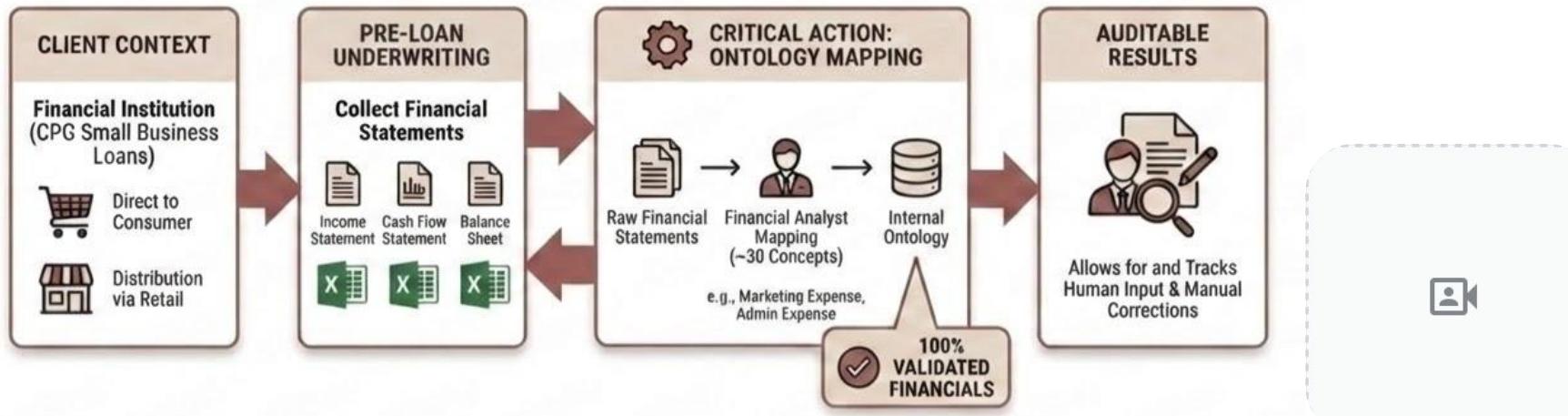


Exgent - A Finance Workflow

- Exgent is designed to solve the problem of evaluating a **company's financials** before a loan is made.
- As companies have different reporting formats, **the accounts must be standardized** so the internal loan evaluation system can process it.
- Exgent automates the **ontology mapping** by ingesting various excel financials and making them available in a format that **internal client systems can process**



Our Goal

Turn excel files into standardized reports. 30+ standard tags need to be associated

	A	B	C
1	Income Stmt		
2			
3		Jan '23	Feb '23
4			
5	Revenue - X Men		
6	4100 - X Men Sales CA	2,051,332.45	1,820,744.74
7	4100-12 - X Men Sales KS		
8	4100-14 - X Men Sales NJ		
9	4101 - Fighting Weight Sales		
10	4203 - Revenue - Software	-	-
11	4102 - Revenue - Consultation	99,250.00	84,525.00
12	4120 - Sales Discounts	(322,587.41)	(273,049.18)
13	4140 - Refund Allowance X CA		
14	4140-12 - Refund Allowance X KS		
15	4140-14 - Refund Allowance X NJ		
16	4140-16 - Refund Allowance Fighting Weight		
17	4140 - Refund Allowance	-	-
18	Total Revenue	1,827,995	1,632,221
19			
20	Cost of Products Revenue		
21	5100 - Cost of Goods Sold-Pharmacy	224,420.64	207,160.20
22	5101 - Cost of Goods Sold-COGS Packaging	19,848.31	45,198.55
23	5103 - Cost of Goods Sold-Merchant Account Fee	123,486.51	115,356.55
24	5104 - Cost of Goods Sold-Chargebacks	7,816.00	4,208.00



