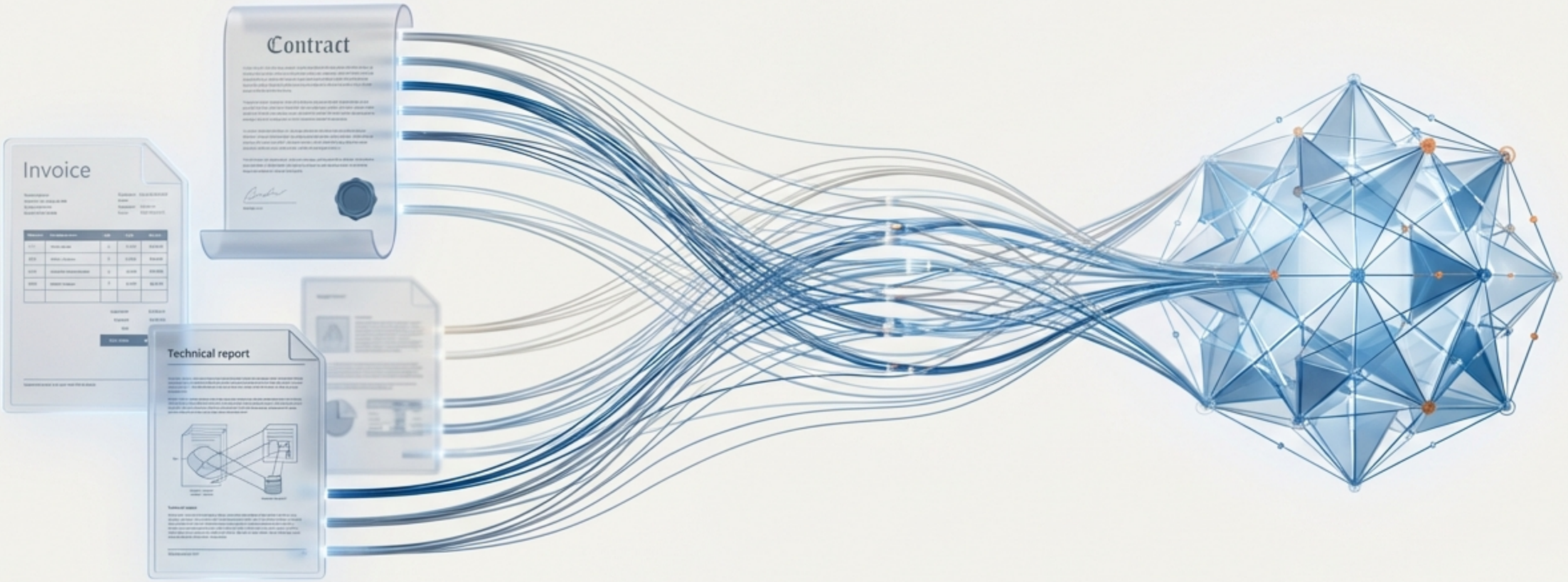
The Modern Playbook for Document AI

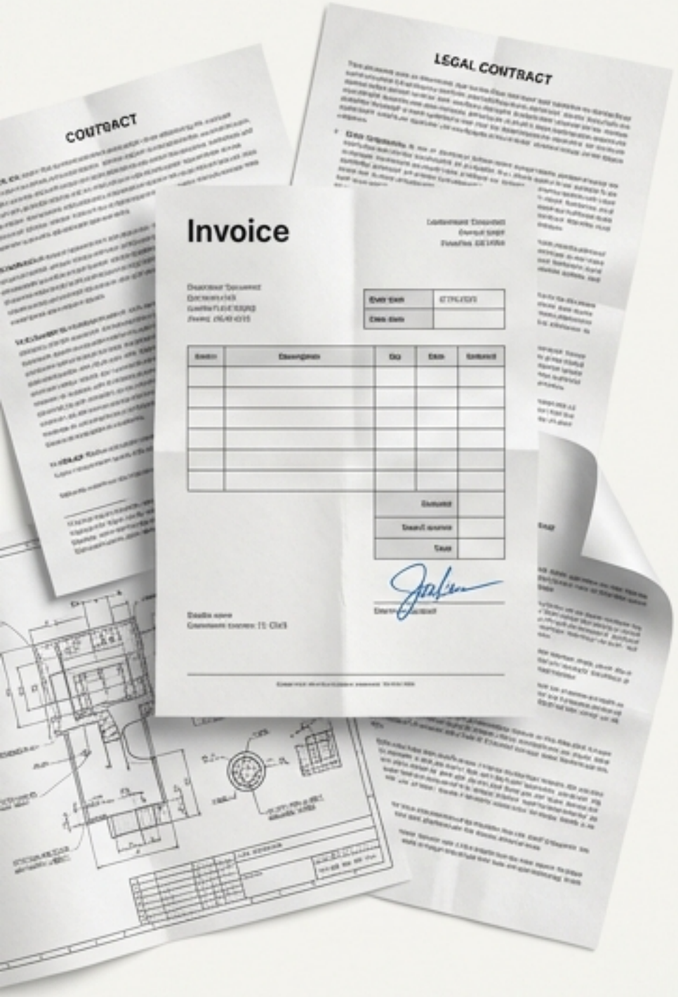
From Extraction to Classification with Azure AI



