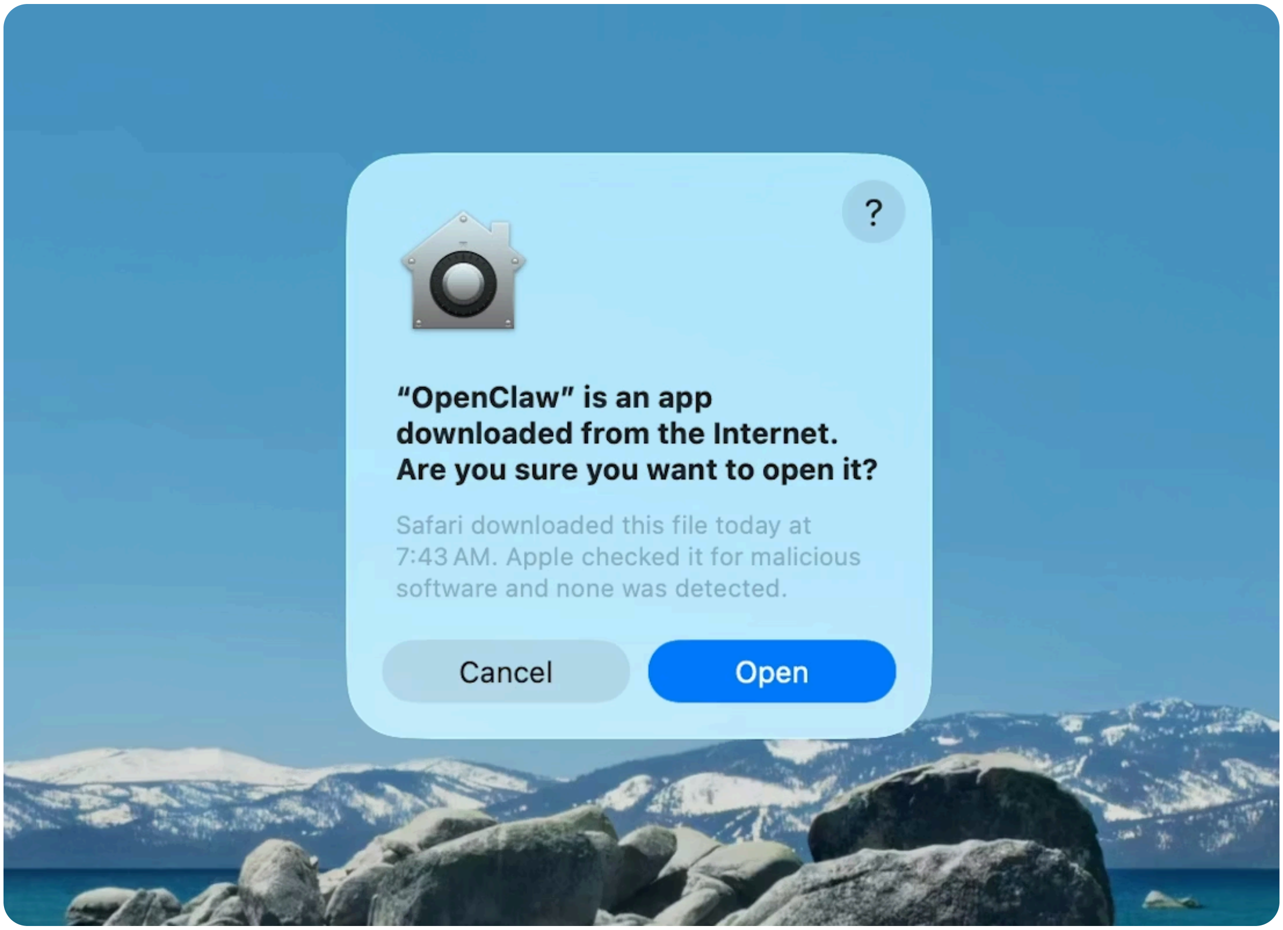
First steps

Onboarding (macOS App)

This doc describes the **current** first-run onboarding flow. The goal is a smooth “day 0” experience: pick where the Gateway runs, connect auth, run the wizard, and let the agent bootstrap itself. For a general overview of onboarding paths, see Onboarding Overview.

