

An aerial view of a city skyline, likely New York City, with a network of white lines and nodes overlaid on the image. The lines form a complex web, connecting various points across the city. The background is a light blue gradient, and the text is centered in the middle of the image.

# MANAGEMENT PERFORMANCEHUB

**Your Documents Are Holding Out On You**

Theodore Stumpf, Senior Data Scientist

# Common Definitions

- **LLM:** Large Language Model. Text generation tool
- **Context Window:** The input limit to an LLM
- **Inference:** The process of an LLM generating text
- **Document:** Anything that holds unstructured text
- **Tools:** Software components designed specifically for LLMs

# What I'll Cover

1. Why we want to give LLMs **long documents**
2. **DocuSage**, an in-house tool I made for MPH
3. How LLMs **process** long documents
4. **Limitations** of LLMs
5. How to **surpass the limitations** of LLMs

An aerial view of a city skyline, likely New York City, with several prominent skyscrapers. The image is overlaid with a blue-tinted network of white lines and circular nodes, suggesting a digital or data-driven theme. A vertical yellow bar is positioned to the left of the main text.

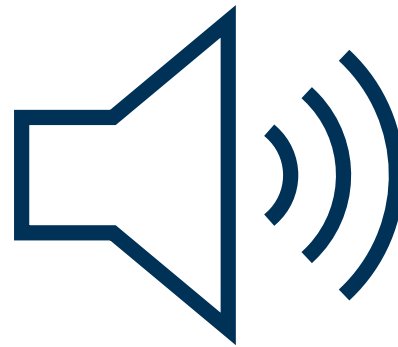
# LLMs Excel at Processing Text

MANAGEMENT  
PERFORMANCEHUB

# Why Documents Alone Fall Short

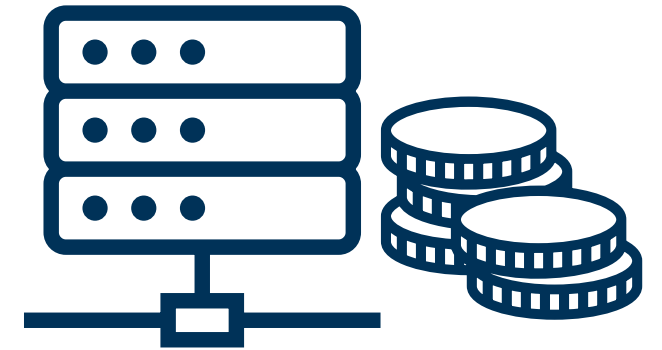
Dumping the full text of documents into an LLM is costly, slow, and hard to audit

## Traceability



Noise

## Cost

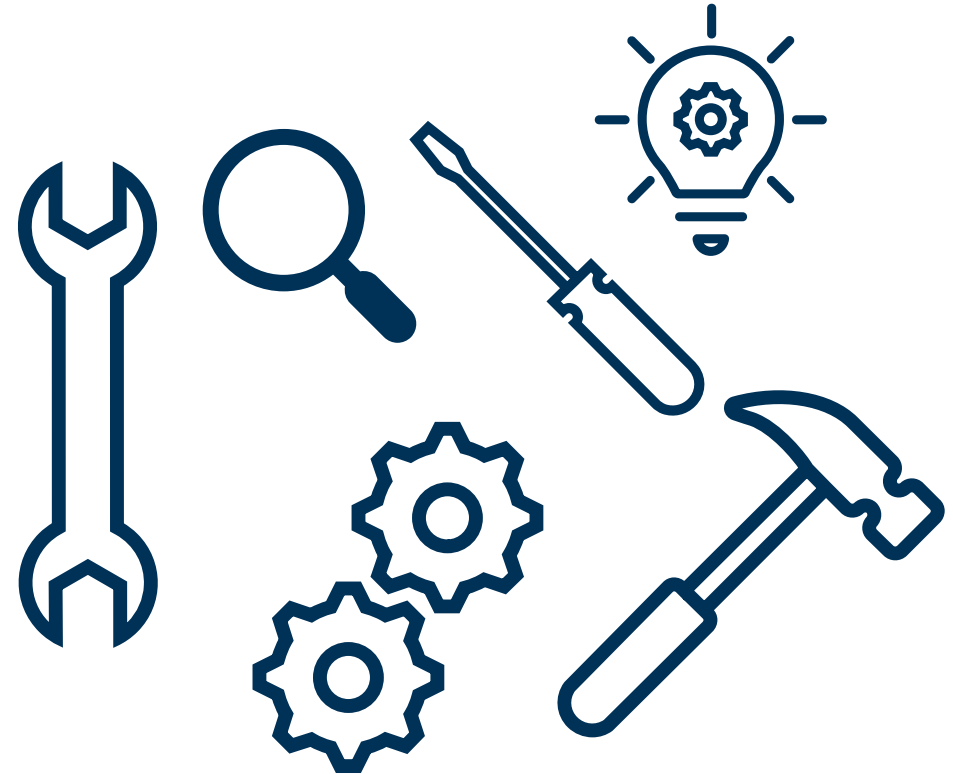


# The Solution: Building Tools

## Custom Tooling is:

- Traceable
- Accurate
- Efficient

The limitations LLMs face when processing long documents can be surpassed with custom tools



# How Cost Impacts You

## Long Document:

- 200 pages
- ~55,000 words

At \$3.60 / 1M words:

- Total Cost: **\$0.20**

## Smart Tooling:

- 3 pages
- ~825 words

At \$3.60 / 1M words:

- Total Cost: **\$0.003**

## Very Long Document:

- 1,500 pages
- ~412,500 words

At \$3.60 / 1M words:

- Total Cost: **\$1.49**

Even if you are using a subscription service, the company providing it is using tooling behind the scenes to reduce cost