Based on the presentation by James Croft, Software Engineer, Microsoft.

The Universal Challenge: From Physical Mess to Digital Structure

Every business runs on documents, but turning them into usable data is a persistent and complex challenge.



Layout Variability: No two invoices, contracts, or reports look the same.

Data Complexity: A mix of structured tables, unstructured paragraphs (terms and conditions), signatures, and handwritten text covering important fields.

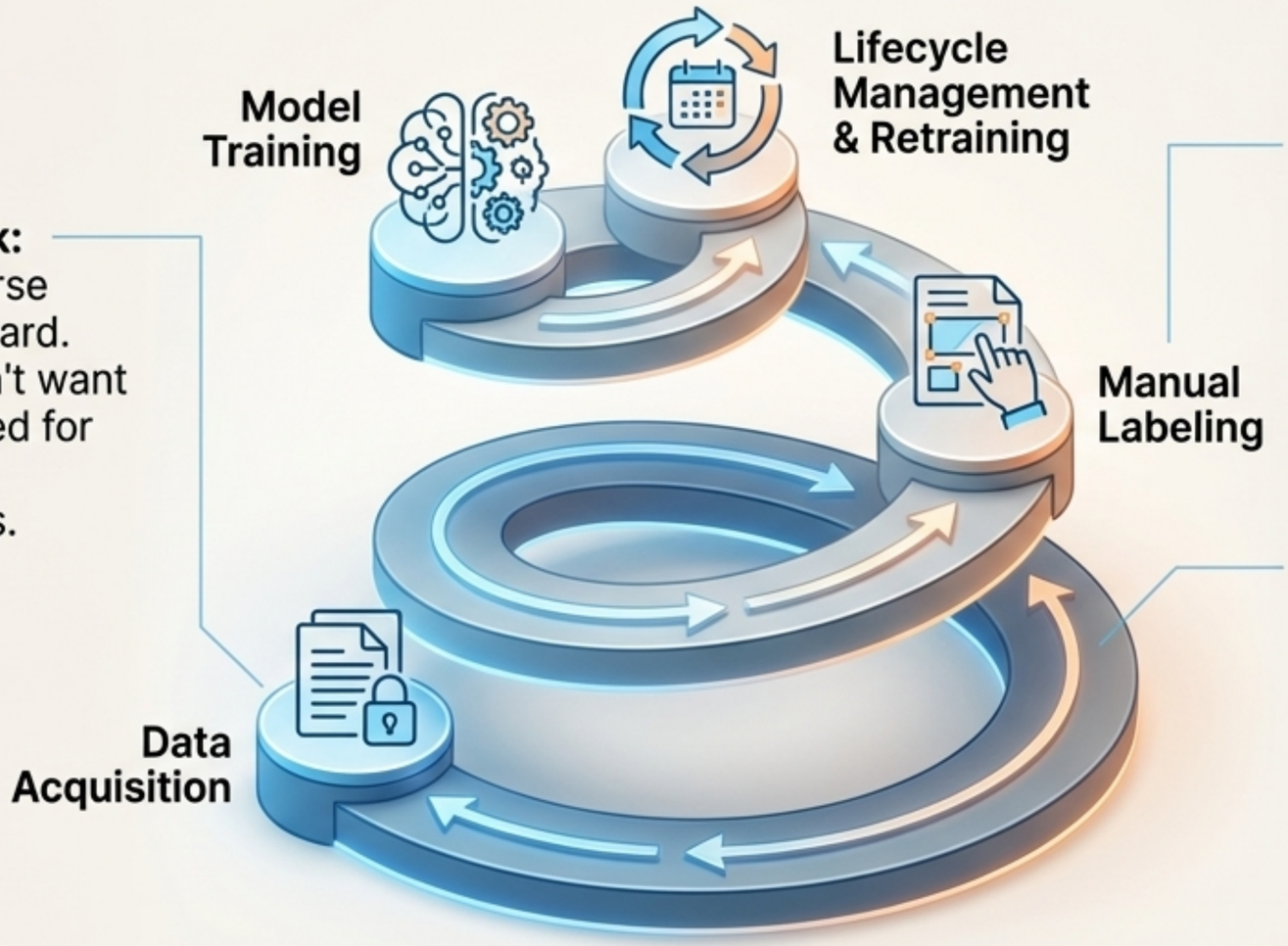
The Manual Bottleneck: Manual review is slow, expensive, and error-prone, yet it's still how many companies operate today.


```
{
  "documentType": "invoice",
  "metadata": {
    "processedDate": "2024-10-26T10:00:00Z",
    "confidenceScore": 0.98
  },
  "data": {
    "customerName": "Acme Corporation",
    "invoiceNumber": "INV-2024-001",
    "invoiceDate": "2024-10-15",
    "dueDate": "2024-11-15",
    "currency": "USD",
    "invoiceTotal": 12500.00,
    "taxTotal": 2500.00,
    "lineItems": [
      {
        "description": "Consulting Services - Q3",
        "quantity": 50,
        "unitPrice": 200.00,
        "total": 10000.00
      },
      {
        "description": "Software License - Annual",
        "quantity": 1,
        "unitPrice": 2500.00,
        "total": 2500.00
      }
    ]
  }
}
```

