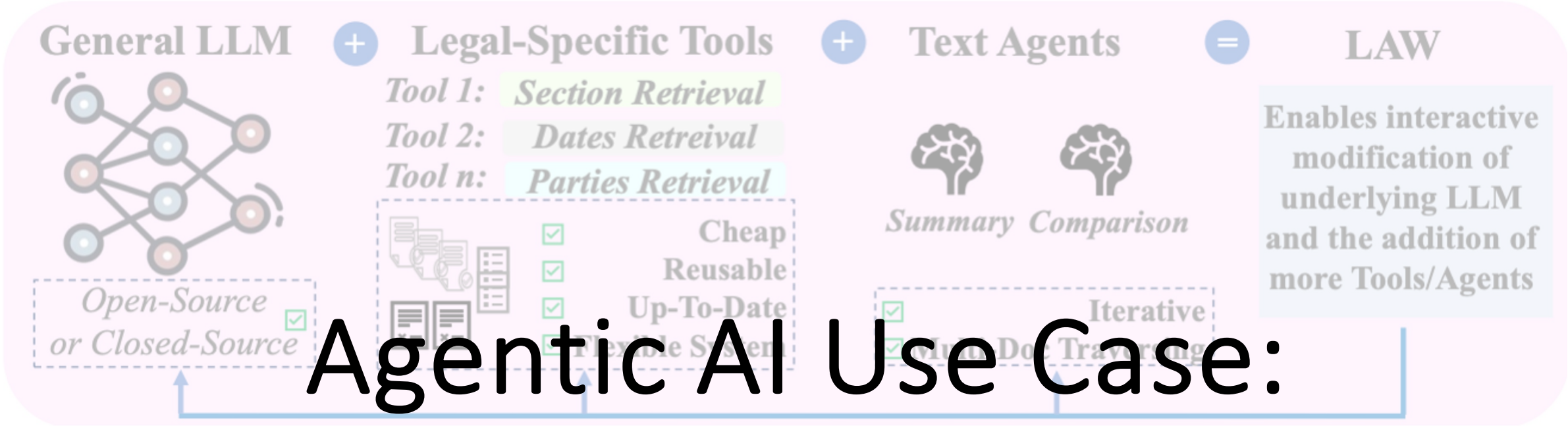


Agentic AI Use Case: Agentic Workflows for Legal Contracts

Yan Xu
Houston Machine Learning

LAW: Legal Agentic Workflows for Custody and Fund Services Contracts

William Watson*, **Nicole Cho***,
Nishan Srishankar*, **Zhen Zeng**, **Lucas Cecchi**, **Daniel Scott**,
Suchetha Siddagangappa, **Rachneet Kaur**, **Tucker Balch**, **Manuela Veloso**
J.P. Morgan AI Research
New York, New York, USA
nicole.cho@jpmorgan.com

FlowMind: Automatic Workflow Generation with LLMs

Zhen Zeng
J. P. Morgan AI Research
New York, NY, USA
zhen.zeng@jpmchase.com

William Watson
J. P. Morgan AI Research
New York, NY, USA
william.watson@jpmchase.com

Nicole Cho
J. P. Morgan AI Research
New York, NY, USA
nicole.cho@jpmorgan.com

Saba Rahimi
J. P. Morgan AI Research
New York, NY, USA
saba.rahimi@jpmorgan.com

Shayleen Reynolds
J. P. Morgan AI Research
New York, NY, USA
shayleen.reynolds@jpmchase.com

Tucker Balch
J. P. Morgan AI Research
New York, NY, USA
tucker.balch@jpmchase.com

Manuela Veloso
J. P. Morgan AI Research
New York, NY, USA
manuela.veloso@jpmchase.com

Use Case: Custody/Fund Service Contract

CLIENT:	TCW ETF Trust
Attention:	Peter Davidson
Address:	515 South Flower Street, Los Angeles, CA 90071
Telephone No:	(213) 244-0533
Email:	peter.davidson@tcw.com
CUSTODIAN:	STATE STREET BANK AND TRUST COMPANY
Attention:	Andrea Sharp
Address:	2495 Natomas Park Drive, Ste. 400, Sacramento, CA 95833
Telephone No:	(916) 319-6688
Email:	andrea.sharp@statestreet.com
with a copy to:	STATE STREET BANK AND TRUST COMPANY 1 Congress Street, Boston, MA 02114-2016 Attention: Legal Department

Agreement between Client (trust of funds) and Custodian (bank)

CUSTODY AGREEMENT

Custody Agreement

This Agreement (the "Agreement") is made as of [●], 2024 (the "Effective Date") **between**:

(1) Each entity identified on Appendix A, whose jurisdiction of formation is identified opposite its name (the "Client"); and

(2) **STATE STREET BANK AND TRUST COMPANY**, a bank and trust company organized under the laws of The Commonwealth of Massachusetts, U.S.A. (the "Custodian").

1 Definitions and Interpretation

Defined terms and the general rules of interpretation agreed by the Parties are set forth in Schedule 1.

2 Appointment of the Custodian

The Client hereby appoints the Custodian to provide the services set out in Sections 3 through 15 below (the "Services") subject to and in accordance with the terms of this Agreement.

3 Safekeeping Securities

3.1 **Holding Securities.** The Custodian will hold Securities delivered or credited to its account under this Agreement directly or through accounts at Subcustodians or CSDs. In turn, Subcustodians will hold Securities directly or through accounts at CSDs.

