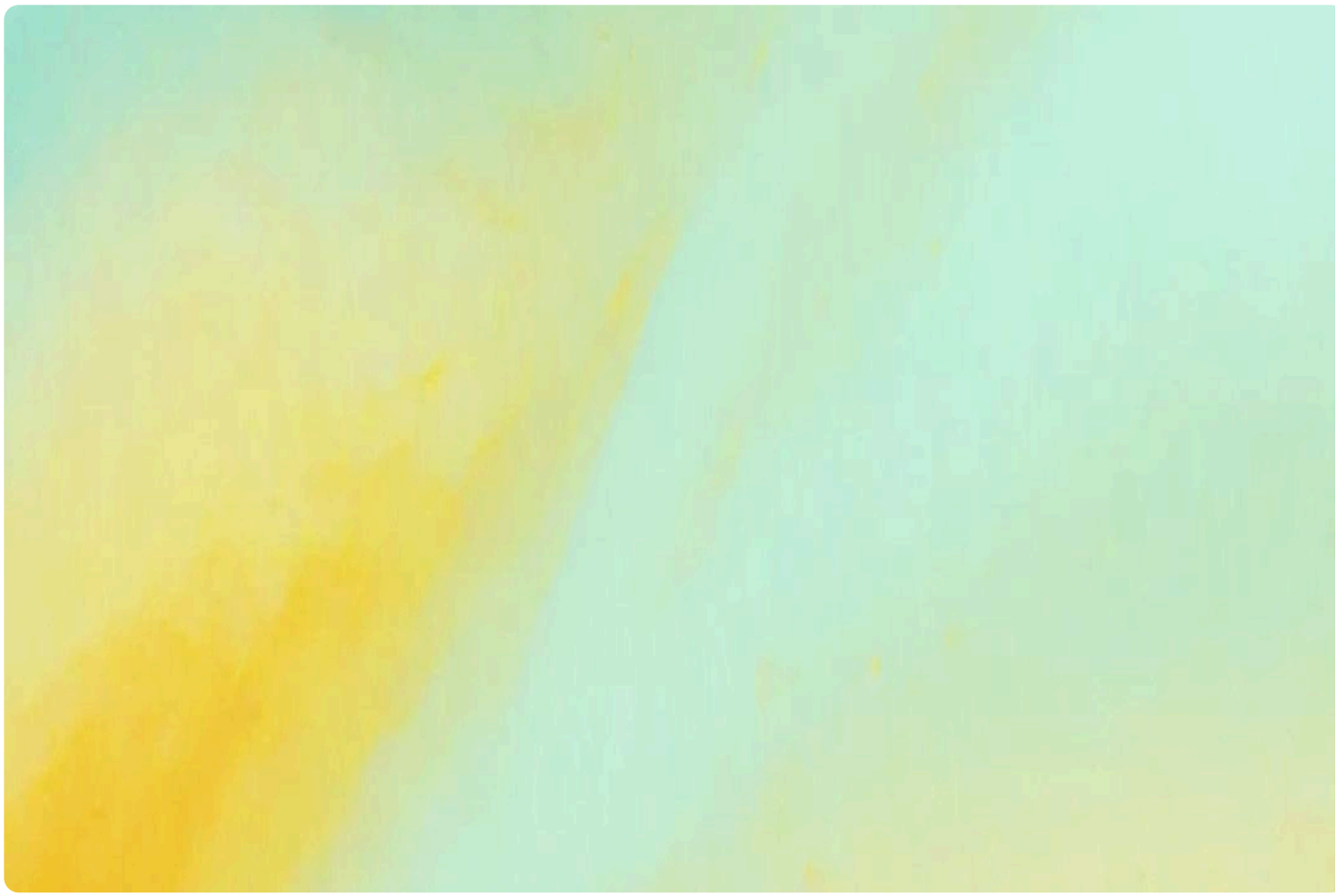
OpenAI

App Security Whitepaper



June 2026

Contents

Contents

Overview

Apps let Enterprise users securely extend OpenAI to third-party applications like Canva, Google Drive, Figma, SharePoint, GitHub, Outlook, and more; list of apps available with OpenAI. This enables AI-assisted access to internal files, emails, documents, workflows, and other data without leaving ChatGPT.

Apps are designed with a secure foundation, leveraging OAuth permissions, encryption, and enterprise-grade compliance to help your data stay protected. Apps are provisioned and governed through the Plugin Directory, enabling administrators to enforce centralized access control and governance across OpenAI services, including ChatGPT, and customer-built solutions leveraging OpenAI APIs. By default no customer data from apps in Enterprise workspaces is used by OpenAI for training, and all retrieval and processing happens within a secure, encrypted pipeline.

App Usage Scope

Apps can be invoked across the OpenAI APIs, Codex, and ChatGPT. Additional product surfaces may be added over time and will inherit the same admin and security controls once the relevant surface reaches Enterprise general availability (GA). Across all surfaces, app calls inherit provider permissions and respect workspace policies and retention settings. While certain apps may be enabled by default, enterprise administrators retain centralized control to disable apps, restrict availability through role-based access controls, and govern user access in accordance with organizational policy.

APIs allow developers to expose apps as tools directly through OpenAI's endpoints. These calls run using a service or user identity, respect the original provider's access controls, and respect existing data retention policies. Admins can control which apps are available in APIs and monitor usage through logs.

ChatGPT supports apps in several ways:

01 In Direct Chat, end users can retrieve data, interact with rich UI experiences, or perform permitted actions through both UI-rendering and non-UI Apps.

---

02 In Deep Research, apps retrieve external context on a request basis with source citations; no bulk pre-indexing or persistent ingestion (unless a Sync app is separately enabled).

---

03 Additional product surfaces may be added over time and will inherit the same admin and security controls once the relevant surface reaches Enterprise general availability (GA).

Security and trust remain our highest priorities. By default OpenAI does not train on app-accessed data in Enterprise workspaces, and we continuously monitor for prompt-injection threats and platform security to help keep your information protected.

App Types

Apps

When an app is invoked, data is retrieved in real time from the customer's source systems and is not persisted or replicated in any long-term storage or search indices. The model processes the retrieved snippets as context. These transient app results are treated like chat conversation inputs, governed by your workspace's retention settings.

If information from an app file is used to supplement an answer, the full file is not uploaded or attached to the conversation. Only the specific retrieved content is processed, and it remains subject to your app's access controls.

Sync Apps

Sync apps proactively index and encrypt administrator-authorized content, enhancing relevance-based search and retrieval as compared to Access apps. Any indexed data respects the user's sharing permissions and the workspace's data retention window. Importantly, no app data is used by default to train OpenAI's models for Enterprise, Edu, or Business customers. This means content accessed via apps remains private to your organization.

MCP Custom Apps

ChatGPT supports custom apps built using the Model Context Protocol (MCP) specification. These MCP custom apps operate with the same security and data-handling principles as app, providing consistent controls and governance across integrations. MCP custom apps may also include an optional widget, which renders developer-provided UI directly inside the ChatGPT conversation to enable richer, multi-step workflows. Widgets run in an OpenAI-controlled sandbox and can present interactive elements (e.g., forms, search results, visualizations) and maintain workflow continuity by passing narrowly scoped UI state back to the model. Widgets do not provide additional permissions beyond what the MCP tools declare; any data they access is limited to what is passed through tool calls or explicit user interactions.

Only workspace owners or admins can add a MCP custom app integration in an Enterprise/Business workspace by using Developer Mode. ChatGPT Developer Mode is a beta feature that provides MCP client support for tools, both read and write. It is powerful and intended for developers who understand how to safely configure and test apps and, where applicable, widget experiences. This helps ensure regular end-users cannot inadvertently add unvetted integrations. Even after a MCP custom app is added by an admin, each user must explicitly link their account (performing OAuth, where applicable) for that tool, before it can be used.

When using Developer Mode, administrators should monitor for prompt injection attacks (where an attacker embeds malicious instructions in content that one of our models is likely to process, such as a webpage, in order to override ChatGPT's intended behavior) as well as other risks, including model errors during write actions that could result in data loss, malicious MCPs that attempt to exfiltrate information, and misleading widget UIs that could encourage unsafe user actions. Widget apps are required to be clearly attributed to the developer and are restricted by default network sandboxing and Content Security Policies; however, organizations should still treat any data sent to a custom MCP backend as shared externally.

While OpenAI does not vet or certify MCP custom apps (including those with widgets), multiple protections are in place, including policy screening of MCP prompts, blocking of known malicious domains, checks on write-capable actions, sandboxed widget execution with restricted network egress, and both online/offline moderation of tool calls. Please note that MCP custom apps are not developed or verified by OpenAI, and MCP custom apps are third-party services that are subject to their own terms and conditions.

Plugins

Plugins are a reusable way to package app integrations, MCP server configuration, skills, workflow guidance, and related agent capabilities into portable artifacts that can be discovered and used across OpenAI surfaces such as Codex and ChatGPT. They are designed to standardize repeatable workflows while maintaining administrative control over which capabilities are made available to users.

It is important to note that app governance and plugin governance are related, layered control models. App governance continues to apply to the underlying app or MCP capability, including authentication, user entitlements, workspace policy, data handling, and action controls. Plugin governance adds an additional layer for how those capabilities are packaged, reviewed, distributed, installed, and enabled within plugin-supported workflows. Accordingly, organizations should evaluate both the underlying app controls and the plugin-level controls when enabling plugin-based workflows.

Plugin Packaging and Trust Boundaries

Codex plugins are packaged as explicit artifacts with a required `.codex-plugin/plugin.json` manifest. The manifest identifies the plugin, declares its version and description, points to bundled components, and provides install-surface metadata such as display name, developer information, capability descriptions, privacy policy URL, terms of service URL, icons, logos, screenshots, and default prompts. A plugin may also include a `skills/` directory, an `.app.json` file for app or connector mappings, an `.mcp.json` file for MCP server configuration, lifecycle hook configuration, and assets used to present the plugin across supported surfaces.

A hook is a Codex extensibility mechanism that runs deterministic scripts at defined lifecycle events, such as before or after tool use, when a permission request is raised, when a user prompt is submitted, or when a turn stops. Lifecycle hooks can be declared in the manifest or loaded from a default `./hooks/hooks.json` file when present. Because MCP servers and hooks can extend the operational surface of the agent, review should include the commands they start, the credentials or environment variables they require, the network destinations they need, and the events on which any hooks run.

This packaging model is designed to make the plugin trust boundary reviewable. Before distribution, organizations can inspect which skills are bundled, which apps or connectors are referenced, which MCP servers may be initialized, what lifecycle hooks are present, what authentication policies apply, and what external services may receive data. Only the plugin manifest belongs under `.codex-plugin/`; bundled skills, MCP configuration, app mappings, hooks, and assets live at the plugin root, making the package structure clear for review and change management.