Standard tags	Income Stmt	Jan '23
Revenue - X Men	Revenue - X Men	
Gross Revenue	4100 - X Men Sales CA	2,051,332.45
Gross Revenue	4100-12 - X Men Sales KS	
Gross Revenue	4100-14 - X Men Sales NJ	
Gross Revenue	4101 - Fighting Weight Sales	
Gross Revenue	4203 - Revenue - Software	-
Gross Revenue	4102 - Revenue - Consultation	99,250.00
Total Discounts	4120 - Sales Discounts	(322,587.41)
Total Cancellation and Returns	4140 - Refund Allowance X CA	
Total Cancellation and Returns	4140-12 - Refund Allowance X KS	
Total Cancellation and Returns	4140-14 - Refund Allowance X NJ	
Total Cancellation and Returns	4140-16 - Refund Allowance Fighting Weight	
Total Cancellation and Returns	4140 - Refund Allowance	-
Net Revenue	Total Revenue	1,827,995
Cost of Products Revenue	Cost of Products Revenue	
Gross Product Cost	5100 - Cost of Goods Sold-Pharmacy	224,420.64
Gross Product Cost	5101 - Cost of Goods Sold-COGS Packaging	19,848.31
Shipping & Fulfillment Cost	5103 - Cost of Goods Sold-Merchant Account Fee	123,486.51
Shipping & Fulfillment Cost	5104 - Cost of Goods Sold-Chargebacks	7,816.00

Start point - Dec 26th 2025

Lets try gemini - How hard can it be?

The screenshot shows a Microsoft Excel spreadsheet with several tabs visible at the bottom: 'General', 'Financial', 'Sales', 'Inventory', 'Purchasing', 'Manufacturing', 'Customer', 'Supplier', 'Bank', 'Report', and 'Gemini'. The main sheet displays financial data for December 2025, including revenue from various product categories like 'Fridge', 'Washing Machine', 'Dishwasher', and 'Oven'. It also shows costs for shipping, taxes, and total sales. A sidebar on the right is titled 'Gemini' and contains a text input field with the following prompt:

```
Given the following financial data for December 2025, identify the revenue from Sales of Fridge, Washing Machine, Dishwasher, and Oven. Lines for Sales of Fridge, Washing Machine, Dishwasher, and Oven should be grouped together in 'Sales' category.
```

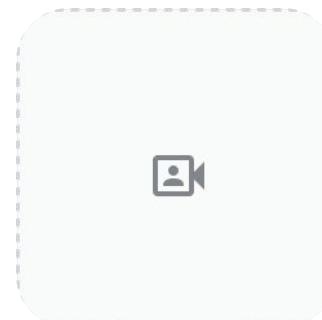
The sidebar also includes sections for 'Summary', 'Details', 'Notes', and 'Actions'.

We can

- Open sheets
- Add prompt
- Ask gemini to insert ontology tags

* Recorded on Jan3rd, with prompt from demo

<https://youtu.be/aaU53DzZsdE>



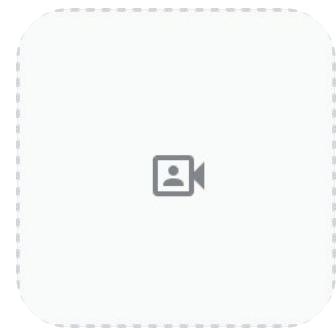
Design choices - Dec 30th

Implement an Agentic multi-step process that follows the Generator -> Verifier pattern

- Users **upload excel files** into a React application
- Users must be able to **interact with the excel sheet** (Similar to canvas in modern AIs)
- Make the process interactive so the analyst remains involved
 - Remove the boring parts of validating financials “such as checking the totals for each date”.
- **Divide and conquer the problem** that is too hard to solve in single step
 - Agent to understand **sheet structure**
 - Another agent to take logical part of sheet **and tag them**
 - Testing on sample data in gemini web app with prompt engineering has been promising..
- Must maintain **Audit trail** and be able to review all changes.

I choose with Python Fast API and React 13 for the frontend, Google ADK for the agent framework.

Github: <https://github.com/sanjayvenkat2000/exgent>



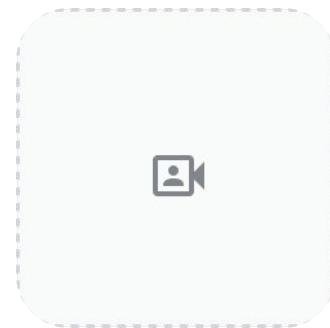
Demo



<https://youtu.be/WT1LHVI8Qgc>

Learning - What the demo achieves

- Models are much better at dealing with csv or markdown than excel files.
 - Matches their training data more closely
- Context engineering is key
 - Small problems with **clean context** and **clear prompt instructions** proved very effective.
 - Asking the **model to rewrite its prompt** for clarity help as always.
- In order to build trust an interactive experience is ideal.
 - Show the user the end result
 - Show the user the steps
 - Allow the user to change the output of the model
 - Maintain an audit trail



