER 003 · RESEARCH

Beyond the *text layer*.

Converted, the text travels and the pictures do not, so anything said about a drawing is generated rather than extracted — a claim, not an extraction. One thread runs through what follows: the conversion record — including the hash this engine does not take — which continues, with the ledger of what is open, in Paper 004. This paper sets out what would have to be true before such a claim reaches a model, and what would count as a result.

PAPER · 003 PUBLISHED · 11 SEP 2026 READING · 7 MIN
LICENCE · CC BY 4.0
Canonical · digital1.foundation/articles/the-text-layer/003-beyond-the-text-layer.html

§1

What the model was shown, six months later

An organisation runs a supplier's drawing pack through a document pipeline, a model reads the result, and a compliance summary quotes a tolerance from it. Six months later the part fails, and someone who owes the organisation nothing asks what the model was shown. Not the drawing — the bytes. Where nothing was recorded, the honest answer is a filename and a date, which is a claim.

Conversion is an event, and it can be recorded like any other. What exists today is more than nothing and less than a record: the sha256 of the source bytes, the converter and its risk rung where one ran, the OCR provider, its egress and a status per page — and no page text, since an audit record carrying the document defeats its purpose.

Three things are absent. No engine-version field appears in the documented response; nothing points a derived statement back to the pixels; and the output side is unhashed — the bytes that went in are fingerprinted, the Markdown that came out is not.

THE REQUIREMENT, PRECISELY

A conversion record must let a reader holding the source bytes recompute the chain: that ***this*** source produced ***this*** Markdown under ***this*** converter; which pages left the machine; which statements were extracted and which generated. What it cannot let the reader recompute, it must name.

§2

OCR reads the glyphs, not the picture

The text a reader sees is one layer of a PDF; figures, scanned annexes, drawings, charts and photographs sit outside it. A picture becomes a figure block holding its position, carrying a caption only if one was printed beside it, never one the engine wrote. A page uploaded natively arrives as an image and its text, which is what lets a reader ask the model about its charts;[1] the converted page cannot be asked, and that is what conversion gives up.

OCR does not close the gap, because it reads glyphs rather than pictures: over a schematic it recovers the labels and leaves the schematic. The provider is handed a raster rather than the document — rasters in, lines out — and each line is capped at `inferred`, because `certain` means the format said so. The difficulty is not local to one engine: a 2026 benchmark on engineering drawings reports models reading a drawing's text far more accurately than its symbols, and calls symbol-centric understanding unsolved[2] — the authors' word, on their own preprint.

A merged cell has no representation in the Markdown emitted[3] — Paper 001 §4 gives the specification — so the cell is flattened and the table named in a warning. The tolerances citable from a drawing are those that were in a table, and not all of even those.

DEFECT · FOUND BY ADVERSARIAL REVIEW OF A CAPABILITY-FIRST REWRITE

A completeness claim survived every check that was not a reader

The conditions are ordinary rather than exotic. The proof sections of a product page were rewritten from problem-framing to capability-framing. No number changed, the claims file diffed `+45 -2` with no source removed, and two sentences came out of it newly false.

```
// the two sentences that came out newly false
"The longest thing in your estate goes in whole."
"…the tolerances you can cite are the ones that were in a table,
   and they are all of them."

// what the checks returned
type checker · test suite · claims diff → PASS

// what was true, one panel away on the same page
the window decides what fits, and intake caps at 2,000 pages
a merged cell is flattened, so a table can lose a tolerance
```

Nothing mechanical caught either, because nothing in the pipeline reads English. The second is the instructive one: it asserted a completeness no claims-file row supported, with the counter-example one panel away on the same page.

Absolutes arrive during positive rewrites, and the strongest sentence in a rewritten section is the likeliest to have overshot.

§3

A picture is a claim waiting to be made

Knowing what is in a picture is several jobs wearing one name: reading values off a chart, recovering the instruments and connections of a piping and instrumentation diagram, following a floor plan or an assembly drawing, naming what a photograph of a site shows. They are one kind of thing — every one of them is generated — and they differ sharply in difficulty and in whether anyone can score them.