Codex installs plugins into a local plugin cache and stores each plugin's enabled or disabled state in `~/.codex/config.toml`. This creates separate control points for catalog inclusion, installation, authentication, and enablement. For enterprise use, this supports a controlled rollout model: curate the marketplace, review the plugin source and manifest, evaluate bundled MCP servers and skills, approve the authentication and installation policy, and enable only the plugins that meet the organization's internal governance requirements.

Plugin Authentication and Access Control

Plugins do not themselves grant access to external systems. When a plugin bundles an app, connector, or MCP server that requires authentication, access continues to be governed by the underlying app or MCP configuration, the user's existing entitlements, and the relevant workspace policies. Where OAuth is used, users or administrators must complete the applicable authorization flow, and requested scopes should be reviewed before enablement. Credentials should be treated as platform or provider secrets and should not be injected into prompts or model-visible instructions.

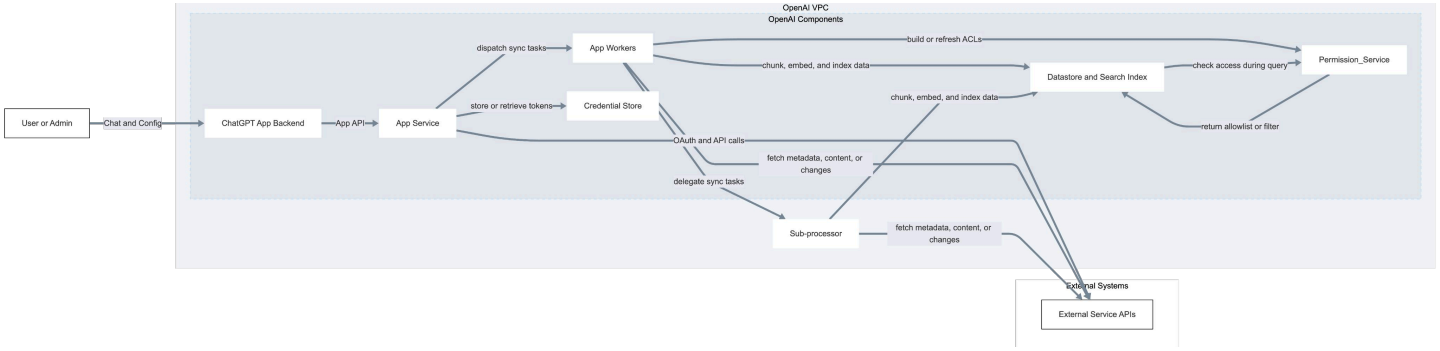
For Codex environments, administrators can govern access through workspace-level Codex enablement, role-based permissions, and managed configuration. Codex supports managed policy enforcement through `requirements.toml`, allowing administrators to constrain security-sensitive settings such as approval policy, sandbox mode, web search mode, and, optionally, which MCP servers users can enable.

Plugin Data Handling

Data accessed through plugin-enabled apps and MCP servers follows the data-handling model of the underlying capability. Skills bundled in a plugin provide workflow instructions and supporting resources; they do not independently authorize access to external systems unless paired with tools, apps, MCP servers, or executable logic that can access those systems.

By default, OpenAI does not train on app-accessed data in Enterprise workspaces. Plugin-enabled workflows should continue to inherit the applicable workspace retention, app data handling, and enterprise governance settings for the product surface where they are used.

Apps Platform Architecture



Apps Architecture Diagram

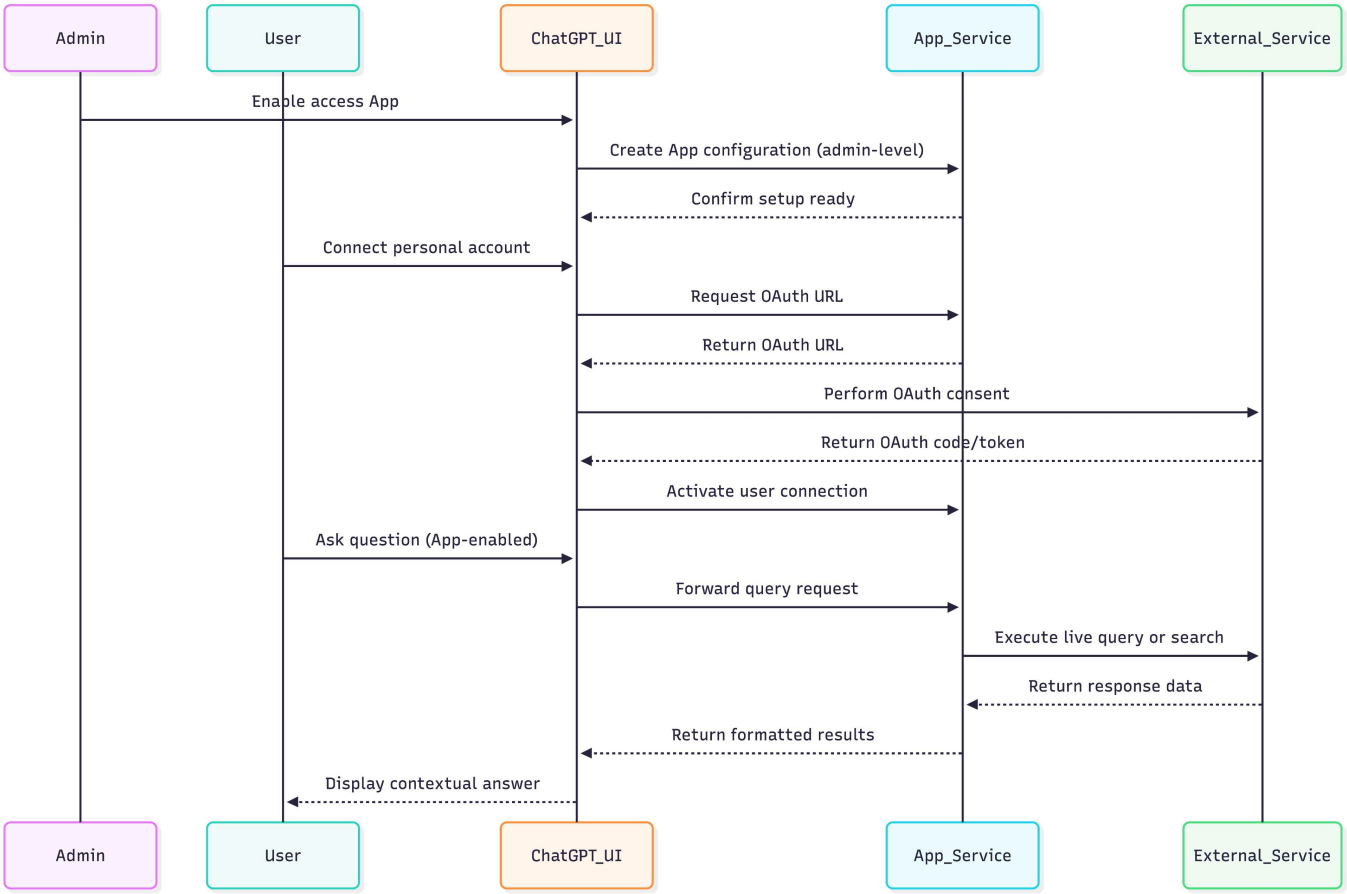
The ChatGPT App Platform operates securely inside the OpenAI VPC and enables ChatGPT to connect with enterprise systems in a controlled and trusted manner. It manages OAuth and identity flows, maintains secure integrations with external applications, and for Sync apps manages file indexing, messages, and permission access to help ensure accurate and contextual access. The platform supports App, Sync, and MCP custom apps, enabling organizations to securely integrate their systems across a wide range of use cases. It also provides the runtime interface that allows ChatGPT to fetch information and execute actions during conversations, powering secure, live interaction with enterprise data and workflows.

Authentication Flows & Setup

Apps Authentication

Apps employ industry-standard OAuth 2.1 flows designed to make it so each user can only grant access to their own data in a connected system. An administrator first configures and grants access at the workspace level for the app to be made available and can further restrict access using role-based access controls (RBAC). Once an app is enabled by a workspace admin, each user must explicitly authenticate with the external service and consent to the specific scopes requested (e.g. read access to files or calendar). Users can find the full list of permissions or scopes on the external service's OAuth screen. ChatGPT never sees passwords and only obtains an OAuth token if the user approves the connection.