The Steep Climb of Building Custom Document AI

Building custom vision models from scratch is a significant engineering challenge, especially for **startups** and **solution providers** who need to **deliver value quickly.**

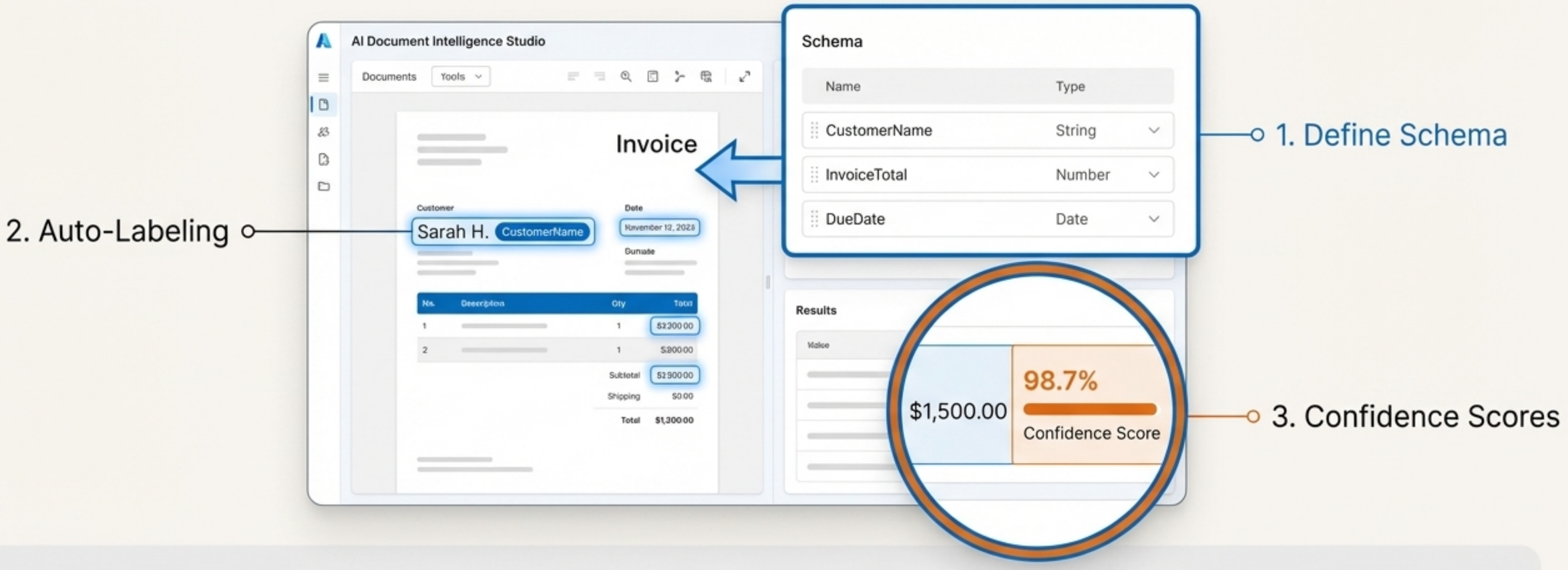
- **The Data Bottleneck:** Acquiring large, diverse training datasets is hard. Customers often don't want their private data used for training, creating a catch-22 for startups.



