WasmEdgeRuntime

Run LLM agents on self-hosted devices

Michael Yuan
x: juntao github: juntao
WasmEdge Runtime: https://github.com/WasmEdge/WasmEdge

Demo: The easiest way to chat with an open-source LLM on your own device

Download and install the software

```
curl -sSfL 'https://github.com/GaiaNet-AI/gaianet-node/releases/latest/download/install.sh' | bash
```

Initialize with a Qwen2 1.5b model

```
gaianet init --config
https://raw.githubusercontent.com/GaiaNet-AI/node-configs/main/qwen2-1.5b-instruct/config.json
```

Start the chatbot

```
gaianet start
```

Chat!

```
http://localhost:8080/
```

https://docs.gaianet.ai/node-guide/quick-start

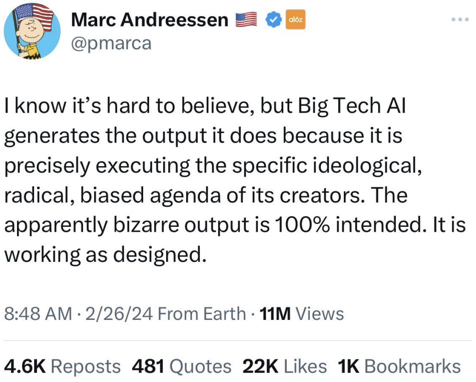
Unique features

- Lightweight
- Portable across OSes / CPUs / GPUs / NPUs
- Supports a wide variety of models
- Simple and transparent model management
- Easily embeddable into applications

The next frontier of AI

Why?

- Privacy and control
- Speed
- Reliability
- Alignment and bias
- Use finetuned model for each agentic task
- Tightly couple models with applications