Apps Auth Flow



App Auth Flow

Sync Apps Authentication

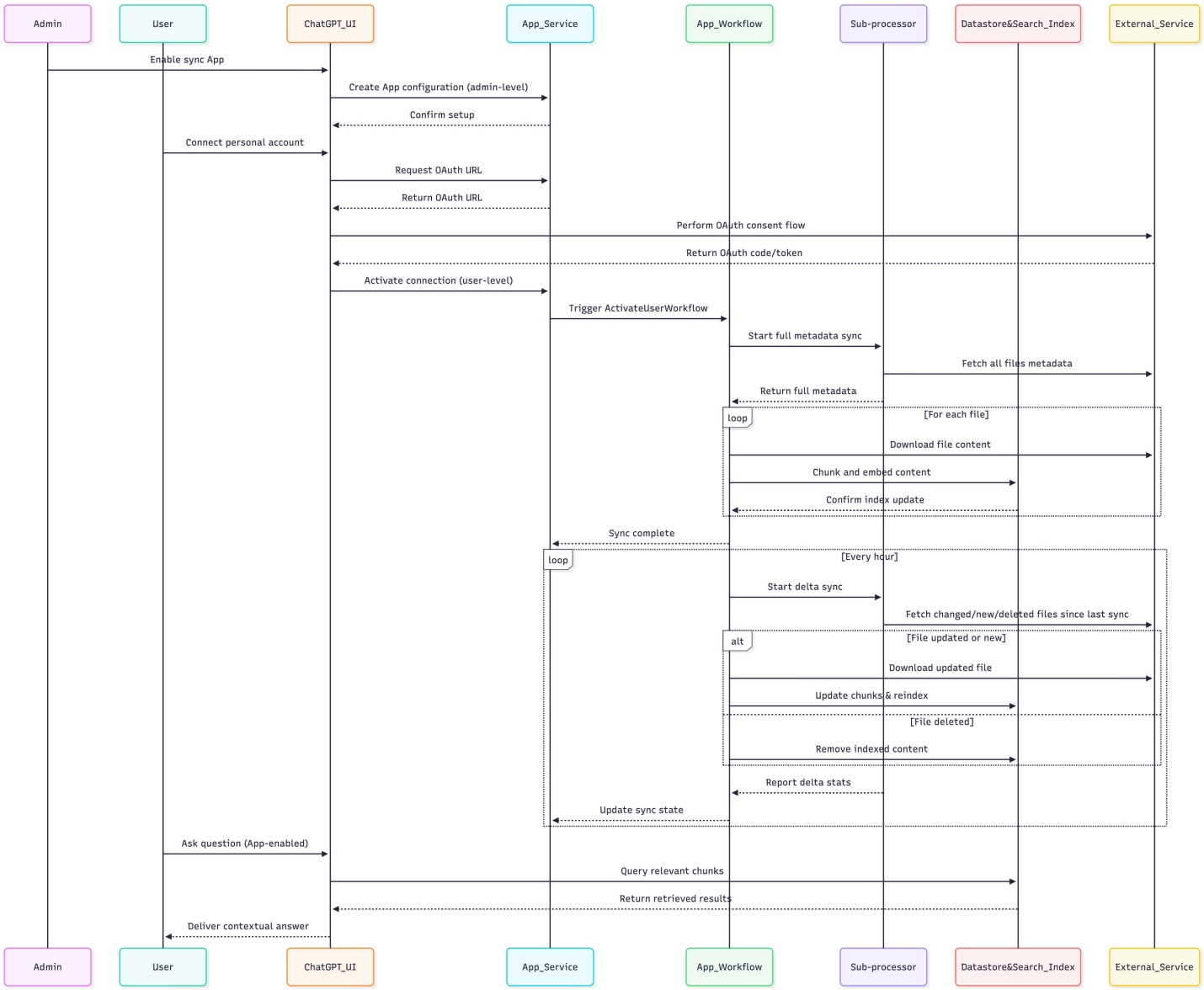
Self-Service Sync App Auth

To enable a Self-Service Sync app, an administrator first configures and grants access at the workspace level for the app and can further restrict access using role-based access controls (RBAC). Once enabled, an individual user must explicitly authorize their own account by completing an OAuth flow directly with the external content system (e.g., Google Drive, SharePoint, Box). This means that each app is user-scoped and permissioned independently. ChatGPT does not see passwords, obtains only an OAuth token if the user approves the connection, and cannot access any data without the user's explicit consent.

Access to content is restricted by the access control list (ACL) provided by the external content provider during OAuth. ChatGPT only indexes the files that the user is already authorized to access, and no elevated privileges are granted to OpenAI beyond the signed-in user's permissions. Content is securely retrieved, chunked, and embedded into a search index limited to administrator-approved data. In addition to the search index, the extracted content chunks are also stored to support retrieval. The indexed and stored content is encrypted at rest and safeguarded by OpenAI's standard enterprise security controls.

Following the initial full synchronization, apps perform hourly delta sync operations to capture file additions, modifications, or deletions. Updated content is re-embedded and reindexed, and content is removed from the index when it is deleted or when user permissions change, to maintain ongoing alignment with the user's access rights. This periodic sync helps ensure users continue to receive contextually relevant responses grounded in the current version of their content while respecting upstream security controls.

Self-service Sync App Auth Flow



Self-Service Sync App Auth Flow

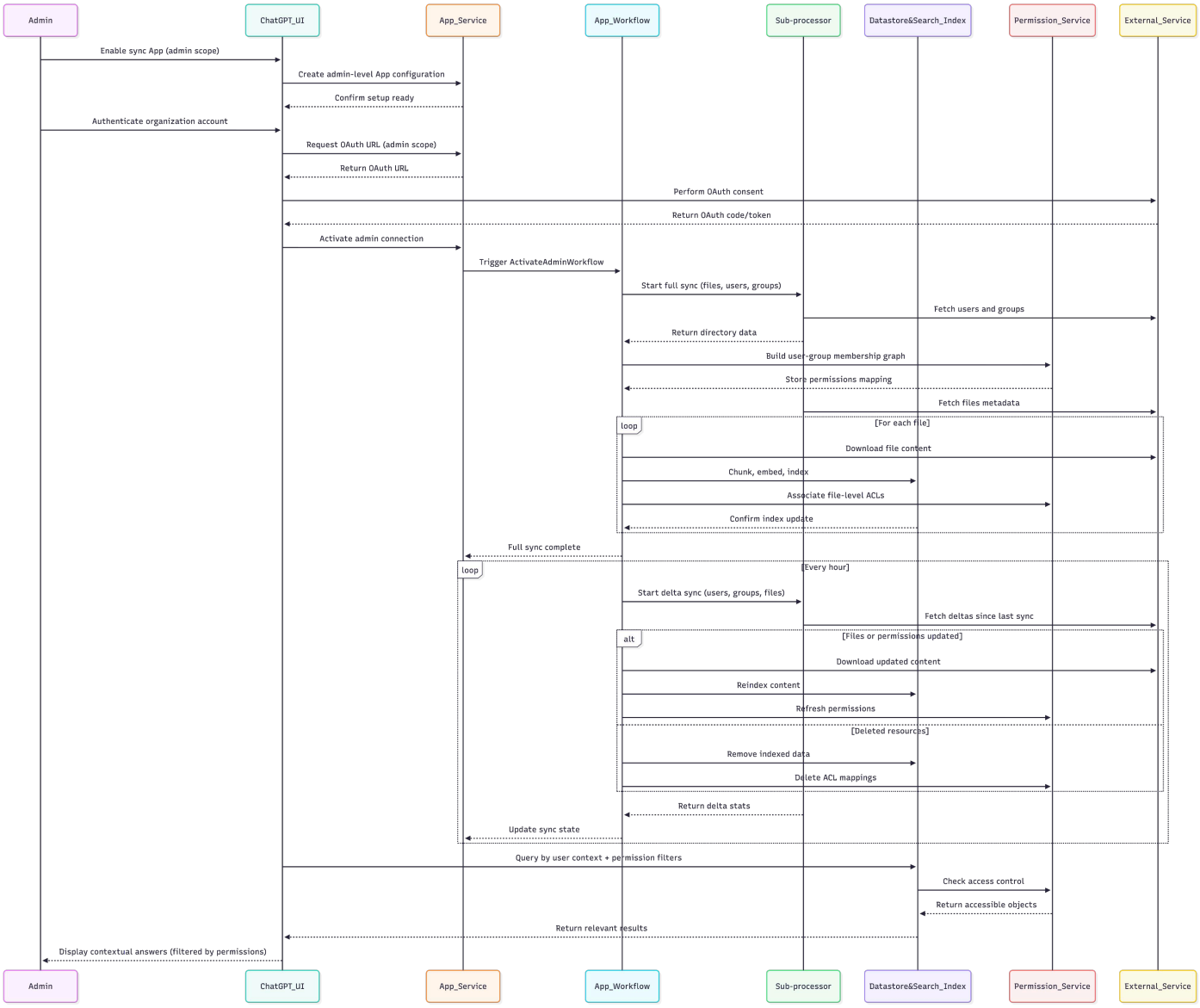
Admin-Managed Sync App Auth

For Admin-Managed Sync apps, workspace administrators centrally authorize and configure access to the app using RBAC-controlled permissions, complete the OAuth flow with privileged enterprise credentials from the external content system, and optionally apply further filtering to scope what data is synchronized. Additional implementation details and security controls for these deployment patterns are provided in the Appendix; Microsoft SharePoint and Google Drive documentation. This centralized authorization allows the app to access a workspace-wide corpus on behalf of users, without requiring each user to individually complete setup.

Access is governed based on file permissions, users, and groups retrieved from the source system. The app securely replicates the provider's permission graph into an encrypted index, ensuring ChatGPT only surfaces content a user is entitled to see. All access enforcement is continually aligned with the external system. Hourly delta sync operations refresh file metadata, user and group membership, and permission changes, and remove access to content when permissions are revoked upstream.

Because authorization and ACL replication are managed administratively, users receive a seamless experience: the app automatically maps each user's downstream permissions (e.g. SharePoint ACLs), but it only becomes usable if the app is enabled for the entire workspace or the user belongs to an RBAC group with access to that app. They can begin querying content immediately, benefiting from contextual responses grounded in their enterprise corpus. No additional action required from the user, and no exposure beyond their assigned entitlements.

Admin-managed Sync App Auth Flow



Admin-Managed Sync App Auth Flow

OpenAI engages a limited number of approved subprocessors subject to strict security, privacy, and compliance obligations.

MCP Custom Apps Authentication

MCP custom apps allow organizations to extend ChatGPT into their own or third-party applications via a secure remote procedure call pattern, where the MCP server is hosted and controlled by the customer or a trusted partner. Because these servers operate outside OpenAI's environment, they are subject to their own security and privacy controls, and organizations should enable only trusted MCP integrations and carefully review requested access scopes.

OpenAI implements multiple safeguards to help protect users even when interacting with unvetted MCP endpoints, including policy screening of MCP prompts, blocking of known malicious domains, checks on write-capable actions, and both online and offline moderation of tool calls. Write operations are designed to always require explicit user confirmation.

We recommend using OAuth 2.0 with Dynamic Client Registration (DCR) to make it so each user explicitly grants access under their existing entitlements; documentation on MCP's specification of authorization can be found here; Authorization - Model Context Protocol. When an MCP server declares a standards-compliant OAuth challenge, ChatGPT will guide the user through:

1. Registration at the MCP server's register endpoint (DCR)
2. Redirect through the /authorize endpoint for user authentication (PKCE included)
3. Secure token retrieval from the /token endpoint
4. Use of the issued token in subsequent requests as a bearer credential

If authentication fails or permissions/scopes have changed, the client halts the call and re-authenticates rather than retrying silently, preventing unintended access. Organizations should not connect to a custom MCP server unless they know and fully trust the underlying application.

IP Egress Ranges: All app traffic (Access, Sync, and MCP custom apps) routes through a set of dynamic IP egress ranges. You can explicitly allowlist the IP egress ranges here: https://platform.openai.com/docs/actions/production#ip-egress-ranges.

Authorization tokens: handling, storage, lifecycle, and deletion

Apps persist the minimal credential material required to perform their work and treat that material as a highly sensitive platform secret. In practice this means two broad classes of credentials: (1) per-user access and refresh tokens used by (per-user) apps, MCP custom apps, and Self-Managed Sync apps to mint short-lived access tokens for on-demand calls, and (2) privileged administrator/service-principal credentials used by admin-managed Sync (indexed) apps to build and maintain an encrypted, administrator-scoped search index. Regardless of type, credentials are used strictly to perform authenticated provider calls and are not injected into prompts or model inputs. The model is unaware of the authentication credentials they leverage through the apps system, which minimizes the possibility for an attacker to use prompt injection to retrieve them.

