



DOCUMENT AI IN PRACTICE

Why Simple LLM Pipelines Fail

Building reliable document-processing systems
beyond impressive demos

Andrei Radulescu-Banu

August 2026 · DocRouter.AI

◆ **Uploading a PDF to an LLM is easy. Building reliable document AI is not. The winning architecture is the one that can adapt when the best model changes tomorrow.**

Contents

1. Uploading a PDF to an LLM is easy. Building reliable document AI is not.
2. Different models have different strengths
3. Quality is only half of the equation
4. Large documents require orchestration
5. Not every page is machine-readable
6. Language and script matter
7. Many document tasks require multiple steps
8. External tools are often essential
9. One platform, many processing paths
10. Flexibility is the real requirement
11. What DocRouter.AI solves

Why simple LLM pipelines often fail when documents become large, messy, and operationally important—and how production systems are designed differently.

Uploading a PDF to an LLM is easy. Building reliable document AI is not.

Almost every major language model now allows users to attach a PDF or Microsoft Word document and ask questions about it. That creates the impression that document processing has become simple:

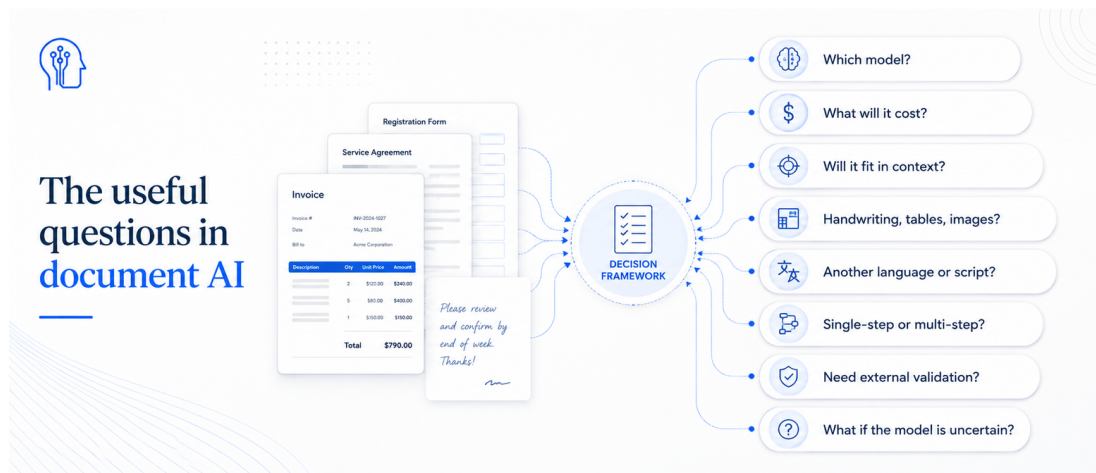
1. Upload a document
2. Write a prompt
3. Receive structured results

For a short, clean document and a one-time task, that may be enough. At production scale, document processing becomes a much more complicated engineering problem. The central question is no longer simply:

Can a language model read this document?

The more useful questions are:

- Which model should process it?
- How much will processing cost?
- Can the entire document fit into the model's context?
- Does it contain handwriting, tables, images, or unusual formatting?
- Is it written in another language or script?
- Does processing require several steps?
- Do results need checking against an external system?
- What happens when the model is uncertain or wrong?



The useful questions in document AI — a decision framework



The easy demo versus real document AI in production

MODEL SELECTION

Different models have different strengths

Language models are not interchangeable. Some are especially strong conversational assistants. Others are optimized for coding, reasoning, speed, multilingual work, or visual understanding.

A model that performs extremely well on software-development tasks may not be the best choice for extracting information from a complex insurance packet. A strong general-purpose conversational model may struggle with dense tables, handwritten notes, low-quality scans, or unfamiliar document layouts.

Document-processing quality can depend on many factors:

- The type of document
- The quality of the scan
- The number of pages
- The complexity of the layout
- The language used
- The expected output
- The amount of reasoning required
- The model's visual and OCR capabilities



What affects document-processing quality?