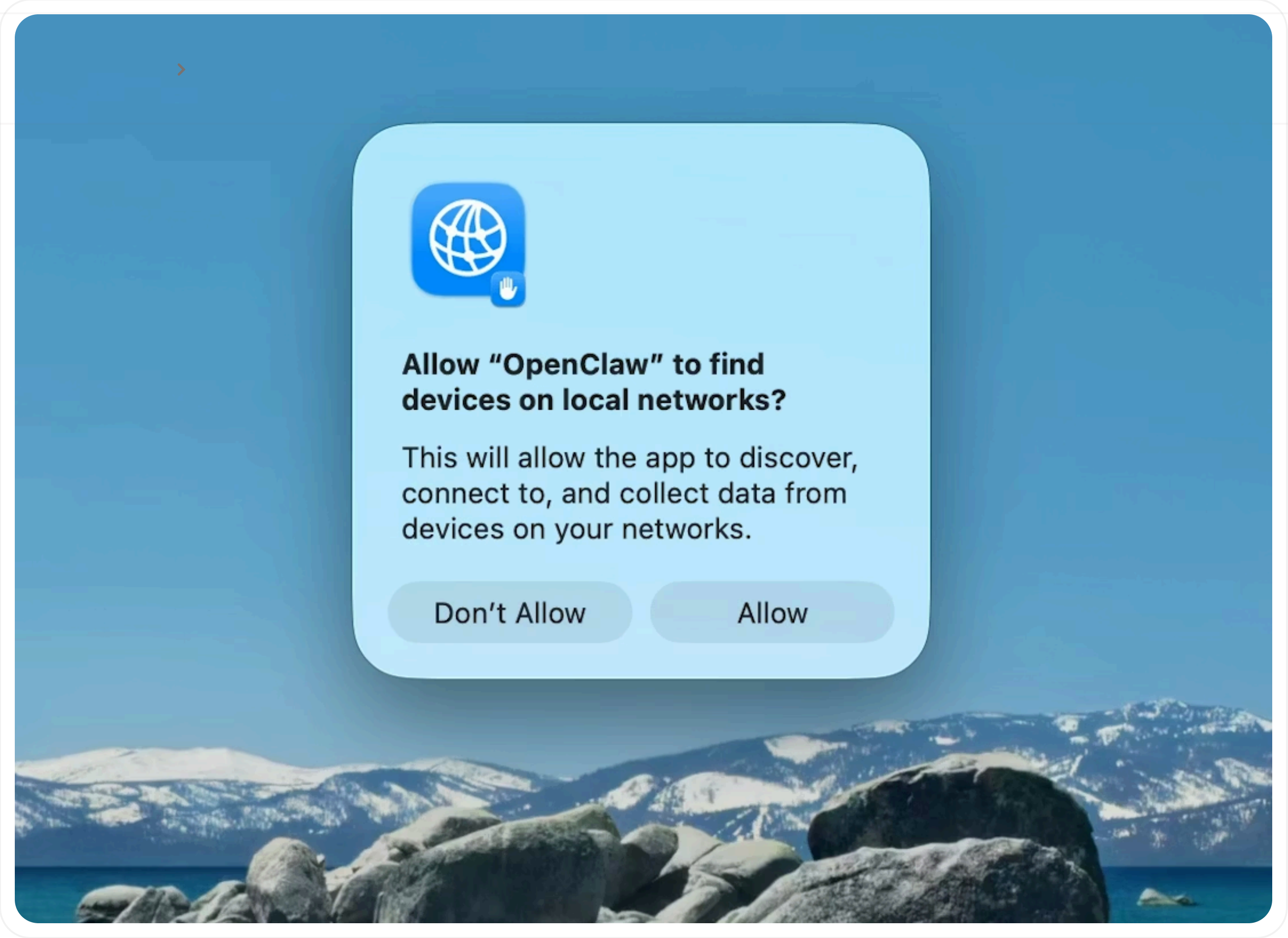
1 Approve macOS warning



“OpenClaw” is an app downloaded from the Internet. Are you sure you want to open it?

Safari downloaded this file today at 7:43 AM. Apple checked it for malicious software and none was detected.

