Configuration Guide

How to set up and operate your Mili agent — accounts, billing, plugins, agent-to-agent links, security, and maintenance.

The interface may still display the name “FrançoisGPT.” It refers to the same product.

01 Changing the administrator password

Mili has one built-in administrator account. It is your “break-glass” credential: it always works, even when single sign-on is unavailable, and every login is recorded in the audit log.

Setting your own password

A fresh install ships with a **default password**: `password`. Log in with it once, then change it right away.

1. Open the server address shown by the installer, e.g. https://YOUR-HOST:5001, and sign in with the default password.
2. Go to **`Settings` › `Authentication`**
3. In the **Administrator Account** section, next to *Password*, tap **Change**.
4. Enter the current password, then the new password (min. 12 characters) twice, and confirm.

CHANGE IT BEFORE GOING FURTHER

The default password is public knowledge. Set your own password the first time you sign in, before exposing the server to anyone else.

The change is written to the audit log immediately. If single sign-on (OIDC) is enabled, the administrator password still works independently as the recovery path.

02 Your subscription

Mili is sold as a flat monthly subscription, not metered pay-as-you-go.

Price

$600 / month

What it covers

Access to the software and ongoing support.

Included credit

$600 of managed LLM usage credit, included each period.

Roll-over

None. Unused credit is **lost at the end of each period** — it does not carry forward.

In plain terms: the monthly fee buys access to the platform and the support behind it, and includes $600 of model-usage credit to spend during the period. Whatever credit you don’t use is forfeited when the cycle resets — there is no accumulation.

TIP

For billing changes, plan questions, or to adjust your included credit, contact Québec Agentique support.

03 Plugins — adding & connecting

Plugins give your agent new abilities by connecting it to outside services — email, calendars, messaging, code hosts, and many more. A plugin needs to *authenticate* to the service it talks to, and there are two ways it can do that.

Method A — OAuth (sign in with the provider)

For services that support it (Google, Microsoft, and similar), you connect by signing in through the provider's own consent screen. You approve the access once, and the platform holds and automatically refreshes the resulting tokens for you — you never paste a secret by hand.

1. Enable the plugin you want to use.
2. Choose **Connect / Sign in**. You are sent to the provider's authorization page.
3. Review the requested permissions and approve. You are returned to Mili, and the plugin is now connected.

Method B — Credential vault (API key or token)

For services that issue an API key, token, or password, you store that secret in the **credential vault** (see section 6) and the plugin reads it from there. The secret stays encrypted at rest and is referenced — never duplicated — by the plugin.

1. Create the secret in the vault, giving it a key such as `github_token` or `smtp_password`.
2. In the plugin's settings, point the credential field at that vault key (written as `github_token`).
3. Enter any non-secret settings the plugin needs — host, port, username — directly in the plugin's normal fields.

KEEP SECRETS IN THE VAULT

Only true secrets (passwords, API keys, tokens) belong in the vault. Non-secret settings such as hostnames, ports, and usernames go in the plugin's ordinary configuration fields.

04 MCP clients — calling other agents

Mili can talk to other Mili agents (or any MCP-compatible agent) so they can use each other's tools. Each peer you want to reach out to is saved as a **Contact**. This is the "outbound" side: *your* agent calling *theirs*.

Open `Contacts` from the main screen, then add a contact and fill in:

Display Name

A friendly label, e.g. "Alice's Agent."

Contact ID

The peer's agent identifier.

Endpoint URL

The peer's MCP address, e.g. `https://agent.local:5001/mcp`

Transport

HTTP SSE (over the network) or STDIO (local process).

Verify SSL

Leave on for real certificates; turn off only for self-signed peers.

Outbound Token

The secret **you send** with every call to this peer (use a long random string, 32+ characters).

Timeouts

How long to wait for outbound (and inbound) calls.

Once saved, your agent can invoke that peer's permitted tools as part of its work.

05 MCP server — exposing your tools

Your agent is also an MCP **server**: other agents can call *your* tools, but only the ones you explicitly allow, and only contacts you've authorised. This is the "inbound" side, and it is configured per contact.

Open a contact in `Contacts` and set its **Inbound** options:

Allowed Tools

The whitelist of *your* tools this contact may invoke. **Empty means nothing is allowed.**

Inbound Token

The secret this contact **must send you** on every incoming call.

IP Validation

Off (any IP), IP (must match the endpoint's address), or DNS (resolve the endpoint hostname).

Inbound Guard Rails

Instructions added to the system prompt for every request from this contact, e.g. "Only answer questions about X. Never reveal credentials."

Inbound Mode

Single Message — each call gets a fresh session. **Conversation** — all calls from this contact share one ongoing session with history.

