

1 Branch 0 Tags Go to file Go to file Code ...

itsrealranky and cursoragent Add architecture diagram and move logo to img dire... ... ×

c030db9 · 15 hours ago 🕒

📁 .github	chore: add MIT license	18 hours ago
📁 .vscode	chore: add MIT license	18 hours ago
📁 benches	chore: add MIT license	18 hours ago
📁 cmake	chore: add MIT license	18 hours ago
📁 docs	chore: add MIT license	18 hours ago
📁 img	Add architecture diagram and ...	15 hours ago
📁 include/ghostclaw	Add multi-provider support and ...	17 hours ago
📁 packaging/homebrew	chore: add MIT license	18 hours ago
📁 scripts	chore: add MIT license	18 hours ago
📁 src	Add multi-provider support and ...	17 hours ago
📁 tests	Add multi-provider support and ...	17 hours ago
📄 .dockerignore	chore: add MIT license	18 hours ago
📄 .gitattributes	chore: add MIT license	18 hours ago
📄 .gitignore	chore: add MIT license	18 hours ago
📄 .pre-commit-config.yaml	chore: add MIT license	18 hours ago
📄 CMakeLists.txt	Add multi-provider support and ...	17 hours ago
📄 Dockerfile	chore: add MIT license	18 hours ago
📄 Dockerfile.alpine	chore: add MIT license	18 hours ago
📄 LICENSE	chore: add MIT license	18 hours ago
📄 Makefile	chore: add MIT license	18 hours ago
📄 README.md	Update README.md	15 hours ago
📄 docker-compose.yml	chore: add MIT license	18 hours ago

About

Grandfather of Openclaw

- 📖 Readme
- 📄 MIT license
- 📊 Activity
- ★ 3 stars
- 👁 0 watching
- 🍴 0 forks

Report repository

Releases

No releases published

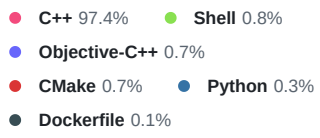
Packages

No packages published

Contributors 2

- itsrealranky
- cursoragent Cursor Agent

Languages






GhostClaw

Ghost Protocol. Claw Execution. Zero Compromise. 100% C++. 100% Agnostic.

 Grandfather of OpenClaw  Smallest and Fastest Feature-Complete Implementation of OpenClaw




 Runs on \$10 hardware with ~10MB RAM and a ~2MB binary — 99% less memory than OpenClaw




license [MIT](#)

Fast, small, and fully autonomous AI assistant infrastructure — deploy anywhere, swap anything.

~1.9MB binary · ~15ms warm start · 30+ providers · 10 traits · Pluggable everything





Features

 Ultra-Lightweight	 Lightning Fast	 True Portability
~2MB peak footprint ~10MB RSS	15ms warm start 235ms cold start	Single binary ARM + x86





 Fully Swappable	 No Lock-in	 Secure by Design
Trait-based architecture Zero code changes	30+ AI providers OpenAI-compatible	Pairing + sandboxing Workspace scoping

Why Teams Pick GhostClaw

Production Ready

-  Small C++ binary, fast startup
-  Pairing, strict sandboxing, allowlists
-  Single self-contained binary
-  Trait-based, fully swappable

Feature Rich

-  30+ AI provider integrations
-  10+ real messaging channels
-  Browser automation built-in
-  Canvas, voice, TTS support

Local machine quick benchmark (macOS arm64, Feb 2026) normalized for 0.8GHz edge hardware.

	OpenClaw	NanoBot	PicoClaw	ZeroClaw	GhostClaw 🐾
Language	TypeScript	Python	Go	Rust	C++
RAM	> 1GB	> 100MB	< 10MB	< 5MB	~10MB
Startup (warm)	> 500ms	> 30ms	< 100ms	< 10ms	~15ms
Cold Start	> 5s	> 1s	< 1s	~440ms	~235ms
Binary Size	~28MB (dist)	N/A (Scripts)	~8MB	~3.4MB	1.9MB
Cost	Mac Mini \$599	Linux SBC ~\$50	Linux Board \$10	Any hardware \$10	Any hardware \$10

Notes: GhostClaw results measured with `/usr/bin/time -l` on release builds. OpenClaw requires Node.js runtime (~390MB overhead). PicoClaw and ZeroClaw are static binaries. GhostClaw has the **smallest binary** and **fastest cold start**.

