



≡ macOS companion app > macOS IPC

macOS companion app

## macOS IPC

**Current model:** a local Unix socket connects the `node host service` to the `macOS app` for exec approvals + `system.run`. A `openclaw-mac` debug CLI exists for discovery/connect checks; agent actions still flow through the Gateway WebSocket and `node.invoke`. UI automation uses PeekabooBridge.

## Goals

Single GUI app instance that owns all TCC-facing work (notifications, screen recording, mic, speech, AppleScript).

A small surface for automation: Gateway + node commands, plus PeekabooBridge for UI automation.

Predictable permissions: always the same signed bundle ID, launched by launchd, so TCC grants stick.

## How it works

### Gateway + node transport

The app runs the Gateway (local mode) and connects to it as a node.

Agent actions are performed via `node.invoke` (e.g. `system.run`, `system.notify`, `canvas.*`).

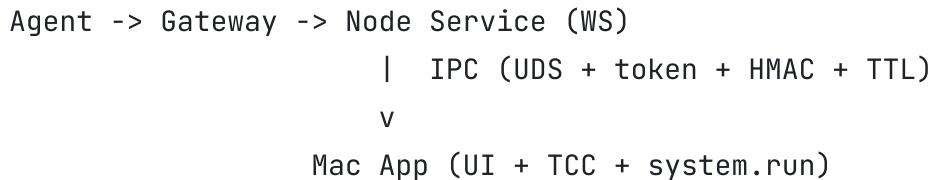
## Node service + app IPC

A headless node host service connects to the Gateway WebSocket.

`system.run` requests are forwarded to the macOS app over a local Unix socket.

The app performs the exec in UI context, prompts if needed, and returns output.

Diagram (SCI):



## PeekabooBridge (UI automation)

UI automation uses a separate UNIX socket named `bridge.sock` and the PeekabooBridge JSON protocol.

Host preference order (client-side): `Peekaboo.app` → `Claude.app` → `OpenClaw.app` → local execution.

Security: bridge hosts require an allowed TeamID; DEBUG-only same-UID escape hatch is guarded by `PEEKABOO_ALLOW_UNSIGNED_SOCKET_CLIENTS=1` (Peekaboo convention).

See: [for details.](#)

## Operational flows

Restart/rebuild: `SIGN_IDENTITY="Apple Development: <Developer Name> (<TEAMID>)" scripts/restart-mac.sh`

Kills existing instances

Swift build + package



Writes/bootstraps/kickstarts the LaunchAgent

Single instance: app exits early if another instance with the same bundle ID is running.

>

## Hardening notes

Prefer requiring a TeamID match for all privileged surfaces.

PeekabooBridge: `PEEKABOO_ALLOW_UNSIGNED_SOCKET_CLIENTS=1` (DEBUG-only) may allow same-UID callers for local development.

All communication remains local-only; no network sockets are exposed.

TCC prompts originate only from the GUI app bundle; keep the signed bundle ID stable across rebuilds.

IPC hardening: socket mode `0600`, token, peer-UID checks, HMAC challenge/response, short TTL.

[◀ Gateway on macOS](#)

[Skills ▶](#)

Powered by [mintlify](#)