3.2 **Client Entitlements and Segregation.** The Custodian will take the following steps to reflect the Client's ownership of Securities and to separately identify the Securities of the Client from the proprietary assets of the Custodian, Subcustodians, and CSDs, in accordance with Local Market Practice:

3.2.1 **Accounts at the Custodian.** Open and maintain on the records of the Custodian one or more securities accounts in the name of the Client or such other name as the Client may reasonably request (each, a "Securities Account") and credit Securities to them;

3.2.2 **Accounts at the Subcustodians or CSDs.** Open and maintain securities accounts at the Subcustodians or CSDs in which the Custodian is a direct participant, cause Subcustodians to open and maintain securities accounts at CSDs in which the Subcustodian is a participant, and cause Securities to be credited to the relevant accounts. Such accounts: (i) may be commingled (or omnibus) accounts for Securities of multiple customers of the Custodian (or Subcustodian, in the case of accounts opened by the Subcustodian at a CSD) or, in limited markets, segregated (or separate) accounts for Securities of the Client; and (ii) must not include any proprietary securities of the Custodian, the Subcustodian or the CSD;

3.2.3 **Physical Securities.** Physically segregate bearer Securities from the proprietary assets of the Custodian, and require that the Subcustodians physically segregate bearer Securities from the Subcustodian's and the Custodian's proprietary assets;

3.2.4 **Registration Names.** Register certificated Securities (other than bearer securities) in the name of the Client or in the name of the Custodian, a Subcustodian, a CSD or a nominee of any of them, or otherwise in accordance with Local Market Practice and the laws and regulations applicable to the Custodian; and

3.2.5 **Records of Transactions; Reconciliation.** Maintain records of the Client's transactions in the Securities Accounts and reconcile its records of clients' securities holdings against the records of its Subcustodians and CSDs in which it is a direct participant in accordance with the Custodian's standard procedures and Local Market Practice. Subcustodians will likewise maintain records of their client's transactions and reconcile their records of the securities holdings of their clients against the records of the CSDs in which they are a direct participant in accordance with the Subcustodians' standard procedures and Local Market Practice.

3.3 **Securities Interchangeable.** Securities of the Client (whether held in separate or commingled accounts) are fungible with all other securities of the same issue held in such accounts by the Custodian and its Subcustodians. This means that the Client's redelivery rights in respect of the Securities are not in respect of the Securities actually deposited with the Custodian or a Subcustodian from time to time, but rather in respect of Securities of the same number, class, denomination and issue as those Securities.

3.4 **Acceptance of Securities.** Except as otherwise agreed in writing with the Client, the Custodian will only accept custody of Securities and other assets that it is operationally equipped and licensed to hold in the relevant market where it provides custodial services either directly or through an existing Subcustodian and may decline to accept custody of certain securities or asset types that it determines present an unacceptable risk profile or that it or its Subcustodians are not operationally equipped or permitted to hold under any law or regulation.

Amendment

AMENDMENT TO CUSTODIAN AGREEMENT

THIS AMENDMENT TO CUSTODIAN AGREEMENT (this "**Amendment**") is made as of July [ ], 2023 by and between **BROWN BROTHERS HARRIMAN & CO.**, a New York limited partnership ("**BBH&Co.**") and **BONDBLOXX ETF TRUST,** a Delaware statutory trust (the **"Fund**") on behalf of each series, separately and not jointly, listed on Annex A to the Agreement as may be amended from time to time. All capitalized terms used but not defined herein shall have the meanings set forth in the Agreement.

WHEREAS, BBH&Co. and the Fund are Parties to a Custodian Agreement dated as of November 29, 2021, as amended (the "**Agreement**"); and

WHEREAS, in accordance with Section 12 of the Agreement, the Parties desire to amend the Agreement.

NOW, THEREFORE, in consideration of the mutual agreements herein contained and other good and valuable consideration, the receipt and sufficiency of which is hereby acknowledged by the parties, each of the parties hereto hereby agree as follows:

1. Annex A to the Agreement is hereby deleted in its entirety and replaced with the attached Annex A. The Parties agree, ratify and affirm that the Funds listed on Annex A attached hereto are and have been added to the Agreement on and as of the respective date indicated.

2. As amended hereby, all terms and provisions of the Agreement are hereby ratified and affirmed as of the date hereof and are hereby extended to give effect to the terms hereof.

3. This Amendment, together with the Agreement, constitutes the entire agreement of the parties with respect to its subject matter and supersedes all oral communications and prior writings with respect hereto. Except as expressly modified hereby, the Agreement shall continue in full force and effect in accordance with its terms and conditions.

4. This Amendment may be executed in any number of counterparts each of which shall be deemed to be an original, but all of which together shall constitute one and the same Amendment.

5. This Amendment shall be construed in accordance the governing law and exclusive jurisdiction provisions of the Agreement.

[Signature Page Follows]

IN WITNESS WHEREOF, each of the undersigned parties has executed this Amendment to the Agreement effective as of the date first above written.

BROWN BROTHERS HARRIMAN & CO.

By: _____
Name:
Title:
Date:

BONDBLOXX ETF TRUST

By: _____
Name: Joanna Gallegos
Title: President, BondBloxx ETF Trust
Date: July [ ], 2023

Use Case: Querying Contracts

Direct extraction

- Effective date
- Master date

Multi-hop reasoning

- Determine termination date
- Retrieve master contract if it is an amendment

Analytical tasks

- Summarize clauses
- Compare clauses across multiple contracts or different time periods

Clause: a specific section or provision that addresses a particular aspect of the agreement or law

Query examples

▶️ Find the termination dates of all contracts from custodian Goldman Sachs.

▶️ Find the master dates of contracts between Trust Investor Counselor Series Fund Inc and Custodian State Street Bank and Trust.

▶️ Compare the fees and expenses clause of the previous contracts.

