Build Your Own Worker in an AI Assistant for your Experiment

Yipu Liao (廖一朴)
Institute of High Energy Physics, CAS, Beijing
On behalf of **Dr.Sai** working group

2026.4.14
@ IHEP, CAS, Beijing
AI and QC Workshop in Spring 2026

Tool-calling in Agents

References	Tools
Dr.Sai-BESIII (version 2)	Tool-calling via Open-Dr.Sai framework; remote HEP tools via **HepAI-DDF**
HEPTAPOD, arXiv:2512.15867	Tool-calling (**MCP**) via OrChestral-AI framework; local HEP tools
LLM4HEP, arXiv:2512.07785	Tool-calling via Snakemake workflow manager; local HEP tools
CoLLM, arXiv:2602.06496	Code generation, automated deep learning pipeline; no tool (framework study)
ColliderAgent, arXiv:2603.14553	**Skills** via Claude Code / Copilot etc; remote HEP tools via **Magnus**
AgenticAI4LEP, arXiv:2603.05735	Tool-calling via Codex / Claude; local HEP tools
JFC, arXiv: 2603.20179	**Skills** via Claude Code; local HEP tools

HEP related, partial

From answer to action: HaiDDF

Actuators

We need to remotely operate a server or use tools from another operating system or environment

- For example, if we want to use BESIII or CEPC software, which are installed IHEP cluster, we need to consider the communication issues between our work client service and the worker servers

From answer to action: HaiDDF

Actuators

We need to remotely operate a server or use tools from another operating system or environment

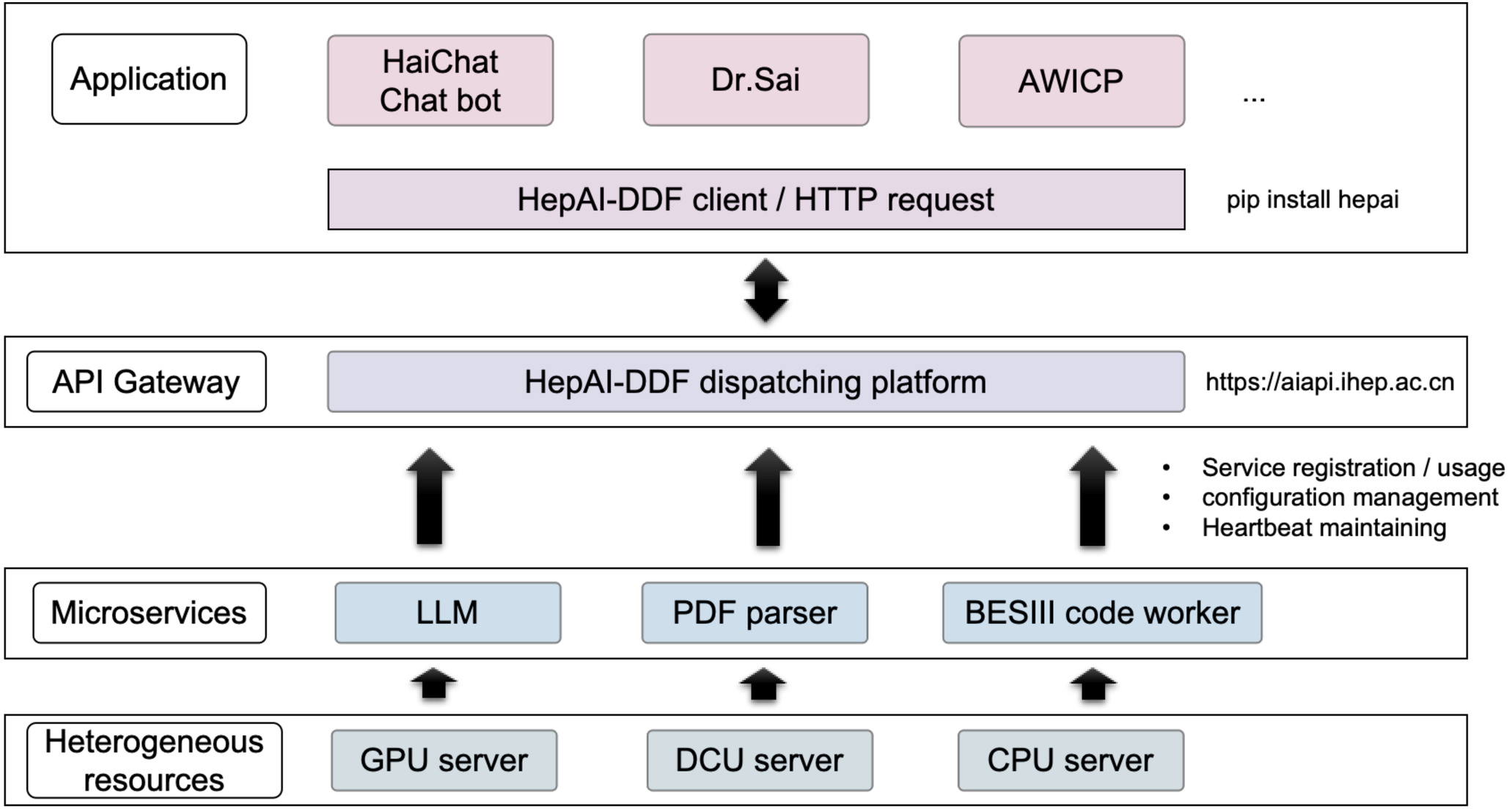
- For example, if we want to use BESIII or CEPC software, which are installed IHEP cluster, we need to consider the communication issues between our work client service and the worker servers

