informer

Whitepaper:

Data Security with Informer Assistants

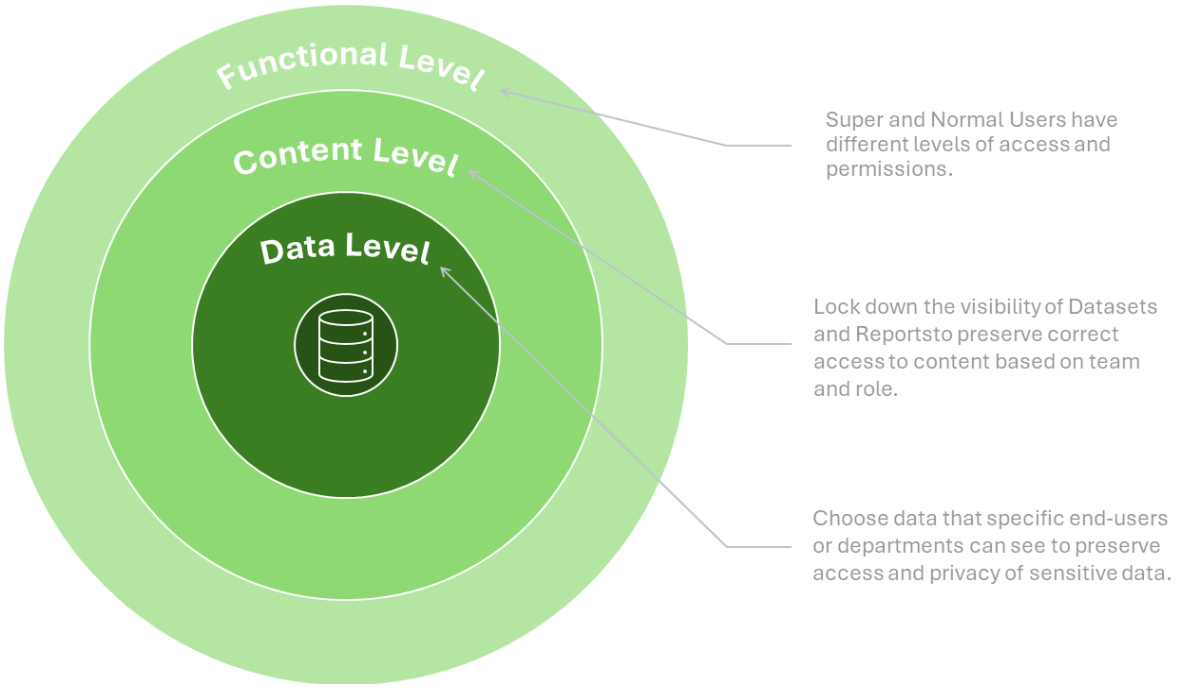
Introduction

Informer Assistants (referred to in this document as “Assistants”) utilize tools and functions, including searching Informer Datasets, unstructured data, third-party Application Programming Interfaces (APIs), and more to facilitate communication between the User, OpenAI, and Informer. Users converse with the Assistant, and Informer coordinates messaging to the underlying AI model. The AI model processes the input, utilizing natural language processing and Informer Functions to generate a response.

Between Informer's native security features and OpenAI's privacy measures, data security, and privacy are ensured, adhering to the principles of Data Governance when interacting with Assistants and OpenAI.

Security in Informer

Informer employs a three-tiered Data Governance security framework (Functional, Content, and Data levels) to prevent unauthorized access, manage sensitive data, and ensure the integrity of information as an authoritative, singular source of truth. Within this Framework, each security level stands as a protective barrier, restricting data access to authorized Users with requisite permissions. User data access permissions are fully maintained when utilizing Assistants.



Functional Level
Functional Level security in Informer categorizes Users into Normal Users, with Team-based access, and Super Users, with full system access. Normal Users are confined to Team-based roles and security and can only access data that is authorized for their use. Conversely, Super Users have full access rights, regardless of Team assignment. A Super User can view all content and access all functionality in the system and only a Super User can define another User as a Super User.

Content Level
Content Level security governs User access based on predefined Team roles, which regulate the creation, utilization, and dissemination of content within Informer. A User is assigned a role within a Team dictating their access privileges. Sharing of data across Teams is restricted to Users with specific roles. By managing User permissions, Teams, and roles, Super Users control data access within Informer.

