QUBER RESEARCH — LANDING.AI ADE

DPT-3

What changes from DPT-2, whether ADE Section survives, what it costs, and a recommendation

Landing.ai released DPT-3, the third generation of its Document Pre-trained Transformer model family, to general availability on July 24, 2026, behind a new Parse v2 API. ADE is Agentic Document Extraction, Landing.ai's document platform; our `quber ade` integration runs on its existing v1 API with dpt-2. This document states what changes from DPT-2, answers whether the Section capability survives, describes the new response format and pricing, and closes with a recommendation. It assumes familiarity with our ADE client at the level of `src/quber/providers/landing/client.py`.

Engineering Reference · Generated 2026-07-31
Primary sources: `docs.landing.ai` · QUE-321

Contents

Audience and Scope

Written for engineers working on quber's document pipeline. The reader knows what ADE Parse returns today under dpt-2 — chunks with bounding boxes plus markdown — and how `quber ade` submits jobs. Everything here is sourced from Landing.ai's official documentation as of July 30, 2026. Claims that could not be verified against a primary source are labeled as such. Source URLs are collected in section 8 and cited by number. One naming note. The capability we informally call Sections is officially named Section, singular: the model family is `section-*` and the SDK method is `client.section()`. This document writes the product name as ADE Section throughout, so it never collides with the document's own numbered sections. A cross-reference such as "section 4" always points at this document.

1 The Findings That Matter

quber uses the ADE platform for two things. Parse turns a filing into chunks our consumers read. ADE Section groups a parse into logical document sections. Five findings decide what DPT-3 means for both, and each is expanded in its own section below.

- **ADE Section does not exist on DPT-3.** The capability runs on its own model family, `section-*`, and remains a preview available only on the existing v1 API. The v2 markdown also omits the anchor tags that ADE Section reads to map its output back to chunks, so v2 output cannot feed it. Nothing in either changelog announces a plan to bring it to v2. The evidence is laid out in section 4.
- **Chunks are gone from Parse v2.** The response replaces the chunk list with one markdown string, a tree describing the document's structure, and a map of bounding boxes. Together with the first finding, both of the capabilities we use are affected: parsing is restructured and ADE Section is absent. The new response is described in section 2.2.
- **No accuracy evidence has been published for DPT-3.** Landing.ai calls the release "a redesign, not just a model update" but publishes no benchmark for it. The widely cited 99.16% DocVQA figure belongs to DPT-2. What is and is not claimed is covered in section 3.
- **Pricing switched from flat to output-scaled.** DPT-2 costs 3 credits per page, flat. DPT-3 costs 1 credit per page plus 0.5 credits per 1,000 output characters, so dense pages cost more than sparse ones. For text-dense financial filings this can exceed the DPT-2 price. The arithmetic is in section 6.
- **The release is a week old, and its response format settled only a week before it.** GA came on July 24, 2026, after breaking response-format changes on July 17. Any dpt-3 preview output we captured before mid-July follows a schema that no longer exists. The timeline is in section 2.1.

> **THE RECOMMENDATION IN ONE LINE**
>
> Stay on dpt-2 for production, measure DPT-3 on our own filings before trusting either its accuracy or its price, and re-evaluate the moment ADE Section appears on v2 or v1 gets a deprecation date. The full argument is in section 7.

2 What DPT-3 Is

DPT-3 is the model family behind Landing.ai's new Parse v2 API. It reads a PDF or an image and returns the document as markdown, plus a tree describing the document's structure and a set of bounding boxes locating each part of it on the page. DPT-3 is not an upgraded model dropped into the existing endpoint. Parse v2 is a different host, a different endpoint, and a different response format. The official overview describes it as "a complete rethink of document parsing for the era of AI agents and agentic workflows" and "a redesign, not just a model update" [1].

Three things changed about how a caller reaches the service:

- **Host** — `api.va.landing.ai` (v1) became `api.ade.landing.ai` (v2) [2, 3].
- **Endpoint** — `/v1/ade/parse` became `/v2/parse`, with asynchronous equivalents under Parse Jobs [2].
- **Inputs** — v2 accepts PDFs and images only. Spreadsheets (XLSX, CSV) still require Parse v1 [1].

2.1 Release timeline

DPT-3 went from public preview to general availability in eight weeks. One week before GA, its response format took changes that break existing callers. The dates below are from the official changelog [4].