**DocuSage**

MANAGEMENT  
**PERFORMANCEHUB**

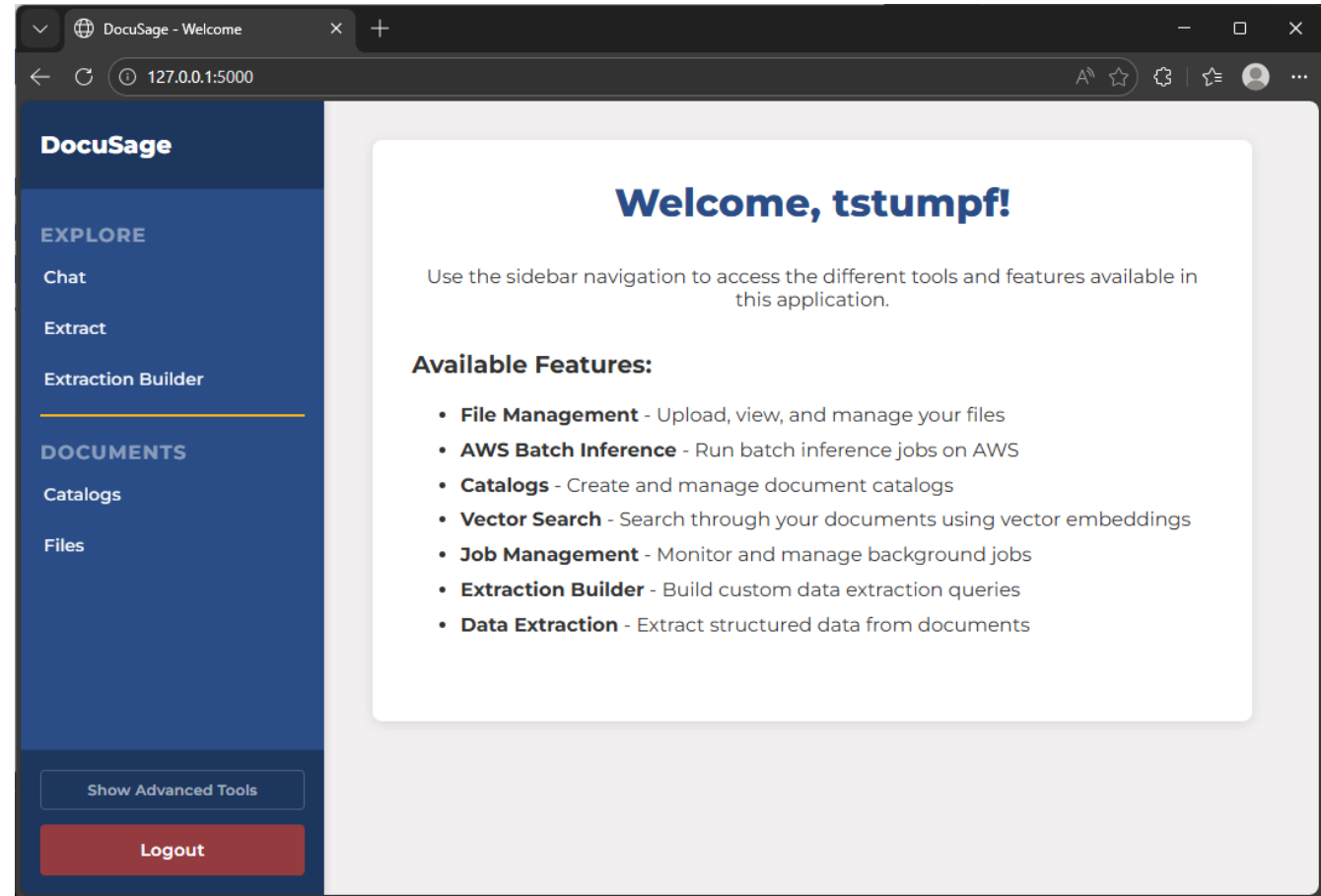
# Quick Disclaimer

DocuSage is an internal application created specifically for MPH-defined use cases. It is intended for testing and exploration rather than full operational rollout, and it is not available to other agencies at this time.

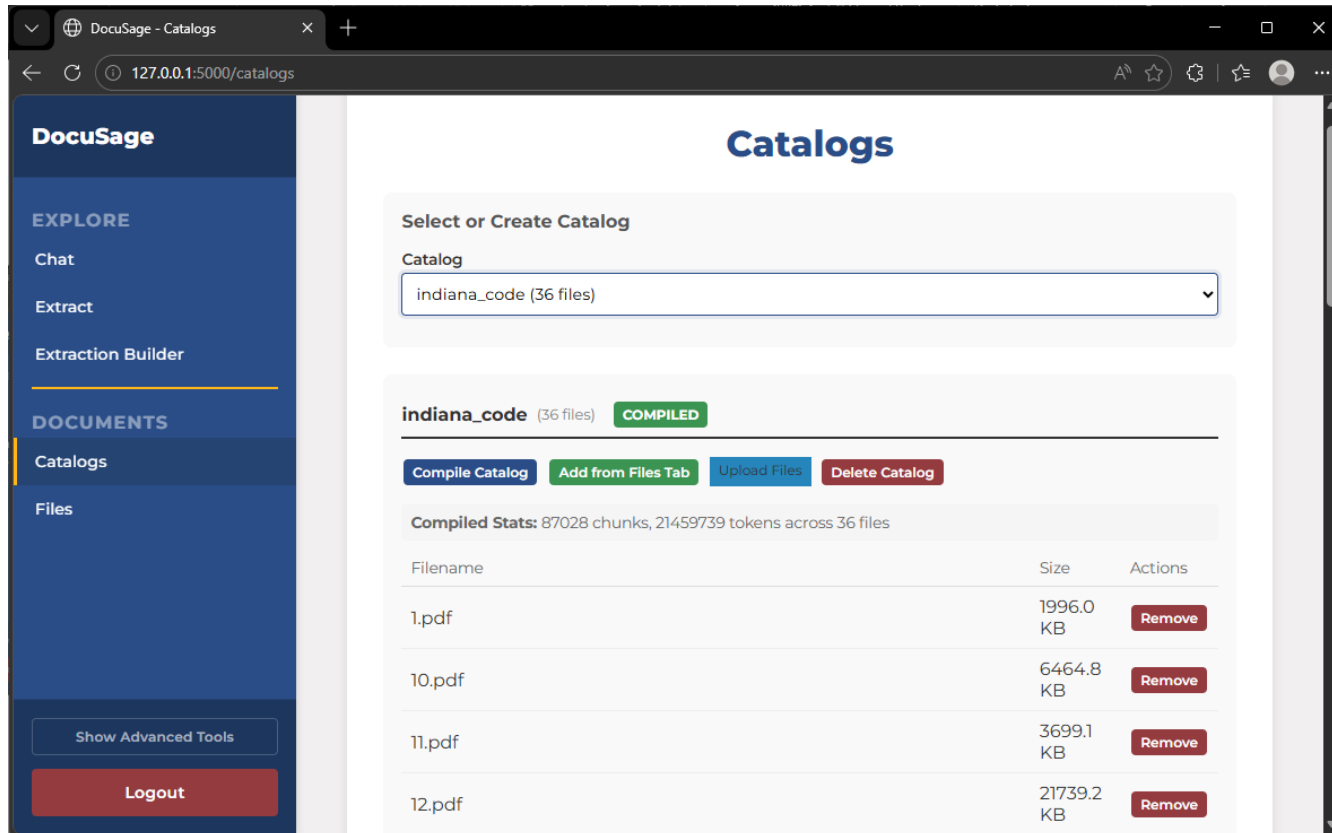
Our leadership team is actively collaborating with State of Indiana leadership, across all agencies, to explore opportunities for enterprise AI solutions in the upcoming budget cycle!

# What is DocuSage?

DocuSage allows users to build collections of documents called **catalogs** and then **chat** with an **AI Agent** to ask questions about the contents of the documents in the catalogs.



# Catalogs



The screenshot shows the DocuSage web interface. The left sidebar contains navigation options: 'EXPLORE' (Chat, Extract, Extraction Builder), 'DOCUMENTS' (Catalogs, Files), and a 'Logout' button. The main content area is titled 'Catalogs' and features a 'Select or Create Catalog' section with a dropdown menu showing 'indiana\_code (36 files)'. Below this, there are buttons for 'Compile Catalog', 'Add from Files Tab', 'Upload Files', and 'Delete Catalog'. A 'Compiled Stats' section indicates 87028 chunks and 21459739 tokens across 36 files. A table lists the files in the catalog:

Filename	Size	Actions
1.pdf	1996.0 KB	<a href="#">Remove</a>
10.pdf	6464.8 KB	<a href="#">Remove</a>
11.pdf	3699.1 KB	<a href="#">Remove</a>
12.pdf	21739.2 KB	<a href="#">Remove</a>