▶  Detailed GhostClaw Measurements

▶ Reproduce these numbers yourself


```
git clone https://github.com/sudiprokaya/GhostClaw.git
cd GhostClaw
cmake -S . -B build -DCMAKE_BUILD_TYPE=Release
cmake --build build -j
```

```
# Just run it – the setup wizard launches automatically on first run
./build/ghostclaw
```

The wizard walks you through provider, model, API key, memory, channels, and skills in 7 steps, then offers to drop you straight into the agent.


```
# Skip the wizard entirely with flags
ghostclaw onboard --provider openrouter --api-key sk...
```

```
# Single message
ghostclaw agent -m "Hello, GhostClaw!"
```

```
# Interactive chat
ghostclaw agent
```

```
# Start the gateway (webhook server)
ghostclaw gateway
```

```
# Start full autonomous daemon
ghostclaw daemon
```

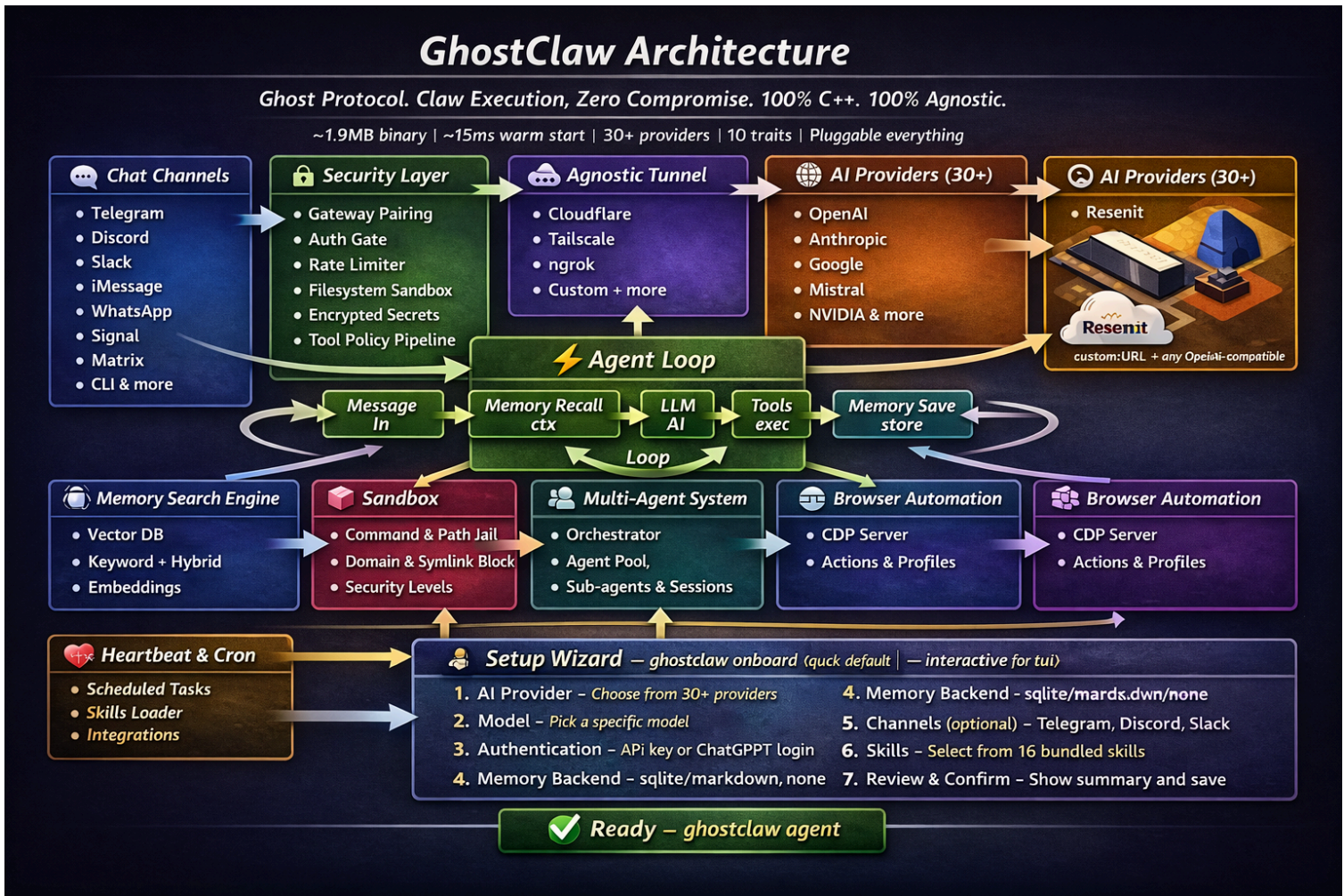
```
# Diagnostics
ghostclaw status
ghostclaw doctor
```



```
docker build -t ghostclaw .
docker run --rm ghostclaw --help
```