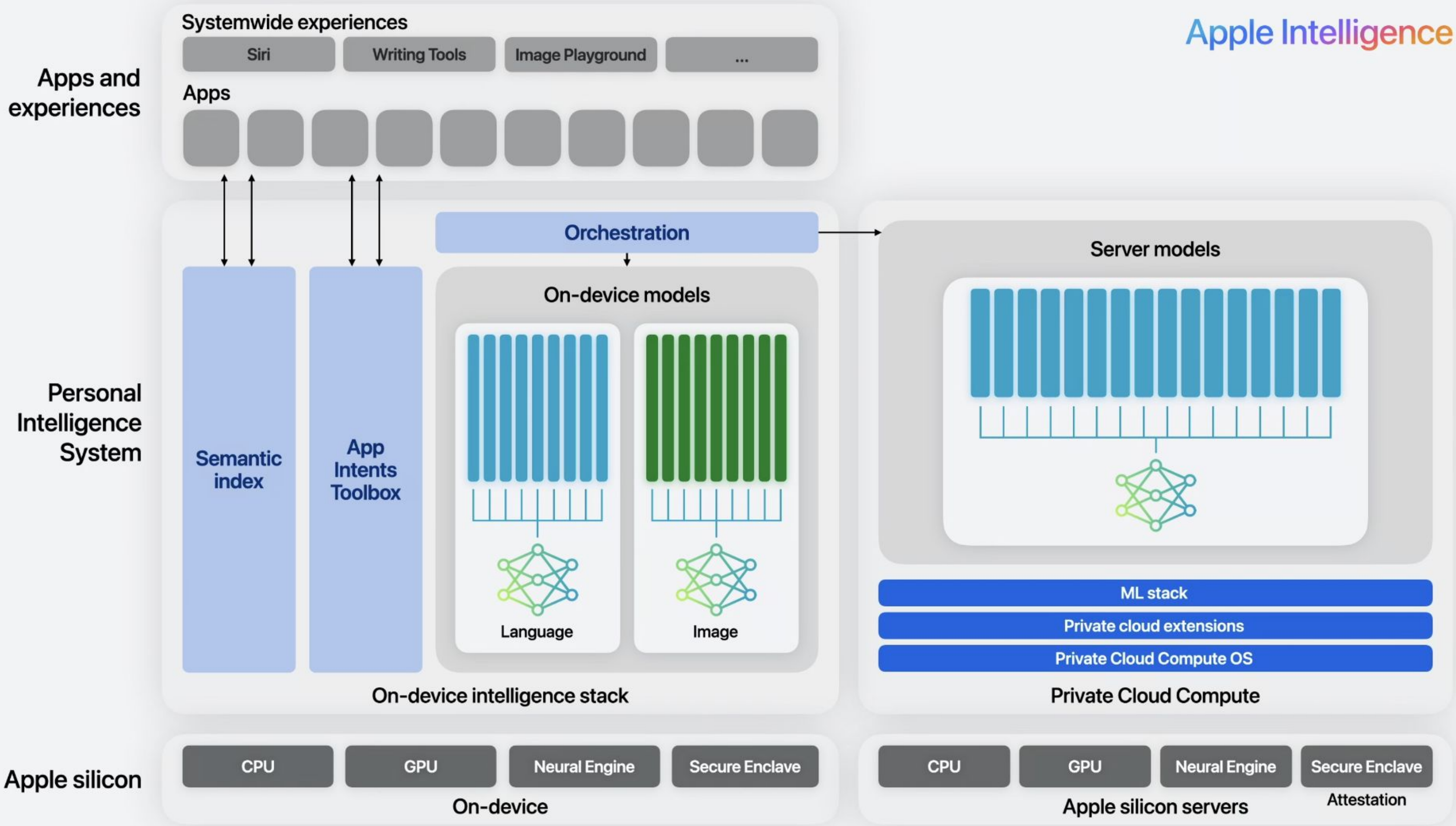
Marc Andreessen 🇺🇸
@pmarca

I know it’s hard to believe, but Big Tech AI generates the output it does because it is precisely executing the specific ideological, radical, biased agenda of its creators. The apparently bizarre output is 100% intended. It is working as designed.

8:48 AM · 2/26/24 From Earth · **11M** Views

4.6K Reposts **481** Quotes **22K** Likes **1K** Bookmarks

Apple Intelligence



Apps and experiences

Systemwide experiences

Siri | Writing Tools | Image Playground | ...