Extraction is deterministic and checkable; description is generated and is a claim. The same bytes give the same Markdown — asserted in the engine's documentation rather than demonstrated by a published corpus, as Paper 004 §4 records — so a doubter can convert the file again and diff. A description cannot be re-run that way: a second run may write another sentence, and either may assert what is not in the picture. A 2018 study found captioning models "prone to 'hallucinating' objects that are not actually in a scene", and a 2023 study finds the current model class inherits it.[4]

One vendor's documentation instructs that image interpretations be verified.[5] So a description reaching the model is ***labelled as derived***, ***kept below*** `certain` — OCR lines are capped at `inferred`, and a generated sentence has no claim to sit higher — and ***carries a way back***: page, bounding box and the sha256 of the raster. It exists only where the pipeline owns the rasterisation: a natively uploaded PDF is rasterised

server-side "at dimensions you don't control", so the coordinates that come back cannot be reliably mapped onto the page. A caption so recorded is locatable, not true.

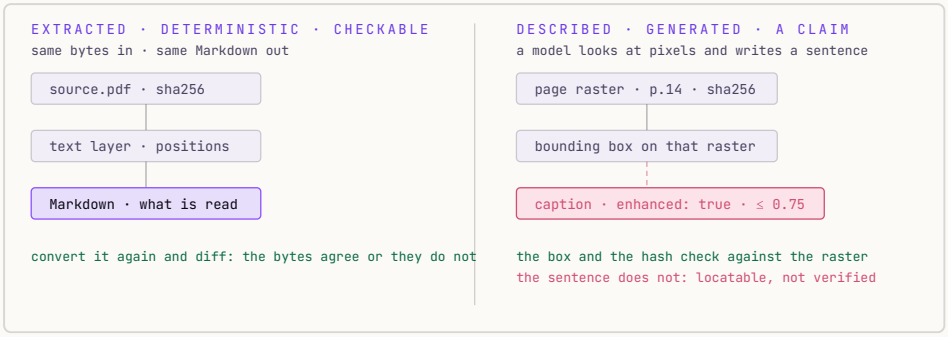


FIGURE 1 Two chains, one source. On the left every link recomputes from the bytes except the last, because the converter records no hash of its output; on the right, page, box and hash check against the raster and the sentence does not. The right-hand chain is not built.

A derived statement needs its provenance committed ***with*** it, and committed injectively: caption, page, box and hash run together by plain concatenation let whoever writes the record move the boundary between them — the defect Paper 002 of The Project Handshake found in its own leaf encoding. A verifier checks the box against the raster, not the sentence against the picture.

§4

What is being researched, and what would count as a result

The direction under research is an enhancement tier over the deterministic baseline, and none of it is built: the engine accepts the option today and ignores it. It would act only on blocks already flagged as needing help — figure alt-text, chart-to-table, reconstruction of flattened merges, semantic chunk boundaries — marking each touched block `enhanced: true`, returning the deterministic output byte-identical when the tier is off, aborting to it at a cost ceiling, and routing model calls through a gateway with an evidence log.

The difficulty is evaluation, and it is uneven:

DERIVED OUTPUT	SCOREABLE	AGAINST WHAT
Chart → data table	Yes	The chart's own data, where the document carries it; otherwise ground truth has to be made first.
Diagram → node and edge list	Yes	Graph-level ground truth, with edge accuracy as the metric; a publicly accessible P&ID benchmark so annotated exists — the first, its authors say.[6]
Flattened table → reconstruction	Yes, with a ceiling	Human layout annotation — and annotators disagree, so their agreement is the ceiling.[7]
Free-text caption	Largely not	High scores on the standard caption metrics do not imply less hallucination.[4] Free-text description resists scoring, which is the honest reason to prefer the rows above.

A piping and instrumentation diagram is drawn in a vocabulary — ANSI/ISA-5.1-2024 establishes "a uniform means of designating instruments and instrumentation systems used for measurement and control"[8] — and a vocabulary invites structured output rather than prose: node and edge lists, tag-to-value statements, each carrying its box and raster hash. A result would be a measurable margin over the baseline on a corpus with ground truth, and for an estate's own drawings no such corpus exists: the published benchmarks are other people's documents, and annotating an estate's own is work that would have to be paid for before anything could be scored. Marked open, all of it.

ON THE PROVENANCE OF THIS MATERIAL

The engine these notes draw on is ***crunch.md***, a Digital One product. Its core engine and MCP server are Apache-2.0-licensed and run offline; the hosted service is proprietary. The repository is to be published at a named milestone rather than readable today, so that licence is a statement of intent, and the engine behaviour these sections describe cannot yet be checked against its code. The references can: each points at a public page or preprint. Deliberately absent: prices, plans and tiers, and deployment internals.

References

1 Anthropic, ***PDF support***, Claude Developer Platform documentation, "How PDF support works", platform.claude.com/docs/en/build-with-claude/pdf-support, read 11 September 2026.

