



RPC and API

RPC Adapters

OpenClaw integrates external CLIs via JSON-RPC. Two patterns are used today.

Pattern A: HTTP daemon (`signal-cli`)

`signal-cli` runs as a daemon with JSON-RPC over HTTP.

Event stream is SSE (`/api/v1/events`).

Health probe: `/api/v1/check`.

OpenClaw owns lifecycle when `channels.signal.autoStart=true`.

See [Signal](#) for setup and endpoints.

Pattern B: stdio child process (legacy: imsg)

Note: For new iMessage setups, use [BlueBubbles](#) instead.

OpenClaw spawns `imsg rpc` as a child process (legacy iMessage integration).

JSON-RPC is line-delimited over `stdin/stdout` (one JSON object per line).

No TCP port, no daemon required.

Core methods used:



```
watch.subscribe → notifications ( method: "message" )  
watch.unsubscribe  
  
send >  
  
chats.list (probe/diagnostics)
```

See [iMessage](#) for legacy setup and addressing (`chat_id` preferred).

Adapter guidelines

Gateway owns the process (start/stop tied to provider lifecycle).

Keep RPC clients resilient: timeouts, restart on exit.

Prefer stable IDs (e.g., `chat_id`) over display strings.

< [voicemail](#)

[Device Model Database](#) >

Powered by [mintlify](#)