WHITEPAPER

# Qonnect AI

*for Qlik Sense*

*and Qlik Cloud*



# Introduction

Business questions rarely wait for a dashboard. In most Qlik environments an answer still travels the same route: open the app, click through filters, find or build the right chart, and pass the result on. AI assistants have become part of daily work, but they cannot see the data that matters most, because that data lives in Qlik, behind your governance.

Qonnect AI closes that gap. It connects the AI tools your people already use to Qlik Sense client-managed and to Qlik Cloud through the Model Context Protocol (MCP), an open standard for connecting AI clients to other systems. People ask in plain language; the assistant works in the apps they may already open, as them, and answers with real figures. Developers can go further: build sheets, charts and master items, and work on load scripts in their own apps.

This whitepaper describes what Qonnect AI is, how it acts in Qlik on each person's behalf, what stays under your control, what reaches the AI provider you choose, and how it is deployed and governed. It is written for the decision-maker and for the architect who will evaluate it. It describes release 2.0.

# What Qonnect AI is

Qonnect AI is software you install and run on a server of your own, inside your network or in your own cloud tenant. It is not a hosted service. It has three parts:

- The MCP server, which AI clients connect to. It offers the tools the assistant uses to find apps, compute answers, make selections, build content and work on load scripts.
- The Qonnect AI Console, the sign-in service and the administration console. People sign in here; administrators decide who may do what.
- A front door, which terminates HTTPS for one host name and is the only part that listens on the network.

One server can serve several Qlik targets: Qlik Sense Enterprise on Windows (client-managed) sites, Qlik Cloud tenants, or both. The Qlik side can be Qlik Cloud alone, with no client-managed site; Qonnect AI itself is still installed on your own server.

It works with AI clients that support remote MCP servers with sign-in, for example Claude (claude.ai, Claude Desktop and Claude Code), ChatGPT, Microsoft Copilot Studio, VS Code and Cursor, and with the Qloud Cover App+ browser extension inside Qlik.