HaiDDF (HepAI-DDF) is a service framework designed specifically for this situation

- It provides a unified deployment and access interface in the most user-friendly form possible
- Also provides interfaces for calling other non-LLM services, such as scientific tools, vector databases, MCP, and so on

From answer to action: HaiDDF

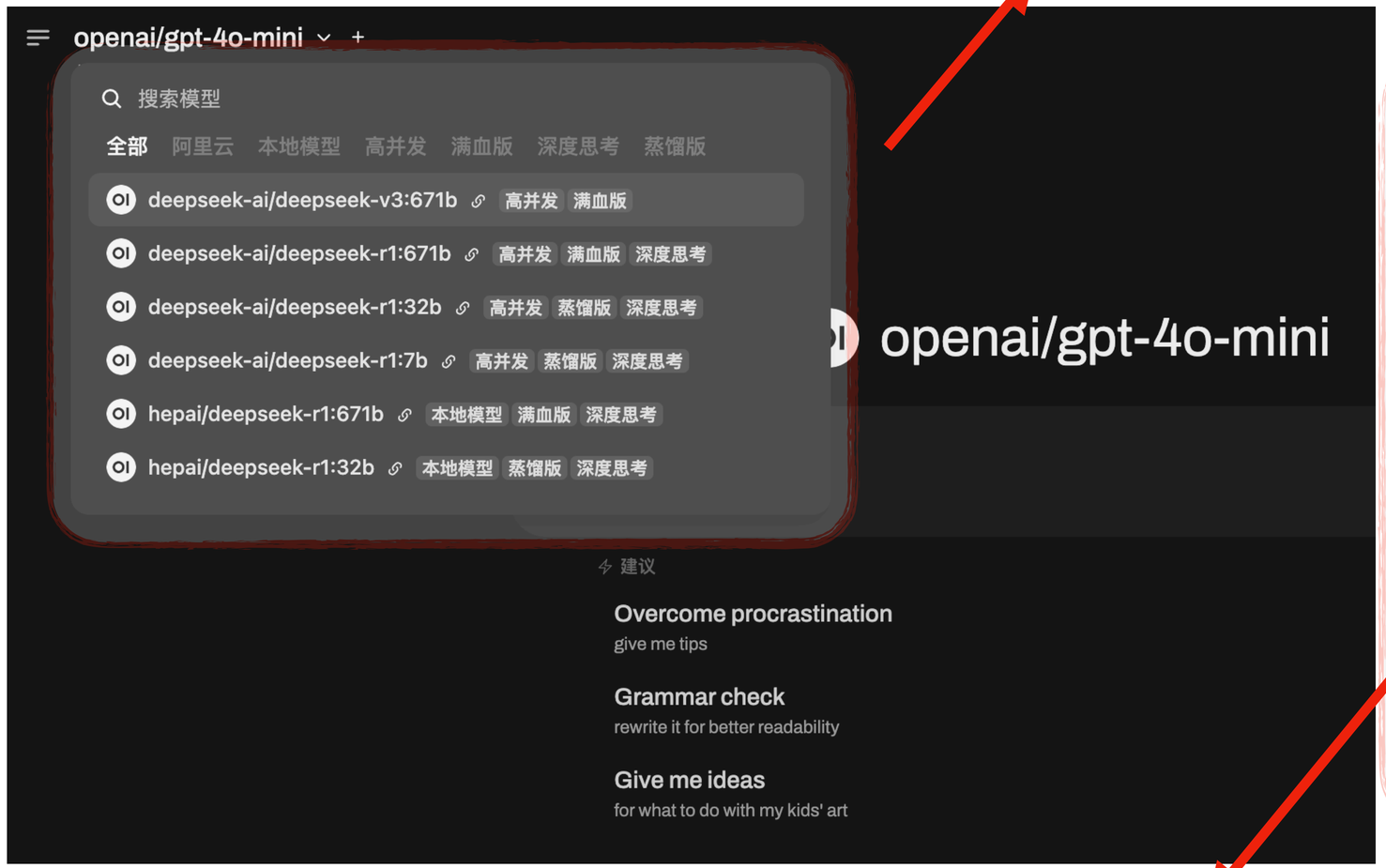
Actuators



Application of HaiDDF

Actuators

The models on **HaiChat** (https://haichatv3.ihep.ac.cn/) are deployed/accessed through HaiDDF