All credential traffic and token exchanges are protected in transit (TLS 1.2+) and tokens are encrypted at rest (AES-256) inside the platform's encrypted storage. OpenAI uses Azure Key Vault for OpenAI-managed keys and Vault handles short-lived DEKs while the cloud KMS protects the master KEKs. Apps support Enterprise Key Management (EKM), allowing customers to retain and manage their own master keys in their Key Vaults. Access to the Secure Vault and related key operations is governed by role-based controls and auditable access logging for accountability.

Runtime handling follows a strict minimization pattern. Short-lived access tokens are minted from stored refresh credentials only as needed, retained in memory for the duration of the upstream call, and discarded/zeroized almost immediately after use. The platform may use bounded, short-lived caching to limit provider load and improve latency, but caches are intentionally brief and revalidated to limit exposure windows. At the platform level, derived encryption keys (DEKs) that protect transient decrypted material are cached for short periods and are rotated on a regular cadence to further narrow any exposure window.

Refresh credentials themselves are treated as offline, sensitive secrets: they are persisted only when necessary, encrypted at rest, and handled in accordance with upstream provider semantics. Platform engineering recognizes provider-specific refresh semantics (including recommendations around single-use refresh tokens) and the app flows are designed to respect provider guidance, replace stored tokens when providers rotate them, and adopt conservative refresh-token hygiene as a defense-in-depth control.

Typical immediate triggers for revocation and deletion include user deletion, a user or administrator toggling an app off, or a user losing permission to the upstream resource. Because admin-scoped Sync apps persist an encrypted index and a denormalized permission graph, they rely on privileged credentials and therefore follow the same fundamental protections described above while operating under explicit admin consent and scoping. Sync apps are delta-first (preferring change notifications and incremental updates), maintain denormalized ACLs so the platform can enforce the source system's permissions at query time, and provide admin controls to re-sync or purge index data when credentials are revoked or scopes change.

Taken together, these practices provide layered assurance: token material is segregated from model inputs, encrypted in transit and at rest, handled with minimal runtime exposure, and subject to programmatic revocation and deletion on lifecycle events. For Enterprise/Edu/Business workspaces, app data is excluded by default from model training and is retained and processed under the same enterprise-grade key-management and retention policies that apply to other ChatGPT Enterprise data.

Provider Throttling & Rate-Limit Impacts

OpenAI's app architecture is designed to minimize and smooth API traffic to third-party provider APIs. We can prevent throttling by: (1) metadata-first discovery, (2) selective content fetches, (3) batching and pacing, (4) delta-based refresh for synced data, and (5) honoring provider rate-limit signals with exponential backoff and jitter. These patterns produce a predictable, low-impact footprint on provider services and scale cleanly for enterprise usage. However, throttling outcomes depend on the provider's own rate-limit policies and a customer's broader API activity across their ecosystem, which OpenAI does not have visibility into or control over.

Core design controls

Metadata-first discovery	Apps identify candidate resources with small search/listing calls that return metadata only, avoiding large transfers at query time. This keeps per-query provider load minimal.
App content fetching	After filtering by type, size and relevance, apps fetch content for the small set of items required to answer the user's prompt.
Batching & paced requests	Listing and content requests are batched and paced to avoid transient spikes. Discovery and indexing use paged APIs rather than unbounded queries.
Delta-first indexing (Synced apps):	Sync apps perform a controlled initial crawl to build a secure index, then maintain freshness by fetching only deltas (new/changed items) on a fixed cadence. Syncs are scoped by admin policy so no unsolicited tenant-wide crawling occurs.
Provider-driven backoff & retry	Apps honor provider rate-limit signals (HTTP 429, Retry-After, RateLimit-Reset) and implement exponential backoff with jitter and retry caps to avoid repeated retries that would worsen congestion. This behavior is implemented in client libraries and pipeline code.

Administrative Controls

Role-Based Access Control (RBAC)

ChatGPT includes granular, administrator-managed Role-Based Access Controls (RBAC) to control which apps an organization can use. In the workspace's app Settings, admins can enable or disable apps individually.

RBAC roles can be defined to grant app access to specific user groups. Availability of each app can be configured at the group level (for example, Google Drive for Group A vs Notion for Group B), allowing different roles to have access to different apps. Custom roles can combine any mix of App, Synced, and MCP custom app settings. Role membership can be managed manually or through SCIM and SSO integration with an identity provider.

Managing Connected Accounts

Domain Claiming:
Enterprises can assert control over corporate app usage by claiming ownership of their corporate email domain (e.g., @acme.com) with a request to your OpenAI Account Director or Account Owner. Once a domain is claimed, only workspaces associated with that verified enterprise can enable apps that require authentication to the enterprise's corporate systems.

When a user attempts to complete an OAuth flow for an app, OpenAI verifies that the email domain associated with the authenticated account matches a domain claimed by the workspace based on our core-SSO API. If the domain has been claimed by a different organization, or if the current workspace is not authorized for that domain, the app authorization is immediately invalidated and the connection request is rejected. This makes it so employees cannot connect corporate systems to personal ChatGPT accounts or unapproved workspaces, guarding against data leakage and maintaining strict boundary control between enterprise and personal environments.

Supported:

- Support multiple domains claimed per enterprise workspace (e.g.: acme.com, acme.co.uk, subsidiary.io).
- Detect and block attempts to add an app by users on the claimed domain outside the approved workspace, and users on an unrecognized domain.
- Continue to surface existing workspace consolidation flows for users who signed in to another workspace with a company email, guiding them to the correct enterprise workspace.

User experience scenarios:

Situation	What Users See	Result
Employee with corporate email in the Enterprise workspace attempts to connect to corporate SharePoint	Success	App works.
Employee with corporate email in a personal or Business (non-domain claimed) workspace attempts to connect to corporate SharePoint	Block + guidance to join Enterprise workspace: "Your company has an enterprise workspace that manages app access. Contact your enterprise admin to get access."	App restricted and no data access.
Personal email (gmail, outlook) tries to connect corporate SharePoint	Block, generic message (will not share IT admin details)	App restricted and no data access.

Domain Restriction:

For supported applications, ChatGPT provides administrative controls that allow organizations to restrict external calls to approved domains. This helps reduce the risk of unintended connections to unapproved third-party endpoints and gives workspace administrators an additional layer of control over how external actions are invoked from within ChatGPT.

For Enterprise and Edu environments, domain restrictions complement existing workspace governance controls for apps and actions by helping ensure that external integrations are limited to organization-approved destinations.

Admins can configure this option by selecting **Manage Domains** from the dropdown in their ChatGPT Workspace App Settings.

Where available, domain restrictions operate alongside existing administrative controls such as workspace app enablement, role-based administration, and action-level governance. This provides organizations with a more granular mechanism to manage how ChatGPT interacts with third-party systems and helps align external connectivity with internal security and compliance requirements. An internal GTM product knowledge response also references domain restriction as part of the Enterprise control surface for Actions.

Availability of domain restriction controls may vary by integration type and supported app surface. Supported applications to date include Microsoft Teams, Microsoft SharePoint, Microsoft Outlook, Gmail, Google Calendar, Google Contacts, Google Docs, Google Sheets, Google Slides, Google Drive, GitHub, and Slack; this is subject to change.

Plugin Directory

The Plugin Directory is OpenAI's official inventory and control center for all apps that interact with ChatGPT and related enterprise surfaces. It provides a clear, consistent record of each app's available actions and security details, including authentication methods, data handling behaviors, and permission scopes. By standardizing how apps are governed for your organization, the Plugin Directory helps ensure that every integration, whether developed by OpenAI, a partner, or a customer, is subject to your oversight.

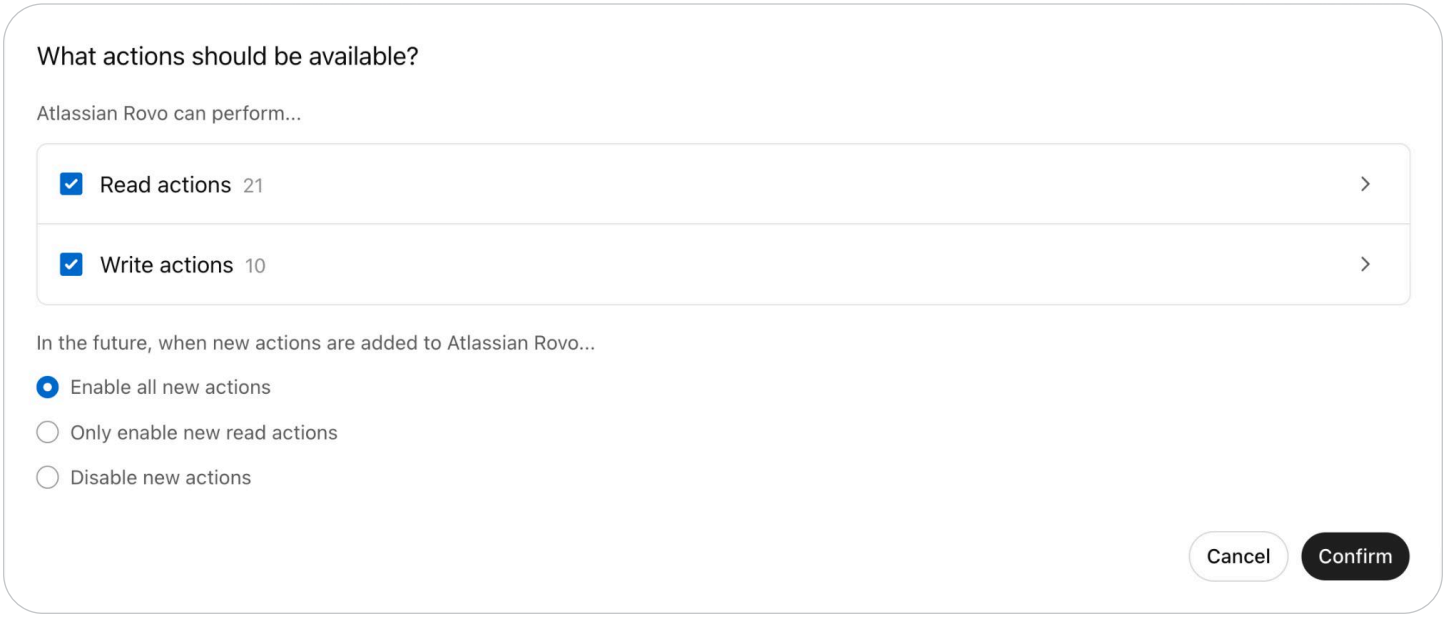
Write Capabilities and Action Controls

ChatGPT apps can support both read and write capabilities, with administrative controls designed to help organizations manage higher-risk actions more deliberately. For Enterprise and Edu workspaces, administrators can configure role-based access controls to allow actions and constraints for apps, including controls that distinguish between read and write behaviors.

