Enhancing Drupal Data Imports with LLMs

Jeremy Barth - https://github.com/criolho/Drupal-Feeds-Using-LLMs

Q. What is this talk about?

We will import 2 sources of Environmental Protection Agency (EPA) data into a custom Drupal site. We'll go beyond traditional web scraping by using an LLM to analyze and supplement the scraped data.

Q. Why would we want to do this?

PDFs and websites are valuable but **unstructured** (from our point of view) sources of data. Drupal, on the other hand, is all about **structured content**. LLMs help us add value over and above what the EPA provides: get better summaries, extract legal citations, get penalty info and assign categories based on our own taxonomy.

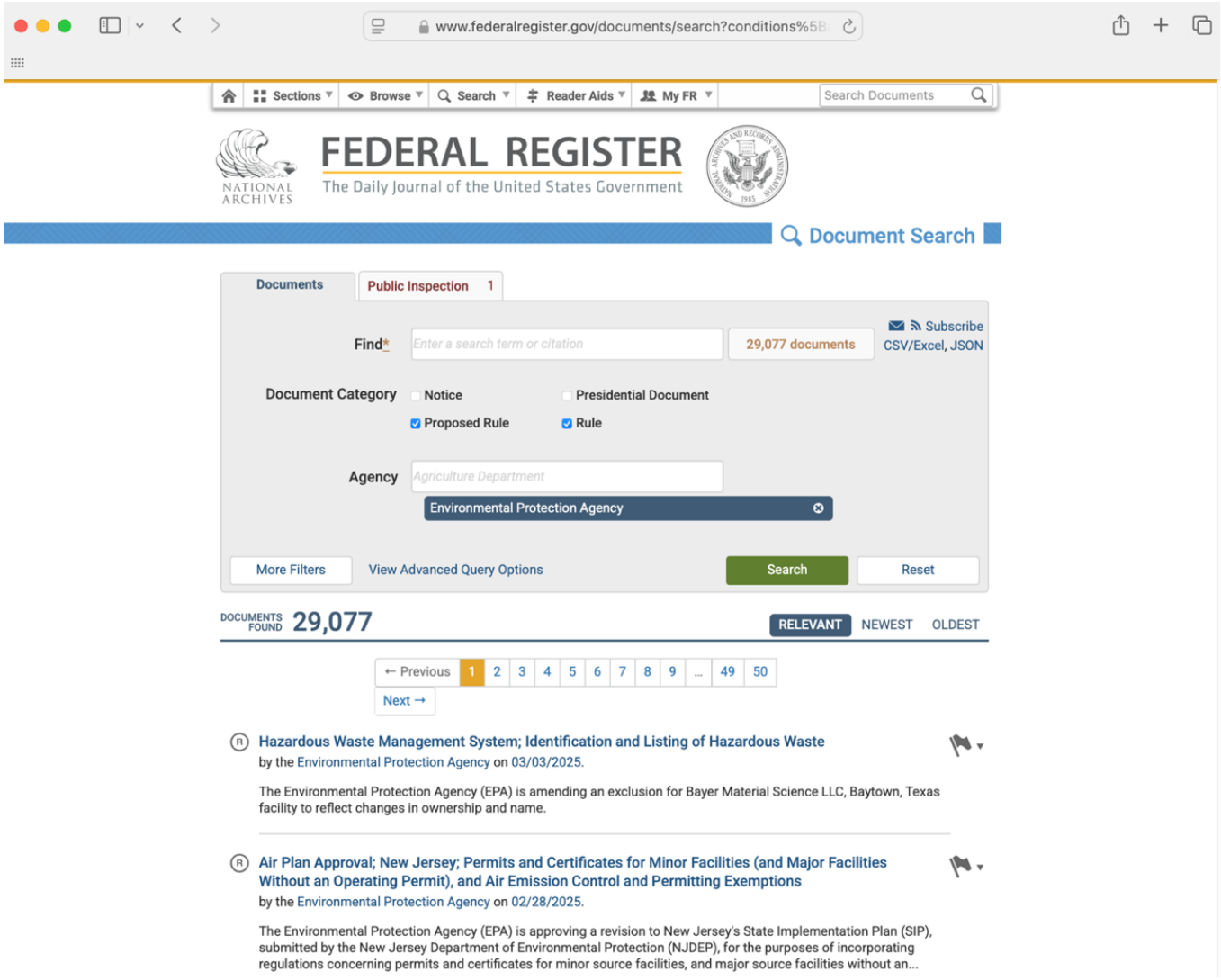
Q. How do we import the data and in what form?