Data Level
Data Level security controls data access in Informer, allowing only certain Team members to manage and restrict data visibility. Creating and sharing Assistants are exclusive privileges reserved for Super Users and Team Admins. Further, Users with the roles of Team Admin, Publisher, or Data Wizard can restrict access to the data in Informer with Mapping Sets and Restricted Fields for Datasources, Dataset-level filters, and row-level security.

OpenAI Security

Informer employs a three-tiered Data Governance security framework (Functional, Content, and Data levels)[MC1] to prevent unauthorized access, manage sensitive data, and ensure the integrity of information as an authoritative, singular source of truth. Within this Framework, each security level stands as a protective barrier, restricting data access to authorized Users with requisite permissions. User data access permissions are fully maintained when utilizing Assistants.

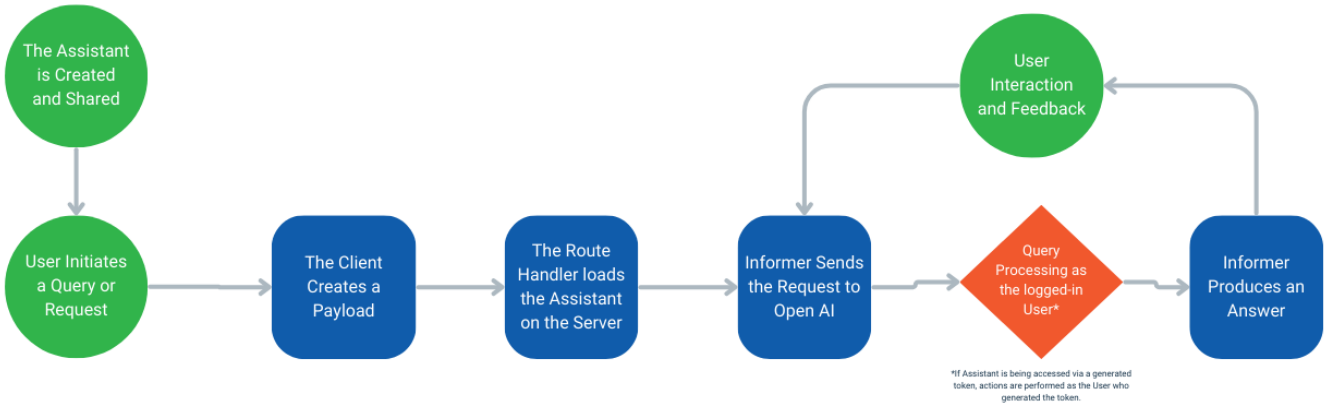
OpenAI employs a comprehensive approach to data security, focusing on protecting data and the integrity of its systems. See OpenAI's Trust Portal for detailed information. While specific technical and operational details may evolve, the core principles and strategies. OpenAI uses include the following:

- Data Encryption: OpenAI ensures that data is encrypted both in transit and at rest; protecting data as it moves between systems (preventing unauthorized interception), and ensuring that stored data is also secure, making it unreadable without the appropriate decryption keys.
- Access Control: Access to data within OpenAI is strictly controlled and monitored, applying the principle of least privilege, meaning individuals are only granted access to the information and resources necessary for their job, minimizing the risk of internal data breaches and unauthorized access.
- Regular Security Audits and Compliance: OpenAI conducts regular security audits to assess and improve its security posture, identify potential vulnerabilities, and ensure that security practices are up to date.
- Anomaly Detection and Monitoring: Monitoring of systems and networks allows OpenAI to detect and respond to unusual activity that could indicate a security threat.
- Incident Response: OpenAI has an incident response plan to address any security incidents or data breaches quickly.
- Data Privacy: OpenAI's commitment to data privacy includes transparency about data usage, ensuring that clients understand how their data is used and have control over their personal information.

OpenAI continuously updates its security practices in response to emerging threats and advancements in security technology.

Securing the Conversation

The interaction between OpenAI, Informer, and the User is a collaborative process designed to provide accurate, relevant, and timely information or assistance to the User. Below is an overview of the entire process:



1. The Assistant is Created and Shared
The process begins with a Super User or a Team Admin creating an Assistant, adding files, Datasets, and Skills, and sharing the Assistant with other Users or Teams.

Only the Assistant's owner, the Team Admin of the owner's Team, or Super Users can add data to an Assistant, either at creation or during editing. Datasets are secured at the field and row level.

2. User Initiates a Query or Request
The User poses a question or makes a request. Only Super Users, Team Admins, and Users who have been given access can use an Assistant.

All requests made by the Assistant are on behalf of the logged-in User, meaning that Users who have access to the data only see the Fields and rows they are allowed to see.

