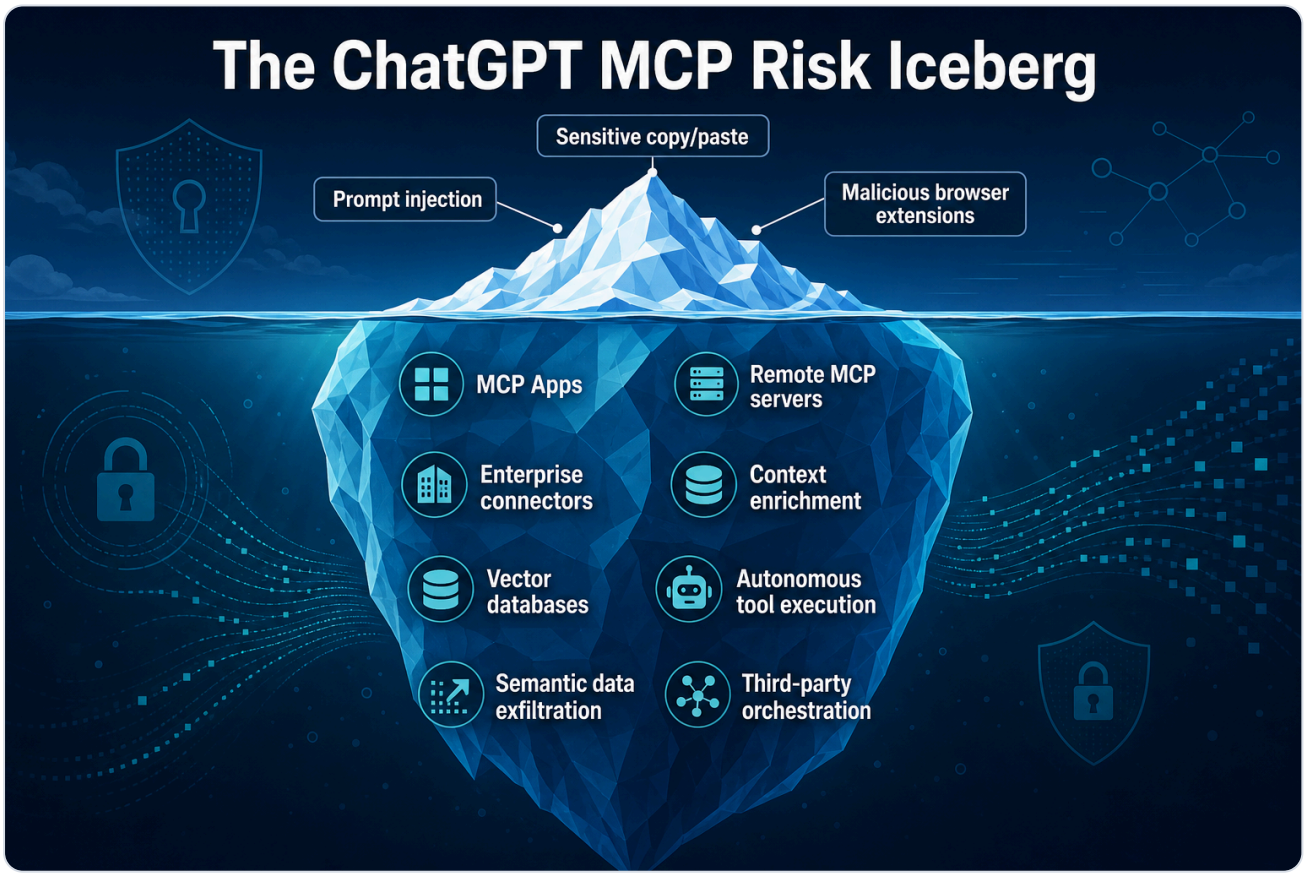
# ChatGPT is becoming part of the enterprise control plane

A lot of security guidance still treats ChatGPT as if it were mainly a website with a clever text box. That view worked when the main risks were copy and paste, document uploads, browser access, prompt injection, and whether staff were allowed to use the service. Those controls still matter, but they do not describe what the platform is turning into.