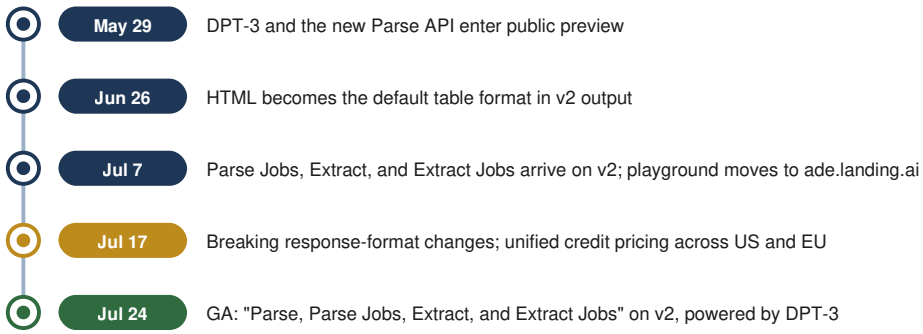

Figure 1 — DPT-3 release timeline, from the official changelog [4].

Preview output is stale. Any dpt-3 responses we saved before July 17, 2026 use a response schema that has since changed in ways that break existing callers. Nothing captured during the preview should be used as a reference for integration work.

2.2 The v2 response

The v2 response splits content, structure, and position into three separate parts. v1 combined all three into the chunk list [1, 5]. The three parts are:

- **Markdown** — one CommonMark 0.31.2 string for the whole document in reading order. Math is LaTeX (`$...$`, `$$...$$`), page boundaries are `<!-- PAGE BREAK -->` comments, and the job id is embedded as a `<!-- doc_id=... -->` comment.
- **Structure tree** — a hierarchy from the document root to pages to blocks to table cells to visual lines. Each block has a semantic id of the form `<type>-<index>` and points into the markdown by a range of Unicode code-point offsets.
- **Grounding map** — bounding boxes keyed by block id, normalized to 0–1, with keys `xmin`/`ymin`/`xmax`/`ymax`. A separate `atomic_grounding` entry gives one box per rendered line inside a block, so a citation can point at a single sentence rather than a whole paragraph.

Figure 2 puts the two generations side by side. The v1 column ends in ADE Section. The v2 column stops short of it, because ADE Section reads anchor tags that v2 markdown does not contain. The full evidence is in section 4.

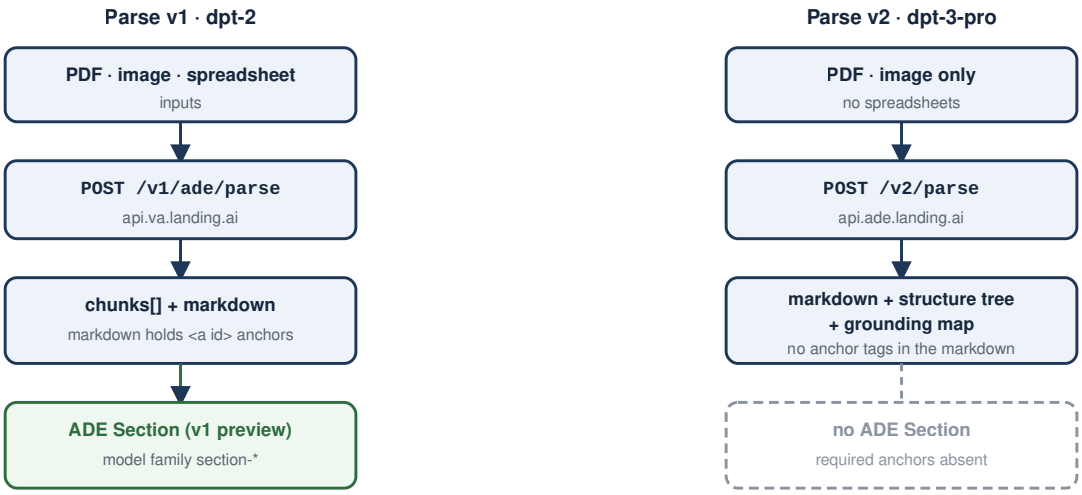

Figure 2 — The two API generations. ADE Section reads v1 markdown by its anchor tags. v2 markdown has none, so only the v1 path reaches ADE Section.

