Outside MCP servers

From the SourceLace manual: Any vendor's MCP server, governed. Many vendors now offer an MCP server for their product, but on its own an MCP connection simply runs whatever tool an AI asks for. Added to SourceLace as a source, it gets the guardrails the rest of your systems have, inside the SourceLa

SourceLace version 0.1.0 · Made November 22, 2026
The latest version is at https://sourcelace.ai/docs

Outside MCP servers

Any vendor's MCP server, governed. Many vendors now offer an MCP server for their product, but on its own an MCP connection simply runs whatever tool an AI asks for. Added to SourceLace as a source, it gets the guardrails the rest of your systems have, inside the SourceLace app and its agents:

- **Per-person sign-in:** each person signs in to the vendor as themselves, so the vendor's own permissions apply.
- **A tool allow-list:** your admin decides which of the server's tools exist at all, and which count as read-only.
- **Held tools:** when the vendor changes a tool your admin allowed, it is held until your admin reviews the new version.
- **Approval before changes:** any tool that may change data runs only after the person sees the exact call and approves it.
- **Every call on the record:** each call is in your organization's audit trail, and your data protection rules apply to what comes back.

Status: Pilot. New or partial, offered to pilot customers. It is switched on per organization; ask your SourceLace contact. It has not been used with a customer's live system yet.

Prefer SourceLace's own connector where there is one. SourceLace's connectors give exact answers that span your systems, previews that show a record's current values, and rules per object and field. An outside MCP server can sit beside one as a second source, for example a vendor's hosted MCP server next to your `salesforce` source, for tools only the vendor offers.

At a glance

	Connects through	Sign-in	Network	Status	First test question
Outside MCP server	The vendor's MCP server address (Streamable HTTP)	Personal sign-in with `oauth`; shared account with a key or token	Allow SourceLace's IP addresses, if the server limits them	Pilot	**Test connection** on **Connections**, then "Which tools does mcp:crm offer?"

Add each source with the usual steps. The network answers are explained in What your network needs.

What to know before you add one

These are the limits of using a server SourceLace did not build. Read them before you allow a tool.

- **We can't fix the vendor's bugs or outages.** When the vendor's server is down, slow or wrong, SourceLace says what it answered, and that is all it can do.
- **"Read-only" labels are the vendor's claim, not a guarantee.** A server can mark a tool as read-only. SourceLace uses that only to suggest a list; your admin decides which tools count as reads. A tool the server marks as destructive is never run as a read.
- **Approvals are per call, not per field.** For a tool that may change data, the person sees the exact tool and arguments before approving, but SourceLace cannot see inside the tool, so it

cannot show the record's current values or check each field the way it does for its own connectors.

- **Answers that span your systems work only for row results.** When a tool answers with records, they become rows you can analyze and combine with other sources (query the tool first, then use its result in a question across sources). Free text stays text.
- **Long lists are read a page at a time, up to a limit.** When a tool sends its answer in pages and lets SourceLace ask for the next one, SourceLace reads the pages one after another, up to 20, and stops at your organization's row and time limits. If a tool has more results than it sends and gives no way to ask for the rest, or the limit is reached, the answer says it is incomplete and why. Add a filter or a shorter date range to get all of it.
- **Quality varies by vendor.** Tool names, descriptions and answers are the vendor's. SourceLace treats all of them as data, never as instructions, but a poorly described tool is harder for any AI to use well.
- **The vendor's agreement governs what their server does.** Requests go to the vendor's server, as the person who signed in. What the vendor does with them is covered by your agreement with that vendor, not by SourceLace's.

How it works

- **Only the tools your admin allows exist.** Every other tool on the server is invisible in SourceLace and refused if anyone names it. A tool the server stops offering simply disappears.
- **Reads** are the tools your admin marks as read-only. People and agents find them with search, see each one's arguments, and run one with a query in the `mcp_json` language: `{"tool": "search_accounts", "arguments": {"name": "Corvanta"}}`. SourceLace checks the arguments against the tool's own description (required ones there, no unknown ones, the right types) before anything is sent.
- **Changes** are every other tool your admin allows. They never run as a query: only as a change, which shows the person exactly which tool will be called with which arguments, and runs once after they approve. Agents follow your agent rules for changes, so record changes wait for a person unless an admin allowed otherwise.
- **A changed tool is held until an admin reviews it.** See Held tools.
- **Every call is audited** in your organization's audit trail, like every other source. Data protection rules apply to what comes back.
- **Answers are data.** What the server sends back only ever becomes rows or text. Nothing in it is followed as an instruction.

Filters for a tool

A tool that lists records can be narrowed at the vendor, so a question across your systems asks for fewer of them. On **Connections**, choose **Edit** on the source and open **Filters the source can apply**. The steps, the meanings, the test and the JSON option are the same as for REST and SOAP sources; the "operation" is the tool, and the "parameter" is one of its arguments.

Example. A vendor's tool `list_tickets` takes an argument `status` and no other required argument. Sign in to the source under **My connections**, add one filter (tool `list_tickets`, field `status`, filtered by `status`, **is equal to**), save, and press **Test**. The test reads the tool's own description to learn the

argument's type, then checks the filter as you. A question such as "Show open tickets next to the customers in Salesforce" then calls `list_tickets` with `status` set to `open`, instead of reading every ticket.