Choosing a model should be treated as an evaluation problem, not as a matter of brand preference.

◆ **The best model is the one that performs reliably on your actual documents and your actual extraction requirements.**

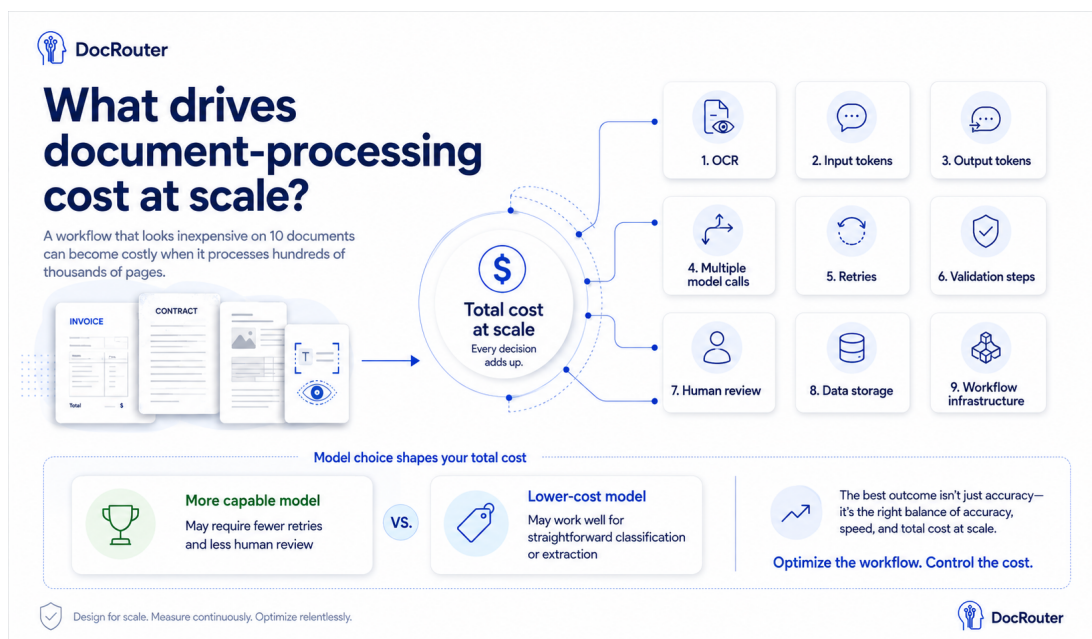
COST AT SCALE

Quality is only half of the equation

The highest-quality model is not always the right model. Document processing at scale can become expensive. A workflow that looks inexpensive when tested on ten documents may become costly when it processes hundreds of thousands of pages.

The total cost can include:

- OCR
- Input tokens
- Output tokens
- Multiple model calls
- Retries
- Validation steps
- Human review
- Data storage
- Workflow infrastructure



What drives document-processing cost at scale?

A more capable model may require fewer retries and less human review. A cheaper model may work perfectly well for straightforward classification or extraction tasks.

◆ **Select the least expensive architecture that produces results at the required level of quality.**

That architecture may use one model for every document. More often, it uses several:

- **Fast, inexpensive model**
Classification and straightforward routing
- **Specialized OCR**
Scanned pages and layout-heavy forms
- **Stronger reasoning model**
Difficult cases and dense packets
- **Human reviewer**