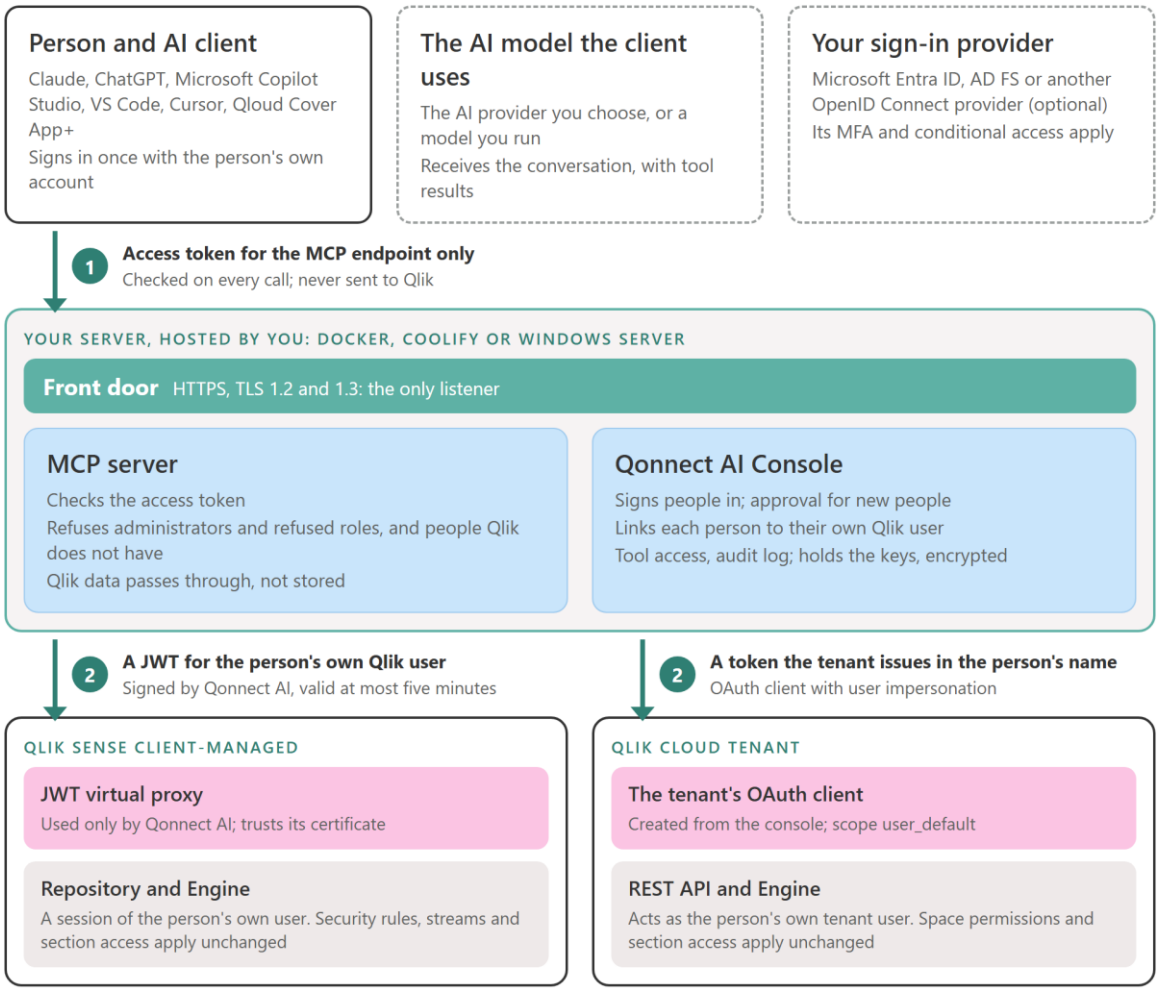



*Figure 1: Qonnect AI runs on your own server. The AI client's access token (1) never reaches Qlik. Qlik sees only the second token (2): on a client-managed site a short-lived JWT for the person's own Qlik user; on Qlik Cloud a token the tenant issues in the person's name.*

# Every call acts as the signed-in person

This is the design decision everything else rests on. There is no shared service user. Every request runs as the person asking, with the rights that person already has in Qlik. Two separate tokens make that work, and Qlik only ever sees the second.

## Token one: who is this person?

People sign in to the Qonnect AI Console, which is an OpenID Connect provider. It signs people in itself with local accounts, or hands off to your sign-in provider: Microsoft Entra ID, AD FS or another OpenID Connect provider, so your multi-factor and conditional access rules apply. The AI client receives an access token for the MCP endpoint only. The MCP server checks its signature, issuer, audience and expiry on every call. A call without a valid sign-in and a linked Qlik user is refused; there is no default user to fall back on.

# Token two: act as them in Qlik

The console links each person to one Qlik user. How the second token is made depends on the target:

- Qlik Sense client-managed. For each call the MCP server signs a short-lived JWT for the person's own Qlik user, valid for at most five minutes, with a private key the console holds. Your Qlik site trusts the matching certificate on a JWT virtual proxy used only by Qonnect AI. Qlik never talks to the console or to your sign-in provider; it checks the signature. Before every new token the server reads the person's Qlik user record with a read-only *directory-lookup identity*, and refuses a person the site does not have, so the virtual proxy never creates an empty Qlik user for a mistyped identity.
- Qlik Cloud. The console matches each person to one tenant user: by default the email they sign in with, with the prefix AZURE\, or a subject linked to them by hand. For each call the MCP server asks the tenant's token endpoint for an access token in that person's name, using the tenant's OAuth client with user impersonation and the scope user_default only. There is no virtual proxy and no signing key. A person with no tenant user is refused, because the tenant issues no token for a user it does not have.

Because every call is the person's own Qlik session, Qlik's own security rules, stream and space permissions and section access decide what each person can see and do, exactly as when they use Qlik themselves. Qonnect AI adds restrictions on top; it never widens what Qlik allows.

# Sign-in and approval

Who may sign in is decided by your sign-in provider and, where you require it, by an administrator. Each sign-in provider has Approval required, on for every provider you add: a person who signs in with it for the first time gets nothing until an administrator approves them, no AI client sign-in, no App+ sign-in and no Qlik tools. Administrators approve or refuse people one at a time or in a selection, can withdraw approval at once, and every decision is in the audit log. Local accounts never wait: an administrator made them.