Limits. Only a tool allowed as a read and needing no other argument can be read as a table. A tool whose only filter is free text in the vendor's own search language (such as `status:open AND owner:ana`) cannot take a filter. A tool the vendor has changed is held until an admin reviews it, and is not read meanwhile.

Held tools

Your admin allows a tool as it is when they allow it: its description, which tells the AI what the tool does, and its arguments. A vendor can change either at any time. SourceLace notices, and holds the changed tool until an admin has looked at it.

- **What is held:** an allowed tool whose description or arguments changed. A harmless change, such as the same arguments listed in another order, does not hold it.
- **While it is held:** the AI is not offered the tool, and a call to it is refused with a message saying it is waiting for an admin's review. Every other tool keeps working.
- **Who is told:** your organization's admins get one email for the source, naming the tools held. On **Connections**, the source's status shows **Tools waiting for review**.
- **The review (admins):** on **Connections**, choose **Review tools** on the source. Each tool shows the allowed version and the new version side by side, description and arguments, with what changed marked. Choose **Allow the new version** to use the tool again as it is now, or **Keep blocked** to keep it held without being asked again. A kept blocked tool can still be allowed later from the same page.
- **Changed again:** a tool that changes again is held again, for a new review. A tool the vendor changes back to the version you allowed is used again.
- **On the record:** holding a tool, and each review, are in your organization's audit trail, with the source and tool names only.
- **New tools** the server starts offering stay invisible until your admin allows them, as always. Taking a tool off the allow-list and allowing it again starts afresh.

How people sign in

Choice (`auth`)	**Who the vendor's server sees**	**When to use it**
`oauth`	**Each person, with their own account**	Recommended, and what most MCP servers offer. The vendor's own permissions then apply to each person.
`api_key`, `bearer`	**One shared account** for everyone who uses the source	Only when the server offers nothing per person.
`none`	Nobody	Public servers only.

With `oauth`, SourceLace finds the server's sign-in service by itself when you save the source, following the MCP standard. If the sign-in service allows it, SourceLace also registers itself there, so there is no client to create. If it does not, have the vendor's admin register an OAuth client for SourceLace with exactly this redirect URL, `https://mcp.sourcelace.ai/connect/callback`, and enter its

client ID.

Shared access: with `api_key`, `bearer` or `none`, everyone who may use the source acts as the same account, so the vendor cannot tell people apart. SourceLace says so wherever the source is listed. Limit who can use it with access by group.

Secrets (a client secret, key or token) are stored encrypted with your organization's key, shown only as "(saved)", sent only to the server itself, and never written to logs or answers.

Add an outside MCP server (SourceLace admin)

1. Ask your SourceLace contact to switch on **Outside MCP servers** for your organization, if it is not on.
2. On **Connections → Add a source**, choose **Outside MCP server** (`mcp`), with a name such as `mcp:crm` and a label.
3. Enter the server's address from the vendor's documentation, such as `https://mcp.corvanta.com/mcp`.
4. Choose how people sign in. For `oauth`, leave the client ID empty to let SourceLace register itself.
5. Save. Then click **Connect** for yourself on **My connections** (for `oauth`), and press **Test connection** on **Connections**. It lists every tool the server offers, which ones you allowed, and the ones the server marks as read-only.
6. Choose the read-only tools. When you have none yet, Test connection offers **Use these read-only tools**, which fills in the ones the server marks as read-only. Check the list, remove anything you are unsure of, and save.
7. To allow tools that may change data, turn on **Allow changes** and name them one by one under **Objects that can be changed**. Each call is previewed and approved by the person.

Option	Example	What it is
`url`	`https://mcp.corvanta.com/mcp`	The server's address. Must start with `https://`. The Streamable HTTP address, not the older one ending in `/sse`.
`read_tools`	`search_accounts, list_notes`	The tools people may run as reads. Nothing else runs as a read.
`auth`	`oauth`	`oauth`, `api_key`, `bearer` or `none`.
`client_id`, `client_secret`		Only if you registered a client by hand. Empty: SourceLace registers itself.
`authorize_url`, `token_url`, `scopes`		Found from the server when you save. Fill them in only if the server does not publish them.
`secret`		The key or token, for shared access.
`key_name`, `key_in`	`X-API-Key`, `header`	For `api_key`: the header or query parameter that carries it.

A tool is either a read or a change, never both. When you change the server's address, SourceLace registers itself again with the new server's sign-in service.

What the vendor's admin does

Often nothing more than switching the connection on. About 10 minutes.

Who: an admin of the vendor's product, with the right to allow integrations or AI apps (the role's name differs by vendor, such as Workspace Admin or Owner).