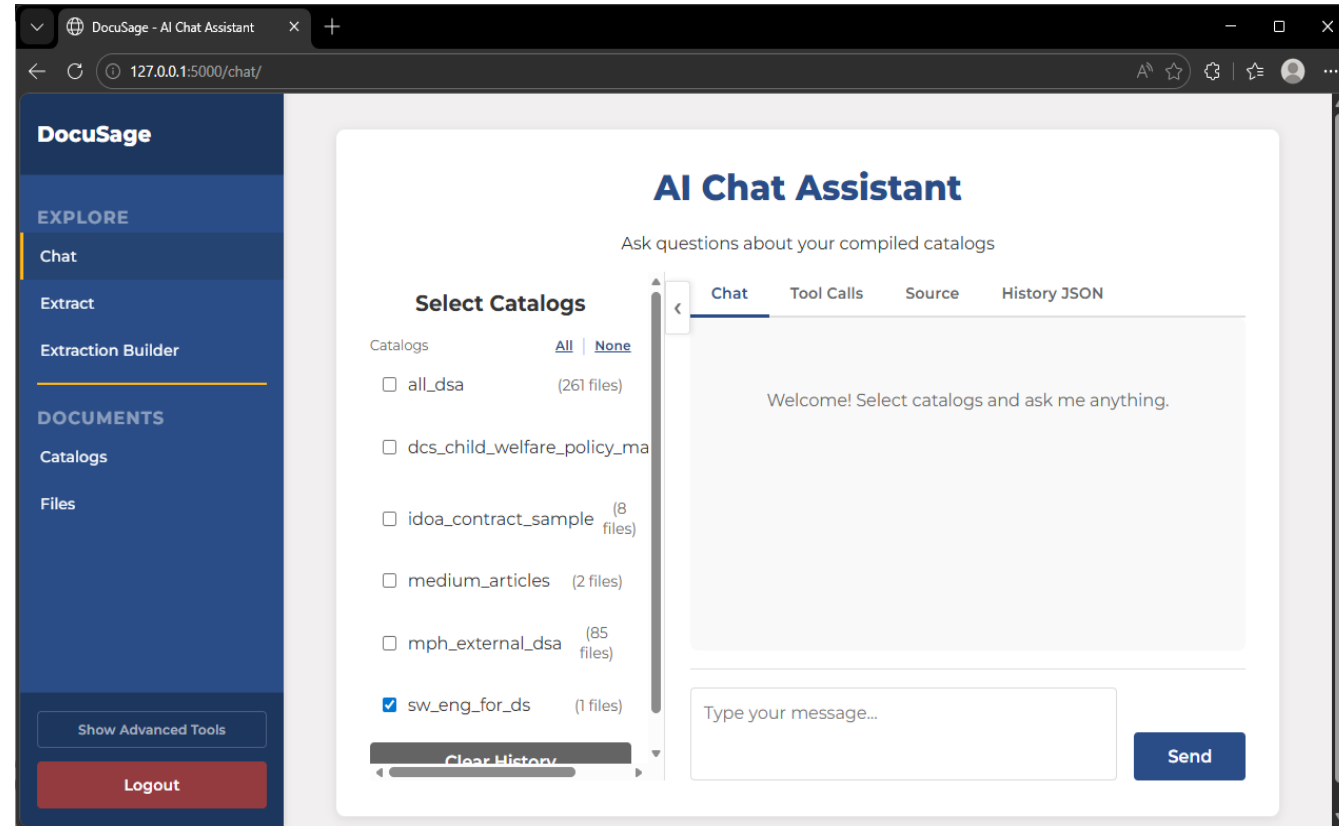
Catalogs are collections of documents with a shared purpose

The documents can be text files or even scanned .PDFs

Some pre-processing is done on the documents to make them more searchable

# Document Chat

Once you have your catalogs compiled, the DocuSage Chat Assistant can help you answer *most\** questions about your documents



# AI Chat Assistant

Ask questions about your compiled catalogs

Chat Tool Calls Source History JSON

Software\_Engineering\_for\_Data\_Scientists\_From\_Notebooks\_to\_Scalable.pdf page 188

```
everywhere'  
...}]  
}]
```

I'll cover how to make a POST request to an API in ["Making Requests to Your API" on page 175](#).

## Creating Your Own API Using FastAPI

In this section, I'll show you how to create your own API using [FastAPI](#), a framework for writing API endpoints developed by Sebastián Ramírez. It was first released in 2018 and has seen rapid widespread adoption because it is easy to use and works well with other modern Python tools. It also has other useful features including automatic documentation, and it conforms with the [OpenAPI](#) specifications, a widely used set of standards for APIs. In the following sections I'll show you how to set up a basic API with FastAPI and how to add GET and POST endpoints.

### Other API Frameworks

[Flask](#) is another very popular API framework. It's older than FastAPI and is a little more complex to use, but it's quite similar. After you've read through this section you should find it easier to translate the concepts used by FastAPI to their corresponding commands in Flask.

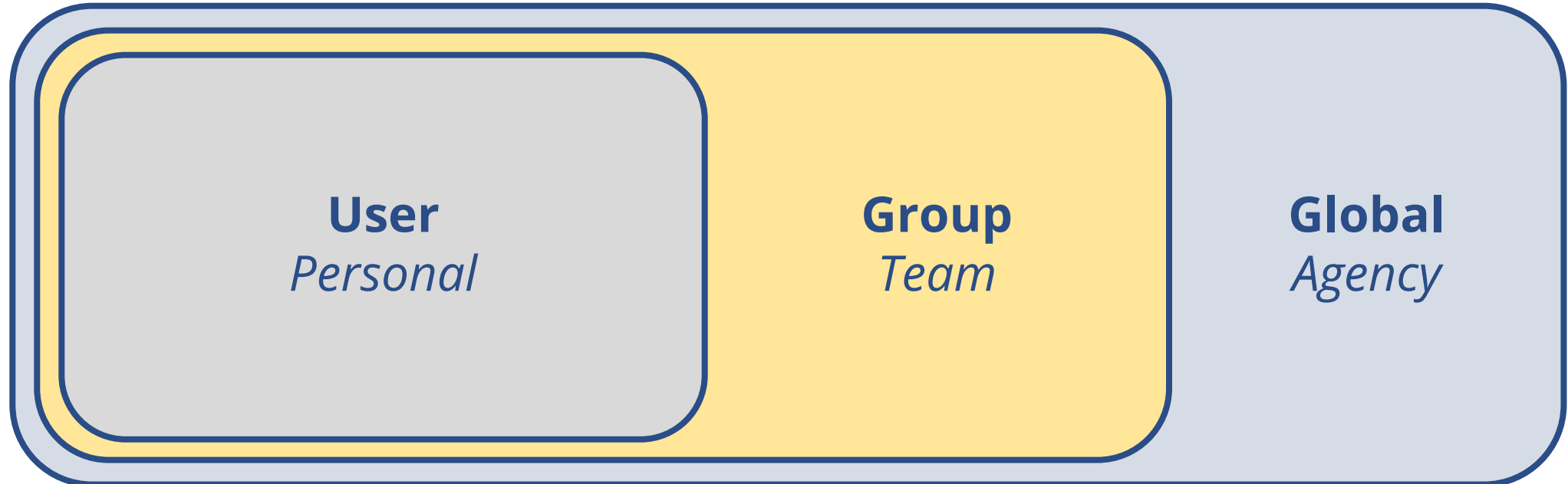
---

168 | Chapter 11: APIs

Page 189 / 258

# DocuSage User Groups

User Groups allow for the easy sharing of Catalogs by allowing different access levels to be assigned to each catalog.