Sign-in does not ask Qlik, so a Qlik site or tenant that is down does not stop anyone signing in. Whether a person may act in Qlik is decided on every call and per target. Groups can be kept by hand in the console or taken from your sign-in provider's groups claim at each sign-in.

# Who is refused, and why

The AI acts with the full rights of the person using it. For an administrator that would include changing anything on the site, so Qonnect AI refuses some people by design:

- **On Qlik Sense client-managed**, a person who holds the **RootAdmin** role, and always Qlik's own INTERNAL service accounts.
- **On Qlik Cloud**, a person who holds **TenantAdmin**, directly or through a group.
- **Further roles** that an administrator names per target under *Refused roles*, for example ContentAdmin. The list is empty by default.

The roles are read before anything is done as the person; if they cannot be read, the call is refused. An administrator can allow a named person per target, and adding one is recorded in the audit log as a security downgrade. A role change in Qlik is seen within five minutes. The recommended practice is a separate, non-administrator account for everyday AI use.

# What people can do

People describe what they want; the assistant picks the tools and acts as them.

- **Discover and ask.** Find the apps and fields a person may use, compute numbers from the data model, make selections, look at a chart as a picture, and try what-if questions with nothing saved.
- **Explain.** Why a number moved, how a data model fits together, and what a master measure means, using the organisation's own definitions where an administrator has written them.
- **Build.** Sheets, charts and master items, in apps the person may change.
- **Script.** Read, check and edit load scripts, with every earlier version kept so it can be compared and restored, then reload the person's own app.
- **Administer.** Reload tasks, publishing, streams, security rules, users and licences on a client-managed site. These tools are off until an administrator grants them one by one.

# Tool bundles and the default tool list

Tools come in bundles, and administrators allow or deny a whole bundle, for everyone, a group or a person, under **People & access → Tool access**. The console shows what each bundle costs an AI client, because a client pays for the whole tool list on every turn.

| Bundle | On a new install |
|---|---|
| Core analytics: find an app, read its model and charts, select values, compute an answer | On |
| Analysis extras: chart pictures, what-if, explain a change, export data | On |
| Master items: master measures, dimensions, visualizations and variables | On |
| Sheet authoring and styling | Off; the console asks once whether people may build sheets |
| Data layer (load scripts and reloads of the person's own apps), app insight, reporting | Off until allowed |
| Administrator tools, and tools marked Preview | Off until granted one by one |

The licence decides the ceiling: **Analytics** for people who ask questions, **Developer** adds building and load-script work, and **Full** adds the administrator tools. Each tier includes the one before it.

## Writes only where the person may edit

Reading works across every app a person may open. Changes are deliberately narrower:

- **Qlik Sense client-managed:** only the person's own unpublished apps. In a published app the AI may only do what Qlik lets any reader do: create and change the person's own private sheets and bookmarks.
- **Qlik Cloud:** unpublished apps in the person's personal space and, where the target allows it, in shared spaces where Qlik says the person can edit, asked with the person's own sign-in at every change and written to the audit log with the space.
- **Never** published apps or apps in managed spaces.

A tool asked to change something outside these limits refuses and changes nothing. A **read-only switch** in the console refuses every tool that changes anything, for the whole deployment or one Qlik target, within about 15 seconds and without a restart.

# What stays under your control

- **The server.** It runs on your infrastructure, with one inbound HTTPS port. Everything else listens on loopback or a private container network.
- **People and access.** Who may sign in, who is approved, which tool bundles each group gets, and immediate offboarding: disabling a person refuses their next call, not when their token expires.
- **Secrets.** The Qlik signing keys, the Qlik Cloud OAuth client secret, sign-in provider secrets and the organisation's AI key are encrypted (AES-256-GCM) under a data key

that a new install keeps outside the data. Local console passwords are stored as argon2id hashes.