- **The Expertise Gap:** Requires specialized AI/ML talent to build, maintain, and update custom vision models.
- **The Brittleness Problem:** Models need constant retraining as document layouts inevitably evolve over time. A model trained on last year's invoice format may fail on this year's.

Entry Point: Rapid Prototyping with AI Document Intelligence Studio

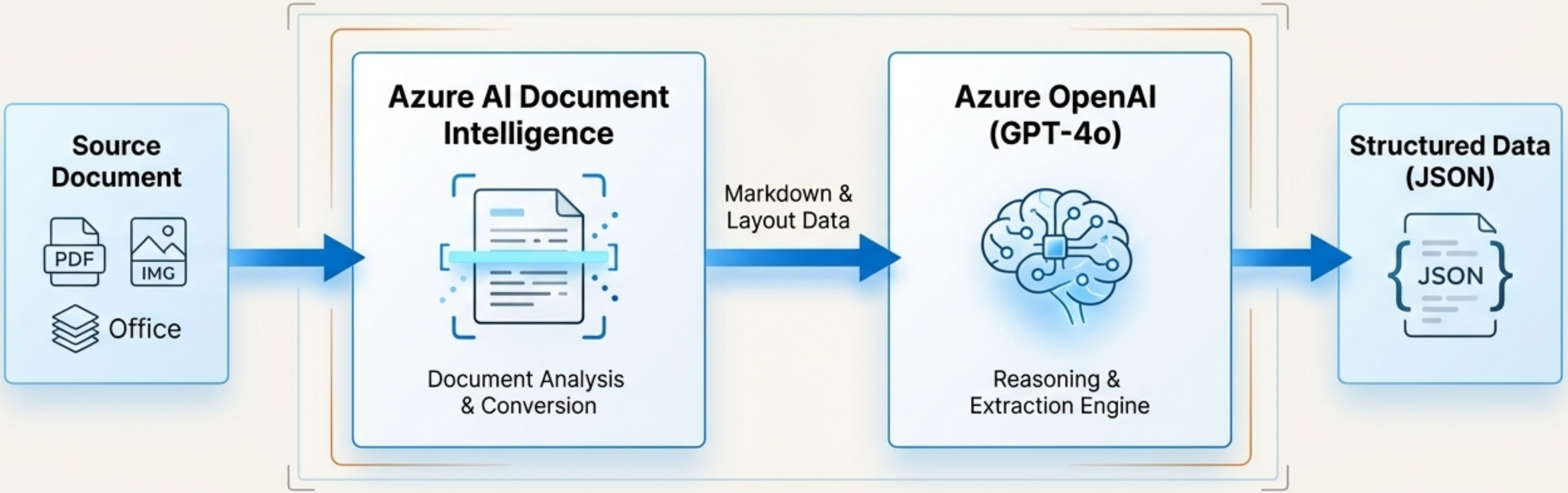
A powerful, low-code environment to quickly define a **schema** and extract data, powered by generative AI for auto-labeling.



Confidence Scores: This isn't just a number; it's the model's internal assessment of its own prediction. Use low scores (<70%) to trigger a human-in-the-loop - the model's internal assessment of its own prediction. Use low scores (<70%) to trigger a human-in-the-loop review, which is essential for building reliable, production-ready systems.

The Pro-Code Toolkit: Our Architectural Building Blocks

For ultimate flexibility and power, we combine the strengths of specialized services into a hybrid, code-first approach.



