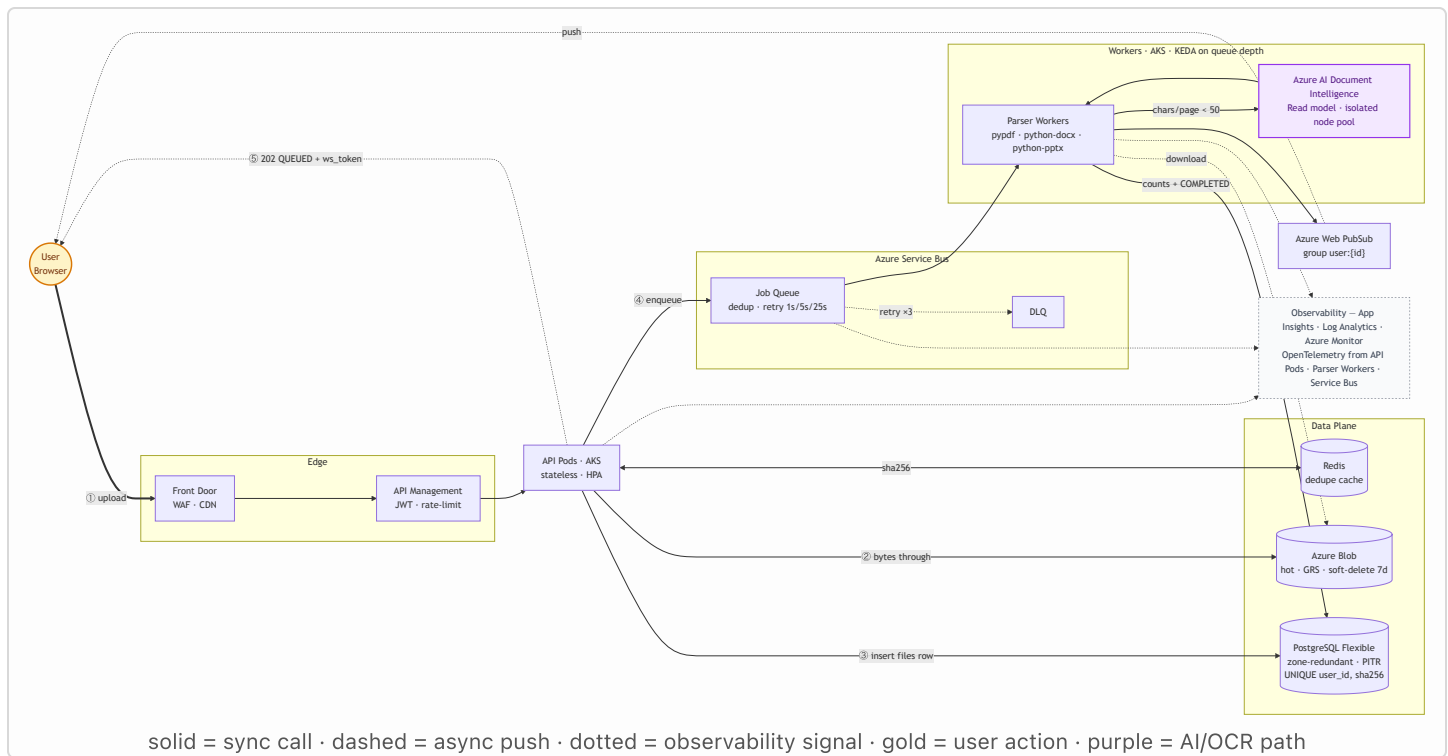


Document Analytics Service — HLD

Authenticated users upload PDF / DOCX / PPTX (≤100 KB) and see per-file page / word / character counts in a per-user dashboard.

Throughput	10K/min burst · 100K/day
File size	≤100 KB
Availability	99.9%
RPO / RTO	15 min / 1 h

Architecture



Request walkthrough.