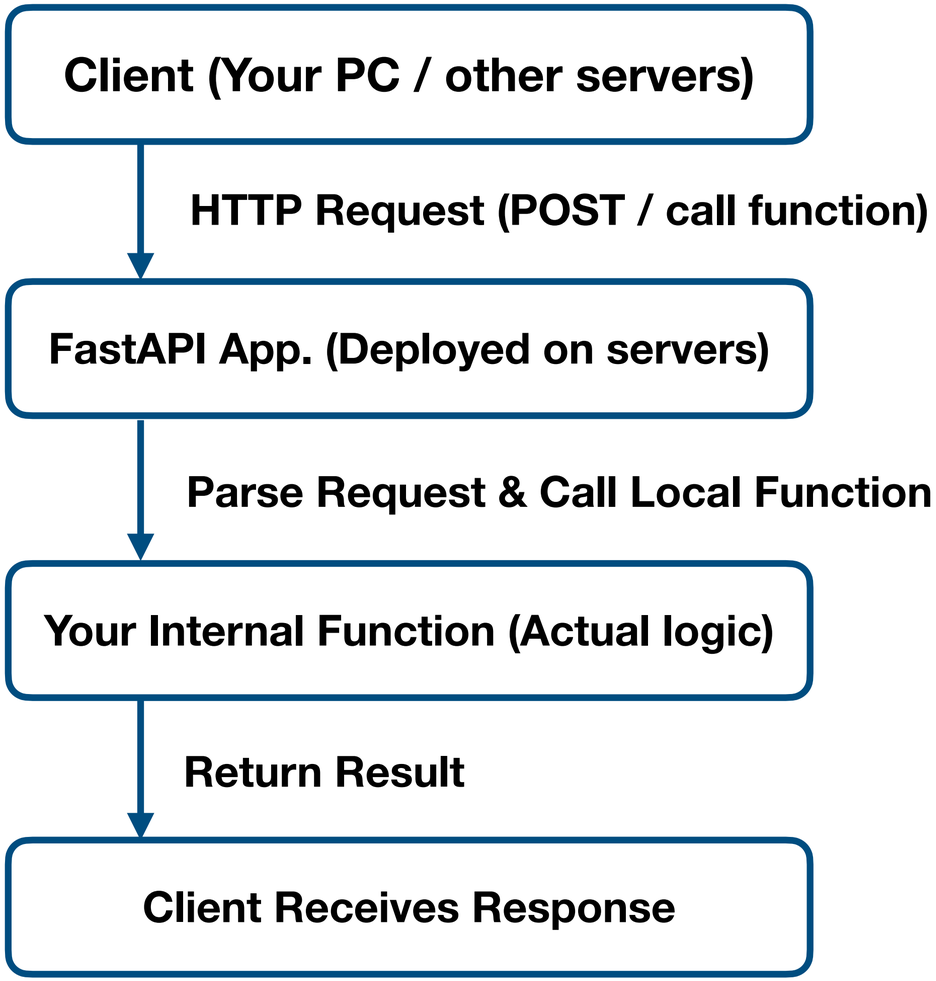


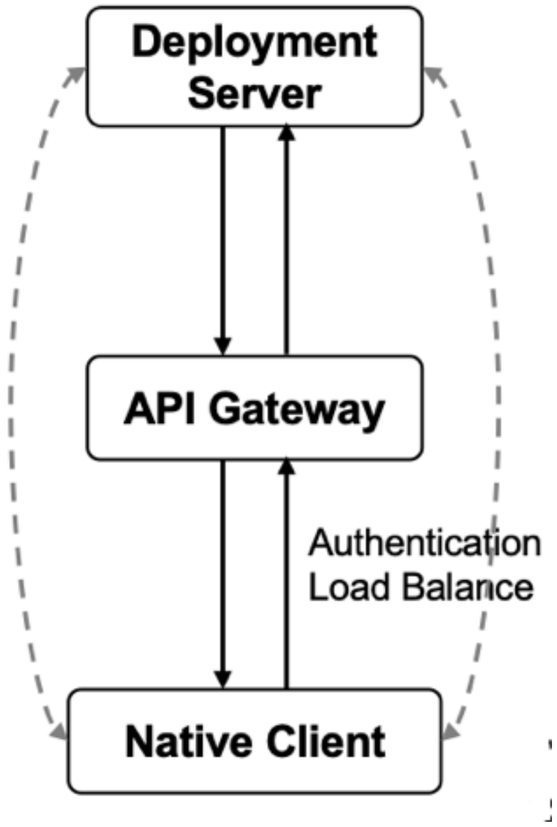

```
Model(id='ark/doubao-vision-pro', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen-max-latest', created=None, object='model', owned_by=None)
Model(id='aliyun/qwq-plus-latest', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen-coder-plus-latest', created=None, object='model', owned_by=No
Model(id='deepseek-ai/deepseek-v3:671b', created=None, object='model', owned_by=Non
Model(id='aliyun/qwen-vl-max-latest', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen2.5-vl-32b-instruct', created=None, object='model', owned_by=N
Model(id='aliyun/qwen-plus-latest', created=None, object='model', owned_by=None)
Model(id='ark/doubao-embedding-large', created=None, object='model', owned_by=None)
Model(id='openai/gpt-4o', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen2.5-vl-72b-instruct', created=None, object='model', owned_by=N
Model(id='aliyun/qwen-turbo-latest', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen3-30b-a3b', created=None, object='model', owned_by=None)
Model(id='openai/o1-mini', created=None, object='model', owned_by=None)