- **AI Document Intelligence**
  The expert in OCR, layout analysis, and document-to-markdown conversion.
- **Generative Models (GPT-4o)**
  The expert in understanding context, inferring data, and generating guaranteed-valid structured output.

Technique 1: Markdown + LLM for Text-Based Extraction

Challenge: Processing a complex invoice with merged table cells and handwritten signatures covering typed text.

Source Document (Visual)

Markdown Output (Text-Based)

```
|                          | ... | ... |
|                          | ... | ... |
|                          | ... | ... |
|                          | ... | ... |
| Payable by: 10/2S/202Z | ... | ... |
|                          | ... | ... |
|                          | ... | ... |
```

Garbled text due to signature obscuration (misread 5 as S, 3 as Z).

How It Works

1. Document Intelligence converts the invoice into a Markdown representation.
2. This Markdown is passed to GPT-4o with a system prompt and a target JSON schema (defined using a Pydantic modetic model in Python).
3. GPT-4o's 'Structured Outputs' feature guarantees a valid JSON response that can be deserialized successfully.

Limitation Highlight

Excellent for text and table data, but it's 'blind' to visual elements. In our test, it correctly inferred a signature *name* was present but incorrectly assumed it was signed—a potential **false positive**. It cannot truly '*see** the signature.

Technique 2: Vision-Powered Extraction for Inference and Layout

Challenge Scenario: A multi-page insurance policy where values must be *inferred* from unstructured text, like a renewal period defined deep within the terms and conditions.