2 A. Kondratenko, M. Birhane, H. E. Hsain, G. Maciocci, ***AECV-Bench: Benchmarking Multimodal Models on Architectural and Engineering Drawings Understanding***, arXiv:2601.04819, January 2026. A preprint; "remains unsolved" is the authors' assessment.

3 GitHub, ***GitHub Flavored Markdown Spec***, version 0.29-gfm (2019-04-06), §4.10 "Tables (extension)", github.github.com/gfm/, read 11 September 2026.

4 A. Rohrbach, L. A. Hendricks, K. Burns, T. Darrell, K. Saenko, ***Object Hallucination in Image Captioning***, arXiv:1809.02156, EMNLP 2018; and Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, J.-R. Wen, ***Evaluating Object Hallucination in Large Vision-Language Models***, arXiv:2305.10355, EMNLP 2023.

5 Anthropic, ***Vision***, section "Limitations", platform.claude.com/docs/en/build-with-claude/vision; and ***Coordinates and bounding boxes***, platform.claude.com/docs/en/build-with-claude/vision-coordinates. Both read 11 September 2026.

6 J. M. Stürmer, M. Graumann, T. Koch, ***From Engineering Diagrams to Graphs: Digitizing P&IDs with Transformers***, arXiv:2411.13929, IEEE DSAA 2025. "First publicly accessible" is the authors' claim.

7 B. Pfitzmann, C. Auer, M. Dolfi, A. S. Nassar, P. W. J. Staar, ***DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis***, arXiv:2206.01062, SIGKDD 2022.

8 ISA, ***ANSI/ISA-5.1-2024, Instrumentation and Control – Symbols and Identification***, purpose sentence from the committee page, isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa5-1, read 11 September 2026.

004

PAPER 004 · RESEARCH

Properties, not *promises*.

Extraction from a text layer is deterministic and checkable; past it, questions are open. Where converted Markdown should live is the corpus owner's question, and **the record is the canonical Markdown and a hash over it**: every index above that pair can be dropped and rebuilt from it. Repetition across an estate is the part an archive could compress, and the archive must reproduce the canonical bytes exactly; no compression figure appears here. Egress declared as a field and written onto the conversion is **a property, not a promise**, because a policy says where pages are allowed to go and a record says where they went.

PAPER · 004 PUBLISHED · 11 SEP 2026 READING · 7 MIN

LICENCE · CC BY 4.0

Canonical · digital1.foundation/articles/the-text-layer/004-properties-not-promises.html

§1

What is open past the text layer

Everything in Paper 001 and Paper 002 describes a floor rather than a finished thing. Extraction from a text layer is ***deterministic and checkable*** — a value can be traced to the row it was printed in — while anything said about a picture is generated, and is a ***claim***. Three questions sit past that line and none is answered here: how a description of a figure might reach a model honestly, labelled as derived, kept below `certain`, carrying a way back to the pixels; where converted text should live once there is a corpus of it; and what an estate's own repetition makes compressible. They are the subject of Paper 003 and of this paper, marked in both as open research rather than as results. None of them exists in the code these notes draw on.

§2

Where the Markdown should live

Today the engine returns Markdown and a structure map and stores nothing, a decision rather than an omission: the moment an engine embeds, it needs a model, a store and a network, and the property being

protected — that documents stay inside the operator's perimeter and jurisdiction — is gone.

So the question belongs to the corpus owner: a document store, one record per source with sections addressable by heading path; a vector index, chunks mapped to embeddings; or both. They are the same kind of thing — **derived indexes over a canonical text** — since an index adds no information and dropping it deletes nothing, because it rebuilds.[1] A vector index is derived twice over, the boundaries a pipeline decision and the embedding a model's output.

The record is the canonical Markdown and a hash over it. Every index above that pair is a derived structure that can be dropped and rebuilt from it. The discipline that follows is exact: whatever an index returns must be re-findable in the record, or it has left the chain — and a store that cannot name the bytes its answer came from has quietly become the record itself. A direction, not a decision, taken jointly by those who own the corpus.

§3 Repetition is the compressible part

Estates repeat themselves — running headers and footers, boilerplate clauses, recurring tables, template pages. Inside one PDF the engine already strips headers, footers and page numbers, and because that is the one pass that deletes text, dropped lines are warned. Across documents it does not look.