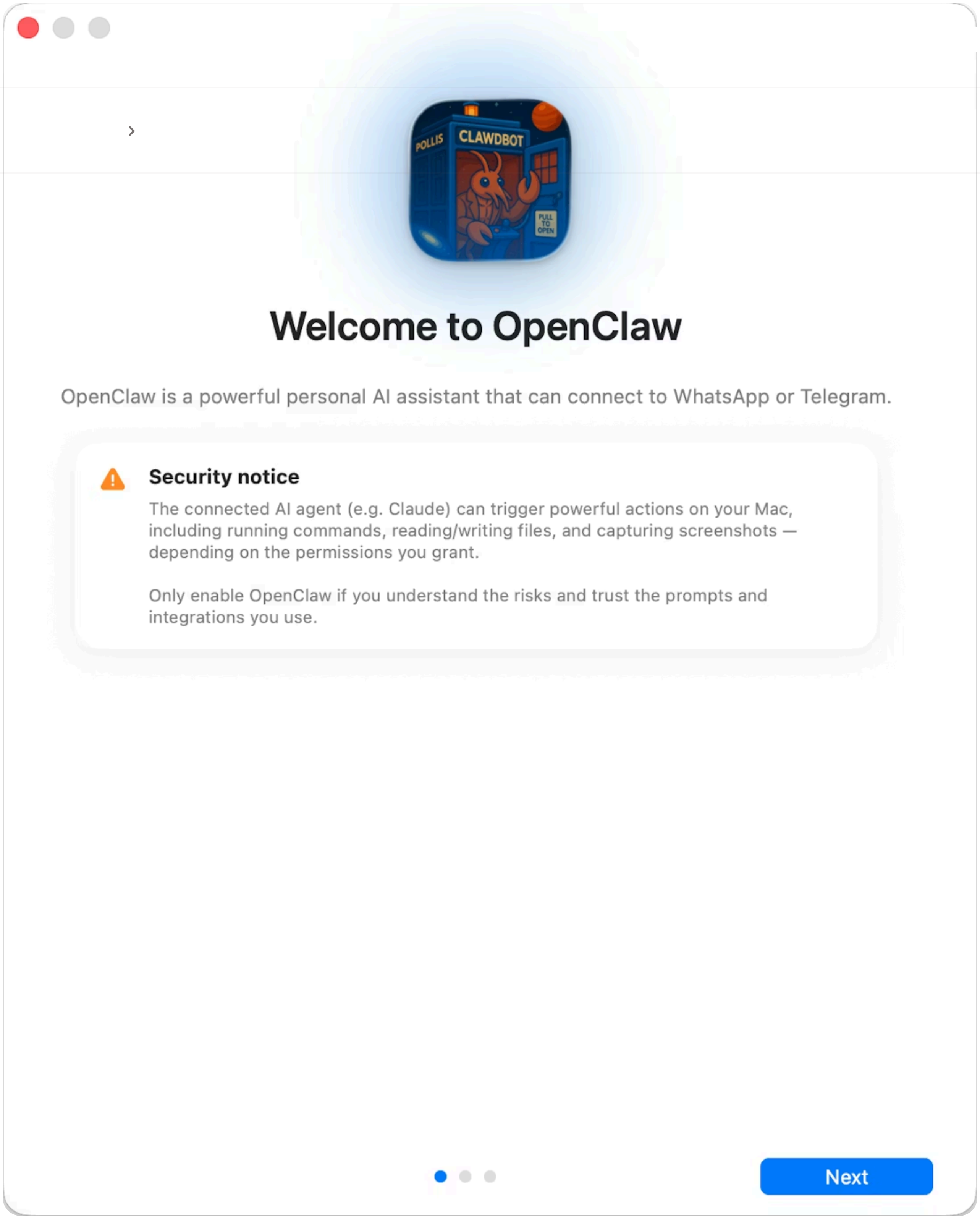
Approve find local networks



Allow “OpenClaw” to find devices on local networks?

This will allow the app to discover, connect to, and collect data from devices on your networks.

3 Welcome and security notice



Welcome to OpenClaw

OpenClaw is a powerful personal AI assistant that can connect to WhatsApp or Telegram.