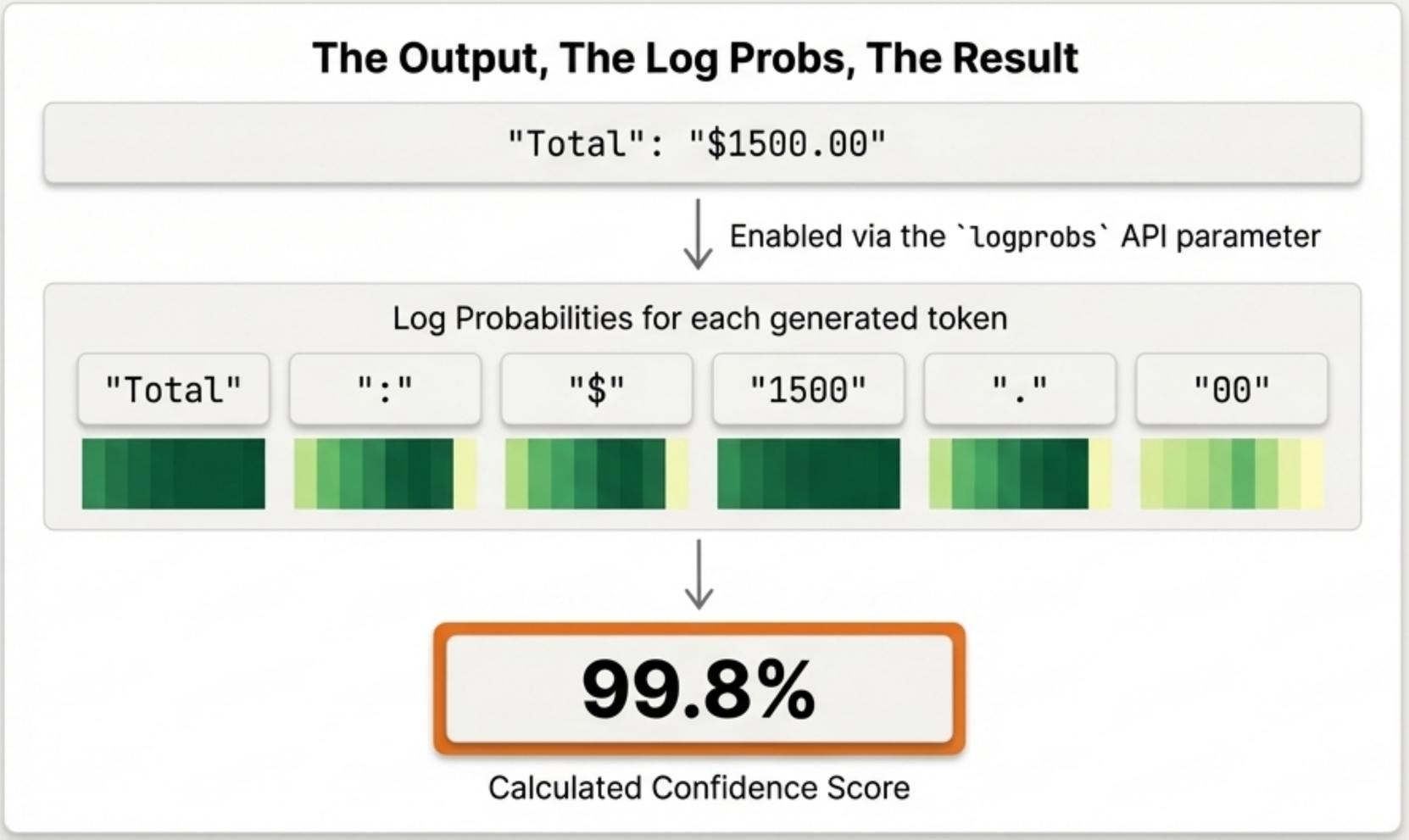
How It Works

1. Each document page is converted into an image (e.g., using a library like `pdf2image`).
2. These images are passed directly to the multimodal GPT-4o model.
3. The model visually analyzes layout, text, and images to extract and infer data based on a prompt that guides it (e.g., 'Some values must be inferred based on rules defined in the policy').

Key Insight: Significantly higher accuracy on complex layouts and inferred data. It can truly *see* signatures, checkmarks, and diagrams, resolving the 'blindness' of the Markdown approach.

The Confidence Conundrum: Re-establishing Trust in Vision Models

Without Document Intelligence's OCR confidence, we need a new way to measure certainty. The answer lies within the model's own token probabilities.



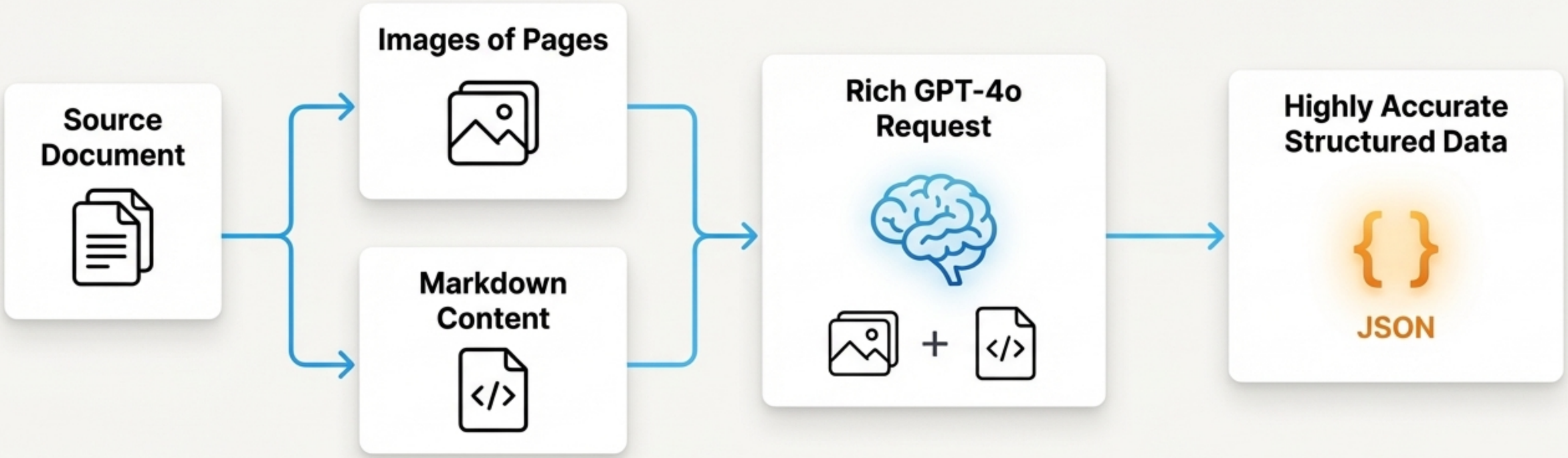
By enabling `logprobs`, we can analyze the probabilities the LLM assigned to each token it generated.

By tokenizing our structured output and mapping it back to these probabilities, we can calculate a robust confidence score for each extracted value.

This is critical for re-enabling human-in-the-loop workflows.

Technique 3: The Hybrid Masterstroke—Vision + Markdown

Combine the pixel-perfect analysis of Vision with the flawless text recall of Markdown for maximum accuracy and robustness.