Domain complexity

- Lengthy unstructured text streams: Legal contracts are often long and lack explicit section declarations
- Limited LLM context windows (maybe of less concern with latest LLM with 1 million token context window): The extensive length of legal documents can exceed the context limits of general-purpose LLMs
- Complex legal jargon: The dense and nuanced language of legal contracts requires specialized understanding that general LLMs may lack
- Cost and Scalability: Providing a cost-effective and scalable alternative to traditional fine-tuned legal LLMs for analyzing a large volume of regulatory contracts (23 years of filings)

Legal LLMs and Platforms

- November 23, 2022 (Seed Round): **Harvey** AI is founded and secures its seed funding round. (While founded earlier, it's a key legal AI player active in 2025)
- March 1, 2023: Casetext unveils **CoCounsel**, the AI legal assistant, developed in partnership with OpenAI using GPT-4 technology.
- October 9, 2023: Clio unveils **Clio Duo**, a proprietary generative AI legal technology designed to be natively built into Clio's offerings, starting with Clio Manage.
- October 26, 2023: **Genie AI** launches its AI legal assistant platform.
- April 23, 2024: **LexisNexis** unveiled its second-generation legal generative AI Assistant on Lexis+ AI.
- October 21, 2024: The paper introducing **LawLLM**: Law Large Language Model for the US Legal System was published at CIKM 2024.
- Various 2025 releases: **LexisNexis** Protégé AI Assistant, A personalized AI assistant with agentic capabilities to streamline drafting, summarization, and legal research across LexisNexis platforms like Lexis+ AI and Lex Machina.
- June 2025: **LexisNexis** and **Harvey** announced a strategic alliance to integrate LexisNexis generative AI technology, primary law content, and Shepard's® Citations within the Harvey platform and jointly develop advanced legal workflows.
- August 13, 2025, Willkie's Global Rollout of **Harvey AI**: A major law firm integrating Harvey's generative AI platform and its Workflow Builder for domain-specific AI workflows.

Harvey AI

Assistant

Ask questions, analyze documents, and draft faster with domain-specific AI.

Vault

Securely store, organize, and bulk-analyze legal documents.

Knowledge

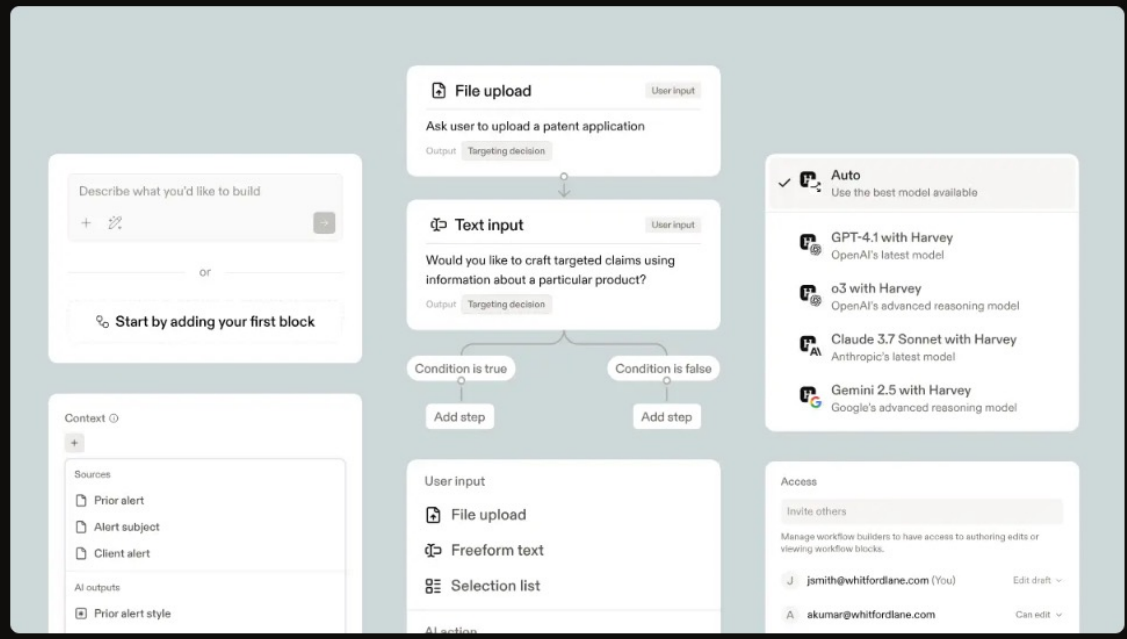
Research complex legal, regulatory, and tax questions across domains.

Workflows

Run pre-built workflows or build your own, tailored to your firm's needs.