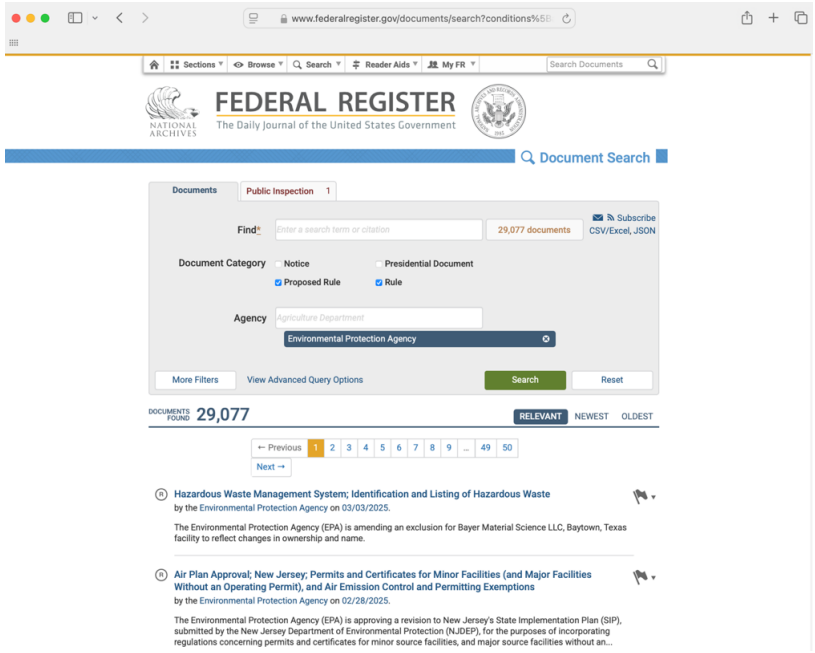
- **Import** - there are many ways to bring content into Drupal, among others webhooks and the Migrate and Feeds modules. Feeds are a simple way to automate routine, scheduled imports and they have an intuitive UI.
- **Data format** - JSON is easy to read by both humans and machines. LLMs also like it.

Q. What are our data sources?

One site we'll be sourcing has an API while the other requires some web scraping.