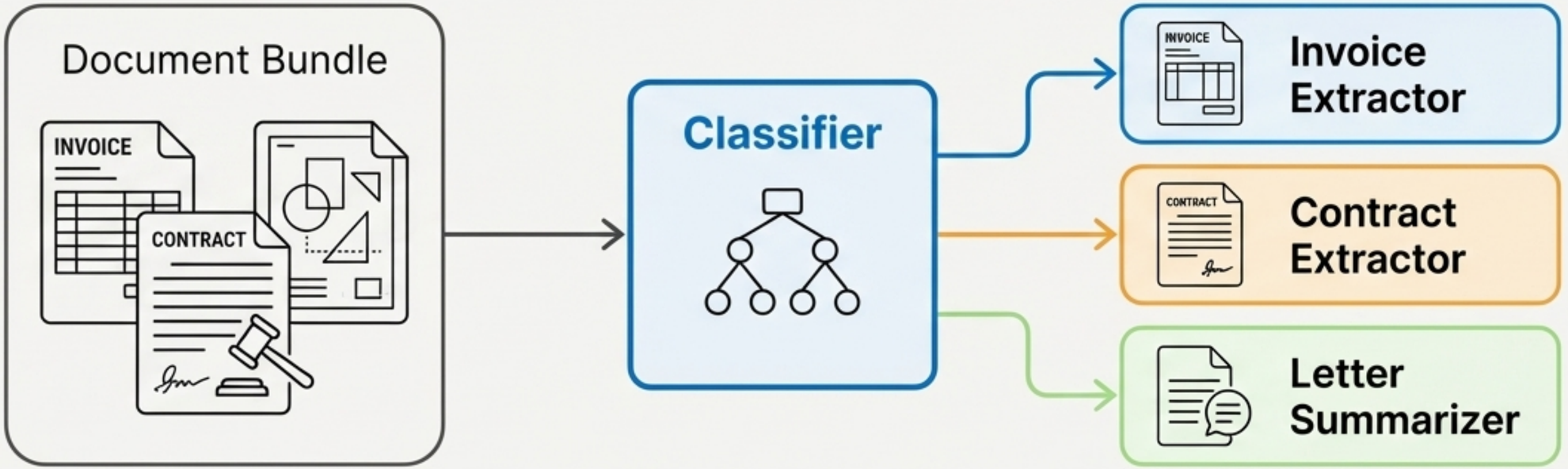


The Synergy

Vision handles spatial layout, signatures, and complex structures where Markdown fails. Markdown provides a perfect transcript of the text, correcting potential OCR errors from Vision. Together, they correct each other's weaknesses, yielding the most reliable results, especially on documents with mixed structured and unstructured data.

Prerequisite: Before You Extract, You Must Classify

In a real-world pipeline, you're not processing one document type; you're processing a bundle. You must first identify the document type (Invoice, Contract, Note) to apply the correct extraction logic.



Fast & Efficient:
Embeddings & Vector Similarity

Highly Accurate:
LLM with Vision

Choosing Your Classifier: Speed vs. Unparalleled Accuracy

Embeddings + Cosine Similarity

How It Works:

Convert document text and classification definitions (e.g., "An invoice contains line items, a total amount, and due date") into vector embeddings. Use cosine similarity to find the closest match.

Pros:

✓ Very fast, cost-effective for large volumes.

Cons:

✗ Accuracy depends on well-defined keywords and text content; can struggle with purely visual distinctions (e.g., two forms that look different but have similar text).

GPT-4o with Vision

How It Works:

Show the model an image of the document (or just the first few pages) and ask it to pick from a list of classifications.