Word Add-In

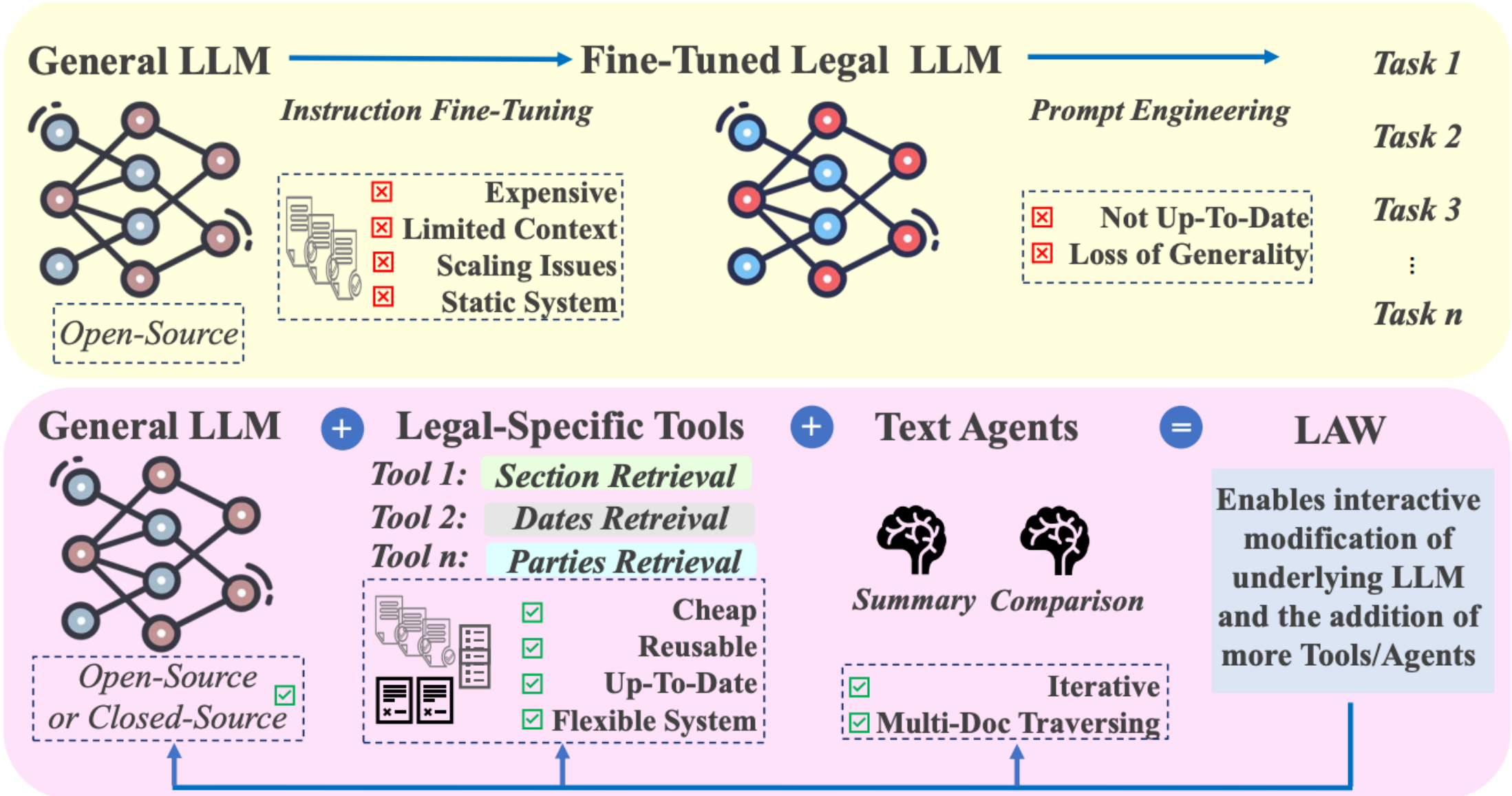
Draft, edit, and review contracts—powered by Harvey, directly in Word.



Introducing Workflow Builder

Design and deploy customizable workflows designed to capture your firm's differentiators and deliver high-quality legal work.

LAW: Legal Agentic Workflows for Custody and Fund Services Contracts

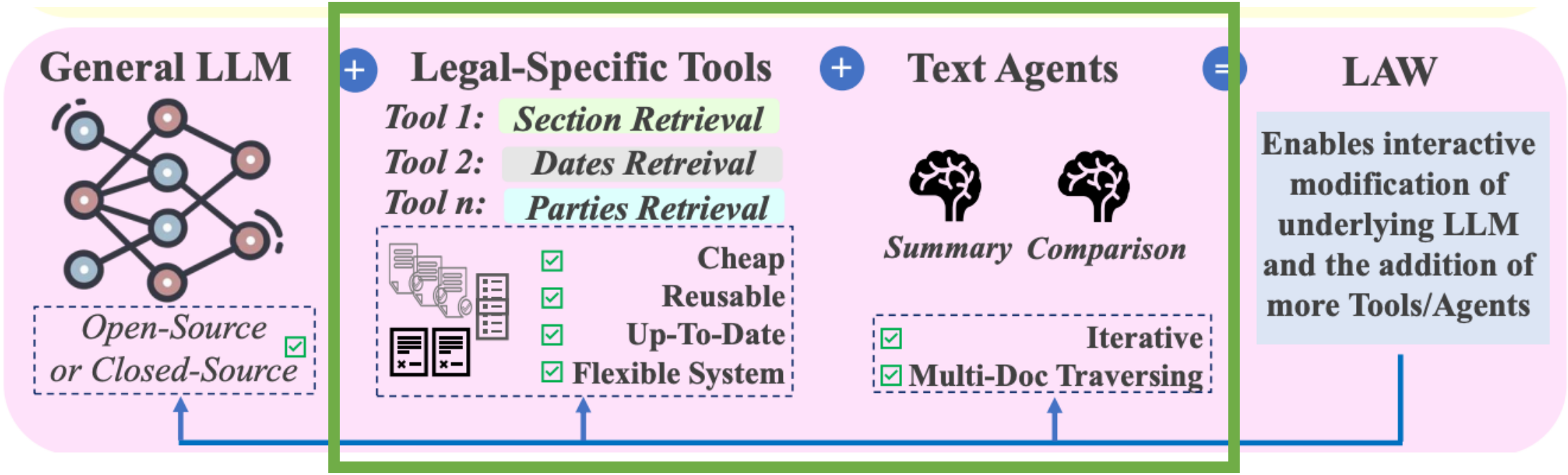


LAW: Ingesting Contracts

- we ingest a total of 22 GB of data from EDGAR (Electronic Data Gathering, Analysis, and Retrieval by SEC) on premise within our knowledge base
  - Scalable Procurement: We ingest at a rate of 112 documents/second - 6.7 hours were spent in terms of sequential processing.
  - AI Metadata Tagging and Search: Each section is made searchable via title recognition algorithms, alongside contextual and visual cues to intelligently chunk each contract for precise retrieval

LAW: Tools

- Direct extraction
- Multi-Hop Reasoning
- Text Agents



Tools: Direct Extraction

- Extracting Dates: Contracts house different types of dates such as the contract's
  - Effective Date (when the current contract is effective)
  - Master Date (when the master/original contract was effective)
  - Dated Date (when the current contract was signed).
  - Detection and extraction of dates is achieved via RoBERTa span detection (Liu et al., 2019), HTML parsing with BeautifulSoup42 , and regular expression heuristics

- Extracting Parties: This tool's objective is to find and extract the associated parties involved in signing the contract. These include the trust of funds and the custodian bank.