Case Study #1 – Federal Register





Site: https://federalregister.gov

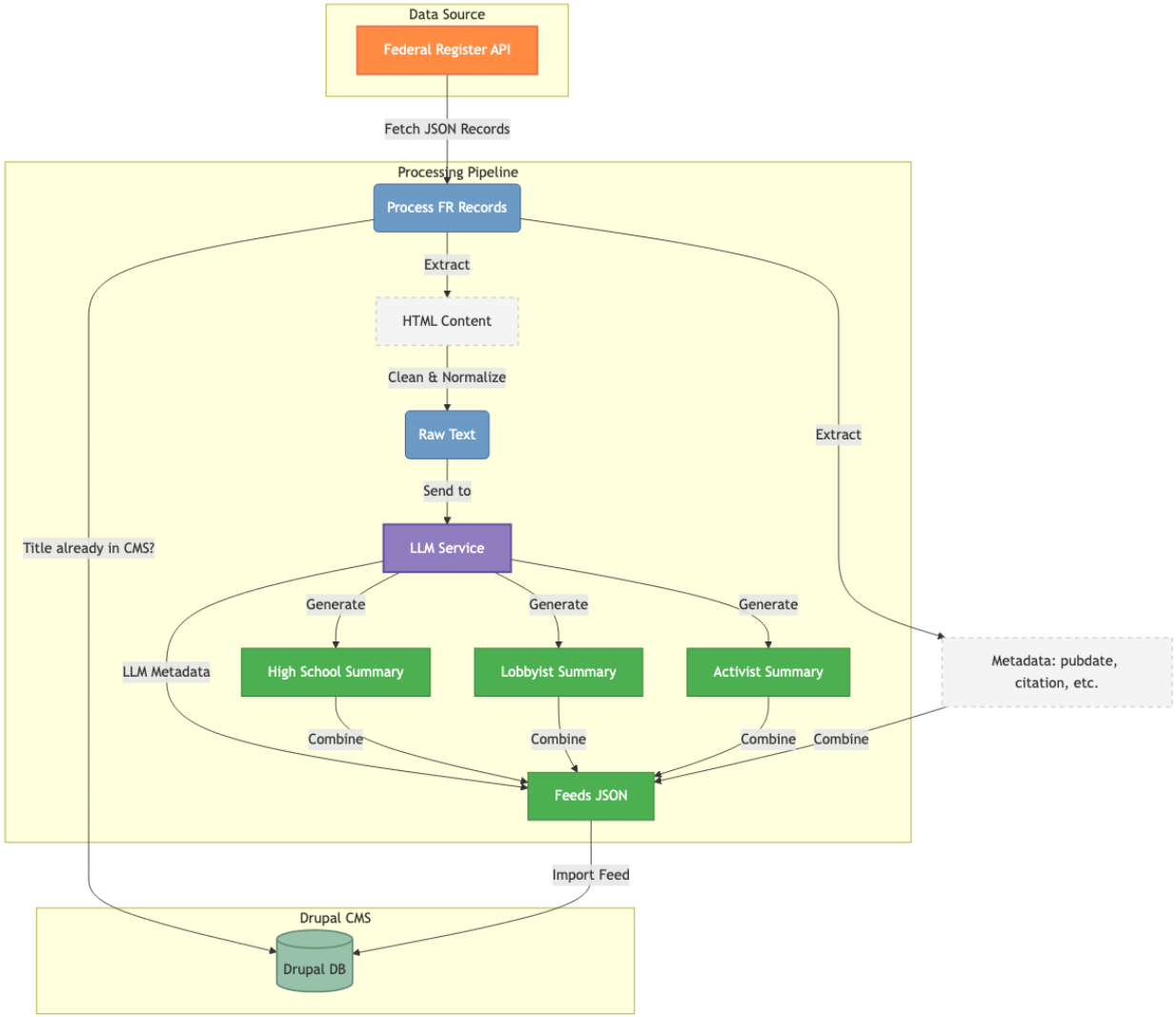
Background: This is where the U.S. government publishes all proposed federal regulations or rules about to go into effect. The site has an excellent UI and is also known for its developer-friendly API and JSON output.

Goals: connect to Federal Register API and get JSON metadata for recently-published rules. Then:

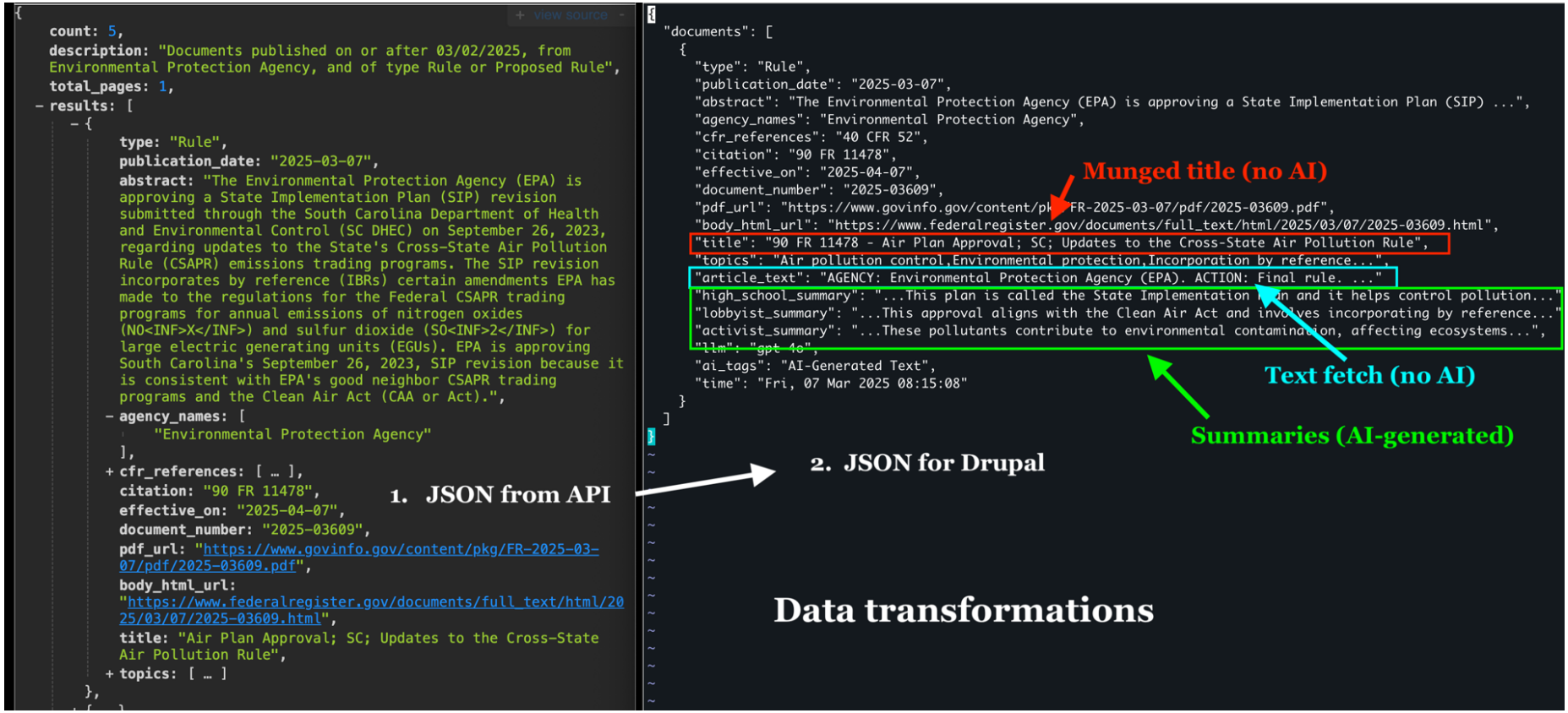
- Make minor transformations to their field data for our matching fields in Drupal, e.g. prepending a citation to their title to match our Drupal node title best practices
- Add 3 LLM-generated summaries individually focused on the needs of: 1) high school students, 2) corporate lobbyists, 3) environmental activists.

Federal Register processing pipeline

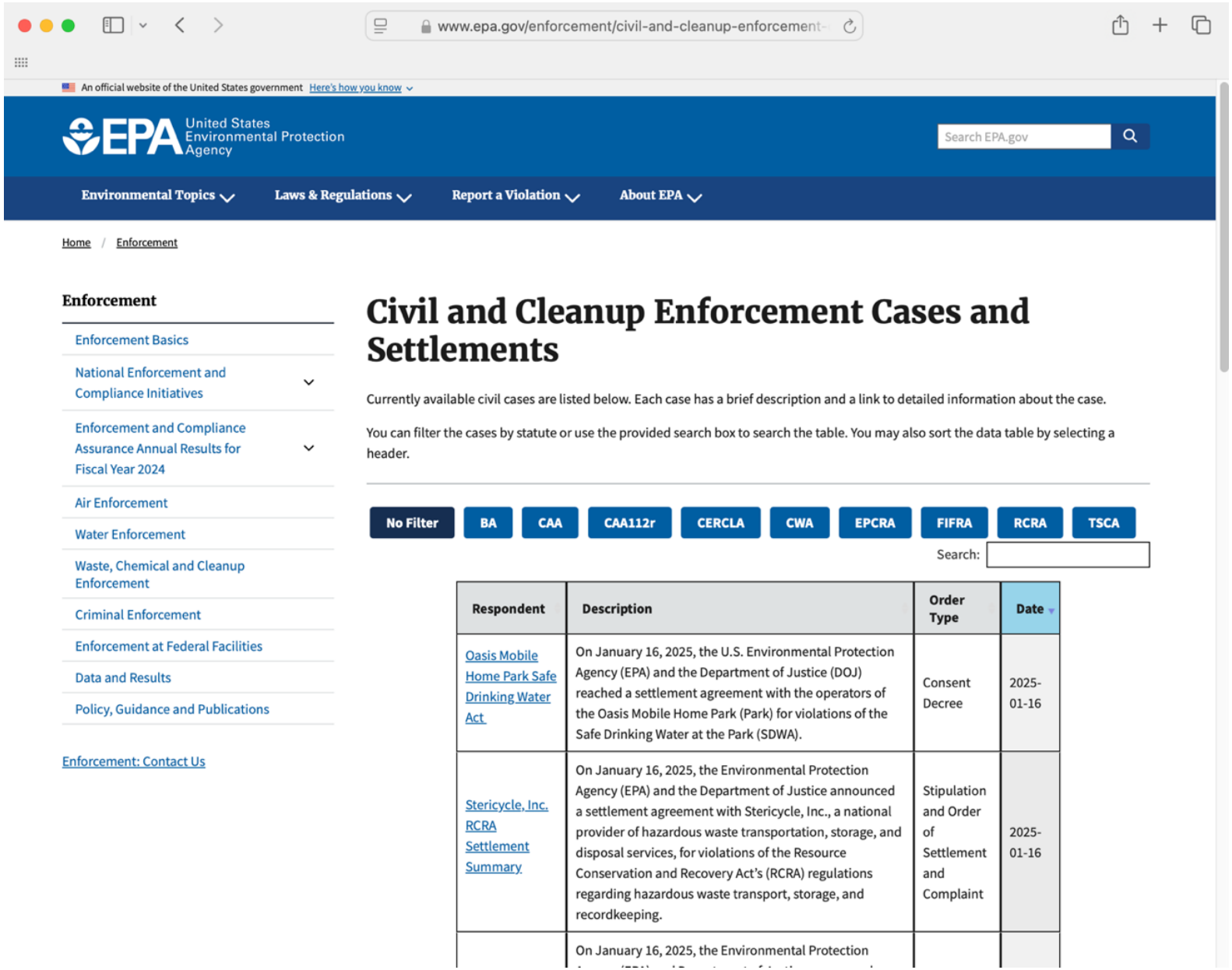
High level overview of the data processing game plan:

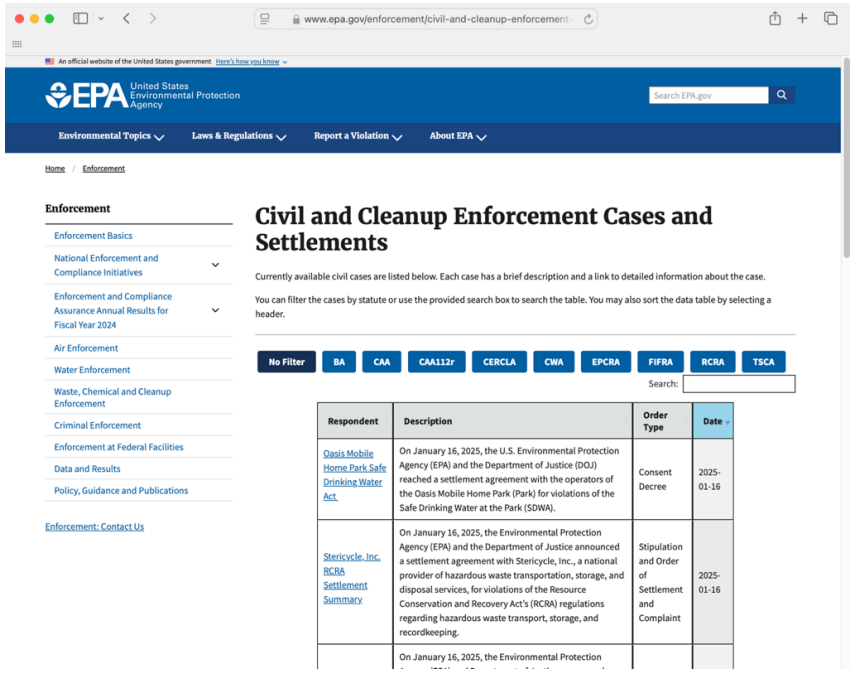


Inputs and Outputs



Case Study #2 – Civil Enforcement Actions by the EPA


www.epa.gov/enforcement/civil-and-cleanup-enforcement-
An official website of the United States government Here's how you know
EPA United States Environmental Protection Agency
Search EPA.gov
Environmental Topics
Laws & Regulations
Report a Violation
About EPA
Home / Enforcement
Enforcement
Enforcement Basics
National Enforcement and Compliance Initiatives
Enforcement and Compliance Assurance Annual Results for Fiscal Year 2024
Air Enforcement
Water Enforcement
Waste, Chemical and Cleanup Enforcement
Criminal Enforcement
Enforcement at Federal Facilities
Data and Results
Policy, Guidance and Publications
Enforcement: Contact Us
Civil and Cleanup Enforcement Cases and Settlements
Currently available civil cases are listed below. Each case has a brief description and a link to detailed information about the case.
You can filter the cases by statute or use the provided search box to search the table. You may also sort the data table by selecting a header.
No Filter