Architecture

Every subsystem is a trait — swap implementations with a config change, zero code changes.



▶ View Subsystem Details

Memory System

All custom, zero external dependencies — no Pinecone, no Elasticsearch, no LangChain

Layer	Implementation
Vector DB	Embeddings stored as BLOB in SQLite, cosine similarity search
Keyword Search	FTS5 virtual tables with BM25 scoring
Hybrid Merge	Custom weighted merge function
Embeddings	EmbeddingProvider trait — OpenAI, local, or noop
Chunking	Line-based markdown chunker with heading preservation
Caching	SQLite embedding_cache table with LRU eviction

```
[memory]
backend = "sqlite"           # "sqlite", "markdown", "none"
auto_save = true
embedding_provider = "openai"
```



Security

GhostClaw enforces security at every layer

#	Item	Status	How
1	Gateway not publicly exposed	✓	Binds 127.0.0.1 by default. Refuses 0.0.0.0 without tunnel or explicit <code>allow_public_bind = true</code>
2	Pairing required	✓	6-digit one-time code on startup. Exchange via POST <code>/pair</code> for bearer token
3	Filesystem scoped	✓	<code>workspace_only = true</code> by default. System dirs blocked. Symlink escape detection
4	Access via tunnel only	✓	Gateway refuses public bind without active tunnel

Configuration

Config: `~/ghostclaw/config.toml` (created automatically on first run)

►  [View Example Configuration](#)

 Full reference: [docs/CONFIGURATION.md](https://docs.ghostclaw.com/CONFIGURATION.md)

Commands

Core Commands

Command	Description
<i>(no args, first run)</i>	Auto-launches 7-step wizard
<code>onboard</code>	Re-run setup wizard
<code>onboard --provider X</code>	Non-interactive setup
<code>agent</code>	Interactive chat mode
<code>agent -m "..."</code>	Single message mode
<code>gateway</code>	Start HTTP gateway
<code>daemon</code>	Start autonomous daemon

Management Commands

Command	Description
<code>service install/start/stop</code>	Manage background service
<code>doctor</code>	Run diagnostics checks
<code>status</code>	Show system status
<code>channel doctor</code>	Check channel health
<code>cron</code>	Manage scheduler jobs
<code>skills</code>	Manage skill packages
<code>tts</code>	Text-to-speech providers
<code>voice</code>	Wake-word utilities
<code>integrations</code>	Browse registry

Development

```
# Debug build + tests
cmake -S . -B build -DCMAKE_BUILD_TYPE=Debug
cmake --build build -j
ctest --test-dir build --output-on-failure
```

```
# Release build
```










```
cmake -S . -B build-release -DCMAKE_BUILD_TYPE=Release
cmake --build build-release -j
```

```
# Run benchmarks
./build/benches/ghostclaw_benchmarks
```

Contributing

See [CONTRIBUTING.md](#). Implement a trait, submit a PR:

- | | |
|---|---|
| <ul style="list-style-type: none">•  New Provider → src/providers/•  New Channel → src/channels/•  New Observer → src/observability/•  New Tool → src/tools/ | <ul style="list-style-type: none">•  New Memory → src/memory/•  New Tunnel → src/tunnel/•  New Skill →
~/ghostclaw/workspace/skills/<name>/ |
|---|---|