Apps

Personal Intelligence System

Semantic index

App Intents Toolbox

Orchestration

On-device models

Language

Image

On-device intelligence stack

Server models

ML stack

Private cloud extensions

Private Cloud Compute OS

Private Cloud Compute

Apple silicon

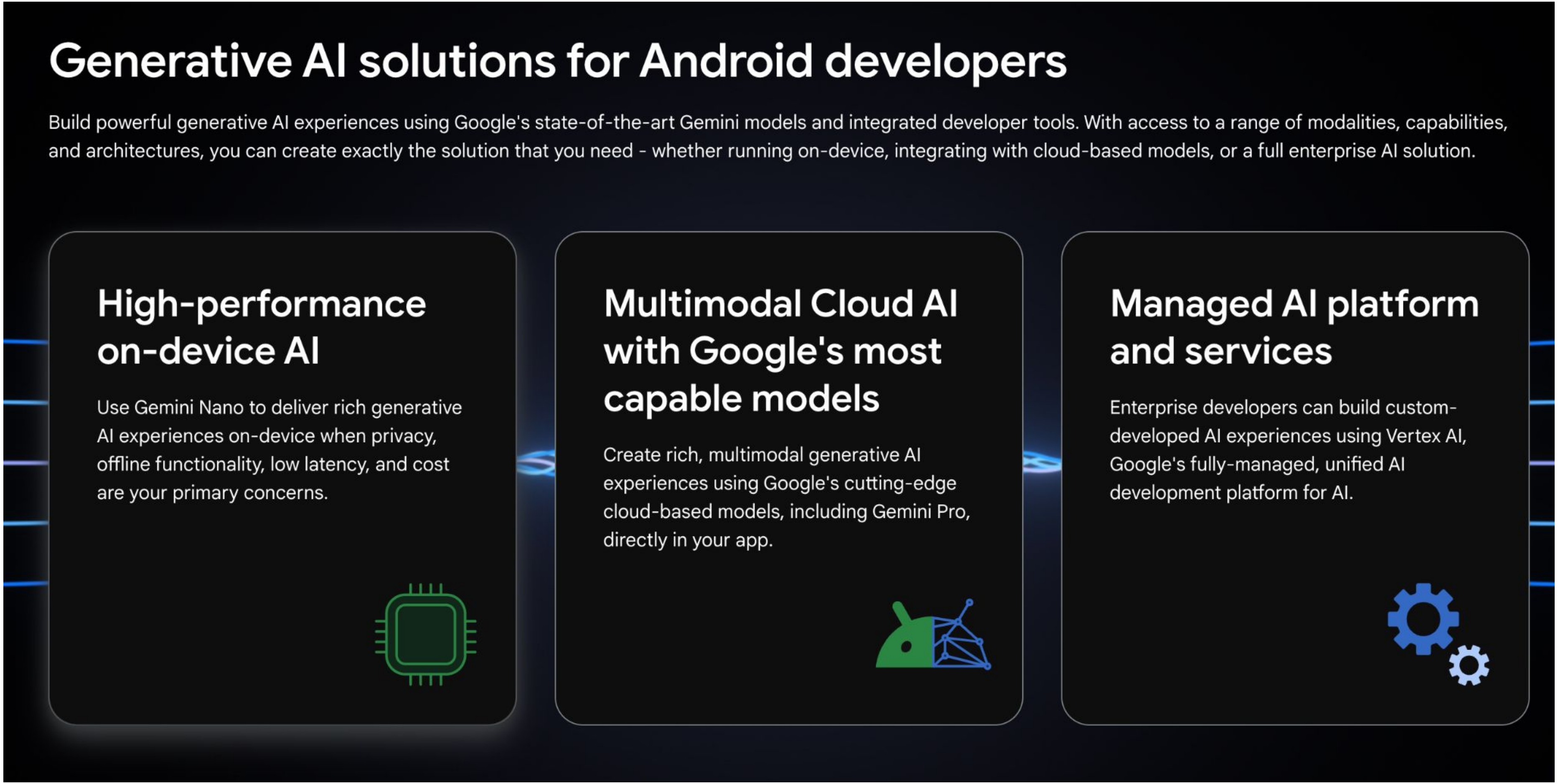
CPU | GPU | Neural Engine | Secure Enclave

On-device

CPU | GPU | Neural Engine | Secure Enclave

Attestation

Apple silicon servers



Generative AI solutions for Android developers

Build powerful generative AI experiences using Google's state-of-the-art Gemini models and integrated developer tools. With access to a range of modalities, capabilities, and architectures, you can create exactly the solution that you need - whether running on-device, integrating with cloud-based models, or a full enterprise AI solution.