BA
CAA
CAA112r
CERCLA
CWA
EPCRA
FIFRA
RCRA
TSCA
Search:
Respondent
Description
Order Type
Date
Oasis Mobile Home Park Safe Drinking Water Act
On January 16, 2025, the U.S. Environmental Protection Agency (EPA) and the Department of Justice (DOJ) reached a settlement agreement with the operators of the Oasis Mobile Home Park (Park) for violations of the Safe Drinking Water at the Park (SDWA).
Consent Decree
2025-01-16
Stericycle, Inc. RCRA Settlement Summary
On January 16, 2025, the Environmental Protection Agency (EPA) and the Department of Justice announced a settlement agreement with Stericycle, Inc., a national provider of hazardous waste transportation, storage, and disposal services, for violations of the Resource Conservation and Recovery Act's (RCRA) regulations regarding hazardous waste transport, storage, and recordkeeping.
Stipulation and Order of Settlement and Complaint
2025-01-16
On January 16, 2025, the Environmental Protection



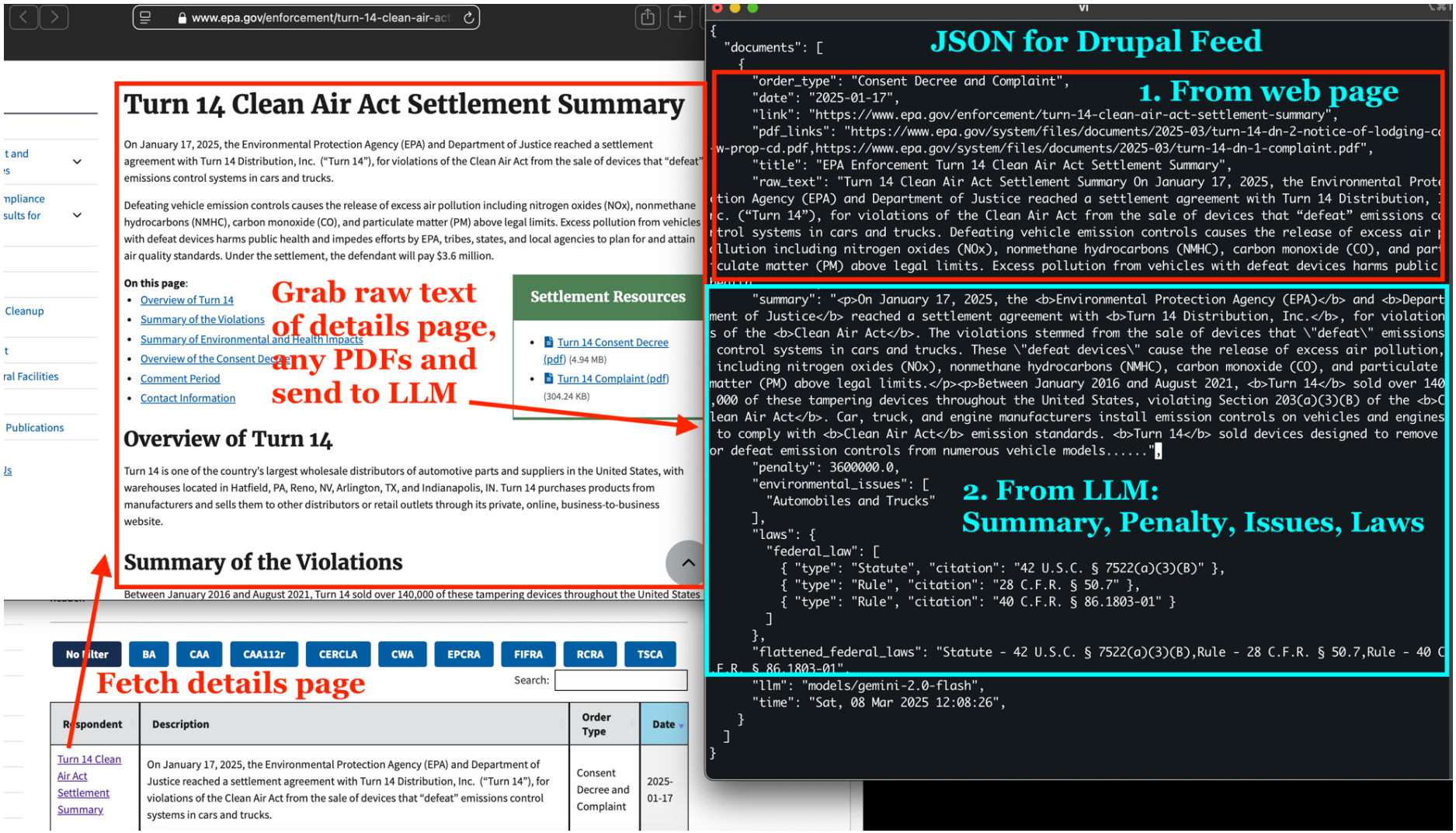
Site: https://www.epa.gov/enforcement/civil-and-cleanup-enforcement-cases-and-settlements