Tools: Multi-Hop Reasoning

- Calculate Contract Lifecycle Contracts typically have a lifecycle during which the provisions are enforced.
  - This tool aims to calculate the termination date of the contract's lifecycle. It uses our existing tool for dates to extract the effective date of the contracts. Next, it searches for the contract's duration or the termination date.
  - If the contract mentions the duration (e.g. 3 years), the tool translates the text into a numerical value. Finally, this numerical value is added to the effective date to generate a termination date.

Tools: Multi-Hop Reasoning

- Retrieve Master Contract: This tool's goal is to differentiate between a master and an amendment agreement.
  - Our tool classifies and retrieves the master contract by comparing the extracted effective date with the master effective date using the tool for dates; if equal, the contract is considered to be the master. If determined to be an amendment, **the tool searches for the master by matching the dates and parties**.

Tools: Multi-Hop Reasoning

- Label Section Titles: this tool directly retrieves the relevant clause or section when queries about particular terms arise.
  - Many contracts lack explicit section declarations and the language across different sections can be highly similar.
  - we employed a fine-tuned t5-large (Raffel et al., 2020) model trained to classify paragraphs into one of 20 potential section labels. These labels cover a broad spectrum of typical clauses found in contracts.
  - As an alternative solution, we trained a t5-base model for title generation instead of classification.

Section Titles
account transactions-
authorized persons
definitions
duties and responsibilities
evidence of authority
fee schedule
fees and expenses
foreign custodian and subcustodian
governing law
indemnification
instructions
limitations and scope of use or liability
miscellaneous
nominees
proprietary information
recitals
standard of care liabilities
subcustodians and securities depositories
successor custodian
termination

Table 2: List of key clauses found in fund custody contracts.

Text Agents

- **Summary Agent:** The summary agent aims to provide a useful summary of legal clauses. Our prompts enable the agent to focus on identifying and preserving key terms such as entity names and dates.
- **Comparison Agent:** The comparison agent's purpose is to understand how particular clauses are different across time or entities. The agent chronologically sorts the sections from different contracts and, in parallel, compares each pairwise set of sections in the list.
  - To handle large bodies of text, the agent also performs repeated summaries on each section to condense the body to a manageable size. The summarized sections are then passed to the comparison agent.

Engineering Infrastructure

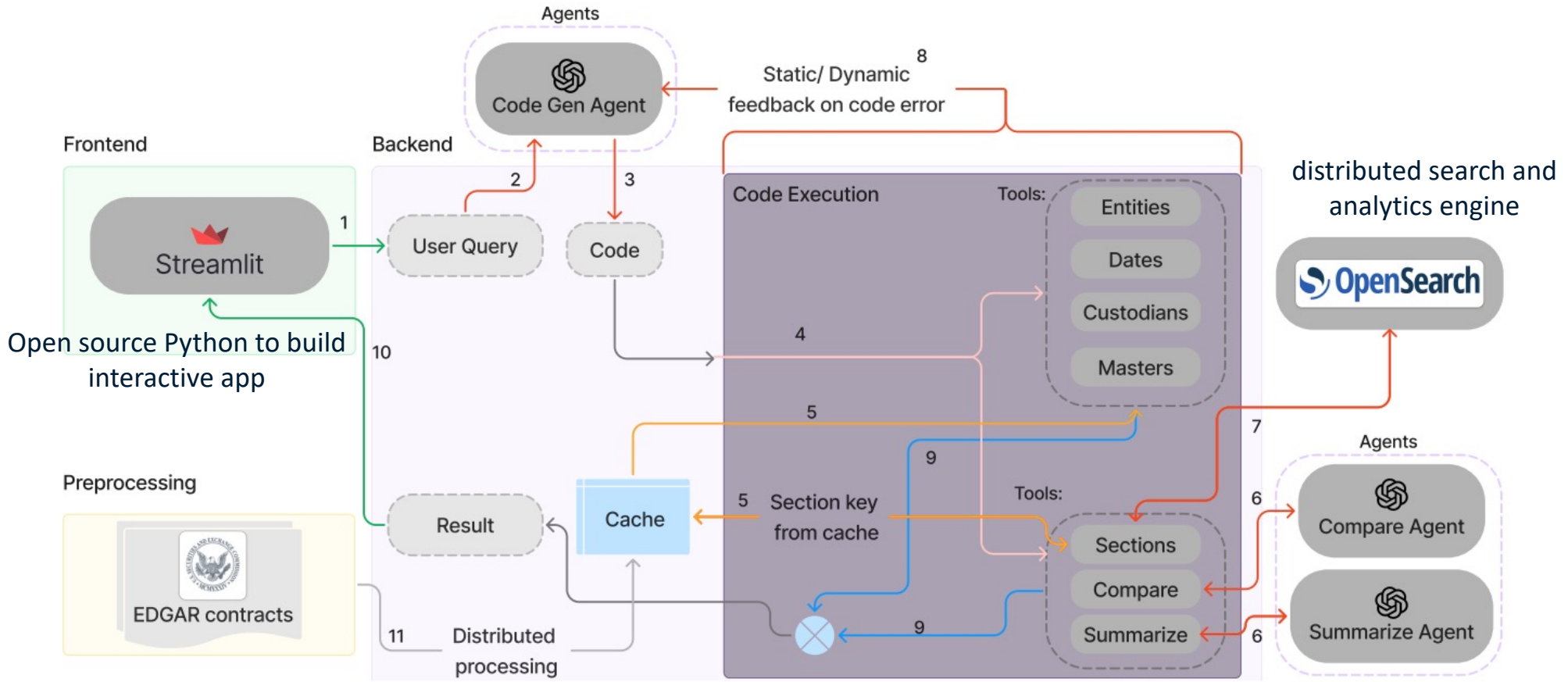
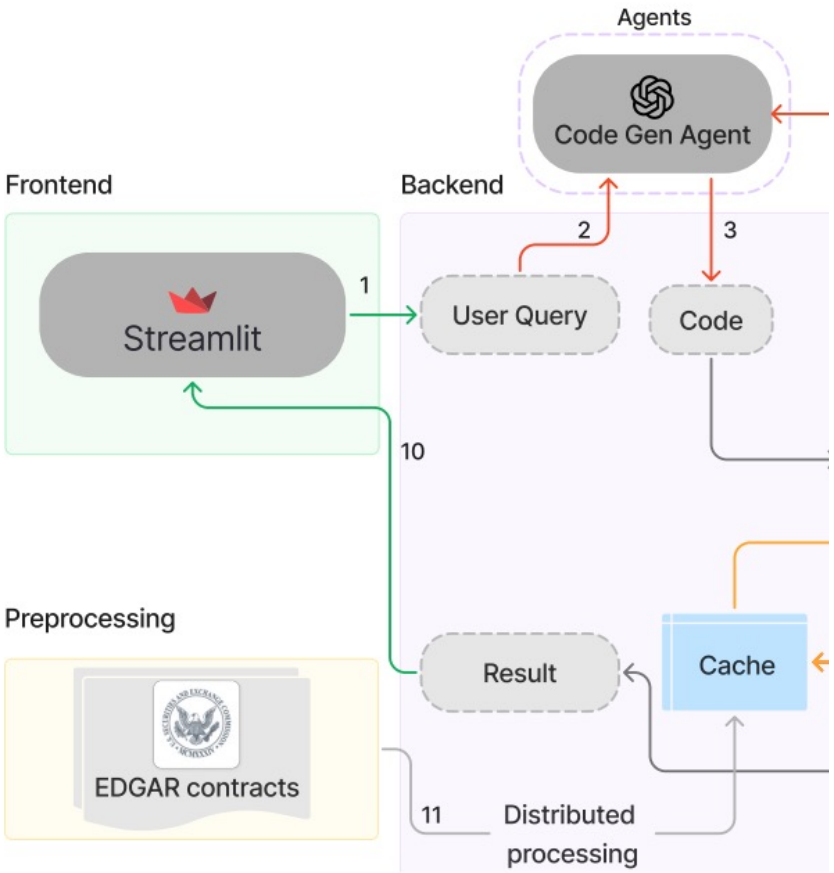