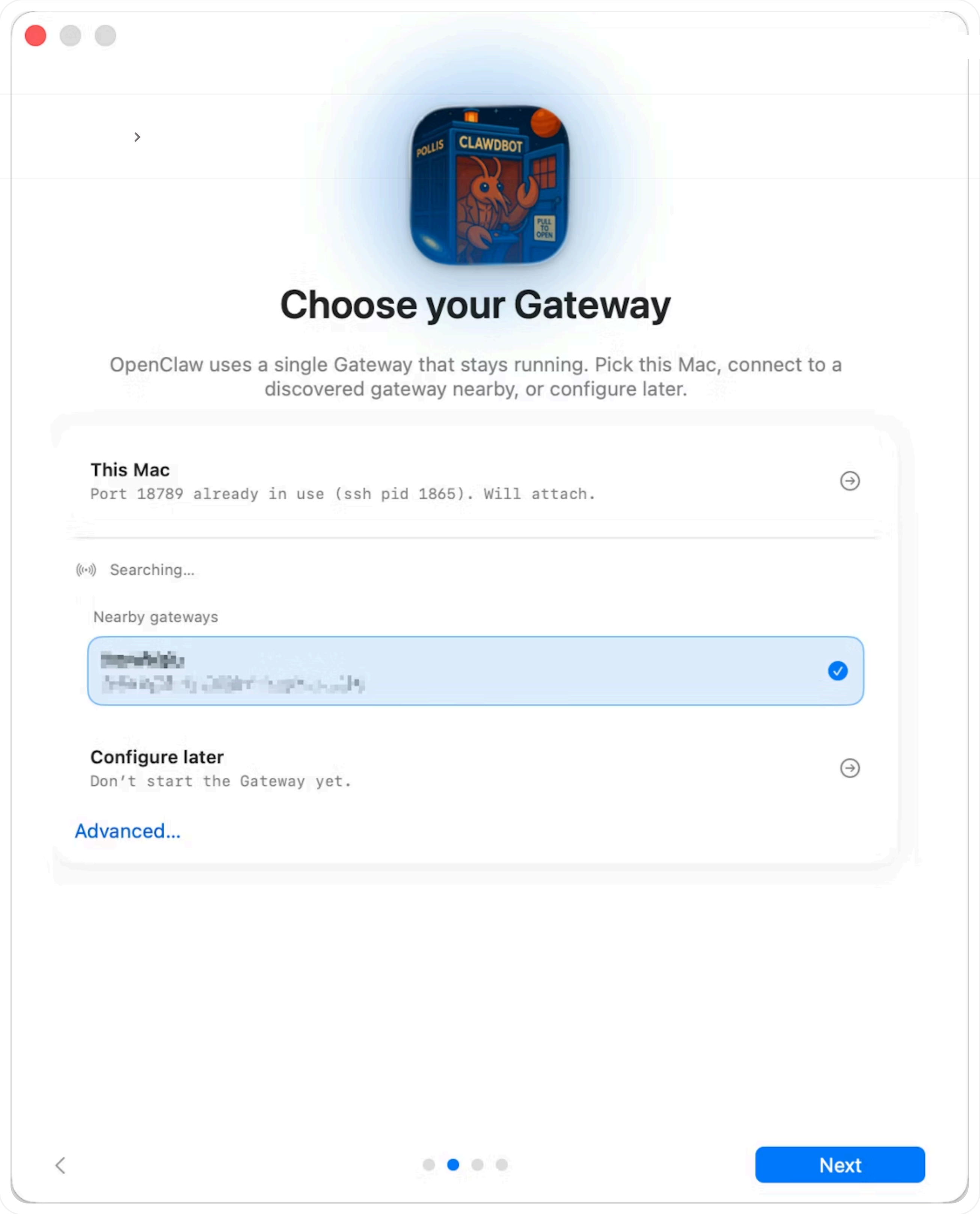
Security notice

The connected AI agent (e.g. Claude) can trigger powerful actions on your Mac, including running commands, reading/writing files, and capturing screenshots — depending on the permissions you grant.

Only enable OpenClaw if you understand the risks and trust the prompts and integrations you use.

Read the security notice displayed and decide accordingly

4 Local vs Remote



Choose your Gateway

OpenClaw uses a single Gateway that stays running. Pick this Mac, connect to a discovered gateway nearby, or configure later.

This Mac
Port 18789 already in use (ssh pid 1865). Will attach.

Searching…

Nearby gateways

[redacted] ✓

Configure later
Don’t start the Gateway yet.

Where does the **Gateway** run?

- **This Mac (Local only):** onboarding can run OAuth flows and write credentials locally.

Remote (over SSH/Tailnet): onboarding does **not** run OAuth locally; credentials must exist on the gateway host.

Configure later: skip setup and leave the app unconfigured.