Background: Centralized source for EPA civil actions for violations. The EPA [historically] enforces a wide range of environmental laws, including the Clean Air Act (CAA) and Clean Water Act (CWA).

Goals:

- Better summaries
- Extract penalty info
- What specific laws were violated?
- Categorize environmental issues

Inputs and Outputs



LLM considerations

Closed models, open source – they're constantly leapfrogging one another. At present, closed source foundation models produce more **reliable structured output**. This may change in the future.

The code here has options for OpenAI's "gpt-4o", Anthropic's "claude-3-7-sonnet-latest" and Google's "gemini-2.0-flash". Currently we default to **Gemini Flash** because it is fast, reliable, inexpensive (as of March 2025: input $0.10 / 1,000,000 input tokens, $0.40 / 1,000,000 output tokens) and has a long context window (1,000,000 tokens) that can readily accommodate multi-hundred page PDFs.

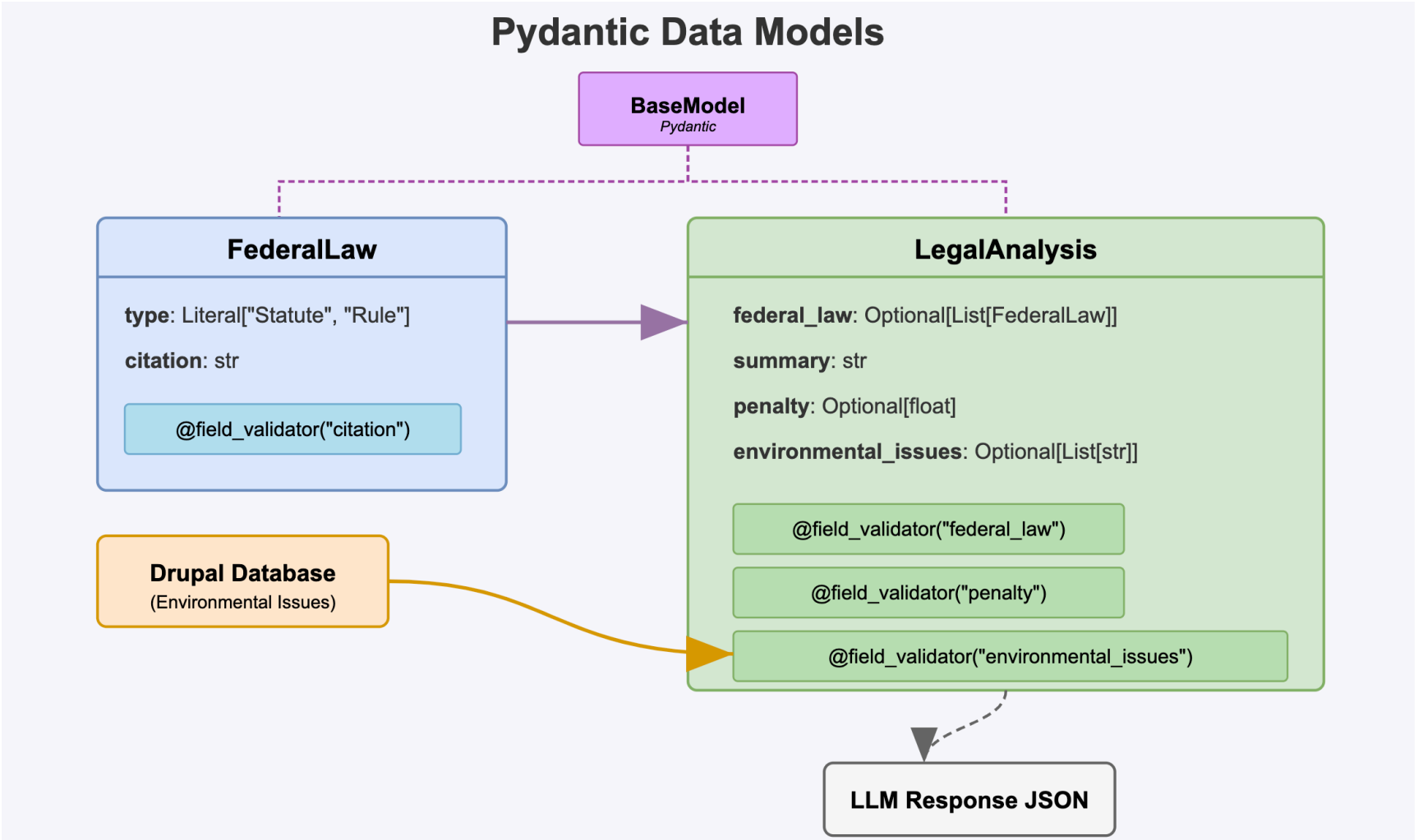
Optimization – a custom title_exists() function checks to see if we already have a node in Drupal, saving on unnecessary LLM calls