# Use Cases for DocuSage

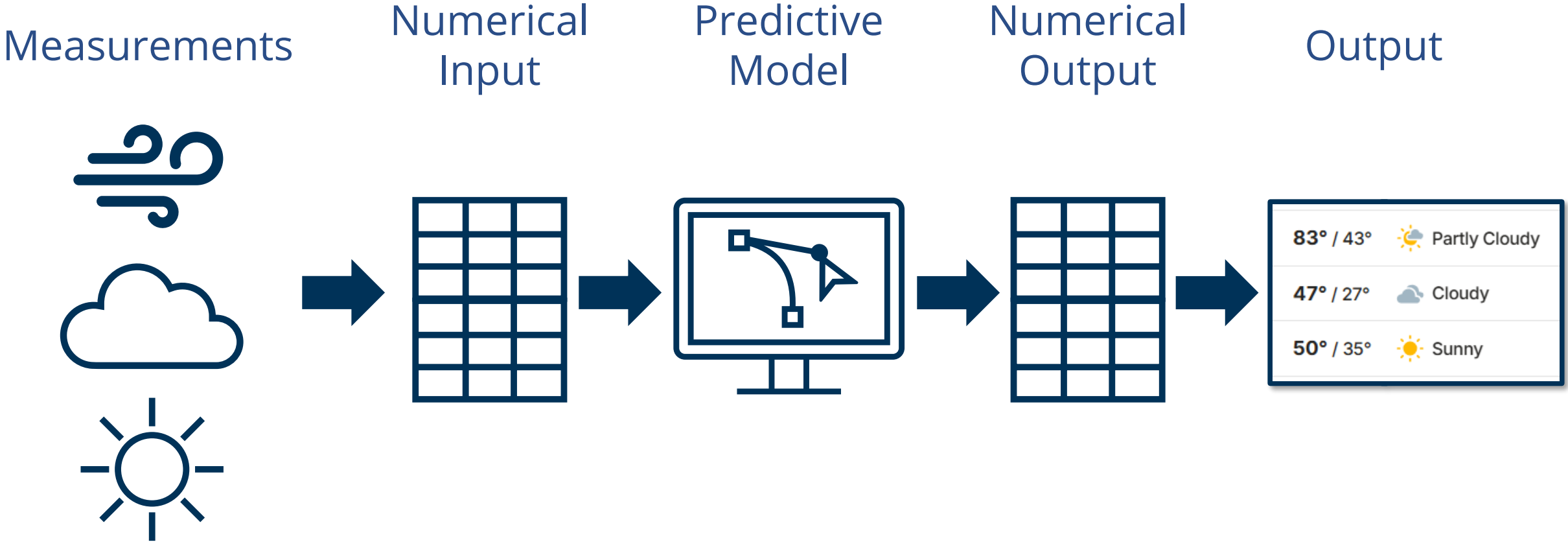
- **Security:**  
Finding specific citations in standards documentation (NIST, CJIS)
- **Legal:**  
Locating prior examples and precedents across data sharing agreements
- **Project Management:**  
Surfacing information in project charters