High-performance on-device AI

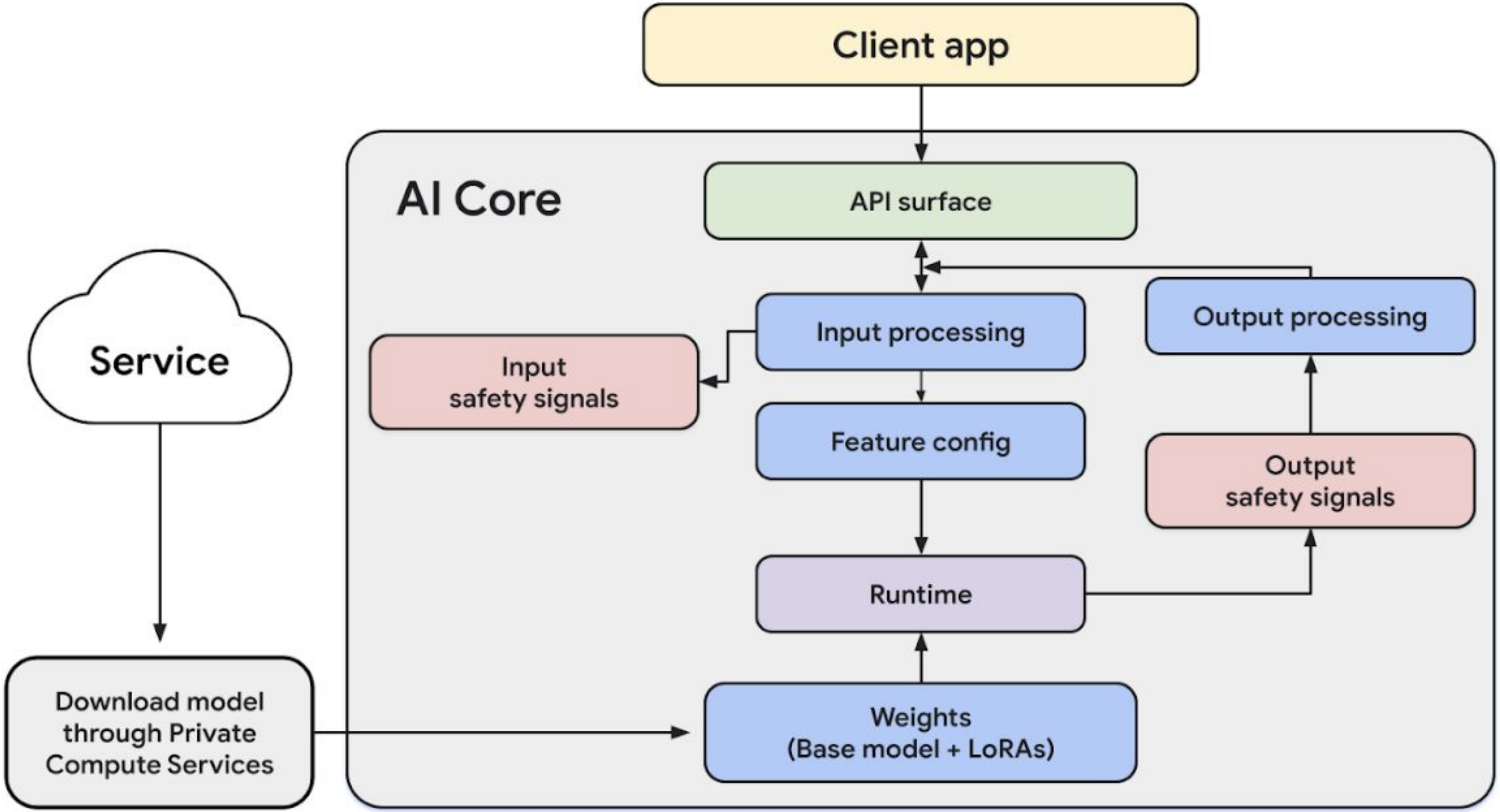
Use Gemini Nano to deliver rich generative AI experiences on-device when privacy, offline functionality, low latency, and cost are your primary concerns.