Uncertain or high-stakes results

LARGE PACKETS

Large documents require orchestration

SIMPLE CASE

10-page PDF

Often processed in a single request

PRODUCTION CASE

500-page packet

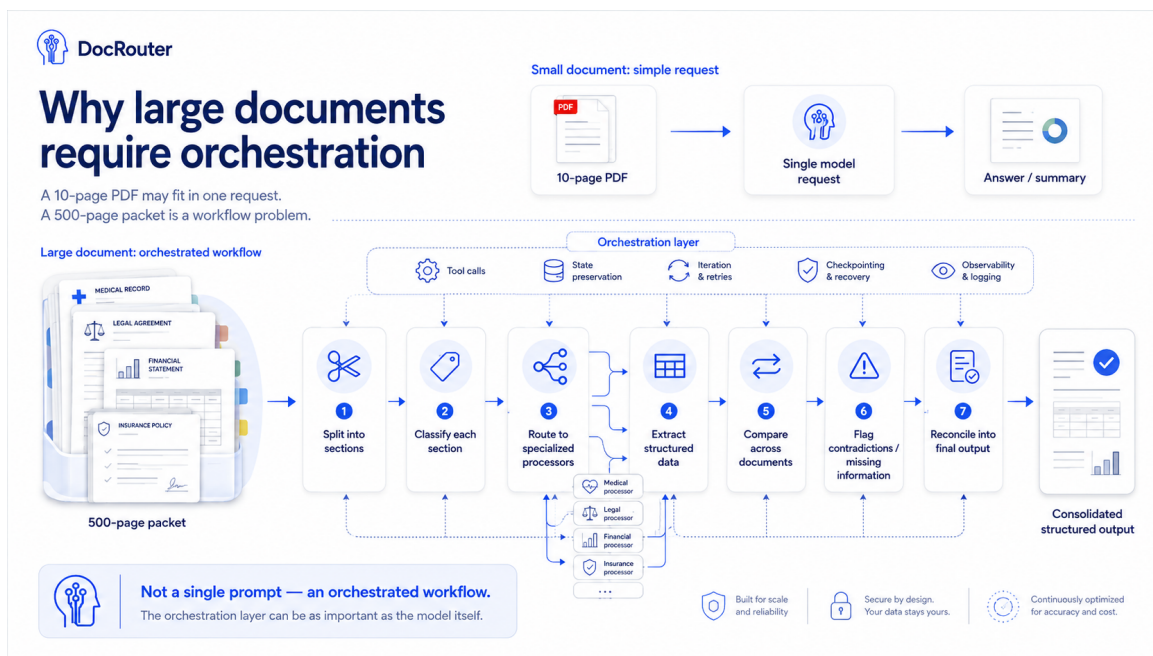
Medical, legal, financial, or insurance—needs a workflow

Even when a model supports a very large context window, passing the entire document in one request may not produce the best results. Important information may be scattered across hundreds of pages. Sections may need to be classified, separated, summarized, compared, or reconciled.

A large-document workflow might need to:

1. Split the packet into logical sections
2. Classify each section
3. Route each document type to a specialized processor
4. Extract structured data
5. Compare information across documents
6. Identify contradictions or missing information
7. Reconcile the results into a final output

This is not a single prompt. It is an orchestrated workflow.



Why large documents require orchestration

Some models and platforms provide internal agentic capabilities that can perform several actions iteratively. In other cases, the surrounding application must manage the steps, preserve state, call tools, handle failures, and combine the outputs.