3 Features and Improvements

The documented gains are in how the output is structured and in what a caller can configure, not in any measured accuracy figure. The additions [1, 5, 6]:

- **Semantic block types** — every block is typed as one of `text`, `table`, `table_cell`, `figure`, `marginalia`, `attestation`, `logo`, `card`, or `scan_code`. Attestation (signatures and stamps) and marginalia (text in the page margins) are new block types in v2.
- **Line-level grounding** — `atomic_grounding` gives one bounding box per rendered line inside a block. v1 locates content only down to the chunk. v2 can locate a single sentence.
- **Configurable output** — three settings. `blocks.table.format` selects HTML (the default) or markdown tables. A per-block-type `markdown` toggle keeps figures or marginalia out of the markdown. `inline_markdown` attaches a markdown slice to each structure node.
- **Typed extraction** — generally available. Extract v2 takes a Pydantic model or JSON schema plus parsed markdown and returns values matching that schema: `client.v2.extract(schema=MyModel, markdown=parsed.markdown)` [7].

> **NO PUBLISHED ACCURACY EVIDENCE**
>
> No official DPT-3 benchmark exists anywhere in the documentation, blog, or press material we could reach. The widely cited 99.16% figure is DPT-2's result on DocVQA, a public benchmark of question answering over document images: 5,286 of 5,331 on the validation split, published September 2025 [8]. Landing.ai's own developer announcement for DPT-3 returns HTTP 404. Every claim sourced to it survives only as a search-index snippet and is unverified, including "faster and less expensive for most customers" [9]. Until Landing.ai publishes numbers or we run our own corpus through it, DPT-3's accuracy relative to DPT-2 is unknown.

Only one DPT-3 tier is documented, with dated snapshots. The lineup across all three DPT generations:

Model string	Status
`dpt-3-pro-latest`	v2 default when `model` is omitted [6]

Model string	Status
`dpt-3-pro`, `dpt-3-pro-20260710`	Alias and dated snapshot of the same tier [6]
`dpt-2-latest` and four dated dpt-2 snapshots	Unchanged, fully supported on v1 [10]
`dpt-2-mini`	Deprecated [10]
DPT-1 line	Shut down March 31, 2026 [4]

The `-pro` infix implies a tier structure, but no non-pro DPT-3 variant is documented anywhere. Whether one is coming is unknown.

4 ADE Section on DPT-3

ADE Section does not exist on DPT-3. The reason is a concrete input mismatch, not simply a feature Landing.ai has yet to ship. Four independent pieces of evidence, each checkable against the cited page:

- **The v2 documentation tree has no ADE Section page.** All four of its pages (guide, models, response, troubleshooting) live under the v1 `/ade/` tree; the `/dpt3/` tree has none [11].
- **ADE Section runs its own model family, not DPT.** Valid model values are `section-latest` and `section-20260406`; DPT-3 is never mentioned as an option [12].
- **v2 output cannot be fed to ADE Section as-is.** ADE Section requires input markdown containing `<a id="..."></a>` reference anchors, which it uses to map its sections back to chunks. Parse v2 markdown contains no anchor tags. Its only embedded markers are page-break and doc-id comments [13, 5].
- **ADE Section itself is still a preview.** Its page warns: "This feature is still in development and may not return accurate results. Do not use this feature in production environments." It entered public preview on April 22, 2026 and has not reached GA [13, 14].

Neither changelog announces any plan to bring ADE Section to v2. The same holds for Split and Classify, which are also v1-only previews [4, 14]. ADE Section is not deprecated either. `client.section()` still works against v1 with dpt-2 parses.

> **One sourcing caveat.** Our fetch of the migration guide came back as a summary rather than the page text, and the summary says ADE Section "has no direct v2 equivalent". The page's own wording could not be retrieved to confirm the phrase. The four points above establish the absence independently. Only Landing.ai's stated intent about the capability's future rests on that summary.

Migrating today would cost us half of what we use the platform for. quber uses two DPT-2 capabilities, parsing and ADE Section. DPT-3 restructures the first and does not offer the second.

5 API Integration and Migration

The SDK we already ship, `landingai-ade` on Python 3.9+, serves both generations. The v1 methods sit on the client root: `client.parse()`, `client.split()`, `client.classify()`, and `client.section()`. The v2 methods sit under a `v2` namespace: `client.v2.parse()`, `client.v2.extract()`, `client.v2.parse_jobs.*`, and `client.v2.extract_jobs.*` [15, 16]. Landing.ai has announced no deprecation of the v1 methods.