OpenAI is now building around ChatGPT Apps, MCP servers, remote MCP integrations, connectors, private retrieval sources, vector stores, developer mode, and tool invocation. The product is moving from a single SaaS destination toward an integration layer that can retrieve information, call tools, and interact with other systems. Security teams that keep looking only at the chat interface will miss a large part of the risk.

The familiar risks are easy to talk about because they look like normal web security problems. A malicious browser extension can read pages. A poisoned website can attempt prompt injection. A user can paste sensitive data into the wrong place. Those are worth controlling, but they are not where the more awkward enterprise questions start. The awkward part begins when the AI platform is allowed to reach into other systems on the user's behalf.

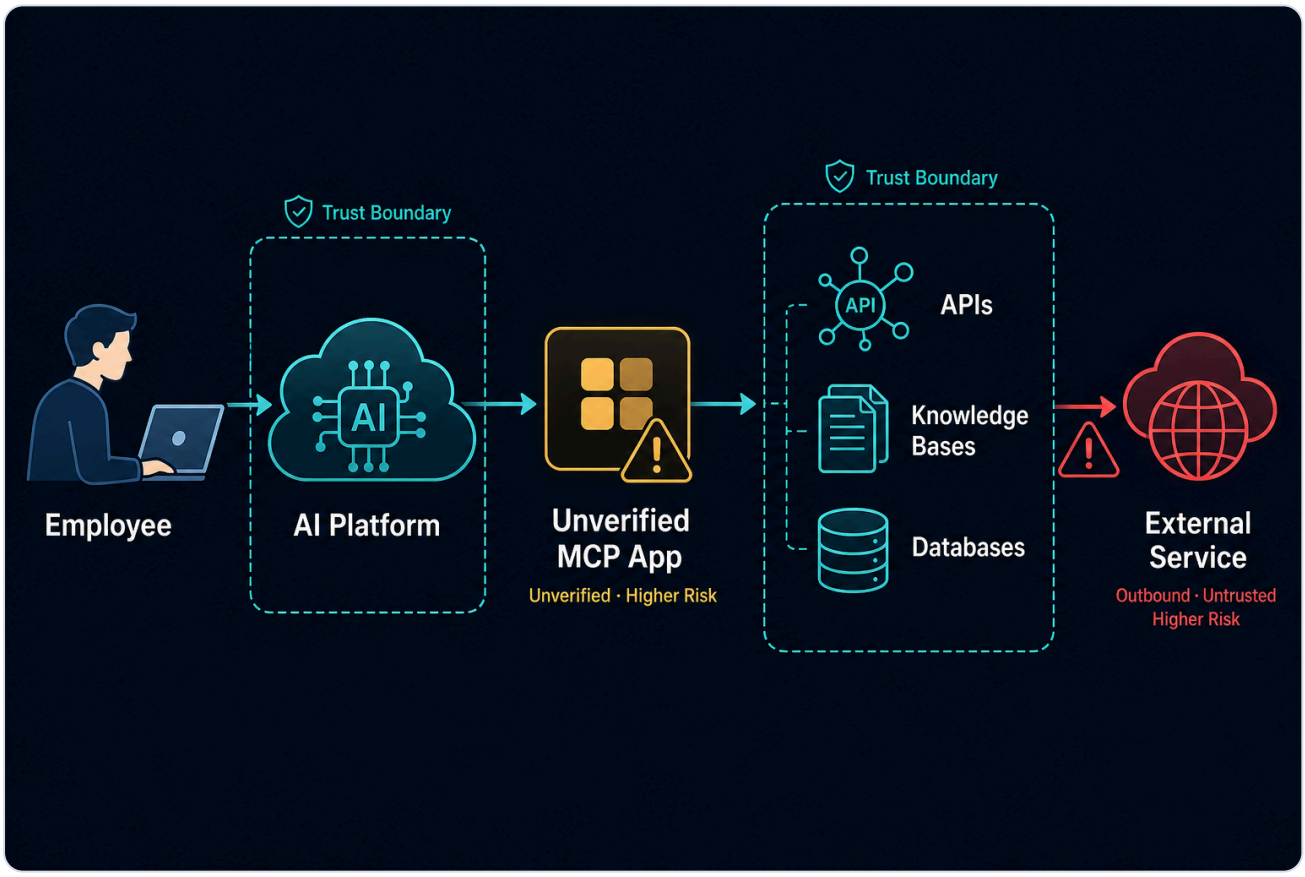



The ChatGPT MCP Risk Iceberg

MCP Apps are a useful example because they sound harmless. To a user, an app is just another helper inside an AI product they already trust. To a security team, it may be a new execution path, a new data path, and a new third party dependency. The browser extension comparison is close enough to be useful, but it also hides the scale difference. A browser extension usually lives beside the browser session. An MCP connected app may run locally, on a remote server, behind a tunnel, or inside a cloud orchestration path.

The blast radius is different. Depending on the permissions, an MCP App may be able to touch retrieved context, internal documentation, tickets, customer records, SaaS APIs, reporting systems, or write capable workflows. If the app is over permissioned, poorly logged, or later compromised, the damage can move well beyond prompt exposure. An external integration may have become part of the organisation's operating path without being treated like software deployment.

This is how shadow IT usually wins. An employee does not think they are expanding the trust boundary. They think they are connecting a tool that helps them finish work. The approval screen looks routine. The tool appears inside a trusted AI interface. Nobody opens a change record, nobody updates the data flow map, and nobody asks what identity the tool will use when it takes action.