For external write actions, ChatGPT requires user confirmation before execution. This provides an additional safeguard for operations that may create, update, or otherwise modify data in connected systems. Together, these controls help organizations apply a more deliberate approval model for write-capable actions rather than treating all available actions as automatically trusted.

This control model is intended to help organizations balance extensibility with oversight. Rather than enabling all available operations by default, ChatGPT supports administrative governance, scoped permissions, and explicit confirmation for higher-impact actions, helping customers manage the risks associated with write-capable workflows in connected systems.

Admin Controls for App Actions



ChatGPT has simplified how admins manage app actions through action-control options at the app level. Admins will be able to quickly enable Read actions and Write actions, or expand them to see individual actions. This gives workspace admins a simpler way to manage the actions users can access while still supporting fine-grained control where needed.

When Read actions are selected, users can access all read actions for that app, and a setting will be selected to automatically enable any new read actions added in the future. When both Read actions and Write actions are selected, users can access all available actions for that app, and a setting will be selected to automatically enable any new actions added to the app in the future.

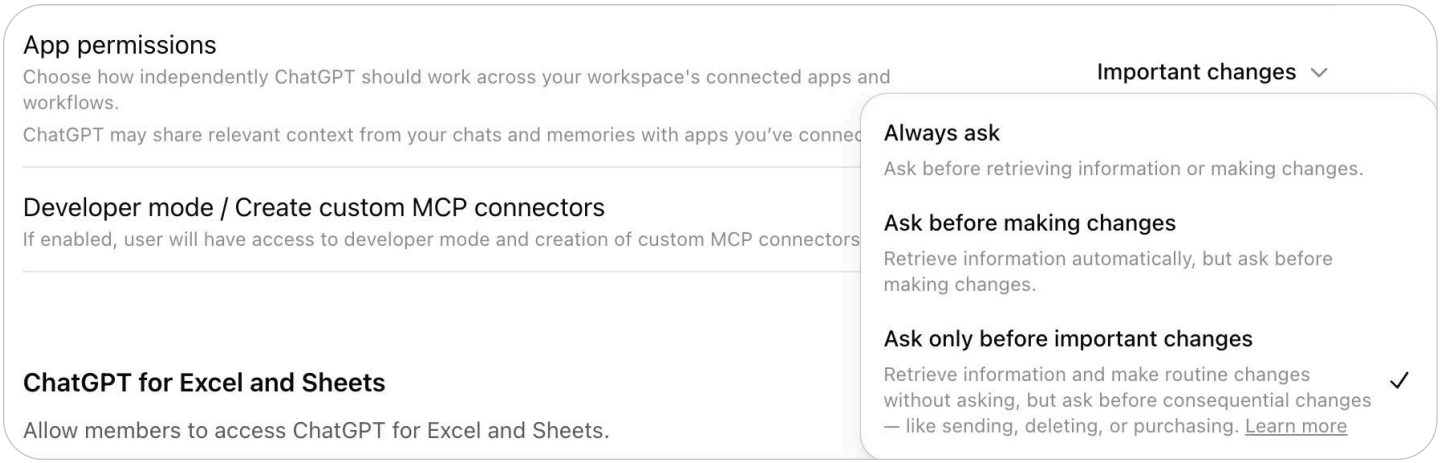
By expanding Read actions or Write actions, admins can continue to manage actions at a granular level. This allows organizations to enable or disable specific actions based on their internal governance requirements, risk tolerance, or rollout strategy.

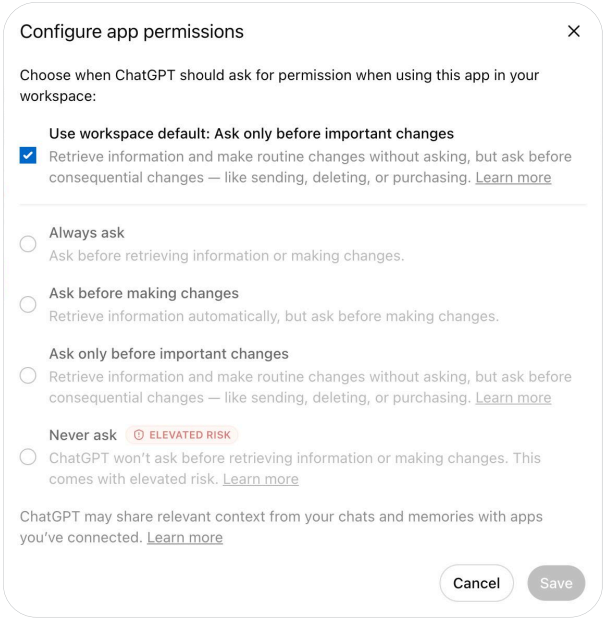
For new actions, admins can choose between enabling all new actions, only enabling new read actions, or disabling new actions. These options are designed to reduce administrative overhead for teams that want app capabilities to stay current as apps evolve. For apps configured to disable new actions, any new actions added in the future must be manually reviewed and enabled by an admin before they become available to users.

Privacy Governance for Apps

In addition to controlling whether users can access specific app capabilities, enterprise administrators can configure privacy protections that govern how ChatGPT invokes those capabilities during a conversation. These controls allow organizations to choose an interaction model aligned with their risk posture. Depending on workspace configuration and product settings, administrators may enforce user confirmation that determines how independent ChatGPT should work with connected apps and workflows.

Administrator configuration may specify whether a user confirmation is required before all app reads and writes, allowing reads without confirmation while requiring confirmation before writes, asking only before important or potentially sensitive actions, or allowing broader app use subject to applicable safety, security, abuse, and policy protections.





These privacy controls are distinct from app permission controls. App permission controls determine which app capabilities are available to users in the first place. Privacy controls govern how ChatGPT may exercise those already-permitted capabilities during an interaction, including when user confirmation is required before an outbound app request proceeds.

Memory and personalization are also relevant to app privacy because they can help ChatGPT determine whether an app should be invoked and how to construct a useful request. For Enterprise workspaces, administrators may control whether memory and personalization-related behavior is available within the workspace.

Importantly, connected apps do not receive a standalone memory store, a persistent feed of user memories, or generalized access to ChatGPT personalization data. An app receives only the information included in the specific outbound request generated to complete the user's task. These boundaries complement existing administrative controls by helping organizations govern not only which app capabilities are available, but also what information may be shared when those capabilities are invoked.

Apps in Codex

Codex supports app and MCP-based extensibility while providing organizations with administrative controls to govern access and configuration. In Business, Enterprise, and Edu environments, Codex can be enabled at the workspace level and administered through role-based permissions, including distinct controls for Codex access, administration, and plugin enablement. **Plugins** are extensions that allow Codex to connect with approved tools, MCP servers, or workflow capabilities inside the Codex environment. This enables organizations to determine which users may access Codex and which administrators may manage Codex policy and configuration.

Codex also supports managed policy enforcement through requirements.toml, allowing administrators to constrain security-sensitive settings. These managed policies can govern settings such as approval policy, sandbox mode, web search mode, and, optionally, which MCP servers users can enable. For Business and Enterprise users, managed requirements can be distributed through the Codex service and applied across Codex surfaces, including the CLI, app, and IDE extension.

It is important to note that ChatGPT app controls and Codex controls are related but distinct. Controls that govern custom MCP connectors in ChatGPT do not directly govern Codex CLI or IDE workflows. Accordingly, organizations should treat ChatGPT apps governance and Codex governance as separate administrative control surfaces, even where both rely on workspace identity and role-based access control.

Data Handling

Data accessed through apps stays within the OpenAI Enterprise trust boundary and is protected by multiple layers of defense designed to prevent unauthorized access or exposure. This section outlines how encryption, permissions enforcement, enterprise key controls, and proactive threat mitigations work together to help customer data remain private and under the organization's governance at all times.

Encryption & Data in Transit

All data handled by apps is protected with encryption in transit (TLS 1.2+) and at rest (AES-256). Retrieved content and index data (for synced sources) are stored in our Azure Cloud environment under the same security controls as other ChatGPT Enterprise data. OAuth credentials (tokens) are stored using strong encryption and audited key management practices.

Enterprise Key Management

Enterprise Key Management (EKM) gives organizations full control over their encryption keys by allowing them to manage their own keys through their preferred Key Management Service (KMS), including Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). Customer data stored at rest within OpenAI's enterprise environment is encrypted using envelope encryption, where customer-managed keys control access to the underlying data. OpenAI never stores or manages these master keys directly. If access is revoked through the customer's KMS, the associated data becomes inaccessible and effectively unreadable. Key usage is logged by the customer's KMS, supporting independent audit and compliance reporting. This provides enterprises with direct cryptographic control over their data across both ChatGPT Enterprise and the OpenAI API, where data controls are built on OpenAI's SOC 2 Type II and ISO 27001 compliant infrastructure.

Currently, EKM enabled workspaces have access to all App and MCP custom apps. The support of Sync apps will come in the future. While a dedicated private network tunnel to customer-managed KMS endpoints is not yet supported, organizations can enable strict network isolation today using IP allowlisting. This gives enterprises strong perimeter enforcement over how and where key access requests may originate while we continue to develop expanded private networking options.

Mutual TLS (mTLS) Support for ChatGPT MCP Connections

ChatGPT supports mutual TLS (mTLS) for connections to custom MCP servers, providing an additional transport-layer mechanism for identifying ChatGPT as the connecting client. When establishing a TLS session to an MCP server, ChatGPT can present an OpenAI-managed client certificate. Organizations that validate client certificates can configure their services to trust the OpenAI certificate chain and use mTLS as an additional control to confirm that inbound connections originate from ChatGPT

This mTLS capability is designed to authenticate the ChatGPT client at the connection layer. It does not replace application-layer authorization or user identity controls. OpenAI continues to recommend OAuth 2.1 for end-user authentication and authorization, with customer systems responsible for validating access tokens, enforcing scopes, and applying resource-specific access policies before executing tool actions.

OpenAI-managed certificates are used for this mTLS flow. ChatGPT does not rely on customer-provided client certificates or custom API keys for this transport authentication model. This design helps provide a consistent, managed trust model for customers that want stronger client identification controls when exposing MCP servers to ChatGPT.