5.1 Breaking changes from v1 to v2

The changes listed in the migration guide, and what each one requires of a consumer [3]:

Change	Consequence for a v1 consumer
`chunks` removed	The guide's words: "Stop reading `chunks`. Walk `structure.children` (pages) and their `children` (blocks) instead." Every consumer that reads chunks must be rewritten.
Bounding-box keys renamed	`left/top/right/bottom` become `xmin/ymin/xmax/ymax`, normalized 0–1.
Markdown consolidated	One string per document; blocks point into it by Unicode code-point offsets. Pages are 1-indexed.
`split` field removed	Page selection moves to `options.pages`.
`custom_prompts` removed	Per-chunk-type instructions (e.g. figure descriptions) have no v2 equivalent.
Password support dropped	v2 does not accept password-protected files. The guide moves the field to `options.password`, but the capability exists only on v1, which gained it in March 2026.
Spreadsheets dropped	XLSX and CSV inputs require Parse v1.

The operational limits changed too. Synchronous v2 parse caps at 50 MiB and 100 pages per PDF, while Parse Jobs allows 1 GiB PDFs and 6,000 pages. Throughput is now allocated as pages per hour by plan tier rather than requests per minute. Extract v2 does not reject requests over the rate limit: it accepts them with a 202 status and completes them at its own pace [17].

5.2 Where our client stands

Our client was written against the preview and already routes by model line. `is_v2_model()` in `src/quber/providers/landing/client.py` sends any `dpt-3*` name through `client.v2.parse_jobs` and everything else through v1. It drops `custom_prompts` for v2. It reads the completed payload from `data` on v1 and from `result` on v2. The GA model strings, `dpt-3-pro-latest` and its snapshots, all match the `dpt-3` prefix, so the routing holds without change. The API key comes from our settings as `ADE_API_KEY` and is passed to the SDK directly.

The client can fetch a v2 response, but nothing downstream can use one. Every consumer of `parse_document()` reads chunks, a format v2 no longer produces. A real migration is therefore not a client change but a rewrite of those consumers against the structure tree.

The failure guard is unverified on v2. Our client refuses partial parses by reading `failed_pages` from the response metadata. That guard was written against v1 responses. Whether v2 metadata reports failed pages under the same key is unverified. Confirm it on a real v2 response before relying on the guard there.

6 Cost

Credits cost $0.01 each, the same in the US and the EU since the July 17 unified-pricing change [18]. The two generations use different pricing formulas, not just different rates [19, 20]:

Operation	Credits
Parse v1 (dpt-2)	3 per page, flat. +1 per page with zero-data-retention. Spreadsheets: 1 per sheet + 3 per embedded image.
Parse v2 (dpt-3)	1 per page + 0.5 per 1,000 output characters. Every Unicode code point in the returned markdown counts, formatting included. The response reports `output_markdown_chars` so a bill can be verified.
ADE Section (v1)	1 per 5,000 input characters + 1 per 1,000 output characters.
Classify (v1)	0.5 per page.

Under v2, the cost of a page depends on how much text that page produces. The official wording: "each page's credit consumption scales with the volume of content returned, so simpler pages consume fewer credits and denser pages consume more" [20].

> **THE BREAK-EVEN POINT — DERIVED, NOT DOCUMENTED**
>
> v2 matches v1's flat price at 4,000 output characters per page. Setting the two formulas equal gives that figure: $3 = 1 + 0.5 \times (\text{chars}/1{,}000)$. Below 4,000 characters v2 is cheaper. Above it, v2 costs more. Two things push our documents toward the expensive side: financial filings are dense, and v2 emits tables as HTML by default, which inflates character counts with markup. The "25–40% cheaper" claim is Landing.ai's own, made in the developer announcement that now returns HTTP 404. It survives only as search-index snippets, and it assumes mixed workloads [9]. For our corpus, the direction of the price change is unknown until we measure `output_markdown_chars` on real filings.

Three plans are offered [18]:

- **Explore** — pay-as-you-go, with 1,000 free credits that expire 90 days after account creation.