1. Client uploads to **Front Door + WAF**; **API Management** validates the Entra JWT and applies per-user rate limits.
2. API pod streams and hashes the body in one pass, checks **Redis** then the Postgres **UNIQUE(user_id, sha256)** constraint; if new, writes to **Blob** and inserts the **files** row; enqueues on **Service Bus**; returns `202 {status: QUEUED, file_id, ws_token}` in <200 ms.
3. KEDA-scaled **Parser Workers** pull the job and run the deterministic parser; PDFs with text density <50 chars/page branch to **Document Intelligence** on an isolated node pool. Worker also renders a first-page thumbnail to Blob; `thumbnail_path` is written to Postgres alongside the counts.
4. Worker writes counts + `status=COMPLETED` to **Postgres** and publishes to **Web PubSub** group `user:{user_id}`.
5. Browser joins Web PubSub group `user:{user_id}` using the scoped `ws_token` from the 202 response, receives the push, and updates the row live; dashboard reads Postgres for lists and CDN-cached thumbnails via Front Door.

Key Decisions

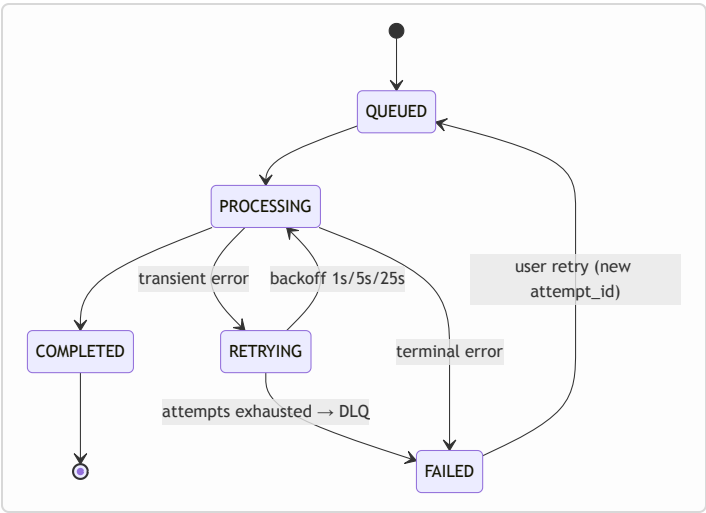
- **Async with push, not sync.** Burst absorption for 10K/min without pre-warming compute; OCR tail latency cannot stall the upload API.
- **Bytes through the API, not SAS presigned upload.** 100 KB files — a SAS handshake adds a round-trip and moves malware scanning, MIME sniffing, and dedupe off the golden path for zero throughput gain.
- **Service Bus, not Event Hubs / RabbitMQ / Celery.** Event Hubs is a partitioned append log for telemetry — wrong shape for per-message ack, DLQ, and scheduled retry; RabbitMQ and Celery burn managed-ops budget that belongs on product.
- **Deterministic parsers primary; AI only for OCR.** Counting text is a solved library problem — fast, free, deterministic. Document Intelligence earns its place only when the input is pixels, not text.
- **Postgres UNIQUE(user_id, sha256) = dedupe truth; Redis = cache.** Redis handles the 99% hot path; the DB constraint makes a duplicate-upload race impossible instead of unlikely.

Format Handling

Format	Page =	Word source	Char source	Parser	Fallback
PDF (text)	physical page	extractable text	extractable text	pypdf	—
PDF (scanned)	physical page	OCR text	OCR text	Document Intelligence	AI path
DOCX	words / 275 (approx)	body text	body text	python-docx	LibreOffice render (roadmap)
PPTX	slide count	shape text	shape text	python-pptx	—

Character = Unicode code points (not bytes, not grapheme clusters). Word/char scope: body text only — headers, footers, and speaker notes excluded in v1.