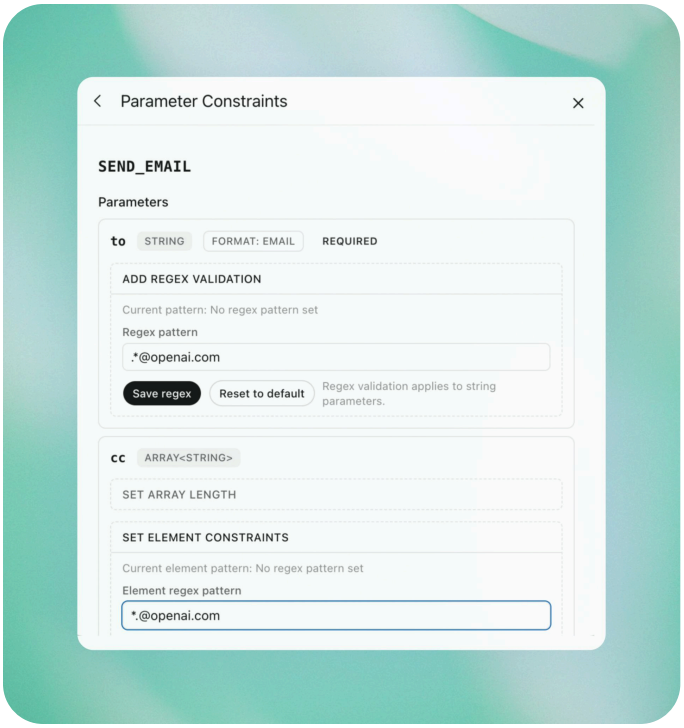
Documentation reference: https://developers.openai.com/apps-sdk/build/auth#mutual-tls-mtls

Parameter Constraints

Parameter Constraints provide an additional control layer for OpenAI Enterprise administrators by allowing organizations to define explicit, enforceable rules for tool inputs. These constraints help make apps and custom MCP servers safer and more predictable by limiting what data can be sent as parameters when a model invokes a tool. For example, an organization could constrain an email tool so that the to: field for a send_email action only accepts addresses within an approved domain, or constrain email search so that the model can only search messages from approved senders.

This approach helps reduce the risk of unintended data exposure and prompt-injection-driven misuse while still allowing models to operate autonomously within clearly defined boundaries. Constraints can be applied across different types of app actions, including but not limited to:

- **Restrict outbound email recipients**: Admins can constrain an Email MCP server so the `send_email` action only allows the `to:` field to include approved domains, such as `@openai.com`. This helps reduce the risk of prompt injection causing messages to be sent to unintended external recipients.
- **Restrict file sharing to trusted domains**: Admins can apply constraints to Google Drive actions such as `share_file`, limiting sharing to recipients within an approved corporate domain. This helps prevent unintended external file sharing and supports data boundary enforcement.
- **Restrict GitHub access to trusted repositories**: Admins can constrain GitHub App actions such as `fetch_issue` so the `repo` field only permits repositories within the organization, for example by using a pattern like `^openai\/.*`. This helps prevent retrieval of data from untrusted or out-of-scope repositories.



Parameter Constraints are designed to combine deterministic enforcement with a more usable experience. Instead of relying only on server-side rejection after an invalid tool call, the model can understand the applicable constraints before attempting an action. When a user request would violate a constraint, ChatGPT can explain the limitation and help the user reformulate the request in a way that complies with the organization's policy.

As part of a broader defense-in-depth approach, Parameter Constraints complement existing protections such as sandboxing, monitoring and enforcement, controls for higher-risk capabilities, protections against URL-based data exfiltration, and enterprise administrative controls.

Secure Tunneling for Internal MCP Servers

ChatGPT supports secure tunneling for organizations that want to connect internal MCP servers without exposing those services directly to the public internet. In this model, customers deploy a tunnel client inside their own network. The ChatGPT connector is configured to use an OpenAI-hosted MCP tunnel endpoint, while the customer-run tunnel client maintains an outbound HTTPS connection to OpenAI and forwards requests to the internal MCP server. This design allows organizations to keep the MCP server on private infrastructure while avoiding direct inbound exposure for the tunnel itself.

The Secure Tunnel model is designed for deterministic request-response workflows and supports deployment patterns such as host-based services, Docker containers, Kubernetes sidecars, or dedicated Kubernetes deployments

From a network posture perspective, the tunnel is designed to require outbound connectivity from the customer environment to the OpenAI tunnel control plane and outbound connectivity from the tunnel client to the customer's MCP server. No inbound ports are required for the tunnel mechanism itself. This can help organizations reduce the operational and security burden of publishing internal services externally while still enabling ChatGPT-connected workflows.

The tunnel client also supports enterprise operational controls. Customers can configure optional mutual TLS (mTLS) between the tunnel client and the internal MCP server, provide custom CA bundles for outbound TLS trust, route traffic through corporate HTTP proxies, and monitor tunnel health through health, readiness, and Prometheus-compatible metrics endpoints. Authentication and authorization for the MCP server and downstream systems remain under customer control; the tunnel provides the transport path, while access decisions continue to be enforced by the customer's own services and policies.

Permission Enforcement

Apps strictly enforce the source system's authorization model, including the user's established ACLs and OAuth-granted permissions, ensuring ChatGPT can only access data the user is already entitled to view.

For Apps and MCP custom apps, no full-document replicas are created, content is fetched just-in-time and only the snippets required to answer a request are processed as transient context. This helps ensure that when access is revoked upstream or a file is removed, subsequent queries through ChatGPT reflect the change.

For Sync apps, only user authorized content (Self-Managed Sync) or administrator approved content (Admin-Managed Sync) is indexed. It is transformed into a secure, encrypted, and permission-aware search index rather than stored as full document replicas. The index is logically isolated per customer and enforced against the user's ACLs and OAuth-granted permissions to help ensure results are only surfaced when access is authorized. Hourly delta syncs update or remove content and permissions so revocations in the source system are reflected in ChatGPT. No indexed data is shared across tenants or used to train OpenAI models by default, preserving enterprise confidentiality and governance.

Together, these controls maintain strict alignment with customer governance policies and help prevent unauthorized access propagation across the ChatGPT surface.

Threat Mitigations

OpenAI proactively mitigates prompt injection risks through multiple layers of defense applied across our model training and runtime systems. We train our models on a clearly defined instruction hierarchy so they consistently prioritize trusted, higher-privilege directives (system and developer instructions) over untrusted user or document content. To strengthen that hierarchy, we incorporate large volumes of synthetic adversarial examples during fine-tuning and evaluation; including cases where malicious instructions are obfuscated or embedded in credible content. These measures have measurably increased resistance to jailbreaking and prompt injection without degrading model performance.

We complement model-level hardening with product-level protections designed to prevent data exfiltration through compromised content or hidden instructions based on a source-and-sink threat model of prompt injection. This includes proactive link-generation controls that block untrusted destinations, explicit user-approval prompts prior to consequential actions (read and write), and internal monitoring systems designed to detect, flag, and block anomalous browsing or app activity. Prompts that could redirect models to leak sensitive data, such as those exploited in the publicized AgentFlayer chain, can be mitigated with monitors and tools deployed across our apps and agents technology. These combined safeguards work to make sure that prompt injection vulnerabilities are both anticipated in design and rapidly mitigated in production.

App and MCP Custom App Data Handling

When data is accessed through an app, information is retrieved securely on demand and only for the duration of the request. Apps do not create long-term storage or search indexes of customer content. Instead, file or system content exists only as temporary context to generate the response, adhering to ChatGPT Enterprise data retention policies. The app enforces the source system's permissions, limiting users' access to information they are already entitled to view. If user access is revoked at the source, that revoked access is reflected through the app.

MCP custom apps operate through a secure remote procedure call pattern in which the customer or trusted partner hosts the server-side implementation. Authentication behavior is determined by that MCP server's configuration. We recommend using OAuth and dynamic client registration. When an MCP server presents a standards-compliant auth challenge, the client halts the call and surfaces a re-authorization flow (no silent retries), and write actions are safeguarded by import-time scanning, online/offline moderation, and explicit user confirmation. We recommend that you do not connect to a custom MCP server unless you know and trust the underlying application.

Across both app types, all data is encrypted in transit and at rest, and app credentials are encrypted and can be revoked by administrators at any time, reinforcing customers' existing governance and compliance controls

Safety Measures for Custom MCP Extensions

OpenAI has designed multiple layers of security controls specifically for MCP custom apps to reduce the risk surface introduced by third-party or self-hosted MCP servers:

1. **Administrator approval** for new or modified tools/actions before they become available for execution.
2. **Online and offline monitoring** to detect prompt-injection or malicious content originating from the MCP server.
3. **Explicit user confirmation** for write or modification actions executed via the MCP server.

For further details on MCP risk and safety mitigations, refer to our MCP documentation.

Sync App Data Handling

Sync apps operate within OpenAI's Enterprise cloud environment, which is hosted on Microsoft Azure and subject to the same production security and compliance controls as ChatGPT Enterprise. This environment is isolated from our public services and adheres to SOC 2 Type II and ISO 27001 standards. It includes strict access controls, encryption at rest and in transit, continuous monitoring, and a documented incident response program.

App data flows through three main secured layers.

- Ingestion Layer: Authorized content is transferred from the source system through encrypted channels, authenticated using standard OAuth-based permissions.
- Indexing & Storage Layer: Content is processed into a permission-aware index stored in OpenAI's enterprise infrastructure. Each customer's index is logically isolated, ensuring data segregation across tenants.
- Retrieval Layer: When a user issues a query, only content they are entitled to access is returned. Access checks are enforced using both the app's permission metadata and workspace-level controls.

Some sync apps support additional configuration options that allow administrators to further limit what content is ingested and indexed. These capabilities may include granular source selection, such as restricting sync to specific locations or content boundaries, as well as honoring source-system metadata related to sensitivity, classification, or access restrictions. Where available, these controls help enterprises apply data minimization principles and align app sync behavior with their internal governance and data handling policies. OpenAI may continue to expand these capabilities across sync apps over time.

App data remains within the multi-tenant ChatGPT Enterprise trust boundary. No app data is shared across customers or used for model training by default. Indexed data is encrypted at rest using AES-256 and is subject to enterprise retention and deletion policies. If an app is disabled, its associated index is promptly deactivated and scheduled for deletion in accordance with OpenAI's data handling and compliance standards.

Sync apps store several categories of metadata to help ensure secure, accurate, and efficient data integration. This includes metadata such as: identity, location, permission and sharing (only for Admin-Managed sync apps), versioning, schema, sync, audit and logging, sync state, and data classification and sensitivity tags.

Skills

Skills provide a reusable way to package task-specific instructions, supporting resources, and workflow guidance into portable artifacts that can help users and AI systems perform repeatable tasks more consistently. They are especially useful for workflows that benefit from structured execution, reusable context, or organization-specific best practices. Rather than requiring users to restate the same instructions repeatedly, skills can help standardize how common work is performed and improve the consistency of outputs over time.