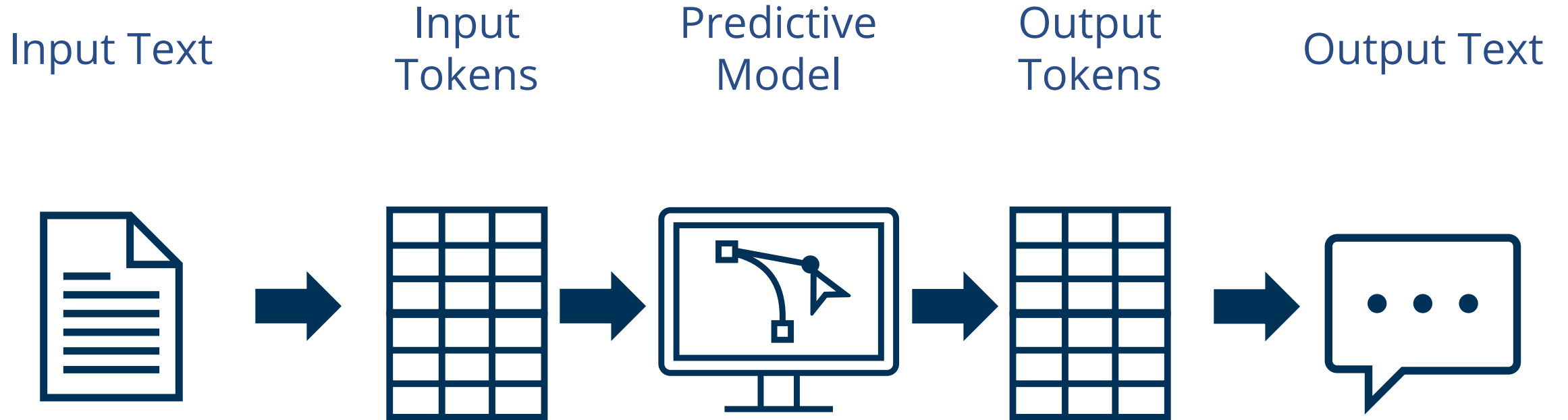
# | How DocuSage (& others) Work

MANAGEMENT  
PERFORMANCEHUB

# Weather Prediction

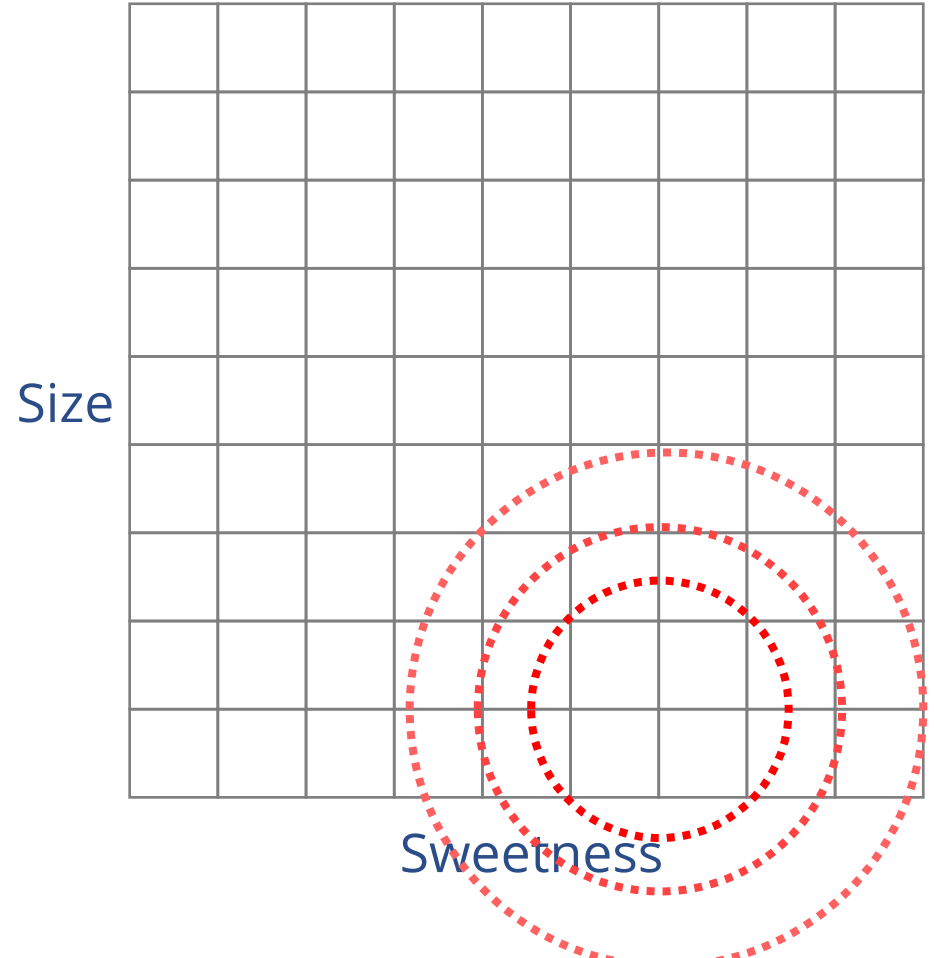


# Large Language Models



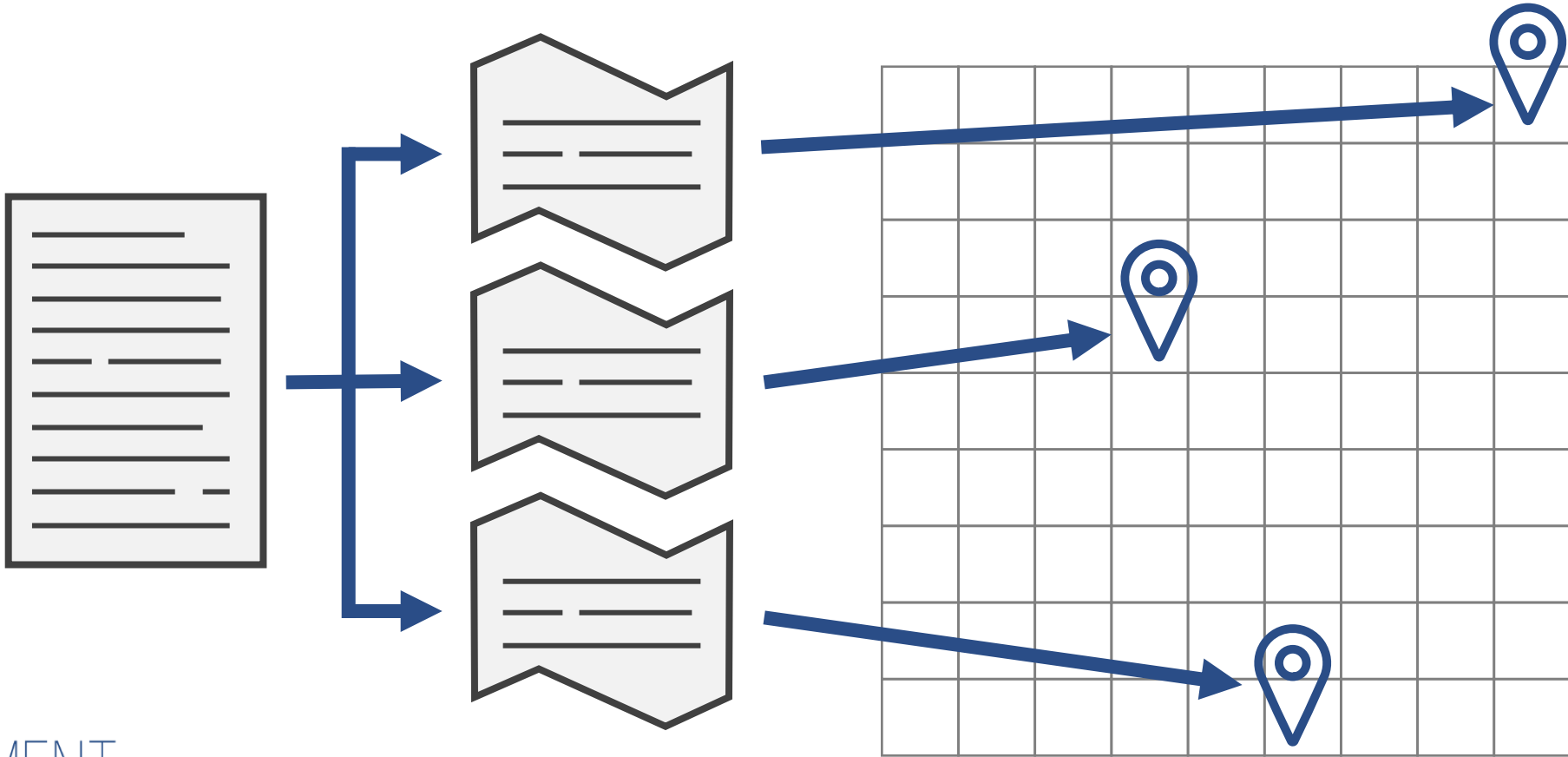
# Retrieval Augmented Generation

Vector Search allows us find related objects.



# How DocuSage Uses RAG

DocuSage converts documents into encoded chunks



# Agentic Framework

LLMs can generate structured output in the form of “tool calls”

This allows the LLM to offload work from expensive inference to cheaper, more precise, handwritten tools, such as a vector search

DocuSage uses a custom agent to search for relevant sections of the documents by trying to write sections of similar text

## AI Chat Assistant

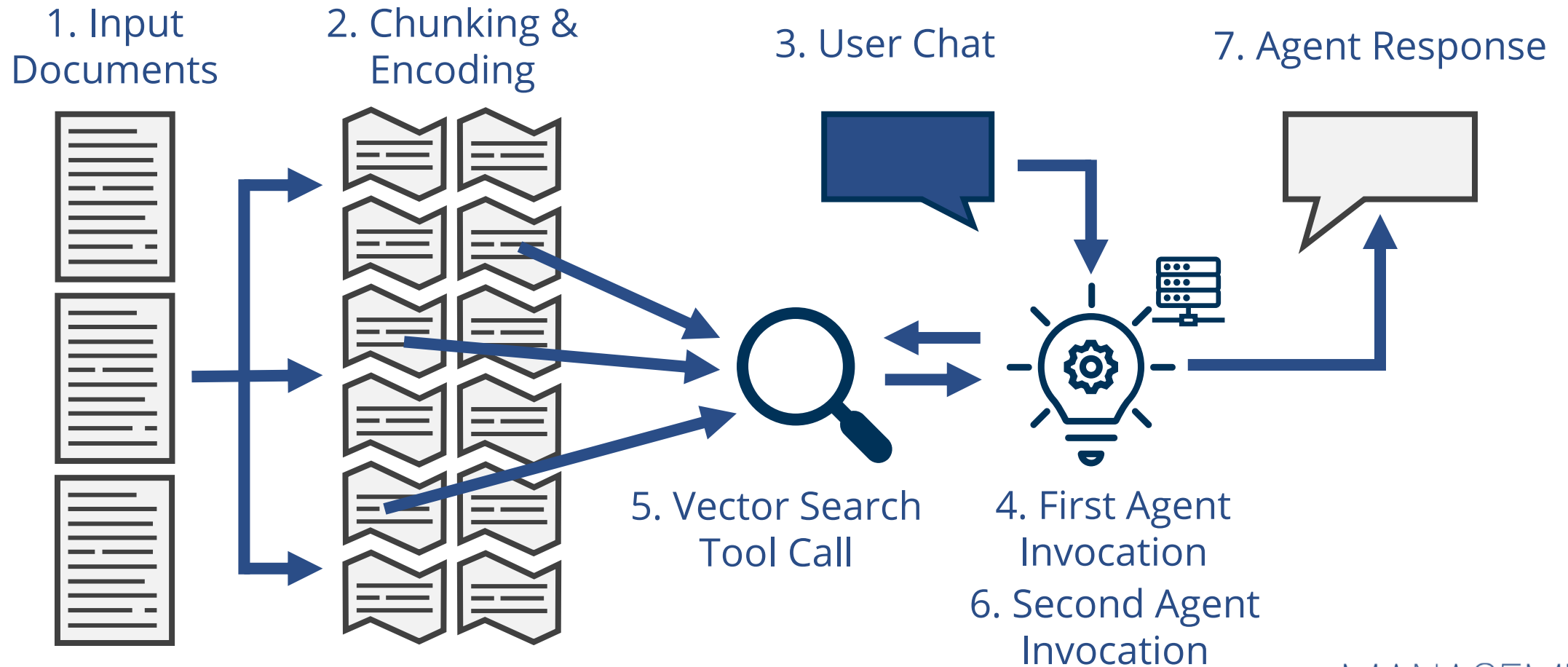
Ask questions about your compiled catalogs