Multimodal Cloud AI with Google's most capable models

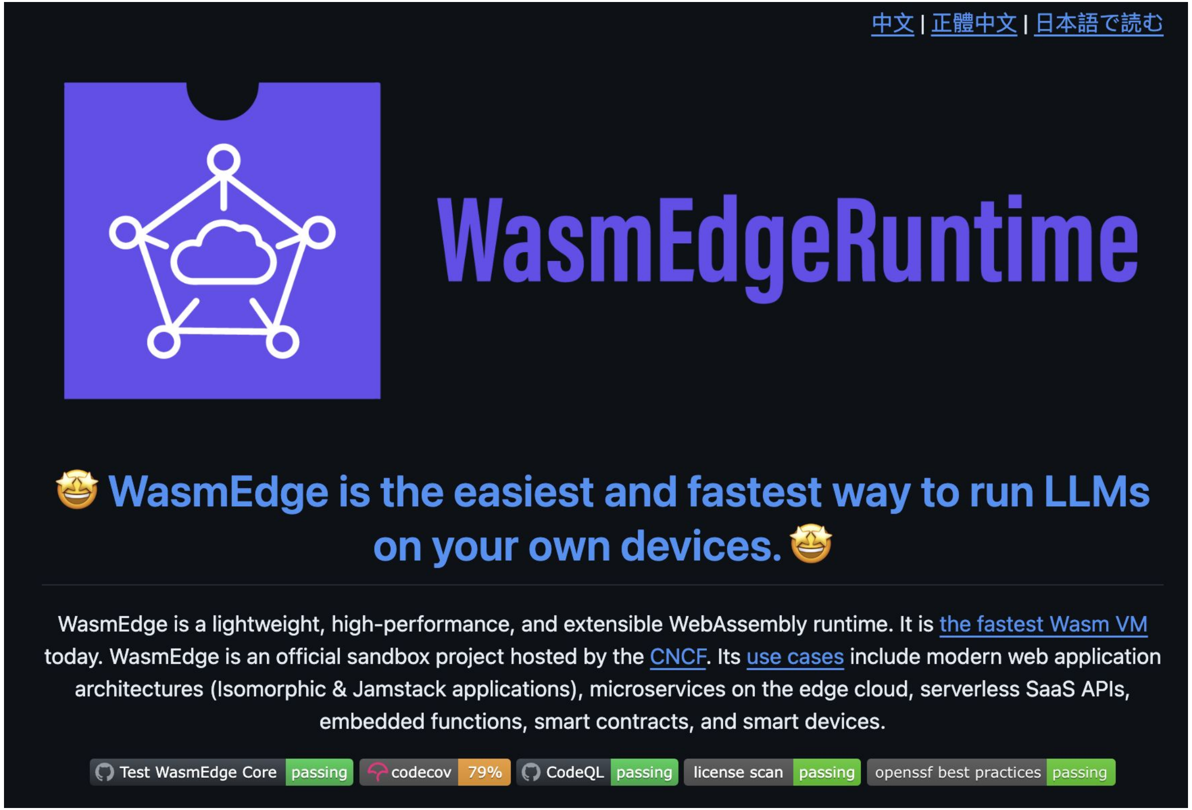
Create rich, multimodal generative AI experiences using Google's cutting-edge cloud-based models, including Gemini Pro, directly in your app.

Managed AI platform and services

Enterprise developers can build custom-developed AI experiences using Vertex AI, Google's fully-managed, unified AI development platform for AI.


Client app
AI Core
API surface
Service
Input processing
Output processing
Input
safety signals
Feature config
Output
safety signals
Runtime
Download model
through Private
Compute Services
Weights
(Base model + LoRAs)