Content-defined chunking sets block boundaries from the bytes rather than from position, so a repeated block gets the same boundaries wherever it sits; content addressing makes a block's identifier the hash of its contents, so duplicates coalesce and corruption shows on read.[2] A repeated clause is then stored once and referenced by hash — cacheable, and more to the point checkable. The chunk sizes and window widths in that work were tuned for source trees over slow links, so no compression figure appears here: none has been measured on a Markdown corpus.

Prompt caching has the same shape and not the same lifetime: a hit requires byte-identical segments, and the default is five minutes, an hour at additional cost.[3] A cache is a cost instrument with a horizon in minutes; an archive is a record. What repetition invites is an archive format for Markdown corpora: dictionary compression across documents, block-level deduplication, an index. The dictionary format RFC 8878 specifies travels out of band, identified by an ID the decoder checks, and that specification leaves its provisioning "being explored".[4]

WHAT MUST NEVER BE TRADED AWAY

The archive must reproduce the canonical bytes exactly. A reference is a commitment, not a summary; a deduplicated block that is "close enough" is an edit, and an edit breaks every hash above it. Content addressing is parameter-sensitive too: identical text under different hash, chunking or encoding settings gets different addresses,[5] so a format has to pin all three. None of this is built.

§4

A record says where pages went, not that the lines are right

Egress declared as a field and written onto the conversion is a property; "documents stay in Europe" is a promise. A policy says where pages are allowed to go, a record says where they went, and the two are indistinguishable while nothing goes wrong.

- **Egress as a recorded fact on the conversion, with a status for every page.** A declared field, checked before a provider authenticates and written onto the result: enforcement of a declaration, not proof of one. Paper 002 §3 gives the rungs.
- **Determinism.** The same bytes give the same Markdown — asserted in documentation rather than demonstrated by a published corpus, the property tests that would pin it planned rather than run, and table column alignment dropped on emission, denting idempotence.
- **Measurement with the counter the claim is about.** Stated against native document upload on one vendor's models; on a vendor billing a flat rate per page the sign reverses. Paper 001 carries the figures and the spread.
- **Estate economics on CPUs inside the building.** CPU-bound and per core, with no GPU in the path.

The fifth property is the record itself, and its limit belongs in the same breath. Paper 004 of The Project Handshake gives the general form: a record shows what was done, not that it was right. A conversion record shows that a page went to a local binary and how confident that binary reported itself to be, not whether the lines it returned are the lines on the page.

```
// what a conversion record carries today · fields present where each step ran
meta.source_sha256     60b30efa…d814c89   // rfc9110.pdf, 194 pages
structure.converted    {converter, converter_version, risk, endpoint}
structure.ocr          {provider, provider_version, egress, endpoint, pages.*}
                       // page text deliberately absent
blocks[*].confidence   1.0 certain · 0.75 inferred · 0.45 guessed · 0.2 salvaged
blocks[*].enhanced     // false on every block: the tier does not exist

// needed, and not there
a hash of the Markdown that came out
the engine release, which the documented response has no field for
per derived block: page · bounding box · raster sha256
an open schema anyone can implement and check
```

Those last four lines are the proposal. What does not exist is an agreed shape: an open schema for conversion records — the fields, the hashes, how a derived block points back to its raster — is the kind of open standard the Foundation being established is meant to steward. These notes propose only that it should exist.

§5

An honest ledger of what is open

Collected here, because a reader deciding what to demand is entitled to them together, each with what would settle it and who decides:

1. **Whether a generated description can be kept below `certain` and still be worth having.** Settled by a tier shipping with every touched block marked and capped, falsified if the cap is lifted. The engine's owners decide.
2. **Whether chart-to-table and diagram-to-graph can be scored on an estate's own documents.** Settled by ground truth that does not exist yet, and a margin over the baseline. The estate owner decides, and pays for the annotation.
3. **Where the canonical Markdown lives, and what is derived from it.** Falsified by an index whose results cannot be re-found in the record. The corpus owner decides.
4. **Whether chunking and dictionary compression pay on Markdown corpora.** Settled by measurement on a named corpus, of which no figure exists, and falsified if body repetition proves smaller than the furniture already stripped. Whoever runs it decides.
5. **Whether a declared egress can be enforced below the engine.** Settled at the operating-system boundary, where a service unit can deny a converter the network; the engine declares and records, and cannot prove. The operator decides.

6 **Whether an open schema can be written so that two implementations agree.** Settled by two independent implementations checking each other's records, and held by a body that owns no product — the Foundation being established, if those establishing it so decide.

Nothing above is committed: the decisions are joint, and these notes take none of them.

