Agentic Document Extraction

Agentic Document Extraction is a **scalable, secure, and layout-agnostic** document processing system. It works across **financial, healthcare, logistics, archival, and consumer product domains**, supports **schema-driven extraction**, integrates with **Snowflake**, and provides both **developer-friendly APIs** and **business-user-ready UIs**

1. Core Offering

- **What it is**:
    - An **API-first service** (REST + Python SDK) for turning **documents & images** into structured information.
    - Outputs both **Markdown** and **JSON**.
    - Provides **visual grounding** → bounding boxes, chunk IDs, and coordinates for every extracted field.
- **Architecture approach**:
    - Layout-agnostic, computer-vision-first (treats documents as images with text, rather than plain OCR).
    - No templates required.
    - Performs well on zero-shot extraction across new/unseen layouts.

---

2. Key Features & Strengths

- **Accuracy**: Validated across financial, healthcare, and logistics industries; reported high performance on difficult docs.
- **Speed**: Can process **hundreds or thousands of pages per minute**.
- **Security & Compliance**:
    - SOC2 and HIPAA compliant; GDPR/ZDR frameworks.
    - **Zero Data Retention (ZDR)** option: data processed in-memory, discarded immediately.

    - **Snowflake-native deployment**: ensures no data movement outside client's environment; integrates with Cortex Search and notebooks.
- **Enrichment**: Adds metadata for downstream use:
    - RAG applications.
    - Schema-based structured field extraction.
    - Figure/chart interpretation and analysis.
- **Economics**: Replaces manual extraction, lowers cost per document.

---

3. Supported Document Types

- **Core**: Invoices (most common use case).
- **Financial**: Loan applications, proof of income, bank/account statements, W2/tax forms, personal financial statements, KYC docs.
- **Healthcare**: Lab results, prescriptions, prior authorizations, medical directives, patient files.
- **Shipping/Logistics**: Bills of lading, customs declarations, certificates of origin, shipping logs.
- **Archival/Historical**: Handwritten, degraded, multilingual documents, property records.
- **Other**:
    - Product labels (extract "organic", "USDA inspected", etc.).
    - Technical/scientific documents (complex layouts, tables, figures, charts).
    - Instructional materials (e.g., IKEA assembly diagrams).

---

4. Demonstrations

A. Visual Playground (Web UI)

- **Parse**: Outputs raw chunks (Markdown + JSON).
- **Extract**: Schema-based structured fields.
    - Auto-suggested schema.

    - Upload custom JSON schema.
- **Chat** (preview): RAG-style interaction with visual grounding.

B. Example Walkthroughs

1. **Optometry prescriptions**: Same data, different layouts; schema extraction demo.
2. **Invoices & Tax Docs**: Consistent field extraction across varied formats.
3. **Medical Directives**: Checkbox detection (life-critical choices).
4. **Shipping Forms**: Extract origins, destinations, product details.
5. **Complex Layouts**: Multi-column text interrupted by figures → handled correctly.
6. **Tables**: Supports merged cells, subtotals, hierarchical structures.
7. **Charts & Figures**: Interprets meaning (e.g., bar charts, Porter's 5 forces, annotated medical images, IKEA assembly diagrams).
8. **Historical Records**: Extracts handwriting, names, occupations, even from degraded scans.
9. **Product Labels (Images)**: Structured extraction of certifications & nutrition claims.

C. Code-Based Demos

1. **Jupyter Notebook Demo** (Python library)
    - Batch parse 6 product images.
    - Custom schema definition → nutrition claims extracted as booleans.
    - Outputs JSON → Pandas DataFrame → Excel.
    - Visual grounding links each field back to original image region.
2. **Streamlit App Demo**
    - Batch process documents in a folder.
    - Simple front-end UI for non-technical users (e.g., loan processors).
    - Generates JSON + Markdown outputs + visual groundings.
    - Fully generic (works with any document type).

---

5. Developer Tools & Resources

- **REST API**: cURL, Python, JS examples.
- **Python Library (preferred)**:
    - Auto-splits & parallelizes long PDFs.
    - Retries + higher rate limits.
    - Handles exception management.
- **Schemas**:
    - Auto-generated based on detected content.
    - Uploadable custom schemas (JSON).
- **Helper Scripts**: GitHub repo with ready-to-run demos.
- **Community**:
    - Docs portal (docs.landing.ai).
    - Discord community with bot + support staff.
    - Trust center (compliance frameworks, security docs).

6. Market Validation

- **Fast**: High throughput, no templates, quick start.
- **Accurate**: Validated on real-world messy docs.
- **Trusted**: Visual grounding + compliance frameworks.
- **Economical**: Cuts manual effort, cost-effective.