Tech stack

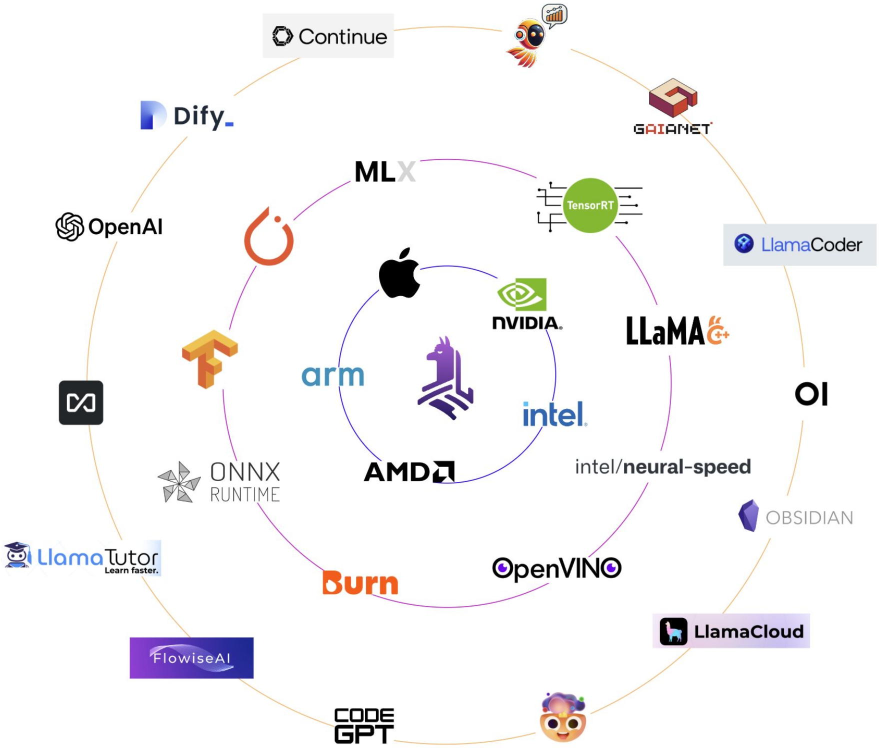


中文 | 正體中文 | 日本語で読む

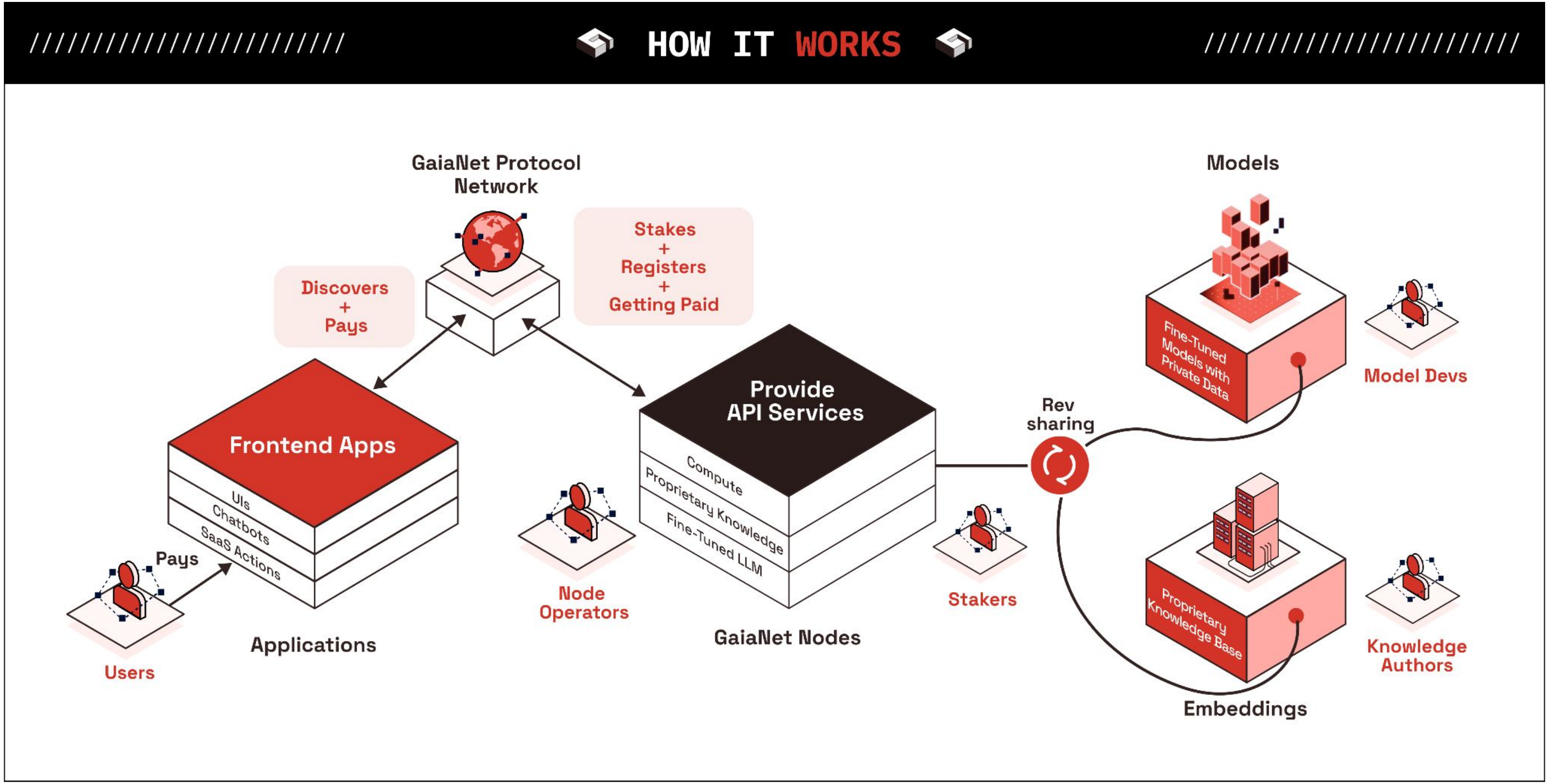
🤩 WasmEdge is the easiest and fastest way to run LLMs on your own devices. 🤩