> ON THE PROVENANCE OF THIS MATERIAL
>
> The engine these notes draw on is ***crunch.md***, a Digital One product. Its core engine and MCP server are Apache-2.0-licensed and run offline; the hosted service is proprietary, and the repository is to be published at a named milestone rather than readable today, so that licence is a statement of intent rather than something a reader can check today. What is published here is method and open questions, not a product argument. Deliberately absent: prices, plans and tiers, and deployment internals.

References

1 M. Kleppmann, ***Turning the database inside-out with Apache Samza***, 4 March 2015, martin.kleppmann.com/2015/03/04/turning-the-database-inside-out.html, read 11 September 2026.

2 A. Muthitacharoen, B. Chen, D. Mazières, ***A Low-bandwidth Network File System***, §3.1, pdos.csail.mit.edu/papers/lbfs:sosp01/lbfs.pdf; and S. Quinlan, S. Dorward, ***Venti: a new approach to archival storage***, FAST 2002, usenix.org/legacy/publications/library/proceedings/fast02/quinlan/quinlan.pdf. Both read 11 September 2026.

3 Anthropic, ***Prompt caching***, Claude Developer Platform documentation, platform.claude.com/docs/en/build-with-claude/prompt-caching, read 11 September 2026.

4 Y. Collet, M. Kucherawy (ed.), ***Zstandard Compression and the 'application/zstd' Media Type***, RFC 8878, February 2021, Informational, §5 and §6, rfc-editor.org/rfc/rfc8878, read 11 September 2026.

5 ***Content Identifiers (CIDs)***, IPFS documentation, "Content addressing": the identifier depends on the hash function, the chunking and the codec as well as the bytes, docs.ipfs.tech/concepts/content-addressing/, read 11 September 2026.

END

THE TEXT LAYER · COLOPHON

Colophon

A

The record

This document is generated from the published pages, which are the record. Where this file and a page disagree, the page is right and this file is stale. Each source below is named with the SHA-256 of the file this document was built from, so a reader can check that what they are holding is what was published; the generator fetches each live page as it builds and reports any that has moved since.

INDEX digital1.foundation/articles/the-text-layer/

001 digital1.foundation/articles/the-text-layer/001-what-the-model-reads.html
sha256 f4174775cf4acd4d44d62c21a5fae14f7bea541846c4ab1ae787e7b25e9e3c49

002 digital1.foundation/articles/the-text-layer/002-the-honest-number-moved-downwards.html
sha256 65e89911d6163057cff937e8adf4470934bec779717983c794862a1ec07381fe

003 digital1.foundation/articles/the-text-layer/003-beyond-the-text-layer.html
sha256 ed1ebdb145c03a25376d828b5a899dda61764b77293bc5c995e98c831398081b

004 digital1.foundation/articles/the-text-layer/004-properties-not-promises.html
sha256 e278bd1289d26c28b9bb5b32b9f78b3e2cb96014a8e769e97f394004e1455378

B

Licence

Every paper in this document is published under the Creative Commons Attribution 4.0 International licence, creativecommons.org/licenses/by/4.0/. You may copy, redistribute and adapt it for any purpose, including commercially, provided you credit ***Stichting Digital One Foundation i.o.*** and link to the page cited above.

C

How it is set

In the six faces the site itself carries and embeds: Fraunces 300 and 600 for titles and the emphasised word, Manrope 400 and 500 for the text, JetBrains Mono 400 for the ledger's own labels, code and running matter. Those six are Latin subsets and do not carry every character the papers use — → ≤ are absent from them. Nothing is drawn in their place and quietly dropped from the text: each is set in a named system face. Every character in every paper was extracted back out of this file and checked before it was written, so the whole document copies, searches and cites as it reads.

Figures are the papers' own inline drawings, re-toned from the site's night ground to this page by a one-to-one swap of the closed house palette, so green still means it holds and rose still means it breaks — and the two are separated in weight of grey as well as in hue, so a photocopy keeps the distinction. The rail down the left of each page is the ledger from the Foundation's homepage; a filled dot marks each numbered section, and the § number sits in the gutter, outside the measure.

The sheet is A4 as the page asks for it, which Chromium rounds to 594.96 × 841.92 pt when it writes the file — 0.1 mm narrower and 0.03 mm taller than the 595.28 × 841.89 pt of the standard. No printer will show it; a document that prints its own hashes should say it anyway.

© MMXXVI Stichting Digital One Foundation i.o.
Amsterdam, the Netherlands

Published under CC BY 4.0.
No public-benefit (ANBI) status is claimed at this time.

digital1.foundation
The Text Layer