◆ **The orchestration layer becomes just as important as the model itself.**

MESSY INPUTS

Not every page is machine-readable

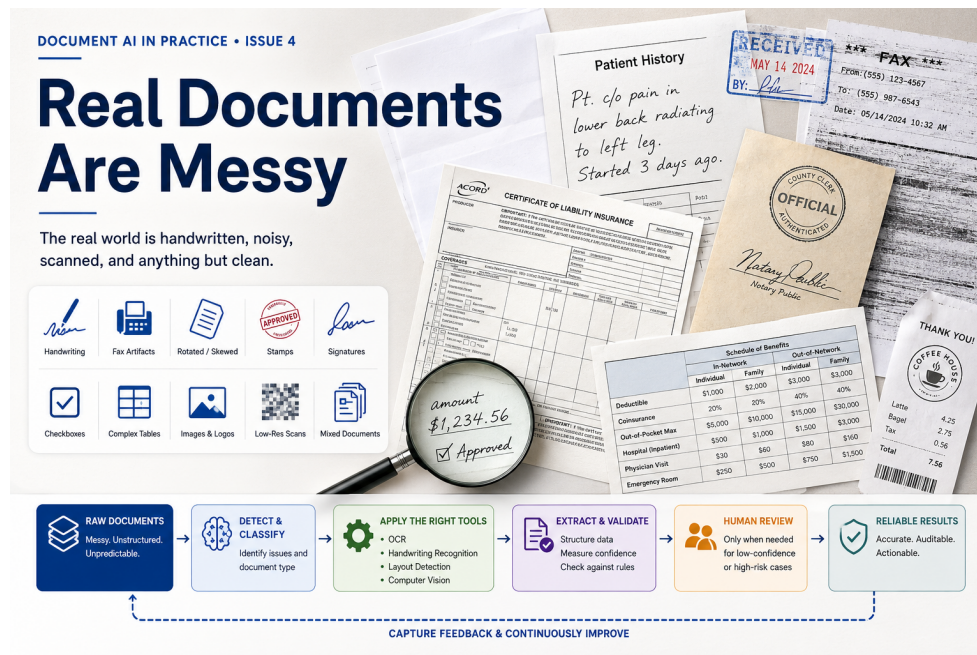
Real-world documents are messy. They may contain:

- Handwriting
- Fax artifacts
- Rotated pages
- Stamps
- Signatures
- Checkboxes
- Tables
- Embedded images
- Low-resolution scans
- Multiple documents combined into one packet

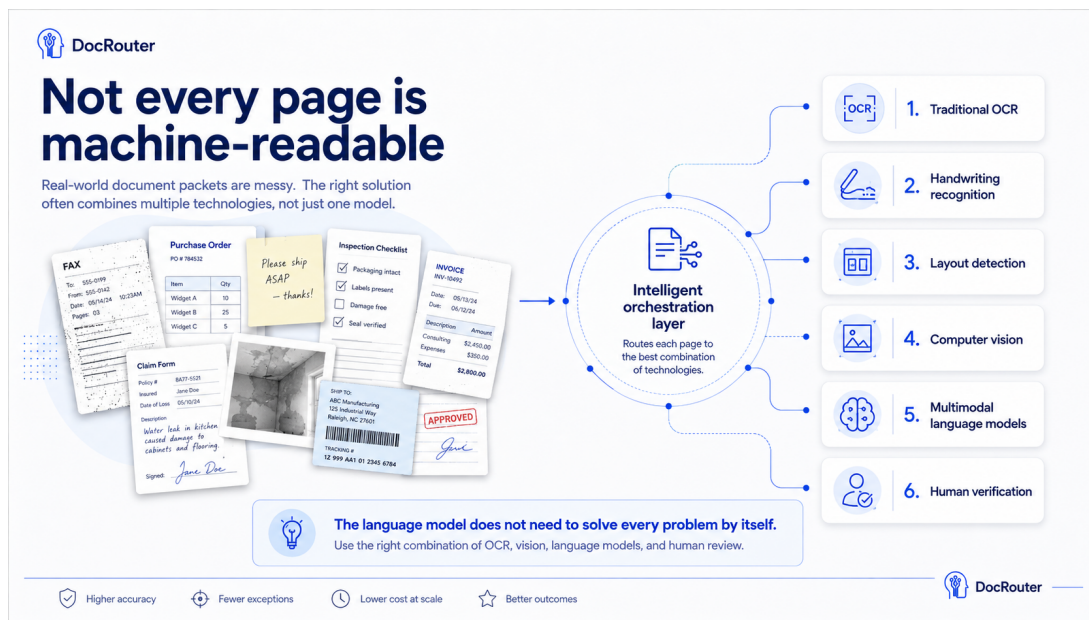
Language models can often interpret many of these elements, but performance varies. Handwriting is particularly challenging, especially when the scan quality is poor or the writing is highly individual.