WasmEdge is a lightweight, high-performance, and extensible WebAssembly runtime. It is the fastest Wasm VM today. WasmEdge is an official sandbox project hosted by the CNCF. Its use cases include modern web application architectures (Isomorphic & Jamstack applications), microservices on the edge cloud, serverless SaaS APIs, embedded functions, smart contracts, and smart devices.

WasmEdge: https://github.com/WasmEdge/WasmEdge



LlamaEdge: https://github.com/LlamaEdge/LlamaEdge



Gaia Network: https://github.com/GaiaNet-AI/gaianet-node

Demo: Enhance the model with your personal knowledge

Download and install the software

```
curl -sSfL 'https://github.com/GaiaNet-AI/gaianet-node/releases/latest/download/install.sh' | bash
```

Initialize with a Llama 3.1 model with a Samsung smartphone manual

```
gaianet init --config
https://raw.githubusercontent.com/GaiaNet-AI/node-configs/main/llama-3.1-8b-instruct_samsung-s24/config.json
```

Start the chatbot

```
gaianet start
```

Chat!

```
http://localhost:8080/
```

https://docs.gaianet.ai/node-guide/quick-start

Demo: Use the local LLM to control your local device

Download and install the software

```
curl -sSfL 'https://github.com/GaiaNet-AI/gaianet-node/releases/latest/download/install.sh' | bash
```

Initialize with a finetuned Llama 3 model with tool call support

```
gaianet init --config
https://raw.githubusercontent.com/GaiaNet-AI/node-configs/main/llama-3-groq-8b-tool/config.json
```