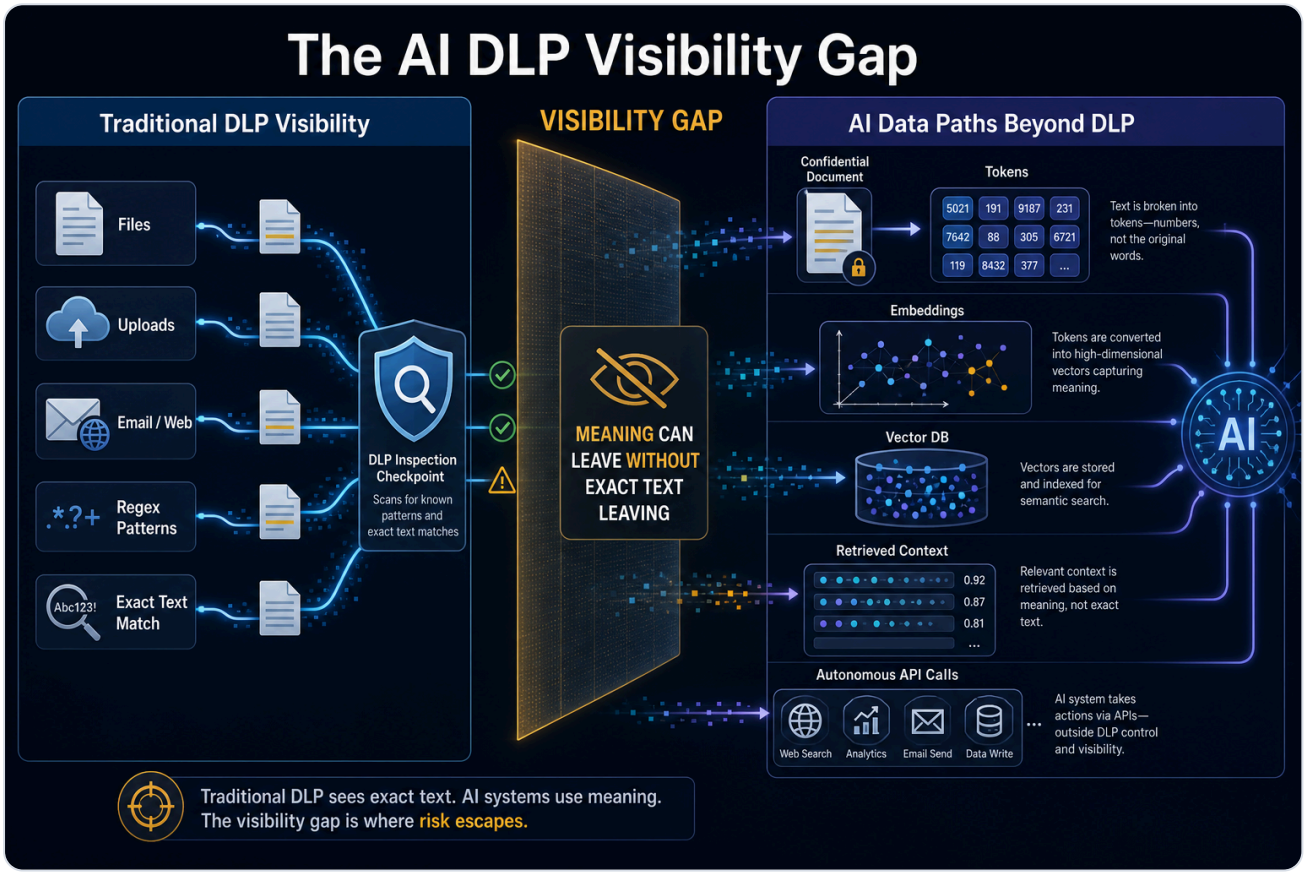



Unverified MCP App flow

The risk does not require a malicious developer. Bad defaults are enough. A connector with broad read access is enough. Weak logging is enough. A trusted third party with a later breach is enough. A retrieval workflow that sends enriched context to an external endpoint is enough. Security incidents often come from boring combinations of normal decisions, not from one obviously evil tool.

DLP is a poor comfort blanket here. It still has a job, especially for direct leakage, but it was built around recognisable movement: files, uploads, identifiers, fingerprints, destinations, and user actions. AI workflows do not always move data in those shapes. A sensitive document may be chunked, tokenised, embedded, summarised, retrieved, mixed with other context, and passed into a tool call. By the time anything leaves, it may no longer resemble the original document.

The meaning can survive the transformation. A customer list converted into embeddings is still derived from a customer list. A confidential design summarised into retrieved context can still disclose the design. A legal strategy split into chunks and passed through an agent workflow can still leak the strategy, even if no single request contains the source document in full.




The AI DLP Visibility Gap

Many existing controls will only see fragments of this. The network may show encrypted API traffic to a sanctioned platform. DLP may miss the original pattern. The SIEM may show a login, but not the tool call. The SaaS admin console may show that a connector exists, but not what context was sent through it or which downstream service received the result. That is not enough for incident response.