Pros:

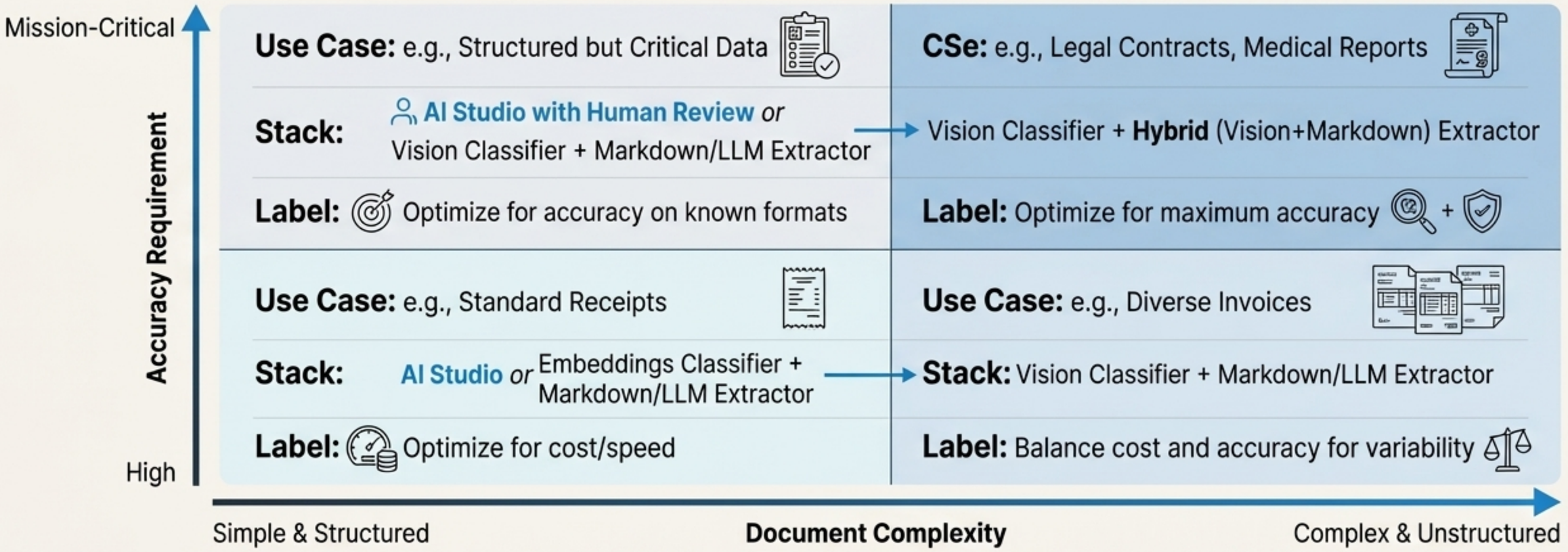
✓ Extremely high accuracy, understands visual layout and form structure.

Cons:

✗ Higher cost and latency per document.

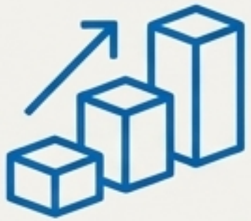
The Architect's Guide: A Decision Framework for Document AI

There is **no one-size-fits-all solution.** Choose the right combination of tools for the job based on your specific requirements for document complexity and accuracy.



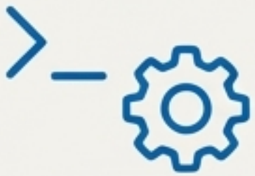
Always balance Accuracy, Speed, and Cost for your specific use case.

Building Production-Ready Solutions: Key Takeaways

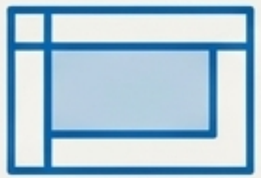


Start Simple, Escalate as Needed: Begin with the easiest effective solution (Markdown + LLM) before adding the complexity and cost of vision-based methods.

Confidence is Non-Negotiable: Always use confidence scores (from Document Intelligence or `logprobs`) to trigger human review. A fully automated system without a human-in-the-loop fallback is a brittle system.



Master the Prompt: Your prompt is your primary tool. Use it to guide the model, define rules for missing data ("provide null""), and specify output formats (e.g., "YYYY-MM-DD").



Mind the Context Window: Be aware of token limits (e.g., ~128k for GPT-4o). An A4 image can consume over 1,000 tokens. For very large documents, you may need strategies like processing only the first few pages for classification.

Go from Theory to Practice

All the **samples** and **detailed analyses** from this presentation are available for you to explore, adapt, and build upon.

Scan to Access the Complete Toolkit:

- Full Python code samples for every technique shown today.
- In-depth articles discussing the nuances of each approach.
- Detailed performance analysis: cost, speed, and accuracy breakdowns.

`github.com/microsoft/some-repo-link`

Join the Microsoft Zero to Hero Community for more deep-dive sessions.

Thank You

Questions & Discussion