Architecture highlights - Server

Storage

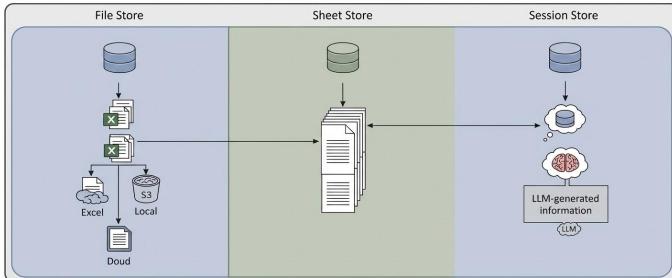
- **File Store:** Stores uploaded files and access control information
- **Sheet Store:** Stores metadata and validation information extracted from the excel files. This is a **versioned store**
- **Session Store:** Stores the LLM chat outputs that are synchronize with the Sheet Store

Agents

- **Sheet Structure Agent:** Purely responsible for understand the sheet structure
- **Tag Agent:** Responsible for tagging
- **Verify Agent:** Uses tools to validate the sheet numbers

Server

- REST inspired endpoints with streaming chat endpoint (all endpoints shown)



GET	/ Read Root
GET	/files List Files
DELETE	/files/{file_id} Delete File
GET	/files/{file_id} Get File Details
GET	/sheets/{file_id} Get Sheet Names
GET	/sheetdata/{file_id}/{sheet_idx} Get Sheet Data By Index
GET	/sheetinfo/{file_id}/{sheet_idx} Get Sheet Info By Index
POST	/sheetinfo/{file_id}/{sheet_idx} Update Sheet Info
POST	/sheetchat/{file_id}/{sheet_idx} Sheet Chat Stream
GET	/sheetchat/history/{file_id}/{sheet_idx} Get Sheet Chat History
POST	/upload Upload File

Architecture highlights - Client

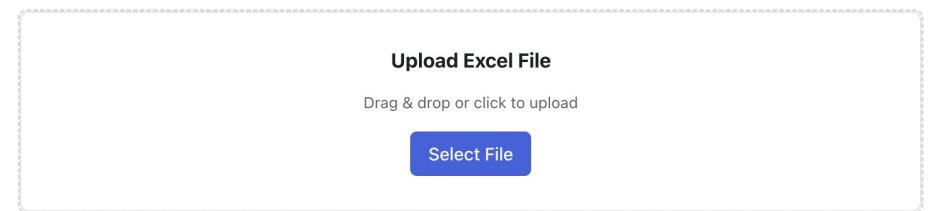
Not Much: “Good react app. Supports streaming for agent”

- **Use best of class libraries:** React apps can break down quick if good design is not followed.
 - Tanstack useQuery is used here to borrow the great work they do.
- Abstract all service calls to a single file, we need to implement authentication etc. (one file to change later).
- Use good state management using providers
 - Providers allow good access without passing props all over the place (prop tunneling)
- Do not use **useEffect(..)**, where possible, as this is an anti-pattern and causes issues very fast that are hard to debug.

Welcome to Exgent

Your professional AI agent designed to handle **very large Excel files** with ease. Exgent's **Primary Objective** is to tag the data into your internal ontology. You can also **converse** with Exgent about information in the files.