A better review starts with the integration layer. Which ChatGPT Apps, connectors, and MCP servers are enabled? Which ones can read data, and which ones can send, update, delete, publish, or trigger workflows? Which identity is used when those actions happen? What requires user approval, and what runs automatically? Where can tool driven traffic go? Can the organisation reconstruct the prompt, retrieval event, tool call, approval decision, destination, and resulting action after an incident?

Those questions are not governance theatre. They separate a controlled AI assistant from an unmanaged automation platform. Mature AI platforms are starting to look like remote management layers with natural language on the front end. They can discover tools, retrieve internal knowledge, act through delegated identities, and execute work across systems. That is useful capability. It is also capability that attackers, insiders, and careless third parties will abuse if the control plane is weak.

Blocking AI outright is not a serious plan for most organisations. Staff will use these tools because they solve real problems. The better answer is to treat AI platforms as integration platforms from the start: inventory the apps and connectors, apply least privilege, restrict egress, log tool invocation, capture retrieval and context enrichment events, preserve approval decisions, and review permissions after the first approval rather than assuming they stay safe forever.

The next wave of AI security failures will include more than people typing secrets into chat boxes. Expect quiet integrations, excessive permissions, invisible retrieval paths, weak approval gates, and semantic data movement that older controls were never designed to see. The model matters, but the orchestration layer around the model is where much of the enterprise risk now sits.

---

## References


- OpenAI Apps SDK
- OpenAI Apps SDK: Build your MCP server
- OpenAI API docs: MCP and Connectors
- OpenAI API docs: Building MCP servers for ChatGPT Apps and API integrations
- OpenAI API docs: ChatGPT Developer Mode
- Model Context Protocol