Model(id='deepseek-ai/deepseek-r1:7b', created=None, object='model', owned_by=None)
Model(id='deepseek-ai/deepseek-r1:32b', created=None, object='model', owned_by=None
Model(id='aliyun/qwen3-235b-a22b', created=None, object='model', owned_by=None)
Model(id='deepseek-ai/deepseek-r1:671b', created=None, object='model', owned_by=Non
Model(id='aliyun/qwen-vl-ocr-latest', created=None, object='model', owned_by=None)
Model(id='aliyun/qwen-long-latest', created=None, object='model', owned_by=None)
Model(id='openai/gpt-4o-mini', created=None, object='model', owned_by=None)
Model(id='aliyun/qvq-max-latest', created=None, object='model', owned_by=None)
Model(id='openai/o1', created=None, object='model', owned_by=None)
Model(id='hepai/deepseek-r1:671b', created=None, object='model', owned_by=None)
Model(id='hepai/markitdown', created=None, object='model', owned_by=['xiongdb@ihep.
Model(id='hepai/deepseek-r1:32b', created=None, object='model', owned_by=None)
Model(id='hepai/mineru', created=None, object='model', owned_by=['xiongdb@ihep.ac.c
Model(id='hepai/bge-m3:latest', created=None, object='model', owned_by=None)
Model(id='hepai/bge-reranker-v2-m3:latest', created=None, object='model', owned_by=
```

