Multi-agent

Presence

OpenClaw “presence” is a lightweight, best-effort view of:

- the **Gateway** itself, and
- **clients connected to the Gateway** (mac app, WebChat, CLI, etc.)

Presence is used primarily to render the macOS app’s **Instances** tab and to provide quick operator visibility.

Presence fields (what shows up)

Presence entries are structured objects with fields like:

- `instanceId` (optional but strongly recommended): stable client identity (usually `connect.client.instanceId`)
- `host` : human-friendly host name
- `ip` : best-effort IP address
- `version` : client version string
- `deviceFamily` / `modelIdentifier` : hardware hints
- `mode` : `ui` , `webchat` , `cli` , `backend` , `probe` , `test` , `node` , …
- `lastInputSeconds` : “seconds since last user input” (if known)
- `reason` : `self` , `connect` , `node-connected` , `periodic` , …
- `ts` : last update timestamp (ms since epoch)

Producers (where presence comes from)

Presence entries are produced by multiple sources and **merged**.

›

1) Gateway self entry

The Gateway always seeds a “self” entry at startup so UIs show the gateway host even before any clients connect.

2) WebSocket connect

Every WS client begins with a `connect` request. On successful handshake the Gateway upserts a presence entry for that connection.

Why one-off CLI commands don’t show up

The CLI often connects for short, one-off commands. To avoid spamming the Instances list, `client.mode === "cli"` is **not** turned into a presence entry.

3) `system-event` beacons

Clients can send richer periodic beacons via the `system-event` method. The mac app uses this to report host name, IP, and `lastInputSeconds` .

4) Node connects (role: node)

When a node connects over the Gateway WebSocket with `role: node` , the Gateway upserts a presence entry for that node (same flow as other WS clients).

Merge + dedupe rules (why `instanceId` matters)

Presence entries are stored in a single in-memory map:

Entries are keyed by a **presence key**.

The best key is a stable `instanceId` (from `connect.client.instanceId`) that survives restarts.

Keys are case-insensitive.

If a client reconnects without a stable `instanceId`, it may show up as a **duplicate** row.

TTL and bounded size

Presence is intentionally ephemeral:

- **TTL:** entries older than 5 minutes are pruned
- **Max entries:** 200 (oldest dropped first)

This keeps the list fresh and avoids unbounded memory growth.

Remote/tunnel caveat (loopback IPs)

When a client connects over an SSH tunnel / local port forward, the Gateway may see the remote address as `127.0.0.1`. To avoid overwriting a good client-reported IP, loopback remote addresses are ignored.

Consumers

macOS Instances tab

The macOS app renders the output of `system-presence` and applies a small status indicator (Active/Idle/Stale) based on the age of the last update.

Debugging tips

To see the raw list, call `system-presence` against the Gateway.

If you see duplicates:

- confirm clients send a stable `client.instanceId` in the handshake
- confirm periodic beacons use the same `instanceId`
- check whether the connection-derived entry is missing `instanceId` (duplicates are expected)

‹ Multi-Agent Routing Messages ›