In the context of ChatGPT apps, apps are complementary to skills rather than a replacement for them. Apps provide access to external systems, enterprise data, and actions, while skills can provide reusable workflow logic that helps users or agents apply those connected capabilities in a more structured and repeatable way. In practice, this means apps can serve as the connection layer to systems and services, while skills help guide how those capabilities are used to complete specific tasks.

From a governance perspective, skills should be treated as managed workflow assets. Because skills may include structured instructions, supporting resources, and in some cases executable logic or scripts, organizations should apply appropriate review before broadly distributing them. Externally sourced or third-party skills should be evaluated carefully prior to use, particularly where they may execute scripts or operate with broad permissions. Organizations should also maintain ownership and review processes so that outdated or inaccurate skills can be updated or deprecated as workflows evolve.

Compliance and Data Residency

Data accessed through apps remains within the ChatGPT Enterprise trust boundary. OpenAI's platform is SOC 2 Type II and ISO 27001 certified, and adheres to GDPR requirements for data handling. (Note: HIPAA-covered data can be supported in apps, through a customer's BAA with their service provider; apps are not yet included under OpenAI's BAA.). All Synced apps currently store their search indexes in U.S. datacenters by default (support for additional regions is on the roadmap).

For MCP custom apps, any data sent via MCP calls is inherently routed to an external service outside of OpenAI's trust boundary. As such, even in a workspace configured for regional data residency, information transmitted to a custom or third-party MCP must be treated as external data sharing and is subject to the data handling, residency, and compliance controls of the receiving provider. Customers are responsible for ensuring that any connected MCP service meets their regulatory and data residency requirements.

Compliance Log Platform

The Compliance Logs Platform provides enterprises with secure visibility into ChatGPT and OpenAI platform activity by enabling authorized administrators and compliance teams to export compliance-relevant data for audit, regulatory, e-discovery, and internal governance purposes. It operates within the applicable OpenAI enterprise trust boundary and is designed for reliable ingestion into customer-managed systems such as SIEMs, data lakes, DLP tools, and e-discovery platforms.

The platform delivers immutable, time-windowed JSONL files grouped by event type, rather than requiring customers to poll multiple object-level endpoints. These files are generated frequently and may include categories such as conversation messages, audit logs, authentication logs, Codex activity, and app-related events. Additional technical details on app-related event schemas are available in the Compliance Log Platform specification, which is only accessible to users who are logged into ChatGPT as an owner or administrator of their workspace. Each event follows a schema that includes a stable event identifier, event type, timestamp, principal, and actor information, enabling customers to correlate activity and maintain an auditable record across systems.

App Log Schema

Each event conforms to the shared Logs Platform envelope described above with these additional top-level fields:

- `type`: Always `APP_LOG`.
- `app_id`: Stable identifier of the App configuration (string).
- `app_name`: Human-readable name of the App at invocation time (string).
- `app_type`: Classification of the App backend. Possible values: `SERVICE` (service-backed connector such as Google or Microsoft), `OPEN_API` (OpenAPI-defined connector), `MCP` (connector backed by an MCP server), `FIRST_PARTY_ECOSYSTEM` (first-party ecosystem connector), `UNKNOWN` (type not captured).
- `conversation_id`: Conversation identifier related to the invocation (string, nullable) – may be `null` for background or system initiated calls.
- `log_type`: Indicates whether the event captures the request or the response phase. Possible values: `request`, `response`.
- `input`: Object containing the request payload sent to the app (present on request events; may be omitted or `null` on `response`).
- `output`: Array of objects containing the normalized response items returned by the app (present on `response` events; omitted on `request`).

Additional app-specific keys may appear; they should be treated as informational and are subject to change. Always rely only on the documented fields above for stable ingestion.

Example request

JSON


```
 "event_id": "31c1b9b0-5b2d-4c2a-9e11-32d7b0c3e9aa",
  "type": "APP_LOG",
  "timestamp": "2025-11-15T20:12:24.551203Z",
  "principal": {
    "id": "be545252-ad04-4cfa-9ca5-deca58416151",
    "type": "CHATGPT_WORKSPACE"
  },
  "actor": {
    "type": "ACCOUNT_USER",
    "user_id": "user-AbCdEf1234567890",
    "user_email": "user@example.org"
  },
  "app_id": "conn-8f92d7c1",
  "app_name": "gdrive",
  "app_type": "SERVICE",
  "conversation_id": "c-7cda9b2e",
  "log_type": "request",
  "input": {
    "query": "List recent architecture decisions"
  }
}
```

Example response

JSON


```
{
  "event_id": "31c1b9b0-5b2d-4c2a-9e11-32d7b0c3e9ab",
  "type": "APP_LOG",
  "timestamp": "2025-11-15T20:12:25.012944Z",
  "principal": {
    "id": "be545252-ad04-4cfa-9ca5-deca58416151",
    "type": "CHATGPT_WORKSPACE"
  },
  "actor": {
    "type": "ACCOUNT_USER",
    "user_id": "user-AbCdEf1234567890",
    "user_email": "user@example.org"
  },
  "app_id": "conn-8f92d7c1",
  "app_name": "gdrive",
  "app_type": "SERVICE",
  "conversation_id": "c-7cda9b2e",
  "log_type": "response",
  "output": [
    {"display_title": "ADR-017 Caching Strategy", "display_url": "https://
confluence.example.org/x/Qw12"},
    {"display_title": "ADR-016 Messaging Bus", "display_url": "https://
confluence.example.org/x/Rx34"}
  ]
}
```

Logs are designed for low-latency ingestion, with a p99 latency objective of less than 30 minutes from event time to inclusion in a log file. Records reflect the state of the data at the time the event occurred, supporting auditability even when the underlying data existed only transiently. Files are immutable once written and retained by OpenAI for a limited 30-day window. Customers requiring longer retention should continuously download, parse, de-duplicate, and store logs according to their own compliance and governance policies.

The Compliance Logs Platform is the forward-looking foundation for enterprise compliance exports. It is designed for ongoing data capture and does not support historical backfill.

Interactive Apps and UI Widgets

The App platform can be extended by allowing customers and approved third-party developers to bring custom workflows and interactive UIs directly into ChatGPT. Interactive Apps are MCP custom apps and operate outside OpenAI's environment through developer-hosted MCP servers, and therefore introduce an external trust boundary; organizations should enable only trusted apps and review declared access scopes carefully. MCP custom apps inherit the same core security posture described in this whitepaper, including tenant/workspace isolation, encryption in transit and at rest, workspace retention policies, Enterprise Key Management, IP-egress ranges for allowlisting and admin-managed enablement through RBAC.

An interactive app typically includes (1) a developer-operated MCP server that exposes tools to ChatGPT, and (2) an optional widget that renders UI within the conversation. MCP custom apps can read data, orchestrate multi-step workflows, and perform write actions. The additional capabilities of widgets are designed to reduce repetitive prompt handoffs while preserving enterprise governance and auditability. MCP custom apps are not designed to receive bulk workspace content or full conversation history unless explicitly included in a specific tool call for a defined user workflow.

Architecture and Trust Boundaries

Interactive apps rely on the same model-to-tool orchestration described in the apps Platform Architecture section. When the model invokes an app tool, the tool call is sent to the app's MCP server (developer backend), which performs the requested operation and returns a structured response. Because MCP servers operate outside OpenAI's environment, OpenAI does not control the developer's downstream storage or retention practices; enterprises should treat app backends as third-party services subject to their own security, privacy, and contractual obligations.

Widgets, when present, run inside an OpenAI-controlled sandbox within the ChatGPT client. Widget code executes in a restricted runtime that enforces Content Security Policies (CSPs) and network sandboxing. By default, widget code can only communicate with its declared MCP endpoints; any broader egress requires review and approval, is logged, and can be revoked if elevated risk or policy violations are detected.

Interactive App Authentication and Access Control

Interactive apps follow the MCP custom app model for identity and least-privilege access. Depending on configuration, apps may authenticate using user-scoped OAuth flows or administrator/service credentials, with downstream permissions continuing to be enforced by the source system. OpenAI does not inject authentication credentials into model prompts; all tokens are treated as platform secrets, encrypted in transit and at rest, minted only when required, and revoked upon defined lifecycle events.

Administrative control is enforced through explicit enablement and RBAC-based governance. Admins must explicitly approve each app and may restrict availability to specific user groups, enabling selective rollout and policy-aligned access control consistent with existing app governance models.

Operational Safeguards for Apps

Third-party apps are expected to comply with OpenAI's policies and technical requirements, and OpenAI may apply automated and programmatic controls to detect and mitigate policy-violating or abusive behavior. Where warranted, OpenAI may restrict an app's availability, limit functionality, remove it from directories, or revoke access to protect users and the platform.

Whisper State (Widget to ChatGPT Bridge)

Some widgets use whisper to keep the model synchronized with the UI state. A whisper is a discreet, app-to-model message sent through the widget runtime that does not appear as a visible conversation turn. It provides concise, structured context about user interactions in the widget (such as selected filters, records, or step completion) so the model can correctly continue multi-step workflows without requiring the user to restate UI choices in chat. Apps are designed to whisper only the minimal state required for the workflow and are not intended to replay unrelated or sensitive conversation content through whispers.

Whispers do not grant new access or privileges and are subject to the same online and offline safety monitoring and policy enforcement as standard user prompts and tool inputs. They cannot bypass RBAC, retrieve additional workspace data, or execute actions directly. Tool calls or write actions must still be generated by the model under the user's existing entitlements and, for consequential actions (including write actions), OpenAI policy requires explicit user confirmation and that the action be subject to applicable platform safety checks and enforcement. Whisper content is treated as untrusted third-party input and is expected to be narrowly scoped to workflow state.

Incident Response to Apps

OpenAI maintains a documented incident response program for Enterprise services (with severity-driven SLAs) and applies the same approach to interactive apps.
If an app is suspected to be malicious or violates platform policy,

OpenAI can:

- Immediately disable the app or widget runtime (kill switch)
- Revoke associated egress exceptions or tighten CSP policy
- Remove the app from directories and prevent future invocation
- Notify affected enterprise admins through established support channels

Security and Trust Resources

OpenAI is committed to earning and maintaining the trust of every organization that extends ChatGPT with apps. The protections described in this whitepaper are part of a broader, independently audited security program covering data handling, encryption, compliance, and operational controls across our platform. For additional detail on OpenAI's security, privacy practices, certifications, and compliance documentation, we encourage you to visit the OpenAI Trust Portal, where enterprises can review continuously updated assurance resources aligned to their governance requirements.