3. The Client Creates a Payload
The Informer client creates a payload containing the User's message and any available UI functions (i.e., tool_calls in OpenAI). The client issues a POST request to an Informer API endpoint (e.g., "/api/assistants/{id}/_chat") specifically for that Assistant. The route looks up the Assistant, ensures the User has read access, and renders a 404 error code if they do not.

4. The Route Handler loads the Assistant on the Server
The server loads the Assistant with the Skills and Knowledge to which it has access. The Assistant accesses any attached Datasets as the logged-in User. The server adds the Assistant's instructions and other tool_calls (e.g., datasetSearch, fileSearch, etc.) to the request.

5. Informer Sends the Request to Open AI

Informer facilitates the interaction between the User and OpenAI. The chat function takes the final payload, containing all the User's messages and tool_calls, and sends it to OpenAI for processing.

6. Query Processing and AI Generation

OpenAI receives the payload from Informer. The AI model processes the input, leveraging its training data and algorithms to either generate a response or select the function to use.

If OpenAI returns a tool_call, Informer runs the function with suggested arguments from the AI model and returns that result to OpenAI. The AI model determines if a function needs to be run again or if it can answer the initial query.

For file processing, Informer computes vector embeddings at file upload time. Vector embeddings transform words or objects into numerical lists to aid AI analysis. Vector embeddings are stored in Informer's PostgreSQL database. Refer to your OpenAI account for the specific data retention policies of your token.

When the Assistant handles an attached Dataset, Informer does not transmit the entire Dataset. Instead, Informer sends only the results of an aggregate query to OpenAI, and only if the results are needed to answer User questions.

7. Informer Produces an Answer

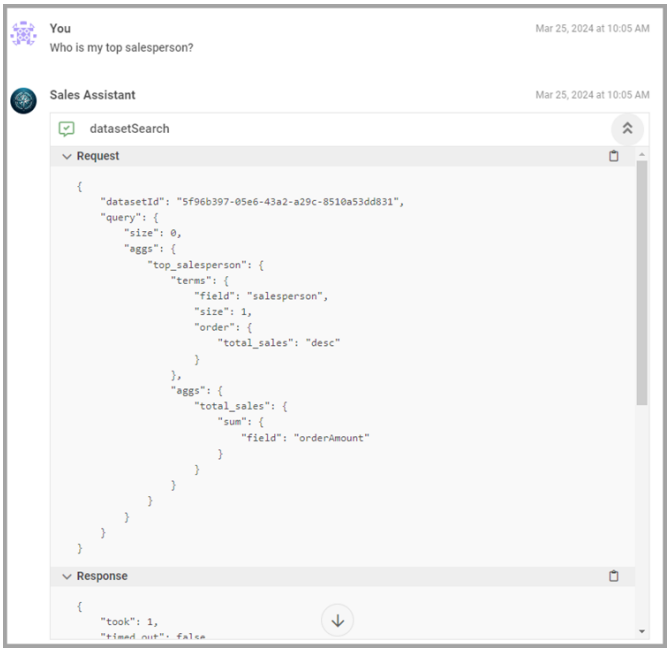
Informer retrieves the final response from OpenAI and presents it to the User. The result is presented to the User in a human-readable way.

8. User Interaction and Feedback

After receiving the response, Users can have follow-up questions, require clarification, or wish to explore the topic further. Users can continue the conversation by providing additional input or feedback. Subsequent questions send the entire conversation with the most recent message added to the end so that the AI model has the full context to generate a response. This iterative process allows for a dynamic and interactive exchange of information between the User and the Assistant.

When an Assistant responds to a question, a question mark icon appears. Clicking the icon prompts the Assistant to explain how it arrived at the answer. This feature can be used to verify the accuracy of the response, understand the Assistant's reasoning process, and refine the ongoing conversation.

Assistants include a Debug mode to show the Skills the Assistant uses to generate responses. Debug mode explains the dialog between OpenAI and Informer as they collaborate to answer questions and the functions the Assistant uses to return a result. Once enabled, a dropdown with the name of the function the Assistant used is added to each of the Assistant's responses along with the responses from OpenAI.



The User's browser stores the conversation in local storage. It is deleted when the User uses the "Start Over" button or accesses the Assistant in an incognito window. This gives Users control over how much information remains in the Assistant and when it is cleared.

Conclusion

The conversation between OpenAI, Informer, and the User is a collaborative process that leverages Informer security and OpenAI API security to ensure that data remains private. Throughout this process, the focus is on delivering accurate and helpful responses while ensuring the privacy and security of data.