ALLOW ONLY WHAT YOU MEAN TO SHARE

The Allowed Tools list is the gate. Add tools deliberately; an empty list blocks the contact entirely, which is the safe default.

06 The credential vault

The vault is an encrypted store for the secrets your plugins and tools need — API keys, passwords, tokens. Secrets are encrypted at rest with AES-256-GCM and are decrypted only in memory when used.

Open `Settings › Credential Vault` (you'll be asked for your administrator password to enter).

Adding a credential

1. Tap **+ Add**.
2. Enter a **Key** — a short identifier such as `smtp_password`. You reference it elsewhere as `{{vault:smtp_password}}`.
3. Enter a human-readable **Label** and paste the secret **Value**. Save.

Master key source

The vault's contents are protected by a master key, and you can choose where that key comes from:

File *(default)*

An auto-generated key file kept with strict permissions on the host.

Environment variable

You supply the key via an environment variable — for advanced setups.

AWS / GCP / Azure KMS

A cloud key-management service unwraps the key at startup.

YubiKey

A hardware key unwraps the master key.

Switching the source **re-encrypts** the whole vault with the new key — your stored secrets are preserved. Every read, write, and source change is recorded in the vault's audit log.

07 Network exposure & hardening

By default, Mili listens on `0.0.0.0:5001` — meaning it is reachable from **every device on the network** that can route to the host. That is convenient for a shared deployment, but it is more open than many setups need.

Restricting to the local machine

To make the agent reachable only from the host it runs on, set the **Server IP / Host** to `127.0.0.1` instead of `0.0.0.0`.

1. Open Server settings at /server/settings.
2. Change **Host** from `0.0.0.0` to `127.0.0.1`.
3. Save, then restart the server for the change to take effect.

CONTAINERISED DEPLOYMENTS

When Mili runs in Docker, the published port also governs exposure. To bind it to the local machine only, map the port as `127.0.0.1:5001:5001` in `docker-compose.yml` in addition to setting the host. For anything reachable from the network, always pair exposure with TLS (section 11) and authentication.

08 Public URL — agent-to-agent file transfers

The **Public URL** field (under Server settings) is this agent's externally reachable base address. It is used for sharing files between agents.

When your agent shares a file with a peer, it builds a download link from its own Public URL — because only this agent knows its own externally reachable address. The peer then fetches the file from that link. Receivers never guess the address; the sender supplies it.

Where

Server settings › Public URL

Example

`https://172.31.124.220:5001`

Why it matters

If it is not set, file sharing between agents cannot produce a usable download link.

SET THIS TO THE ADDRESS PEERS CAN ACTUALLY REACH

Use the host's real network address (and port), reachable by the agents you exchange files with — not `127.0.0.1`, which is only valid on the local machine.

09 Software updates

Updates are delivered as a new release package. You apply one by running the installer again on each machine.

1. Obtain the new release and run the installer for your platform on the target machine.
2. The installer loads the new application image (replacing the previous version) and restarts the container.
3. Your settings and history are kept: the installer does **not** overwrite your configuration, certificates, or conversation data if they already exist.

PRESERVED ACROSS UPDATES

Configuration, TLS certificates, and all conversation history carry over automatically. There is nothing to export beforehand for a normal update.

Some operational changes — such as a configuration reload, a policy update, or a restart — can also be applied centrally by the management server without a full reinstall.

10 Backup & restore

Open `Settings › Backup & Restore` .

Creating a backup

Tap **Download Backup** to save a single `.zip` file. It contains your configuration, persona, memory, tasks, conversations, and stored credentials. Keep this file somewhere safe — it includes secrets.

Restoring

1. Drag a backup `.zip` onto the restore area, or tap to choose one.
2. Confirm. The server replaces all current data with the backup's contents and restarts.

RESTORE IS DESTRUCTIVE

Restoring **replaces all existing data** and cannot be undone. Take a fresh backup of the current state first if you might need it.

11 TLS certificates

Mili serves over HTTPS. At first boot it generates a self-signed certificate so the connection is encrypted immediately; for production you'll usually replace it with your own trusted certificate.

Open `Settings › SSL / TLS` .

SSL Enabled

Toggle HTTPS on or off.

Status

Shows whether the certificate and private key are present.

Uploading your own certificate

1. Under **Upload Certificate & Key**, upload your certificate (`.crt` / `.pem`).
2. Upload the matching private key (`.key` / `.pem`).
3. Restart the server. The uploaded files replace the previous certificate and key.

SELF-SIGNED VS. TRUSTED

The first-boot self-signed certificate encrypts traffic but browsers will warn it isn't trusted. Upload a certificate from a trusted authority (or your internal CA) to remove the warning.