- **Team** — from $250/month in credit packs. This tier makes zero-data-retention available, billed on v1 parses as the +1 credit per page in the table above, and adds HIPAA support with a signed Business Associate Agreement.
- **Enterprise** — custom pricing.

Prepaid credits expire one year after purchase [18].

7 Assessment

Stay on dpt-2 for production. Nothing in this release argues for moving, and the lost capability and the price risk argue against it. The four reasons, in order of weight:

- **DPT-3 removes one capability we value and restructures the other.** ADE Section is absent, with no announced path to v2 (section 4). Chunk-based parsing, the format our consumers read, is replaced by the structure tree (section 5.1). A migration would cost us a rewrite and lose us a capability.
- **There is no accuracy evidence to set against that loss.** A "complete rethink" with no published benchmark is a claim, not a result (section 3). DPT-2's DocVQA figure remains the only measured number Landing.ai has published for either family.
- **The price may move against us.** Output-scaled pricing favors sparse documents. Ours are dense, and the HTML table default inflates character counts most on table-heavy pages, which are the pages our extraction exists to read (section 6).
- **Nothing forces a move.** v1 is fully supported, dpt-2 snapshots continue, and no deprecation has been announced (section 5). Waiting costs us nothing.

DPT-3 does get three things right, which is why it is worth re-checking later:

- **Line-level grounding is the feature our provenance work wants.** quber already traces each extracted table value back to the place it is printed on the page. `atomic_grounding` would let an ADE-side citation point at one printed line instead of one chunk.
- **The structure tree is a cleaner contract than chunks.** Typed blocks with stable ids, offsets into one markdown string, and a separate grounding map is what we would design ourselves.
- **Extract v2 takes a Pydantic schema natively**, which matches our stack exactly.

7.1 Recommended next steps

- **Measure, on our documents.** Run two or three filings from our corpus through both generations. Compare table fidelity by eye and cost by `output_markdown_chars` against the 4,000-character break-even. This is a day of work, and it settles the one unknown in section 6: which side of the break-even our filings fall on.
- **Verify the v2 failure contract** before any production use: confirm whether failed pages are reported, and under what key (section 5.2).

- **Watch two changelog events**, either of which reopens this assessment. ADE Section, or a successor, appearing on v2 would make DPT-3 attractive. A deprecation date on Parse v1 would make migration mandatory on a schedule.

8 Sources

All sources fetched July 30, 2026. Bracketed numbers in the text refer to this table.

#	Source
1	DPT-3 overview — `docs.landing.ai/dpt3/overview.md`
2	Parse v2 reference — `docs.landing.ai/dpt3/parse.md`
3	Migration guide — `docs.landing.ai/dpt3/migration-guide.md`
4	DPT-3 changelog — `docs.landing.ai/dpt3/changelog.md`
5	Parse v2 response format — `docs.landing.ai/dpt3/parse-response.md`
6	Parse v2 input options — `docs.landing.ai/dpt3/parse-input.md`
7	Extract v2 — `docs.landing.ai/dpt3/extract.md`
8	DPT-2 DocVQA result — `landing.ai/blog/superhuman-on-docvqa-...` and `github.com/landing-ai/ade-docvqa-benchmark`
9	DPT-3 developer announcement — `landing.ai/blog/dpt3-parse-announcement-for-developers`. **Returns HTTP 404;** claims sourced only to it are unverified search-snippet material.
10	Parse v1 model list — `docs.landing.ai/ade/ade-parse-models`
11	Documentation index — `docs.landing.ai/llms.txt`
12	ADE Section models — `docs.landing.ai/ade/ade-section-models.md`
13	ADE Section guide — `docs.landing.ai/ade/ade-section.md`
14	ADE v1 changelog — `docs.landing.ai/ade/ade-changelog.md`
15	Python SDK guide — `docs.landing.ai/dpt3/ade-python.md`
16	SDK repository — `github.com/landing-ai/ade-python`
17	Rate limits — `docs.landing.ai/dpt3/rate-limits.md`
18	Pricing and plans — `docs.landing.ai/ade/ade-pricing.md`

#	Source
19	v1 credit consumption — `docs.landing.ai/ade/ade-credit-consumption.md`
20	v2 credit consumption — `docs.landing.ai/dpt3/credit-consumption.md`

DPT-2 press release. The only item under `landing.ai/news` remains the September 30, 2025 DPT-2 announcement. No comparable press release exists for DPT-3.