The right approach often combines several technologies:

- Traditional OCR
- Handwriting recognition
- Layout detection
- Computer vision
- Multimodal language models
- Human verification



Real documents are messy—handwritten, noisy, scanned, and mixed



Not every page is machine-readable

◆ **The language model does not need to solve every problem by itself.**

GLOBAL OPERATIONS

Language and script matter

Global document-processing systems must also handle multilingual content. A model may perform well in English but produce weaker results in another language. Performance can vary further when documents use non-Latin scripts or mix several languages on the same page.

The system may need to determine:

- What language is present
- Whether multiple languages are used
- Whether translation is required
- Whether to extract before or after translation
- Which model performs best for that language
- Whether original text must be preserved for audit



Multilingual processing should be tested by language, document type, and script—not assumed from a model's general language-support claims.

WORKFLOW DESIGN

Many document tasks require multiple steps

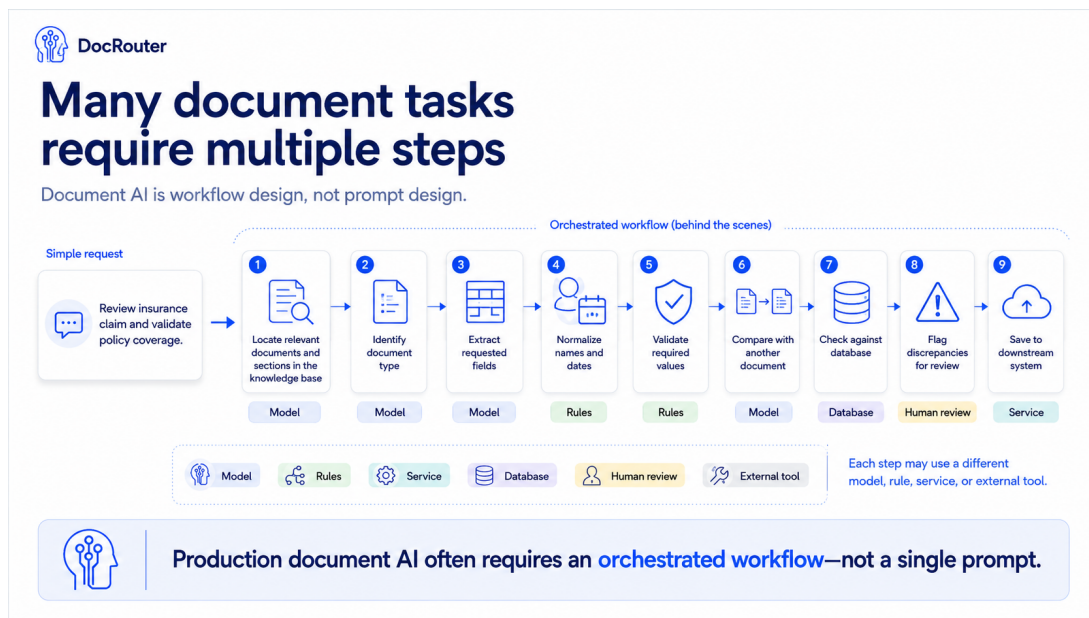
A document-processing task may sound simple:

Review insurance claim, and validate policy coverage.

A production workflow may require much more:

1. Identify the document type
2. Locate the relevant section
3. Extract the requested fields
4. Normalize names and dates
5. Validate required values
6. Compare the information with another document
7. Check the result against a database
8. Flag discrepancies for review
9. Save the result in a downstream system

Each step may use a different model, rule, service, or external tool.



Many document tasks require multiple steps

◆ Document AI should be viewed as workflow design rather than prompt design.

SYSTEMS INTEGRATION

External tools are often essential

Language models process the information placed in their context, but many business decisions depend on information outside the document. For example, the workflow may need to:

- Verify a customer against a database
- Confirm that an identifier exists
- Retrieve an insurance policy
- Check a payment amount
- Compare a document against a contract
- Validate an address
- Look up historical records
- Update a claims or case-management system

The model must therefore be able to interact with external tools and systems. That introduces additional engineering requirements:

- Authentication
- Permissions
- Error handling
- Audit trails
- Data validation
- Retry logic
- Human approval

A useful document-processing platform must coordinate both unstructured documents and structured systems.