- **What is stored.** Qlik data passes through and is not stored, with named exceptions such as load-script versions and briefing results. Prompts and answers are not stored, in the database or in the logs. There is no telemetry and no update check.
- **The console's administrators.** Local administrators can add two-step sign-in with an authenticator app, and the console can require it of all of them.

## Audit

- **Administration changes** in the console are recorded: who changed what and when, for accounts, groups, sign-in providers, tool access, Qlik targets, keys and settings.
- **Administration done through an AI client**, such as publishing an app, changing a reload task or allocating a licence, is recorded with the person, the Qlik identity it acted as, the target, the object and the outcome. Destructive changes run only with a confirmation the server issued for that exact change. If the console cannot be reached, the record waits in a spool on disk and is sent when it can be, marked as late.
- **Every tool call can be recorded.** With *Record every tool call* on, the console keeps one record per call: who, as whom, the target, the tool, when, the outcome, the duration, the AI client and a summary of the arguments with secrets left out. Never the result. It is off by default; records are kept 90 days by default and can be exported as CSV.

Qlik's own logs also show every action under the person's own Qlik user.

# What reaches the AI provider you choose

An honest answer matters more here than a reassuring one. A tool's result is returned to the AI client that called it, and the client sends it to the model it uses. So whatever a tool returns reaches that model, and that can include row-level Qlik data the person may see, personal data included. Each result is limited to what that person may see in Qlik. Nothing is sent to an AI provider unless a person, or a briefing they set up, asks.

Which provider receives it depends on the setup you choose:

| Setup | Who calls Qonnect AI | Where tool results go |
|---|---|---|
| Cloud chat app: Claude, ChatGPT, Microsoft Copilot Studio | The AI vendor's cloud, which must reach the server from the internet | To that vendor, under its terms and your agreement with it |
| Local client with a cloud model: Claude Code, VS Code, Cursor, App+ | The person's computer or browser, inside your network | To the model provider the client is set up with |
| A model you run: App+ or another client pointed at an OpenAI-compatible model on your network | The person's computer or browser, inside your network | To the model you run, as long as it is the only provider the client is set up with |

The third setup has been run with a real self-hosted model through the console's AI proxy only, from a test client that sends what App+ sends. App+ itself in a browser has not been run with one yet, so prove it on your own site first.

Qonnect AI has no mask for personal data in tool results in 2.0. Qlik section access decides, per person, which rows and fields a Qlik session can see, and every call is that person's own session, so what section access withholds never reaches a tool. Keep personal data out of the apps AI users work in where you can, and cover the AI provider in your data processing agreement.

# Deployment and operations

## Where it runs

| Path | For | Front door |
|---|---|---|
| Linux with Docker | A 64-bit Linux server with Docker Engine and Compose; an offline bundle carries the images for a server with no internet access | Caddy, included |
| Coolify | Organisations that already run Coolify | Coolify's own proxy |
| Windows Server | Windows Server 2019, 2022 or 2025, a server of its own; one signed setup.exe, no Docker, no internet access needed to install | Caddy, installed as a Windows service |

A small server is enough: 2 vCPU, 4 GB RAM (8 GB recommended) and 40 GB disk on Windows Server. One console and one MCP server per deployment is the supported maximum.

## What the Qlik side needs

- Qlik Sense client-managed, November 2024 or later: a RootAdmin imports a setup page in the QMC and applies it from the Hub, which creates the JWT virtual proxy (the Qlik proxy restarts once), and a read-only directory-lookup identity. Each person needs the Professional or Analyzer access they would need in the Hub. Tested on November 2024, May 2025 and November 2025. Building sheets needs Qlik's Dashboard bundle, an optional part of the Qlik Sense install: the cards Qonnect AI builds are its layout containers. Qonnect AI checks for it before it writes one and says so when it is missing.
- Qlik Cloud: a tenant administrator gives the console a short-lived API key once, and the console creates the tenant's OAuth client, stores its secret and deletes the key; or the client is created by hand. Nothing restarts on the tenant. The console can also remove what it created.