Pydantic and Instructor

Pydantic and Instructor are Python libraries that work together to: a) tell LLMs how you want their reply to be structured, e.g. JSON with certain fields and types of data; b) validate the data before passing it on for downstream use. In particular, we want anything GenAI-related to pass muster before we import it to our Drupal CMS.

You define a “data model” and Pydantic helps ensure that it’s followed, e.g. that certain fields must be present or are merely optional. You can also defined custom validation methods.

Pydantic Data Models



FederalLaw

Represents individual federal law citations
Either statutes (U.S.C.) or rules (C.F.R.)
Validated to ensure correct format

LegalAnalysis

Top-level analysis container
Holds structured data from legal documents
Used for document categorization and summary

Pydantic Validation Flow

Field validators normalize and check values against constraints
Drupal database provides a realtime source of valid categories
Converts empty lists to None for consistency in JSON output

Pydantic Field validators

These are optional functions you can write that automatically run against fields before Pydantic signs off on the data. You can use them to simply confirm that data matches certain criteria, or you can modify the data to conform your needs. The point is that this is a **structured way of enforcing standards** for your data.

What we're choosing to validate:

- **citation** – make sure legal citations are in a standardized format, e.g. 40 C.F.R. §§ Part 1039" should be transformed to "40 C.F.R. § 1039
- **penalty** – make sure it's a numeric float with at most 2 decimal places
- **environmental_issues** – suppose we have a Drupal taxonomy we're already using. We can fetch the terms dynamically from the Drupal DB and use them both to tell the LLM what terms we want it to look for and then to make sure that's what it did.

Miscellaneous “Best Practices”

There are few “standards” yet for how websites should manage AI-generated content. You may want to consider:

- Creating a vocabulary “AI Tags” to help you keep track of nodes to which you’ve applied GenAI. For example: AI-Generated Text, AI-Generated Categories, or AI-Generated Entity Extraction
- It is quite likely that you’ll use different LLMs over time. You may want to have a vocabulary for these as well with terms such as: claude-3-7-sonnet-latest, gpt-4o, gpt-4o-mini, gemini-2.0-flash
- If you’re going to the trouble of extracting lots of raw text for use in GenAI, even though in its raw state it may not be suitable for end users it might be good for a) fulltext search; b) future GenAI passes over the same nodes
- People are justifiably wary of what’s being pushed on them – consider including preamble / info text such as “This article contains an AI-generated summary” that fully informs people what they’re getting.