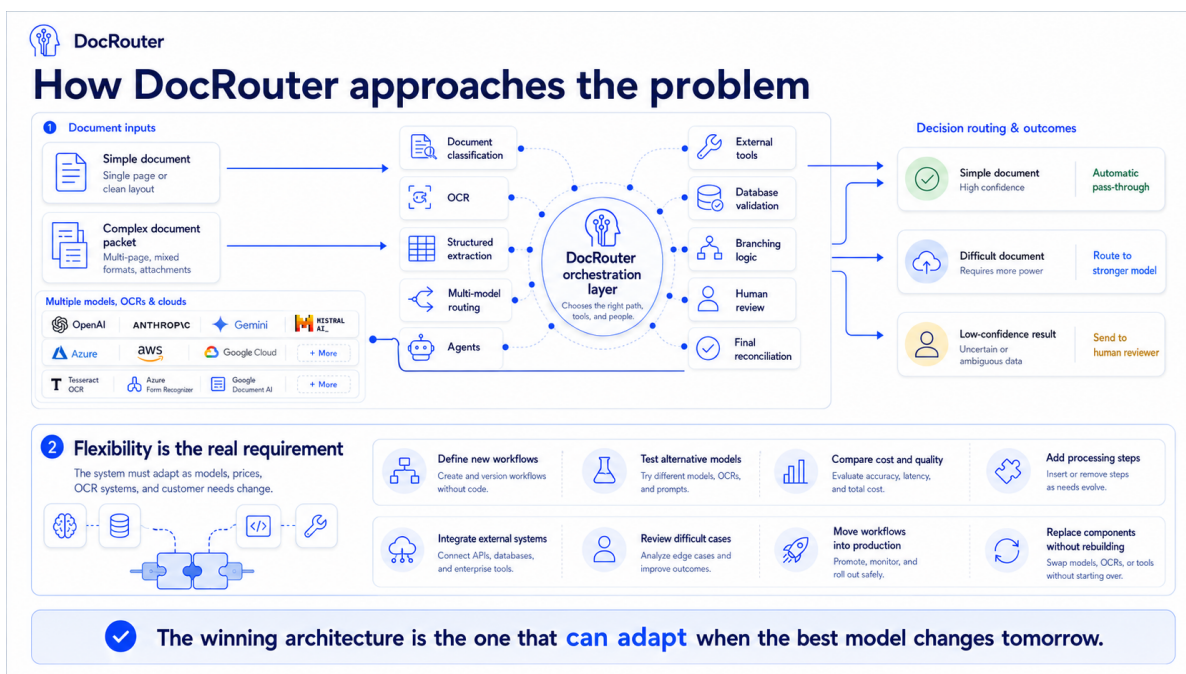
DOCROUTER

One platform, many processing paths

DocRouter is designed around the idea that no single model, cloud, or processing pattern is right for every document. It integrates with multiple cloud providers, OCR systems, and language models. A workflow can use a direct, single-shot model call for a simple task or a multi-step process for a large and complex document packet.

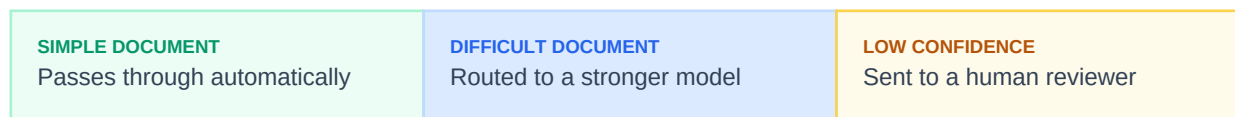
DocRouter workflows can include:

- Document classification
- OCR
- Structured extraction
- Multi-model routing
- Agents
- External tools
- Database validation
- Branching logic
- Human review
- Final reconciliation



DocRouter orchestration: multi-model routing, tools, and human review

This flexibility makes it possible to start with a straightforward pipeline and add more sophisticated processing only where it is needed.