## Upgrades and the way back

A new release comes as a new bundle. On Docker, setup.sh --upgrade backs up, pulls and starts the new release and prints the way back. On Windows Server, the new setup.exe upgrades in place, takes a backup first and rolls back on its own if the new version does not come up; Undo-QonnectAIUpdate goes back to the previous version. Backups know the version they were taken from, and an older console refuses to start on data a newer one migrated, before it changes anything. Going back restores the data from before the upgrade.

## Running it day to day

Administrators work in the Qonnect AI Console: people and approvals, tool access, Qlik targets, AI connections and the audit log. Home lists what needs an administrator, such as people waiting for approval or a target without a directory-lookup identity. The online guide at docs.stretchqonnect.com covers installation, sign-in, the tools and upgrades. Support: servicedesk@stretch.dk.

# Inside Qlik with Qloud Cover App+

Qloud Cover App+ is a browser extension that brings the assistant onto the Qlik page: a chat next to the sheet that knows which app and sheet the person is on, page actions such as making a selection or opening a sheet, the Sheet Builder (draw a layout and the AI builds it, on client-managed Qlik Sense and on Qlik Cloud), and load-script help. With Fleet mode, IT rolls App+ out by browser policy and sets its AI models, shared templates and budgets centrally from the same console. When App+ uses the organisation's AI key, its calls go through the console, the key never reaches the browser, and the console meters each call without storing the prompt or the answer.

# Known limitations

Stated plainly, so your evaluation does not have to find them:

- **No external penetration test yet.** The product has had internal security reviews only. An external test is planned.
- **A model you host** has been run through the console with a test client that sends what App+ sends; App+ itself in a browser has not been run with one yet. One model call may take five minutes at most.
- **AD FS** sign-in and groups have been run against a live AD FS with one domain; accounts of a second domain have not. Groups from Microsoft Entra ID have not been run against a live tenant.
- **Two directory accounts with the same e-mail address** (a person's normal and elevated account) become one console account, whose Qlik user follows the account that signed in last. Set the provider's Email claim to a value each account has alone, such as upn for AD FS, before anyone signs in.
- **Qlik Sense November 2026 has not been tested yet.**
- **Only single-node Qlik sites have been tested.** For more than one node, plan the install with us and run a pilot first.
- **A server only people inside your network can reach**, used with App+ alone, has not been run end to end yet, and neither has a cloud chat client reaching a server that admits only listed addresses.
- **Building sheets needs Qlik's Dashboard bundle** on a client-managed site. On a site installed without it, the tools that write its objects refuse with a sentence saying so.
- **Reloading your own app through the assistant** needs a Qlik security rule that lets the person create a reload task. Without it the reload is refused; a test reload still works.
- **Approval is per person:** there is no pre-approval by group or e-mail domain yet. Restrict the app at your sign-in provider to admit only the people you choose.
- **Two-step sign-in covers the console's local administrator accounts only.** People who sign in through your provider get whatever multi-factor sign-in it requires; the console does not yet check that a second factor was used.
- **One server, no high availability.** One console and one MCP server per deployment is the supported maximum.
- **On Qlik Cloud, every tool asks the tenant for the caller first,** also the tools that answer from the console alone (knowledge search, the guides and the business context). While the tenant cannot be reached or is limiting requests, those tools are refused too.
- **On a Qlik Cloud tenant that hides subjects** from the OAuth client, *Check everyone* marks people the tenant has only by e-mail as *not confirmed*. Calls are not affected.
- **The identity-mismatch records App+ sends** are not kept in the audit spool while the console is down.
- **Installs made before the data key moved out of the data keep it there,** and backups taken before the encryption still hold those secrets in plaintext: encrypt or remove them.

# Summary

Qonnect AI puts an AI assistant in front of the Qlik you already run, client-managed, Qlik Cloud or both, without a shared service account and without a second permission model. Every call acts as the signed-in person under Qlik's own rules; IT decides who may sign in, which tools each group gets and where the AI may write; and the software runs on your own server. What reaches the AI provider is what you choose, under your agreement with that provider.

To see it on your own data, or to plan a proof of concept, contact Stretch Qonnect at servicedesk@stretch.dk or www.stretchqonnect.com.

# From Data to Action