State Machine & Failure Codes



Code	Retriable	Meaning
FILE_CORRUPTED	No	Bytes don't parse as claimed format
FILE_PASSWORD_PROTECTED	No	Encrypted, extraction impossible
FILE_T00_LARGE	No	Exceeds 100 KB (caught at API)
FILE_FORMAT_UNSUPPORTED	No	Magic bytes don't match extension
PARSER_TIMEOUT	Yes	Retry with extended time budget
PARSER_OOM	Yes	Retry on larger pod
DOWNSTREAM_UNAVAILABLE	Yes	Blob / DB / Doc Intelligence transient
UNKNOWN	Yes (3x)	Any other error, then terminal

Data Model

```
CREATE TABLE files (
  file_id          UUID PRIMARY KEY,
  user_id          UUID NOT NULL,
  filename         VARCHAR(512) NOT NULL,
  size_bytes       INT          NOT NULL,
  sha256           CHAR(64)     NOT NULL,
  mime_type        VARCHAR(64)  NOT NULL,
  blob_path        VARCHAR(1024) NOT NULL,
  thumbnail_path   VARCHAR(1024),
  status           VARCHAR(16)  NOT NULL DEFAULT 'QUEUED',
  page_count       INT,
  word_count       INT,
  char_count       INT,
  parser_used      VARCHAR(32),  -- pypdf | python-docx | python-pptx | doc-intelligence
  failure_code      VARCHAR(32),
  failure_message  TEXT,
  created_at       TIMESTAMPTZ  NOT NULL DEFAULT NOW(),
  updated_at       TIMESTAMPTZ  NOT NULL DEFAULT NOW()
);

CREATE UNIQUE INDEX files_dedupe      ON files (user_id, sha256);
CREATE INDEX files_user_recent      ON files (user_id, created_at DESC);
CREATE INDEX files_worker_sweep ON files (status)
WHERE status IN ('QUEUED', 'PROCESSING', 'RETRYING');
```

Idempotency. POST /files accepts an Idempotency-Key header (client-generated UUID); API stores it with the request and returns the original 202 on retry. The UNIQUE(user_id, sha256) constraint is the second line of defense.

Observability & SLOs

- **Signals.** RED metrics per service; distributed traces via OpenTelemetry SDK → Application Insights; structured logs → Log Analytics; correlation ID propagated Front Door → worker.
- **Alerts** (Azure Monitor → PagerDuty): DLQ depth > 50, parse-fail rate > 1% (5 m window), p95 result latency > 5 s, upload API 5xx > 0.1%.
- **DLQ triage.** Depth > 50 pages ops; poison messages are triaged daily, root-caused, fixed at code level, and re-queued via an admin endpoint. Files stay in FAILED with the original failure_code until successful re-processing.
- **SLOs.** Upload API p95 < 200 ms · result available p95 < 5 s · 99.9% availability · RPO 15 min (Blob GRS + Postgres PITR) · RTO 1 h.
- **Rate limits.** APIM policy: 100 uploads/min/user, 5000/day/user; bulk imports go through a separate batch endpoint (roadmap).

Caveats & Trade-offs

- **No legacy .doc / .ppt binary formats.** Requires LibreOffice — deferred; users get a clean FILE_FORMAT_UNSUPPORTED.
- **DOCX page count is words/275.** True pagination needs a rendering engine; LibreOffice-headless is the roadmap fix when accuracy matters.
- **Office Online preview sends SAS URL to Microsoft rendering.** Fine for general docs; regulated tenants get self-hosted LibreOffice → PDF instead.
- **Single-region v1.** Active-passive across paired regions is the natural next step; Blob GRS and Postgres geo-replica already lay the groundwork.
- **Enrichment (summary / topics / entities via Azure OpenAI) is roadmap.** Architecture accommodates it as a second worker deployment fanning off Service Bus — no re-plumbing.
- **Malware scan is roadmap.** Azure Defender for Storage on the Blob container is planned for v1.1; a FILE_MALICIOUS terminal code slot is reserved in the failure taxonomy.
- **OCR-branch heuristic is not universal.** <50 chars/page misfires on scanned PDFs with partial embedded OCR; a shadow-parse metric in production tunes the threshold.
- **OCR cost.** ~\$1.50 per 1,000 pages (Document Intelligence Read); at a 5% OCR-fallback rate on 100K/day, ~\$225/month — detailed in the cost model doc.

OpenAPI spec, cost model, CI/CD, and multi-region topology available in follow-up docs.