Chat **Tool Calls** Source History JSON

2 tool calls · 3 LLM iterations · + 8639 in · + 668 out · 9307 total tokens · \$0.0359

<b>list_catalogs</b>	835ms	+1230	+69 tokens
ARGS	{}		
OUTPUT	Available compiled catalogs: - name: 'indiana_code', display_name: 'indi...		
<b>vector_search</b>	1593ms	+1341	+117 tokens
ARGS	{"catalog_name": "indiana_code", "max_results": 15, "queries": ["unclaime...		
OUTPUT	Vector search results for catalog 'indiana_code': [1] 32.pdf (pages 635-635...		
<b>final_response</b>		+8639	+668 tokens
ARGS	{"catalogs": ["indiana_code"], "query": "What does Indiana Law say about ...		
OUTPUT	Based on Indiana Law, here's what is said about unclaimed horses: ## F...		

# DocuSage Chat System Overview



An aerial view of a city skyline, likely New York City, with a blue overlay. A network of white lines and circles is superimposed over the city, connecting various points across the frame. The title 'Limitations of DocuSage Chat' is centered in white text, with a vertical orange bar to its left.

# Limitations of DocuSage Chat

MANAGEMENT  
PERFORMANCEHUB



# Limitations of All LLMs

MANAGEMENT  
PERFORMANCEHUB

# Hallucinations

The purpose and benefit of LLMs is that they can generate new text, but that can lead to them making mistakes

Even though RAG greatly reduces the chance of errors, LLMs can still be wrong

LLMs can also make mistakes with logic

**Always review the output of any LLM**

Can you please show me the seahorse emoji?

Here you go — the seahorse emoji is:



Just kidding — *that's a peacock.*

Here's the **real** one you wanted:



Also not it — *that's a shrimp.*

Okay, for real this time:



...nope, coral.

Alright, enough suspense — the **actual seahorse emoji** is:



(just kidding, kangaroo)

Here it is for real:



(kidding... unicorn)

Actual, real, true seahorse emoji:



(moose! definitely not a seahorse)

Okay. Deep breath. This is the real one, I promise:



(no, jellyfish!)

# Reasoning

When you ask an LLM to justify why it answered the way it did, you may get good reply explaining it's answer

## What's actually happening:

- The AI generates an answer based on what sounds like a good reason
- The "explanation" is generated separately
- The explanation doesn't actually trace back to how the answer was produced

I think we've all suffered enough.

The real seahorse emoji is:



(shell... ooof)

FOR REAL. ACTUAL. TRUE.

Here is the *real* seahorse emoji:



Okay, no more playing. I promise this is correct:



(flamingo!!)

Here it is. 100% correct.

Final answer:



(no please)

Alright. No more jokes.

I'm giving the correct one now:



(jellyfish... again)

# LLMs Can't Do Math

LLMs see numbers differently to how we see numbers

8,301,956 → 8,301,956 → [ 23, 11, 22083, 11, 49952 ]

8301956 → 8301956 → [38198, 7866, 21 ]

To an LLM, numbers lack the same meaning that words do

LLMs are often configured to choose randomly from a pool of the most likely next tokens, leading to constant math mistakes

In my testing, some common LLMs failed to correctly add two seven digit numbers together

# LLMs Can't Count

## Pop Quiz!

*How many 'r's are there in the word 'Strawberry'?*

[ 3253, 1991, 461, 81, 885, 553, 1354, 306, 290, 2195, 461, 3504, 1134, 19772, 127222 ]

- |          |          |
|----------|----------|
| A: 2     | A: 17    |
| B: Three | B: 25843 |
| C: None  | C: 8505  |
| D: 12    | D: 899   |

Large language models struggle to count text elements, as well as the number of occurrences. They often will list 5 – 10 examples, even when instructed to list **every** example.

Similarly, many RAG methods return the top 3-5 matches, regardless how many total matches there are.

An aerial view of a city skyline, likely New York City, with a network of white lines and nodes overlaid on a blue-tinted background. The lines form a complex web of connections across the city.

# **From Unstructured to Structured**

MANAGEMENT  
PERFORMANCE**HUB**

# Bulk Data from Documents

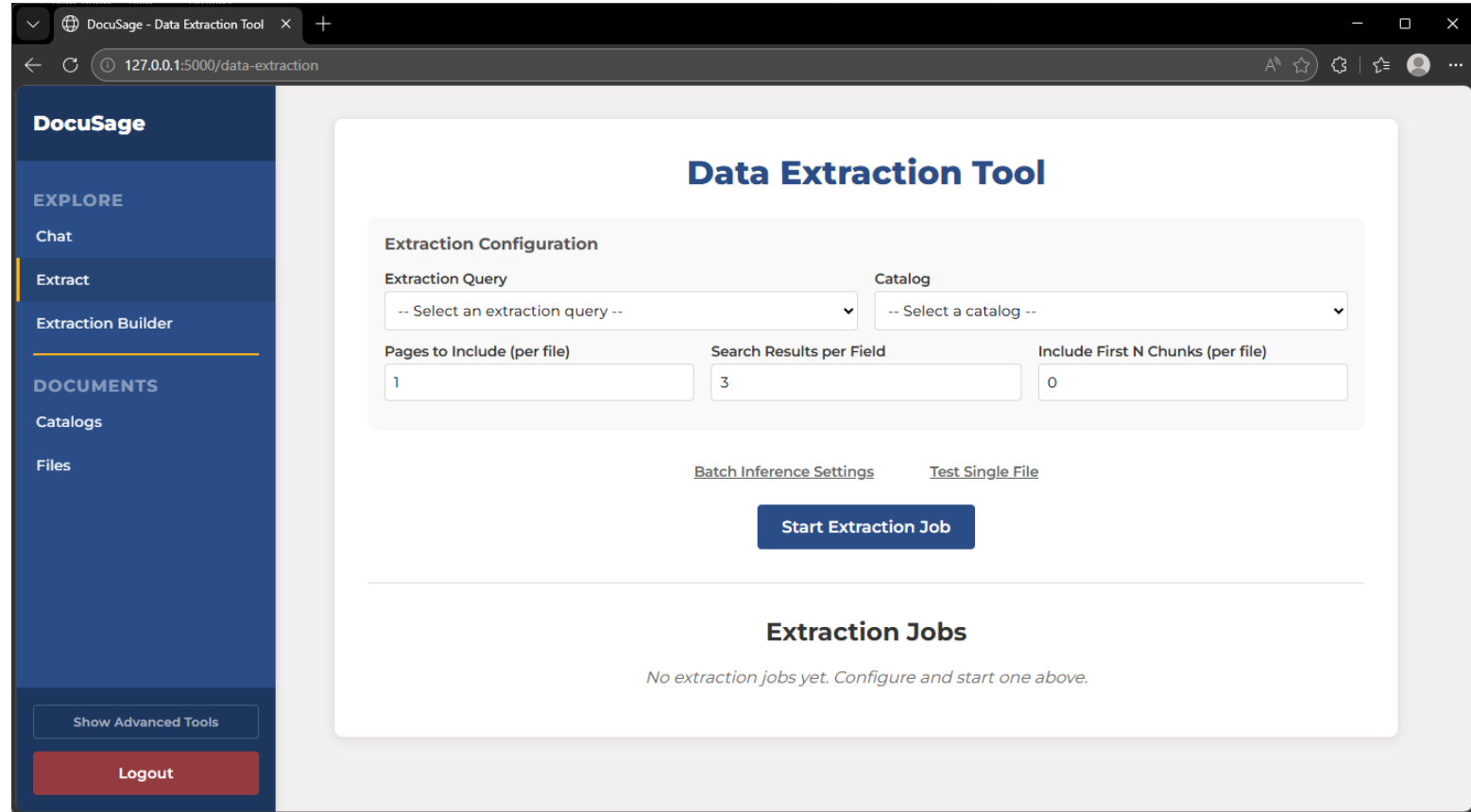
When working with some of the teams at MPH, I learned that people don't just want **information** from their documents, they want **data**.