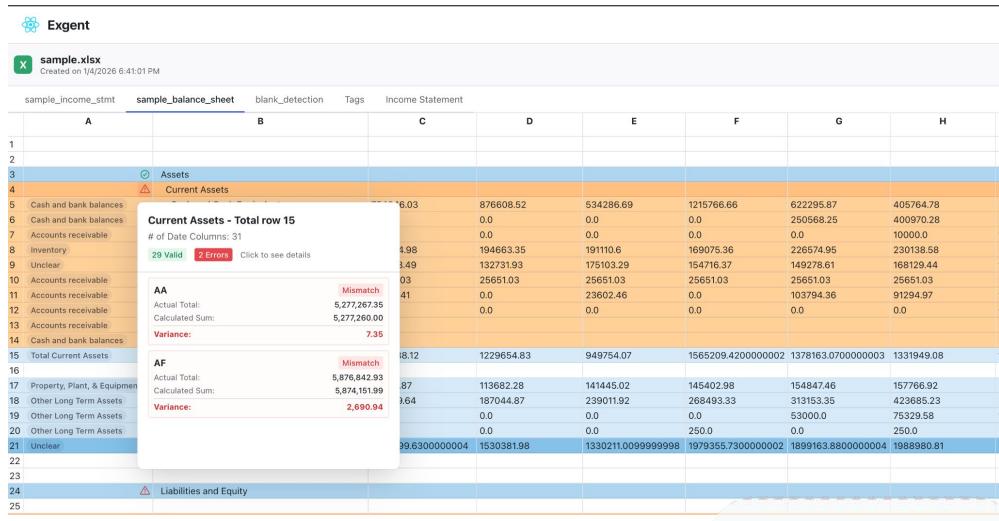
[Large File Support](#) [Internal Ontology Tagging](#) [Instant Insights](#)



Architecture highlights - Client

Synchronizing UI state

- Ag-UI and A2UI are emerging standards. Needs a few more months to flush out.
- **Use a custom LLM message to tell the UI “what to update”.**
- Should migrate to standards to get UI reuse?



The screenshot shows a spreadsheet application window titled "Exgent" with the file "sample.xlsx" open. The sheet "sample_balance_sheet" is selected. The data is organized into sections: Assets, Current Assets, and Liabilities and Equity. A tooltip is displayed over the "Current Assets" section, indicating a "Mismatch" error with the message "Actual Total: 5,876,842.93" and "Calculated Sum: 5,874,151.99". The "Variance" field shows a value of 2,690.94. The "Income Statement" tab is visible at the top of the sheet. The overall interface includes standard spreadsheet features like rows, columns, and a toolbar with tabs for sample_income_sheet, sample_balance_sheet, blank_detection, Tags, and Income Statement.

A	B	C	D	E	F	G	H	
1								
2								
3	Assets	Current Assets						
4	Cash and bank balances	16.03	876608.52	534286.69	1215766.66	62295.87	405764.78	
5	Cash and bank balances	0.0	0.0	0.0	0.0	250568.25	400970.28	
6	Accounts receivable	0.0	0.0	0.0	0.0	0.0	10000.0	
7	Inventory	1.98	194663.35	191110.6	169075.36	226574.95	230138.58	
8	Unclear	1.49	132731.93	175103.29	154716.37	149278.61	168129.44	
9	Accounts receivable	0.03	25651.03	25651.03	25651.03	25651.03	25651.03	
10	Accounts receivable	41	0.0	23602.46	0.0	103794.36	91294.97	
11	Accounts receivable	0.0	0.0	0.0	0.0	0.0	0.0	
12	Accounts receivable	0.0	0.0	0.0	0.0	0.0	0.0	
13	Accounts receivable	0.0	0.0	0.0	0.0	0.0	0.0	
14	Cash and bank balances	7.35						
15	Total Current Assets	18.12	1229654.83	949754.07	1565209.4200000002	1378163.0700000003	1331949.08	
16	AF	Mismatch						
17	Property, Plant, & Equipment	5,876,842.93	87	113682.28	141445.02	145402.98	154847.46	157766.92
18	Other Long Term Assets	5,874,151.99	87	187044.87	239011.92	268493.33	313153.35	423685.23
19	Other Long Term Assets	2,690.94	164	0.0	0.0	0.0	53000.0	75329.58
20	Other Long Term Assets	0.0	0.0	0.0	250.0	0.0	250.0	
21	Unclear	99,630,000,0004	1530381.98	1330211.0099999998	1979355.7300000002	1899163.8800000004	198890.81	
22								
23								
24	Liabilities and Equity							
25								



Closing Thoughts

Forbes

INNOVATION

The RPA Hype Cycle Is Over -- Here's The Reality



By [Eric Johnson](#), Former Forbes Councils Member.

for [Forbes Technology Council](#), COUNCIL POST | Membership (fee-based)

Published Dec 18, 2019, 08:40am EST

The RPA trend lasted from 2014 to 2020.

I guess Agentic AI is a follow-on, that actually has legs.