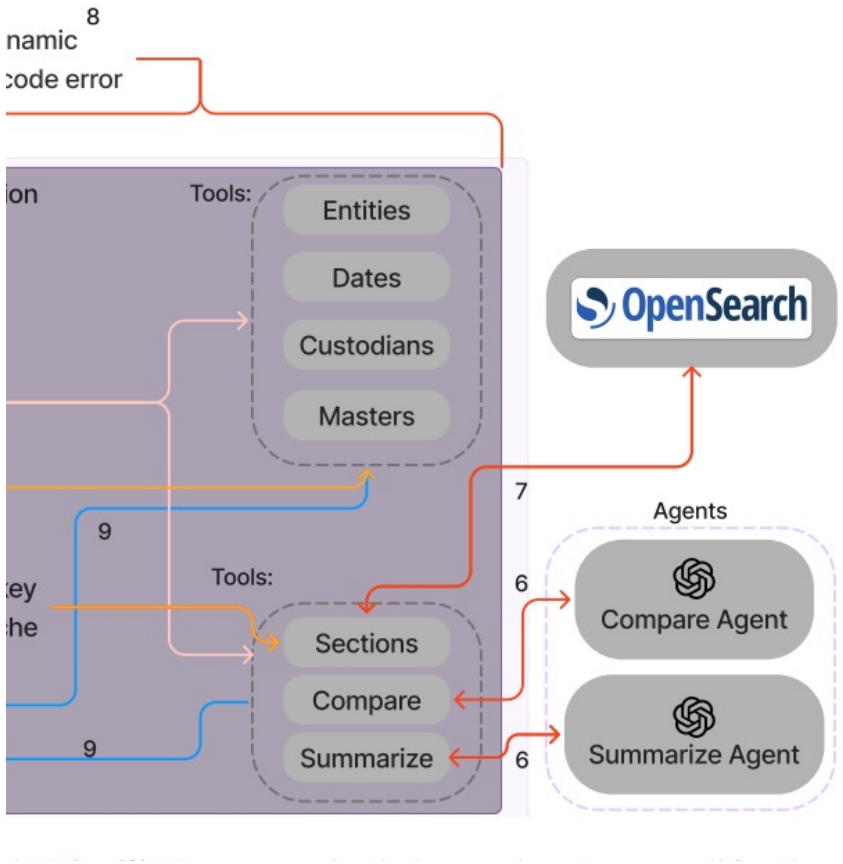
Figure 2: System Overview of **LAW**. **(1)** User query input on front-end (§6); **(2)** Query manipulation and custom modification added to prompt and sent to the code generation agent (§6); **(3)** Chat completion return from the code generation agent; **(4)** Execution of backend API tools (§4); **(5)** Tool retrieval of information from internal cache (§6); **(6)** Calls to text agents (§5); **(7)** Calls to multi-node OpenSearch cluster for text retrieval (§6); **(8)** Feedback on code runs back to the code generation agent in case of failure; **(9)** Concatenation of final output; **(10)** Final text output is rendered on the UI; **(11)** EDGAR contracts undergo continuous, offline, distributed processing to update our internal cache and OpenSearch systems (§3).

Engineering Infrastructure



- User query:
  - We compose user queries with two parts : (1) the entity of interest; and (2) the task to be executed
  - The possible tasks to apply on the entities include: (1) Explore all contracts; (2) Find {master agreements, master dates, termination dates, parties, clause X}; (3) {Summarize, Compare} clause X. These tasks are motivated by legal use cases.
- Caching:
  - To reduce runtime latency, our system batch pre-processes data extraction. This includes features related to the involved parties or dates.

Engineering Infrastructure



- Section Search:
  - The large volume of legal text cannot be directly stored in our cache when retrieving contract sections. Instead, our contracts are segmented and indexed in an OpenSearch distributed datastore provided by AWS
  - For each contract, we retrieve the top 20 most relevant sections using the BM25 ranking algorithm.