Enterprise Enablement Best Practice / Guidance

App enablement in ChatGPT has different stages of controls and governance, and organizations should consider what level of granularity they want to control apps, instead of simply treating it as a single feature toggle. The goal is to make enterprise data and workflows available to users while preserving the existing identity, access, security, and compliance controls that already exist in an organization. Apps inherit upstream permissions, workspace policy, and retention behavior, and so the most effective deployments begin with foundational identity controls and then expand through role-based rollout, app-specific restrictions, and audit validation.

Before enabling apps broadly, organizations should establish their core workspace control principles. This may include enabling SSO, verifying corporate domains, and configuring SCIM or group-based identity management so that app access can be assigned through RBAC if it suits the organization's security posture. Roles and groups seeded through SCIM and SSO integrations allow organizations to assign users to specific categories of controls based on their job functions and data entitlements. These roles can be used to grant access to specific apps by a workspace administrator, allowing for pilot app rollouts and custom apps to be limited to the right set of users.

Starting in Summer 2026, new Enterprise and Edu workspaces are seeded with a set of our most popular, enterprise-ready apps as being enabled. Before a workspace administrator opens access to other users, they should curate the list of apps and ensure the right ones are set to be accessed by the roles.

Organizations may also claim their corporate domains early in the rollout. Domain claiming helps ensure that users cannot connect enterprise systems from personal or unapproved ChatGPT accounts. For apps that support domain restrictions or approved-domain settings, administrators should enable those controls at the time the app is enabled so that account linking is limited to organization-approved identities and domains.

A best practice is to roll out apps in phases using a growing list of RBAC groups. Rather than enabling an app across the full workspace on day one, organizations may create a pilot cohort composed of users with different permission profiles and business roles. That pilot group should test a defined set of prompts, actions, and workflows designed to surface the range of workflows that are possible based on the user's access levels. For example, multiple users with different levels of access in the source system may run the same prompts and confirm that ChatGPT's outputs appropriately reflect their underlying entitlements. This allows the organization to validate ACL enforcement and user experience before broader rollout.

When enabling individual apps, administrators should review access while ensuring enablement empowers users to do more by leveraging the models to do more under their supervision. Admins may review which roles can access the app and also choose the individual actions that are available to the model being used. Actions are categorized into Read and Write actions, and the administrator interface makes it easy to select/deselect all of each category, or individual actions within them. The administrator may also choose to enable new actions that are added in the future.

Parameter constraints are also available to be applied as limited ranges of values that users can pass to specific actions, such as restricting recipients, domains, repositories, or other tool parameters to approved patterns.

Some apps provide a sync capability that provides optimized search and retrieval functionality for a source. When available, Admin-managed Sync provides the ability to preseed the enterprise workspace with enterprise-specific context and allow users to access that data from Day 1, without having to individually log into the source. Administrators can pick and choose the scope of data that's synced, providing more granular management of the scope of data that users can send to the model. An administrator may allow users to individually opt into sync via individual user-authenticated sync.

For organizations that are not enabling sync apps initially, Enterprise Key Management can add an additional layer of cryptographic control for supported app types. EKM is available for Apps and MCP custom apps today.

Finally, organizations should enable auditing and post-deployment oversight as part of setup, and not after rollout has begun. Enterprises should configure the Compliance Logs Platform so security, compliance, and platform teams can review app usage, associated model requests, and relevant metadata. This allows organizations to validate that enabled apps are being used as expected, monitor how internal content influences outputs, and investigate whether usage remains aligned to organizational policy.

In practice, the most mature enterprise rollout pattern is:


Establish identity in Groups and Roles


Create domain controls


Set up access and action controls


Allow for sync if the app allows and where justified


Pilot new apps with RBAC cohorts


Maintain ongoing audit review throughout.

This allows organizations to scale app enablement deliberately while preserving least privilege, accountability, and alignment with existing enterprise governance controls.

Enterprise App Enablement Checklist

01 Establish baseline workspace controls

- [ ] Enable SSO for the workspace.
- [ ] Verify corporate domains.
- [ ] Configure SCIM or group-based identity management.

02 Set up domain governance

- [ ] Ask OpenAI Account Representative to claim corporate domains early in the rollout.
- [ ] Identify which apps support domain restrictions or approved-domain settings.
- [ ] Enable domain restrictions where supported so users can connect only organization-approved identities and domains.

03 Roll out with phased RBAC cohorts

- [ ] Create a pilot cohort before broad app enablement.
- [ ] Include users with different downstream permission profiles and business roles.
- [ ] Define a standard set of prompts and workflows to test.

04 Review and constrain app actions

- [ ] Review each app's available actions before enablement.
- [ ] Choose the appropriate sets of Read and/or Write actions to be available for each app.
- [ ] [OPTIONAL] Enable Sync if faster retrieval is desired and the app supports it.

05 [OPTIONAL] Apply parameter constraints for higher-risk workflows

- [ ] Identify actions that could create data-sharing, routing, or write risks.
- [ ] Apply parameter constraints where appropriate.
- [ ] Confirm constraints align to internal policy and approved usage patterns.

06 [OPTIONAL] Enable Enterprise Key Management for ChatGPT workspace

- [ ] If sync apps are not being enabled initially, evaluate enablement of Enterprise Key Management for ChatGPT workspace.

07 Enable audit and post-deployment oversight

- [ ] Configure the Compliance Logs Platform.
- [ ] Establish a process to validate that enabled apps are being used as expected.
- [ ] Monitor for policy misalignment or unexpected data access patterns.

Appendix

The appendix consolidates official references cited throughout this security whitepaper on OpenAI's trust and security posture for apps. It includes app setup guides and ChatGPT administrative controls, such as RBAC, IP allowlisting, EKM, and related operational policies, to support secure rollout and day-to-day app governance for admins.

Topic	URL
App Overview	https://help.openai.com/en/articles/11487775-connectors-in-chatgpt
Global Admin Console	https://help.openai.com/en/articles/12289294-coming-soon-global-admin-console
MCP Custom Apps Developer Mode	https://platform.openai.com/docs/guides/developer-mode
App Prompt Injection Mitigations	https://developers.openai.com/api/docs/mcp#prompt-injection-related-risks
Admin-Managed SharePoint Sync App Site/ Folder Filtering	https://help.openai.com/en/articles/12143177-sharepoint-connectors-on-chatgpt
Admin-Managed Google Drive Sync App	https://help.openai.com/en/articles/10929079-google-workspace-admin-managed-setup
OpenAI Sub-Processor List	https://openai.com/policies/sub-processor-list/
MCP Dynamic Client Registration (DCR)	https://modelcontextprotocol.io/specification/2025-03-26/basic/authorization#2-4-dynamic-client-registration
Authorization - Model Context Protocol	https://modelcontextprotocol.io/specification/2025-06-18/basic/authorization

Topic	Section	URL
App IP Egress Ranges		https://platform.openai.com/docs/actions/production#ip-egress-ranges
ChatGPT Role-Based Access Controls (RBAC)		https://help.openai.com/en/articles/11750701-rbac
ChatGPT Enterprise Key Management (EKM)		https://help.openai.com/en/articles/20000957-ekm-kms-key-lifecycle
OpenAI / AWS EKM Integration Instructions		https://help.openai.com/en/articles/20000947-openai-aws-ekm-integration-instructions
OpenAI / Azure EKM Integration Instructions		https://help.openai.com/en/articles/20000951-openai-azure-ekm-integration-instructions
OpenAI / Google Cloud Platform (GCP) EKM Integration Instructions		https://help.openai.com/en/articles/20000949-openai-gcp-ekm-integration-instructions
ChatGPT IP Allowlist		https://help.openai.com/en/articles/12111596-ip-allowlisting-for-chatgpt
MCP Overview		http://platform.openai.com/docs/mcp
Compliance API		https://help.openai.com/en/articles/9261474-compliance-api-for-enterprise-customers
OpenAI's Trust Portal		https://trust.openai.com

Revision History

The revision history provides a transparent record of all material updates to this security whitepaper, enabling readers to track changes, enhancements, and clarifications across published versions.

Update June 2026; revisions:

- Removed the “Real-time” qualifier from app terminology, simplifying references to this app type as “Apps” and retiring the naming convention introduced in the prior whitepaper revision.
- Added a Plugins section describing how plugins package app integrations, MCP server configuration, skills, workflow guidance, and related agent capabilities.
- Added new administrative control sections for app write capabilities and app action controls.
- Added new guidance on Apps in Codex, clarifying how Codex app/MCP extensibility and managed configuration relate to ChatGPT app governance.
- Added new MCP security sections covering mutual TLS (mTLS), parameter constraints, and Secure Tunneling for internal MCP servers.
- Added Domain Restriction guidance and updated connected-account governance language, including that GitHub now respects domain-claiming controls, whereas GitHub domain claiming was not respected at the time of the first whitepaper release.
- Update to Plugin Directory to remove outdated “Registry” / “Provisioning” terminology.
- Removed AgentKit from the app usage scope and deleted the prior AgentKit workflow description, reflecting that AgentKit has been sunset.
- Removed the prior Private Networking MCP Servers section and replaced it with more specific controls for mTLS, parameter constraints, and Secure Tunneling.
- Removed the legacy Apps in Agent section, as the product direction has moved away from that framing and toward Workspace Agents.
- Updated compliance language to deprecate the legacy Compliance API section and add the Compliance Logs Platform as the forward-looking compliance export model.
- Added a Skills section describing how skills complement apps and how they should be governed as managed workflow assets.
- Added Security and Trust Resources, Enterprise Enablement Best Practice / Guidance, and an Enterprise App Enablement Checklist to support customer rollout and governance planning.

Update 15 Dec 2025; revisions:

- OpenAI will refer to all external connector integrations as Apps moving forward. Access Connectors, Sync Connectors, MCP Connectors will all be referred to as Apps. Any “connector” text changed to “app”.
- Updated “Access apps” (fka connectors) to” Real-time apps”
- Update to the MCP Custom Apps section to include Interactive Apps, which can be experiences that have interactive UI components.
- Added clarifying language that MCP custom apps are a third party service; MCP custom apps are not developed or verified by OpenAI, and MCP custom apps are third-party services that are subject to their own terms and conditions.
- Updated language on supporting apps through a customers’ BAA with a service provider, not through OpenAI’s BAA
- Added section, Interactive Apps and UI Widgets, providing deeper details about these potentially interactive experiences in ChatGPT
- Updates on App coverage in the Compliance API

Update 19 Nov 2025; revisions:

- Updated diagrams with high quality images
- Slightly updated language on admin-managed connectors; made it clear that admin-managed connectors do not enable the connector for every user by default, rather the admin-managed connector auto maps ACLs and if the connector is enabled for the workspace/rbac group then there is seamless usage for users in enterprise workspace