Start the API server

```
gaianet start
```

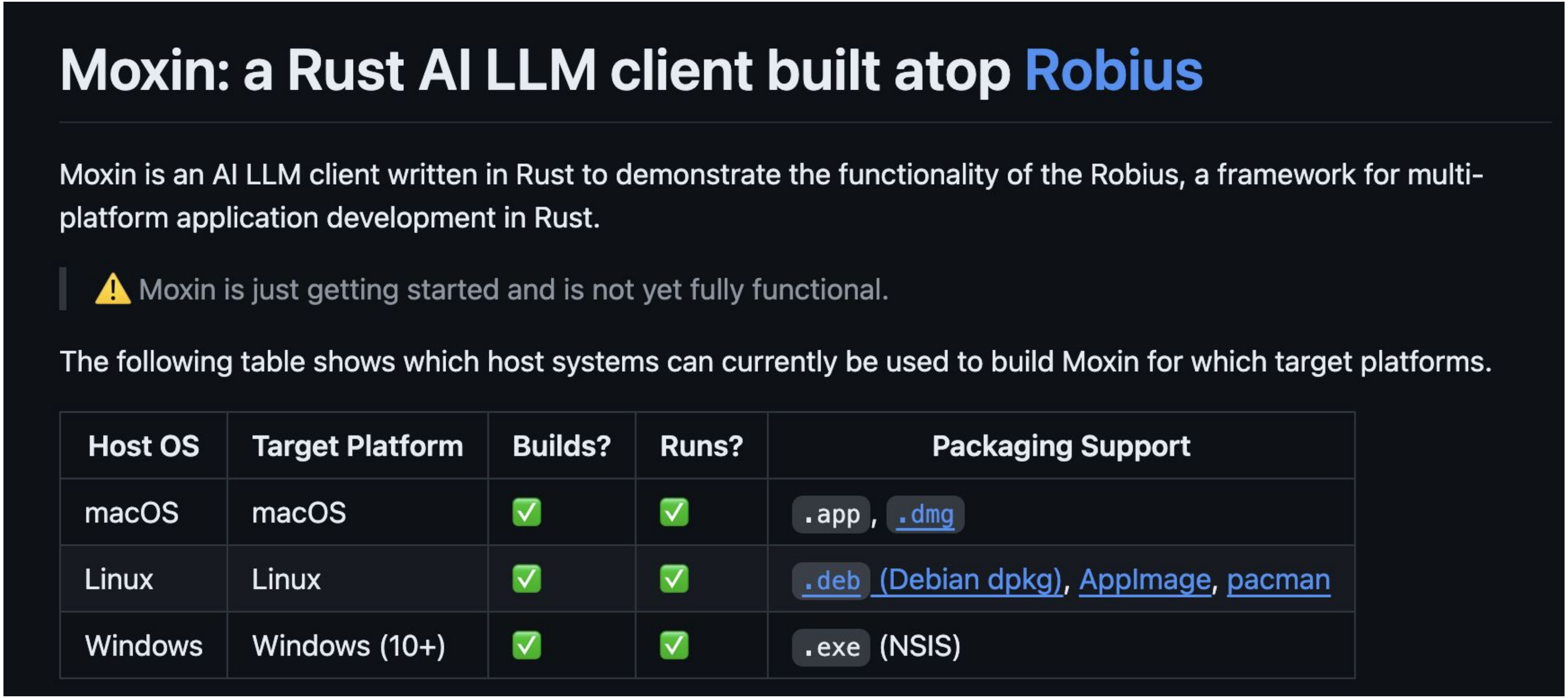
Run the agent

```
git clone https://github.com/second-state/llm_todo
cd llm_todo
pip install -r requirements.txt

export OPENAI_MODEL_NAME="llama-3-groq-8b"
export OPENAI_BASE_URL="http://127.0.0.1:8080/v1"

python main.py
```

I want an UI!



Moxin: a Rust AI LLM client built atop Robius

Moxin is an AI LLM client written in Rust to demonstrate the functionality of the Robius, a framework for multi-platform application development in Rust.

> ⚠️ Moxin is just getting started and is not yet fully functional.

The following table shows which host systems can currently be used to build Moxin for which target platforms.

Host OS	Target Platform	Builds?	Runs?	Packaging Support
macOS	macOS	✅	✅	`.app`, `.dmg`
Linux	Linux	✅	✅	`.deb` (Debian dpkg), AppImage, pacman
Windows	Windows (10+)	✅	✅	`.exe` (NSIS)

Moxin: https://github.com/moxin-org/moxin

Thank you

Learn more:

https://github.com/WasmEdge/WasmEdge
https://github.com/LlamaEdge/LlamaEdge
https://github.com/GaiaNet-AI/gaianet-node