Engineering Infrastructure: Code Agent

- The code generation agent is responsible for mimicking planning by generating a series of function calls, thinking steps, that breaks the question into steps semantically linked to our tool calls.
- Use FlowMind framework.

FlowMind: Automatic Workflow Generation with LLMs

Zhen Zeng
J. P. Morgan AI Research
New York, NY, USA
zhen.zeng@jpmchase.com

William Watson
J. P. Morgan AI Research
New York, NY, USA
william.watson@jpmchase.com

Nicole Cho
J. P. Morgan AI Research
New York, NY, USA
nicole.cho@jpmorgan.com

Saba Rahimi
J. P. Morgan AI Research
New York, NY, USA
saba.rahimi@jpmorgan.com

Shayleen Reynolds
J. P. Morgan AI Research
New York, NY, USA
shayleen.reynolds@jpmchase.com

Tucker Balch
J. P. Morgan AI Research
New York, NY, USA
tucker.balch@jpmchase.com

Manuela Veloso
J. P. Morgan AI Research
New York, NY, USA
manuela.veloso@jpmchase.com

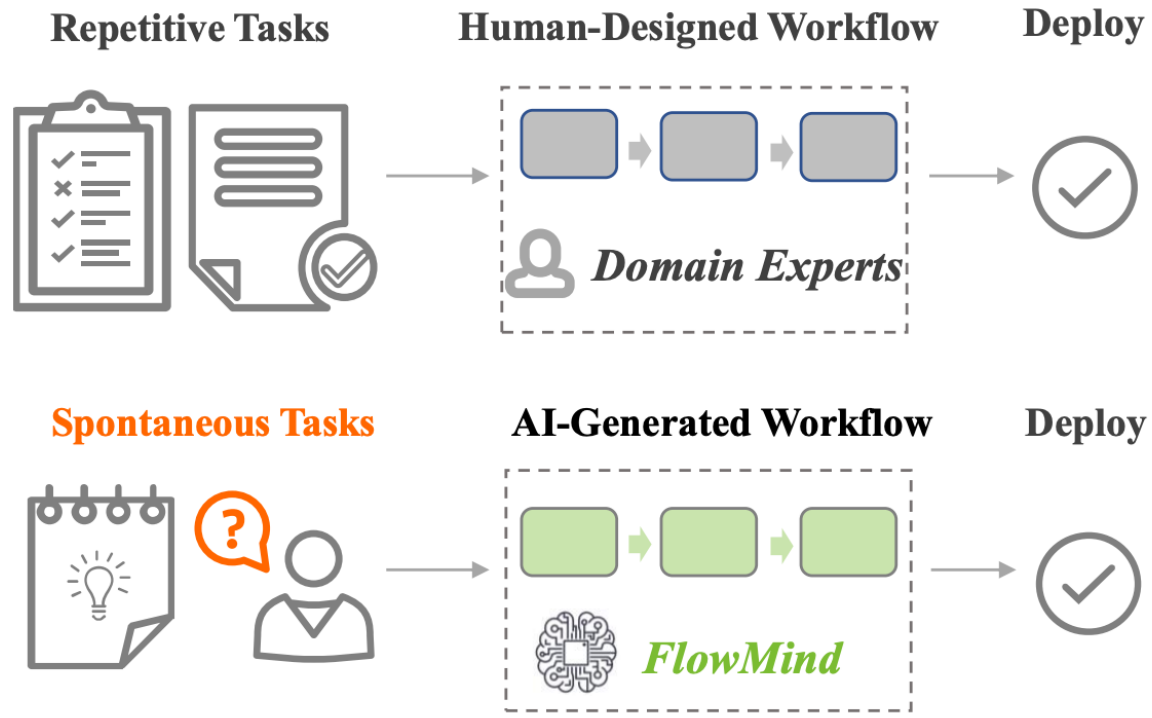


Figure 1: FlowMind automates spontaneous tasks demanded by users through on-the-fly workflow generation, advancing beyond traditional automation of repetitive tasks designed by domain experts.

Engineering Infrastructure: Code Agent

Imagine we are working with a document bot. The job of this bot is to respond to information queries from user.

The main functions you can use are:
-**get_all_reports():** Returns all N-CEN reports.
-**get_report(fund_name):** Returns the N-CEN report that contains the fund specified by the fund_name.
-**segment_report(report):** Returns parsed blocks from the input report, each block describes a fund.
-**fetch_block(report, fund_name):** Returns the corresponding block for the fund in the input report.
-**extract_entity(block, entity_label):** Extracts the name of the entity that is in the specified entity_label from the input text block.
-**extract_value(block, value_name):** Extracts the numeric value as specified by the value_name from the input text block.