The goal is not to place a human in every workflow. It is to involve a human only when automation cannot produce a sufficiently reliable result.

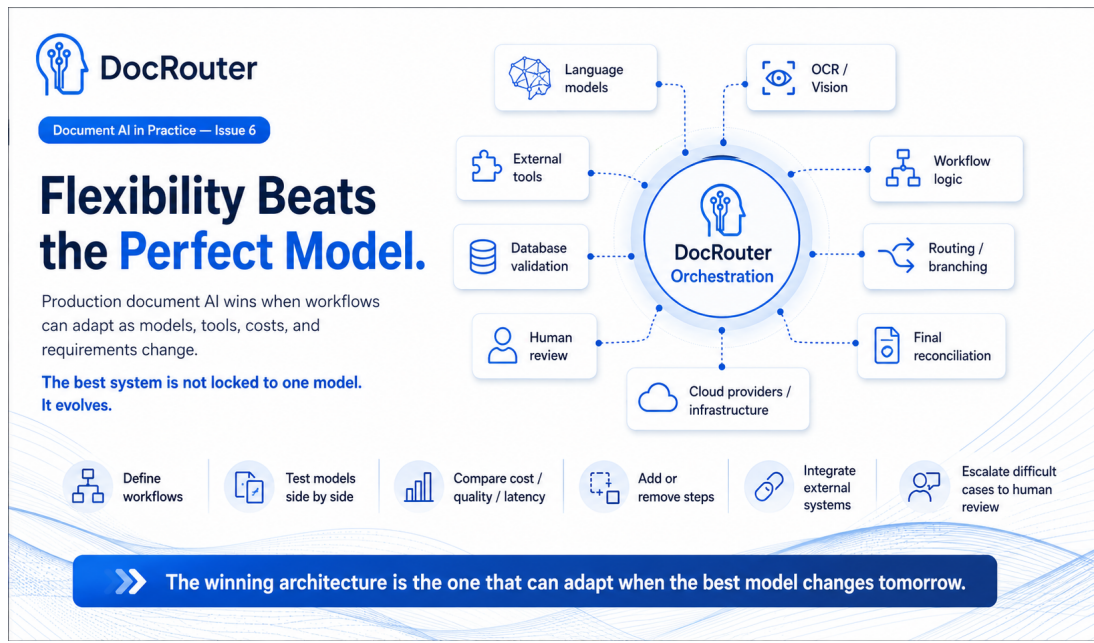
ARCHITECTURE

Flexibility is the real requirement

Document AI is evolving quickly. Models improve, prices change, new OCR systems appear, and customer requirements become more sophisticated. A production system must therefore make it easy to:

- Define new workflows
- Test alternative models
- Compare cost and quality
- Add processing steps
- Integrate external systems
- Review difficult cases
- Move workflows into production
- Replace components without rebuilding everything

◆ **The winning architecture is the one that can adapt when the best model changes tomorrow.**



Flexibility beats the perfect model

PRODUCTION CAPABILITIES

What DocRouter.AI solves

These are not open research questions for us. They are the problems DocRouter.AI was built to address in production:

- ✓ Evaluate language models on real documents—not vendor benchmarks
- ✓ Choose OCR or a multimodal model based on the page, not a single default
- ✓ Process packets with hundreds of pages through orchestrated workflows
- ✓ Compare quality, latency, and cost across models and pipelines
- ✓ Build reliable multi-step workflows with branching, tools, and state
- ✓ Use human review only for uncertain or high-stakes results
- ✓ Validate model output against external systems before it becomes a business decision

Document AI is no longer limited by whether a model can read a PDF.

With DocRouter.AI, you can process diverse documents reliably, economically, and at scale.

— Andrei Radulescu-Banu · DocRouter.AI · August 2026