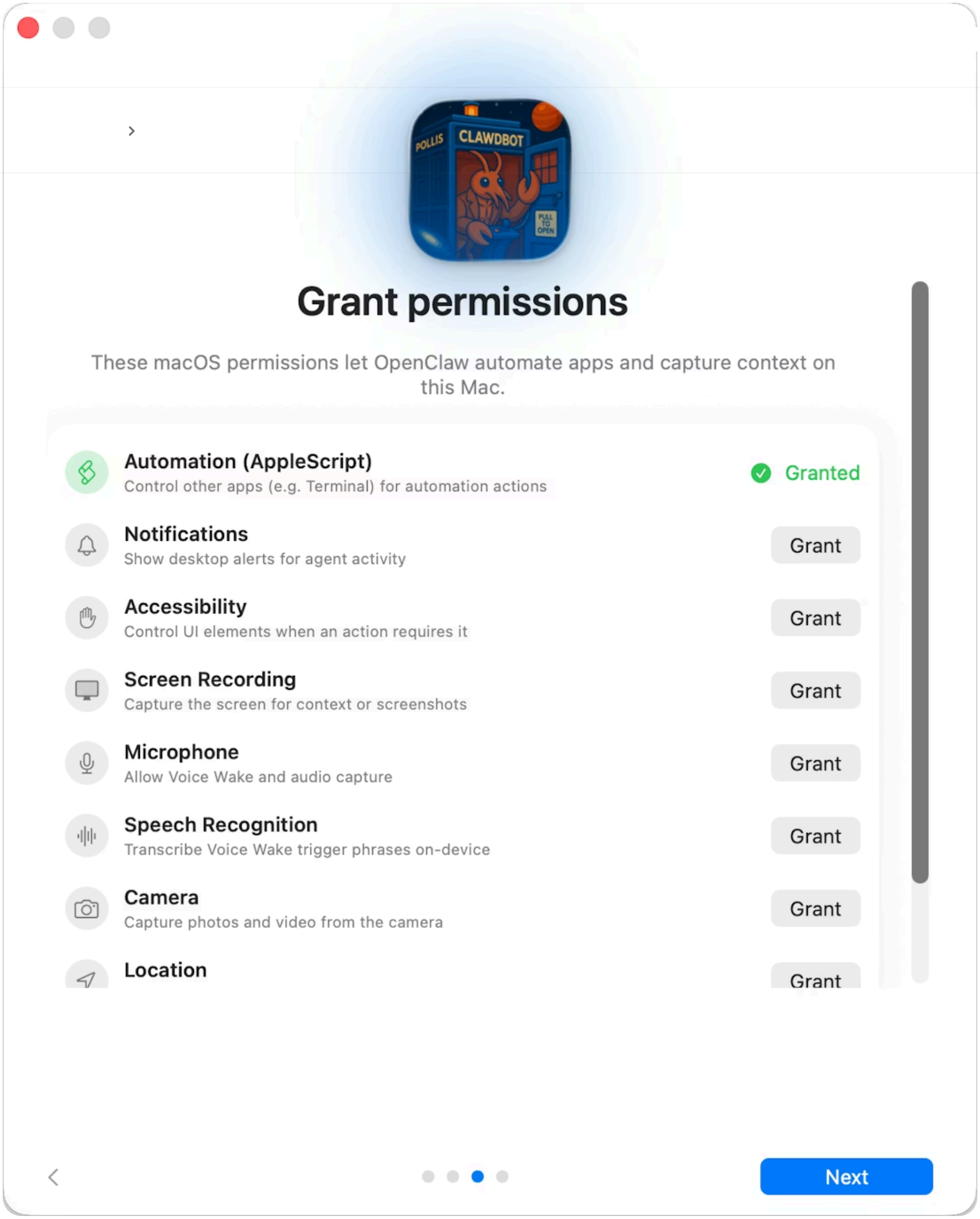
Gateway auth tip:

The wizard now generates a **token** even for loopback, so local WS clients must authenticate.

If you disable auth, any local process can connect; use that only on fully trusted machines.

Use a **token** for multi-machine access or non-loopback binds.

5 Permissions



Grant permissions

These macOS permissions let OpenClaw automate apps and capture context on this Mac.

- **Automation (AppleScript)** — Control other apps (e.g. Terminal) for automation actions — ✓ Granted
- **Notifications** — Show desktop alerts for agent activity — Grant
- **Accessibility** — Control UI elements when an action requires it — Grant
- **Screen Recording** — Capture the screen for context or screenshots — Grant
- **Microphone** — Allow Voice Wake and audio capture — Grant
- **Speech Recognition** — Transcribe Voice Wake trigger phrases on-device — Grant
- **Camera** — Capture photos and video from the camera — Grant
- **Location** — Grant

Choose what permissions do you want to give OpenClaw

Onboarding requests TCC permissions needed for:

- Automation (AppleScript)

- Notifications
- Accessibility
- Screen Recording
- Microphone
- Speech Recognition
- Camera
- Location

6 CLI

ⓘ This step is optional

The app can install the global `openclaw` CLI via npm/pnpm so terminal workflows and launchd tasks work out of the box.

7 Onboarding Chat (dedicated session)

After setup, the app opens a dedicated onboarding chat session so the agent can introduce itself and guide next steps. This keeps first-run guidance separate from your normal conversation. See **Bootstrapping** for what happens on the gateway host during the first agent run.

‹ Onboarding: CLI Personal Assistant Setup ›