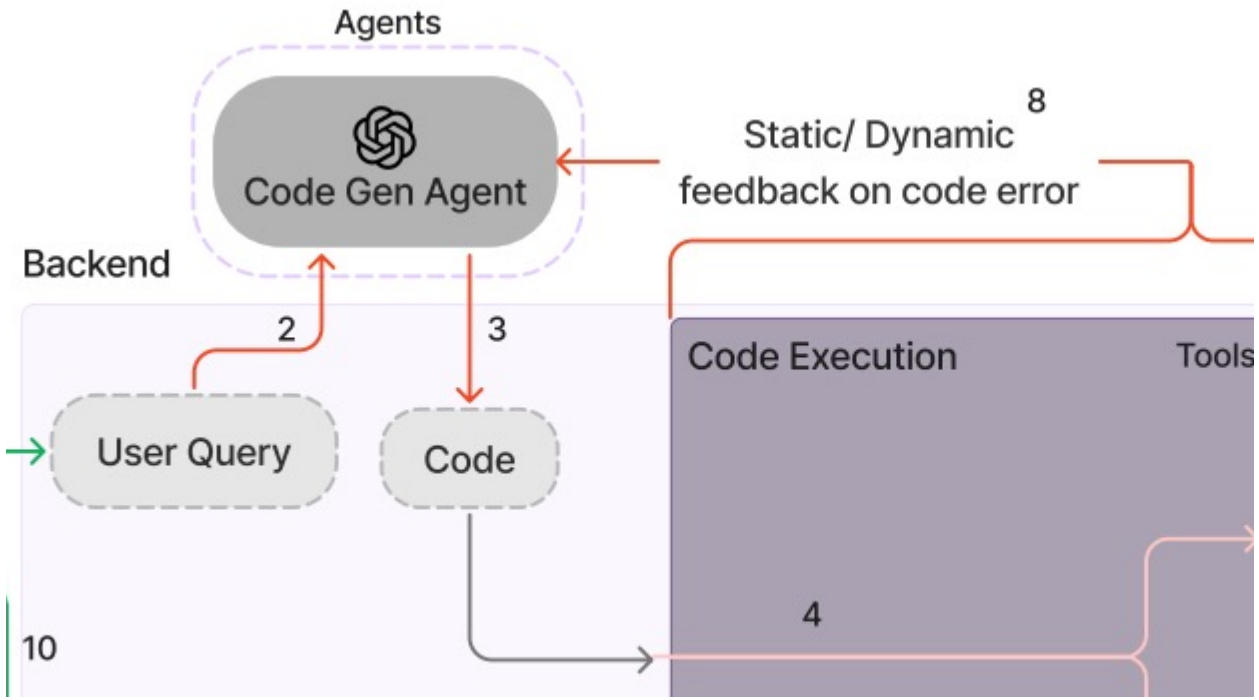
Wait for user queries, then write python code (with modularization) and use these functions to respond. let me know once you are ready for user queries.

Before LLM takes any questions from users, we first give a ***lecture*** to LLM in stage 1.

1. Setup the context;
2. Describe tools;
3. Ask to write code to use the tools

Figure 3: Before an LLM takes any queries or tasks from users, we first give a *lecture* to it. We show an example of such a lecture above. The proposed generic lecture recipe includes: 1) setting up the context, 2) enumerating the available APIs with each function declaration, parameters, and high-level descriptions, and 3) prompting the LLM to write workflow code using these APIs.

Engineering Infrastructure: Code Agent



- Code Generation Agent: Our system prompts the agent to generate Python code that can resolve the user's query, the prompt includes
  - tool names,
  - Descriptions
  - Examples with CoT
  - Self-Refine
- LAW employs generated a three-tier system to generate and validate code:
  - Syntax Validation: Performs pre-execution checks to verify code syntax, types, and security constraints.
  - Hallucination Detection: Ensures generated code only calls tools that exist in LAW's toolset with valid parameter signatures.
  - Runtime Validation: Implements specialized error handling that captures and categorizes execution failures for targeted remediation.

A query example

Query: Compare the *authorized persons* clauses for Fund *BNY Mellon International Equity Income Fund*.

Customized suffix: Only compare subsequent clauses of five sampled non-empty contract sections. Ensure that there are also contracts for this entity choice.

```
# Step 1: Retrieve all agreements for the
    specified fund.
agreements = get_agreements_for(funds='BNY
    Mellon International Equity Income Fund')

# Step 2: Check if any agreements were found.
if agreements:
    # Step 3: Extract the clauses for the
        specified section from each agreement.
    clauses, urls, _ =
        get_section_v2(agg_list=[agreement[1]
        for agreement in agreements],
        section_name='authorized persons')

    # Step 4: Check if any clauses were found.
    if clauses:
        non_empty_clauses = [clause for clause
            in clauses if clause != '']
        num_clauses = len(non_empty_clauses)
        if num_clauses > 5:
            interval = num_clauses // 5
            selected_clauses =
                non_empty_clauses[::interval]
        else:
            selected_clauses = non_empty_clauses

        # Step 5: Compare the selected clauses.
        output = get_comparison_v1(
        list_agreement_tuples=[(agreement[0],
            agreement[1]) for agreement in
            agreements],
            text_list=selected_clauses)
    else:
        output = "No 'authorized persons'
            clauses found for Fund 'BNY Mellon
            International Equity Income Fund'"
else:
    output = "No agreements found for Fund 'BNY
        Mellon International Equity Income
        Fund'"

# Step 6: Return the output.
output
```

Model performance

The main source of error in obtaining termination dates was the hallucination of non-existent dates.

User Query	**LAW**	**Baseline**
Retrieval	*Hit Rate*	
Explore all contracts	94.4	71.8
Find master agreements	100.0	65.4
Find master dates	93.3	36.2
Find termination dates	95.4	2.5
Find parties	100.0	16.3
Analytical	*BERTScore F1*	
Summarize clause X	89.5	68.1
Compare clause X	71.9	-

Table 1: A comparison of LAW with a simulated `gpt-3.5-turbo` baseline. For retrieval-type queries we measure the hit rate/recall calculating the percentage of correct retrievals compared to the ground truth. For analytical-type questions, we measure text similarity using BERTScore's (Zhang* et al., 2020) F1 metric. The contextual embeddings for BERTScore are obtained using the `bert-large-uncased` model.

Conclusion

- LAW leverages reusable, domain-specific tools and text agents, for interacting with complex financial-legal contracts, particularly in custody and fund services.
- Offer a cost-effective and flexible alternative to static, expensive fine-tuned legal LLMs, enabling continuous updates and extensibility
- Streamline workflows for both legal experts and lay-users, enhancing efficiency and accuracy in understanding complex legal documents

Upcoming meetups – Open to proposals and guest speakers!

- Agentic AI real-world use cases
- Latest trends in AI research
- Build AI agents: Hands-on sessions
- Panel discussions

Slides posted at:
https://github.com/YanXuHappygela/LLM-reading-group

Recordings posted at:

YanAITalk

@yanaitalk · 3.41K subscribers · 70 videos

Make machine learning easy to understand! ...