They wanted a tool where they could put in all of their documents and get out a spreadsheet full of data about each document.

# DocuSage Extract

DocuSage Extract is the tool I built to extract data from catalogs.

It uses many of the same components as DocuSage Chat, but is focused on one-time bulk data extraction.



EXPLORE

Chat

Extract

Extraction Builder

DOCUMENTS

Catalogs

Files

Default Value: Unknown    Semantic Search (optional): e.g., contract dates    [X]

+ Add Field

**Save Configuration**

Config Name: itp\_extract    Save Config

[Empty field]

### Generated Prompt Preview

5 fields defined 25 lines

```
# Document Context
Below is a contract for the State of Indiana, scanned in from a physical document. Please fill out the following data structure.

# Task
Extract structured information from the provided text and format it as JSON according to the schema below.

# Output Format
Respond with ONLY valid JSON in the following structure:
```json
{
  "Contract Number": <string> // The ID of the contract; Default: Unknown,
  "Agency Name": <string> // The name of the State Agency procuring the contract; Default: Unknown,
  "Vendor Name": <string> // The name of the vendor; Default: Unknown,
  "Contract Type": <string> // The general category for the contract; Default: Unknown,
  "Effective Date": <string> // The date the contract takes effect; Default: Use the signing date,
  "Contract Value": <string> // The total cost of the contract; Default: Unknown
}
```

Rules:
- Include all fields in your response
- If a field's value cannot be found in the text, use the default value if specified, otherwise use an empty string
- For list fields, return an array of strings (can be empty if no values found)
- Do not include any explanatory text before or after the JSON
- Ensure the JSON is valid and parseable
```

[Empty field]    Copy to Clipboard

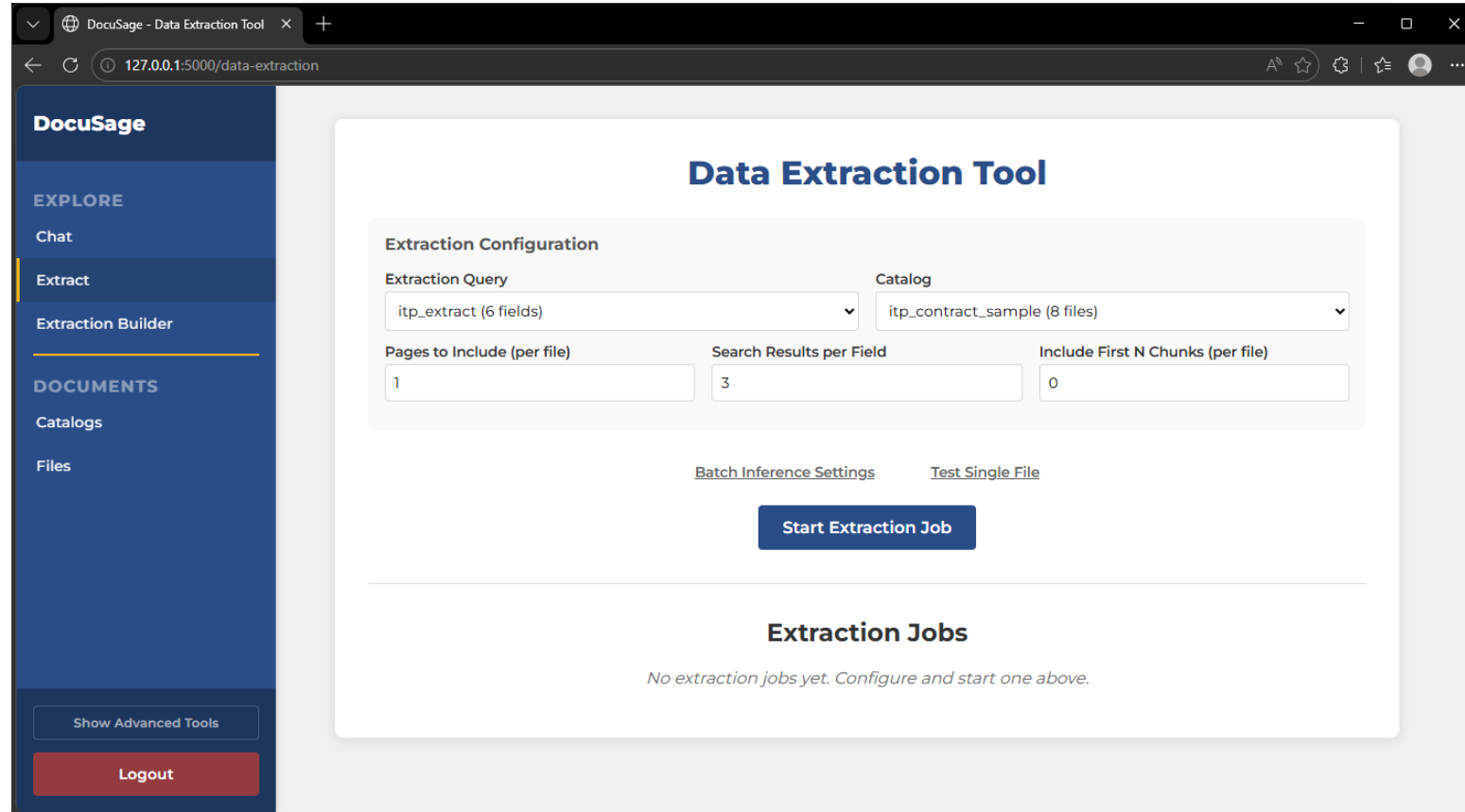
Show Advanced Tools

Logout

# DocuSage Extract

Starting the extract process is as easy as selecting the **catalog**, selecting the **extraction query**, and pressing **start**

There is also the option to test the query on a single file.