1. **Find the server's address.** In the vendor's documentation, look for its MCP server for remote clients. Use the **Streamable HTTP** address, usually ending in `/mcp`. Some vendors still list an older address ending in `/sse`: SourceLace does not use that one.
2. **Allow the connection.** Many vendors turn MCP access or AI apps off until an admin allows them, in their admin or security settings. Allow it for the people who will use the source.
3. **Sign-in, usually nothing to do.** SourceLace registers itself with the vendor's sign-in service when you save the source. If the vendor does not allow that, register an OAuth client for SourceLace in the vendor's developer or admin settings:
   - redirect URL exactly `https://mcp.sourcelace.ai/connect/callback`;
   - grant type **Authorization code** with PKCE;
   - the scopes the vendor's documentation names for its MCP server.

   Copy the **client ID** (and the **client secret** for a confidential client) into the source.
4. **For a shared key instead:** create a key or token with the least access needed, and note the header or query parameter the vendor names for it.
5. **Let SourceLace in.** If the vendor limits access by IP address, allow SourceLace's (ask support for the list).

Check that it worked: click **Connect** for yourself on **My connections** (with `oauth`), then press **Test connection**: it lists the server's tools.

If it did not work: look up the message in When something goes wrong. The standard is described at Model Context Protocol.

Safety

An outside MCP server is an address your admin types in, so every request gets the same guard as REST and SOAP sources: only `https://`, only addresses on the public internet in SourceLace's cloud (cloud metadata addresses always refused), redirects never followed, answers limited to 5 MB, each call limited to your organization's query time, and at most `queries.api_requests_per_minute` requests a minute per source. Nothing of SourceLace's own (cookies, tokens, keys) is sent: only the source's own sign-in.

When something goes wrong

What you see	What to do
"Outside MCP servers as sources are switched off for your organization. ..."	Ask your SourceLace contact to switch it on.
"Set the option url: the outside MCP server's address, ..."	Enter the server's address from the vendor's documentation.

What you see	What to do
"SourceLace could not find ...'s sign-in service. ..."	The server does not publish its sign-in details. Enter `authorize_url`, `token_url` and a client ID from the vendor's documentation, or use a shared key if the vendor offers one.
"...'s sign-in service does not let SourceLace register itself. ..."	Have the vendor's admin register an OAuth client with the redirect URL above, and enter its client ID.
"...'s sign-in service does not offer PKCE (S256), which SourceLace requires."	This server's sign-in is not safe enough for SourceLace. Ask the vendor, or use a shared key.
"... is listed both as read-only and as a change. ..."	Keep the tool in one list only.
"... changed the tool ... (its description or arguments), so SourceLace holds it until an admin reviews the new version. ..."	The vendor changed the tool. An admin reviews it under **Connections → Review tools**; until then, use another tool.
"... is not waiting for review with this version any more. ..."	The vendor changed the tool again while you reviewed it. Reload the list and review the newest version.
"... has no tool ... you may use. Use search_schema."	The tool is not allowed, or the server no longer offers it. An admin can allow it.
"... may change data, so query cannot run it. ..."	Use `propose_change`; the person approves the exact call.
"The server now marks ... as destructive, so SourceLace will not run it as a read. ..."	An admin should review the source's read-only tools with **Test connection**.
"... needs .... Use describe_object to see its arguments." or "... has no argument ...."	Fix the query's arguments.
"...'s tool ... failed. It said: ..."	The vendor's own reason follows. Try again, or ask the vendor.
"... did not answer in a form SourceLace understands (MCP over Streamable HTTP). ..."	Check the address: it must be the server's MCP address, not its website.
"That looks like the older SSE address. SourceLace uses Streamable HTTP: ..."	Use the vendor's Streamable HTTP address, often the same address ending in `/mcp` instead of `/sse`.
"... does not take MCP requests at its address (HTTP 405). ..."	The address is not the server's Streamable HTTP address. Use that one, often ending in `/mcp`.
"... no longer accepts your sign-in."	Connect the source again on **My connections**.
"...: ... leads to a private or reserved network address, so SourceLace did not call it."	SourceLace's cloud only calls addresses on the internet. Use the server's public address.
"... refused the source's saved key (HTTP 401). Ask an admin to check it."	The saved key or token is wrong, expired or revoked.
"... does not allow this for your account (HTTP 403)."	The server refused your account, or the key lacks the permission.
"... found no MCP server at its address (HTTP 404). An admin should check it."	The address is wrong.
"... answered with a redirect (HTTP ...), which SourceLace never follows."	The server moved.
"... is limiting requests (HTTP 429). Try again later."	The server is rate limiting.
"... has more results than it sent in this answer, and SourceLace cannot ask it for the next page. The answer is incomplete."	The tool sends only its first page and has no way to ask for the next.
"SourceLace read the first 20 pages of ... and stopped there. The answer is incomplete."	The list is longer than SourceLace reads in one question.

What you see	What to do
"... stopped answering after page N of ..., so the answer is incomplete."	The server refused or failed on a later page (for example a rate limit). The pages already read are kept.
"... took longer than the query time limit." or "... failed (HTTP ...). Try again later; if it keeps failing, tell your admin."	The server was slow or had a problem.
"... answered with an error: ...."	The server refused the tool call and said why.
"Outside MCP server sources have no records by id. Query the tool that reads one, with its arguments."	A record-by-id call was made; MCP servers only offer tools.