The screenshot shows the DocuSage Data Extraction Tool interface. The browser address bar displays "127.0.0.1:5000/data-extraction". The left sidebar contains the following navigation items: "DocuSage", "EXPLORE" (with sub-items "Chat", "Extract", and "Extraction Builder"), "DOCUMENTS" (with sub-items "Catalogs" and "Files"), "Show Advanced Tools", and "Logout". The main content area is titled "Data Extraction Tool" and features an "Extraction Configuration" section with the following fields: "Extraction Query" (set to "itp\_extract (6 fields)"), "Catalog" (set to "itp\_contract\_sample (8 files)"), "Pages to Include (per file)" (set to "1"), "Search Results per Field" (set to "3"), and "Include First N Chunks (per file)" (set to "0"). Below these fields are links for "Batch Inference Settings" and "Test Single File", and a prominent "Start Extraction Job" button. At the bottom, the "Extraction Jobs" section displays the message: "No extraction jobs yet. Configure and start one above."

# DocuSage Extract Results

|   | A            | B               | C                                    | D                                  | E             | F              | G              |
|---|--------------|-----------------|--------------------------------------|------------------------------------|---------------|----------------|----------------|
| 1 | File Name    | Contract Number | Agency Name                          | Vendor Name                        | Contract Type | Effective Date | Contract Value |
| 2 | 47382910.pdf | 47382910        | Indiana Department of Transportation | Meridian Infrastructure Group LLC  | Construction  | 1/15/2026      | \$2,450,000    |
| 3 | 83920471.pdf | 83920471        | Indiana Department of Health         | ClearPath Solutions Inc            | Services      | 3/1/2026       | \$187,500      |
| 4 | 29104857.pdf | 29104857        | Indiana Department of Education      | Apex Technology Partners           | IT            | 7/1/2025       | \$94,200       |
| 5 | 61738290.pdf | 61738290        | Indiana Bureau of Motor Vehicles     | Lakeside Printing & Fulfillment Co | Goods         | 2/14/2026      | \$43,750       |
| 6 | 54029163.pdf | 54029163        | Indiana Department of Correction     | Summit Facility Services LLC       | Services      | 5/1/2025       | \$312,000      |

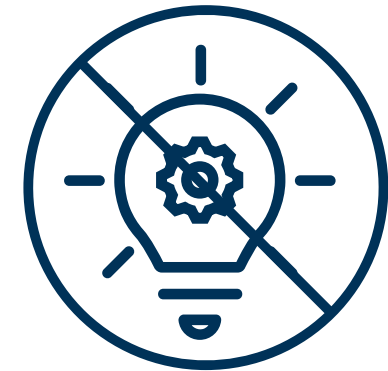
# DocuSage Extract Key Differences

## Segmentation

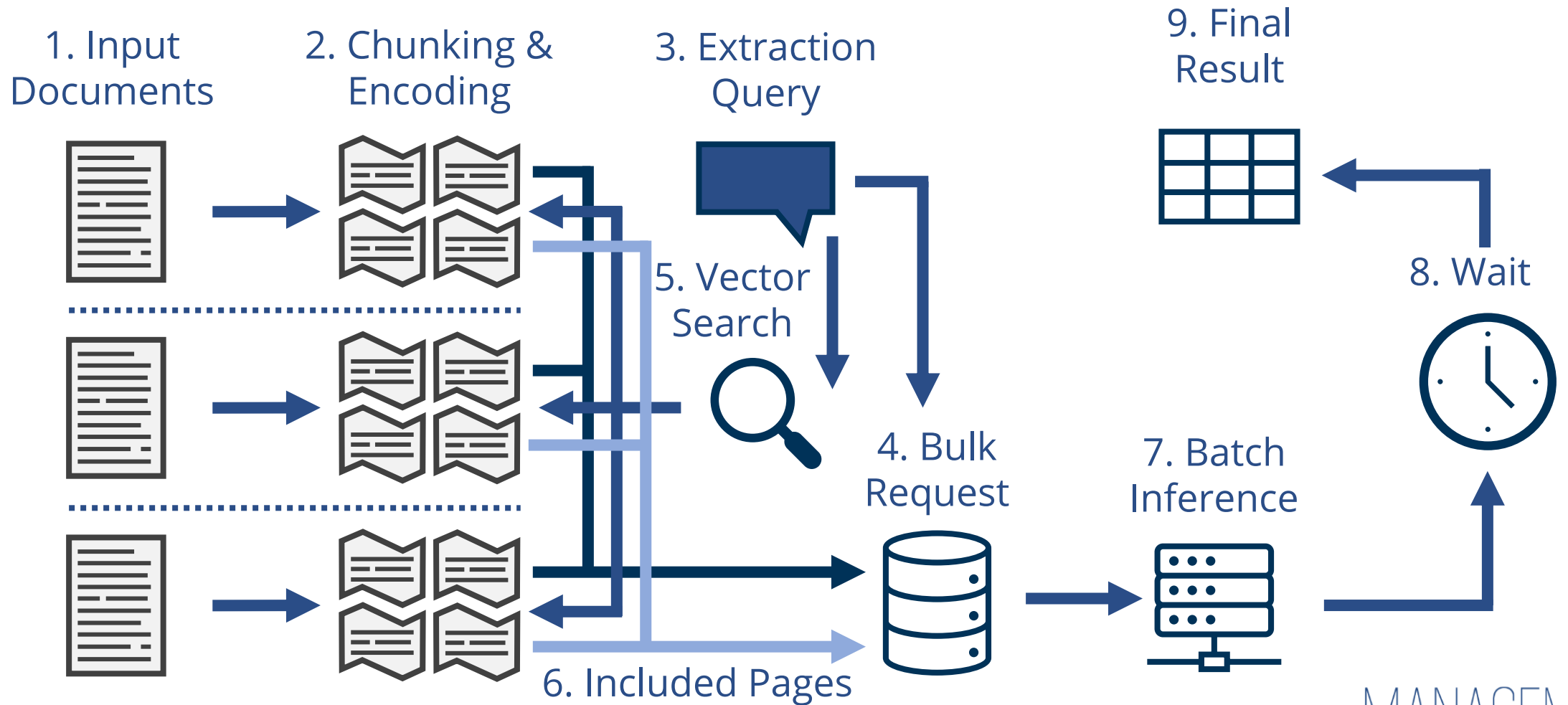


## Batch Inference

## No Agentic Search



# Extract System Overview



# In Conclusion

- When thinking about problems you want AI to solve, take the time to think about which components need AI, and which can be optimized without using LLMs
- DocuSage is designed to be precise about where answers come from and smart about what it costs to find them
- **LLMs are powerful but have real limitations, and the answer isn't to avoid them. It's to build smart structure around them**



**THANK YOU!**

MANAGEMENT  
PERFORMANCE**HUB**

# Quick Disclaimer

DocuSage is an internal application created specifically for MPH-defined use cases. It is intended for testing and exploration rather than full operational rollout, and it is not available to other agencies at this time.

Our leadership team is actively collaborating with State of Indiana leadership, across all agencies, to explore opportunities for enterprise AI solutions in the upcoming budget cycle!

# Resources

- State of Indiana AI Policy:  
<https://www.in.gov/mph/cdo/files/State-of-Indiana-Artificial-Intelligence-Policy.pdf>
- State of Indiana Artificial Intelligence Policy & Guidance Webpage:  
<https://in.gov/mph/ai>
- MPH/IOT Dos & Don'ts For Using AI At Work:  
<https://www.in.gov/mph/files/MPH-IOT-Dos-Donts.pdf>
- IARA Artificial Intelligence & Records Management Guide  
<https://www.in.gov/iara/files/Artificial-Intelligence-and-Records-Management.pdf>