Meantime, we also deploy many embedding models, PDF parsers, and other tools

Remote Model in HaiDDF

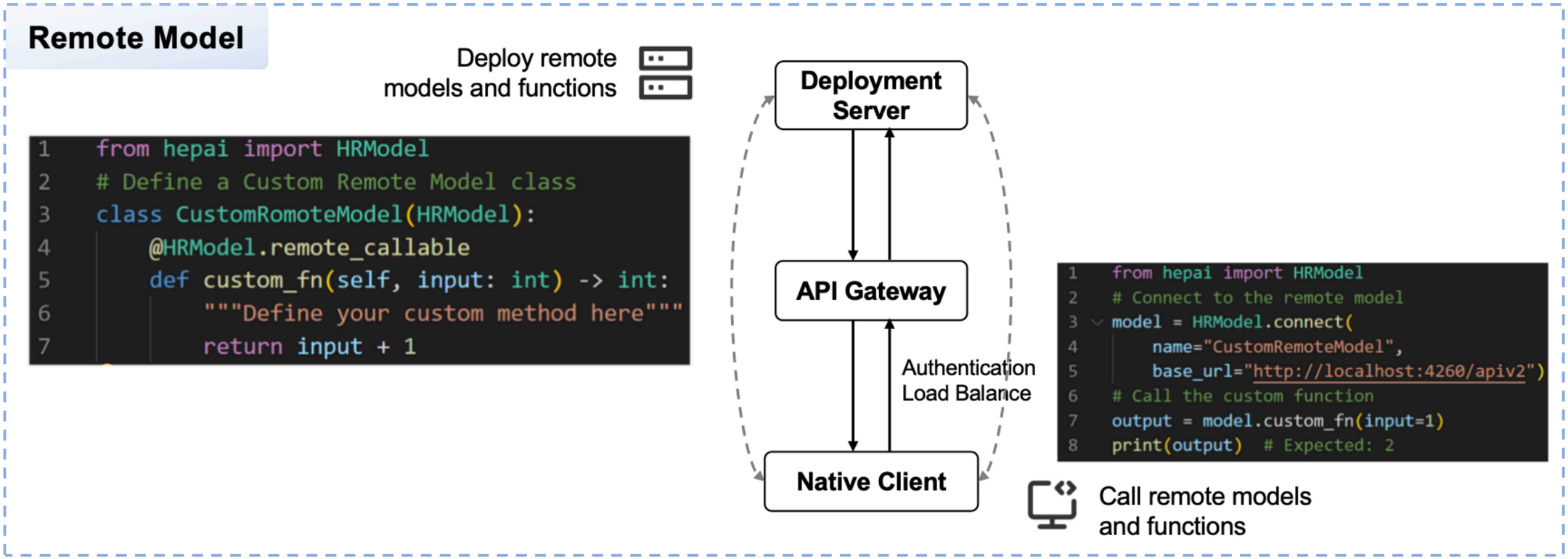
Actuators





Remote Model in HaiDDF

Actuators

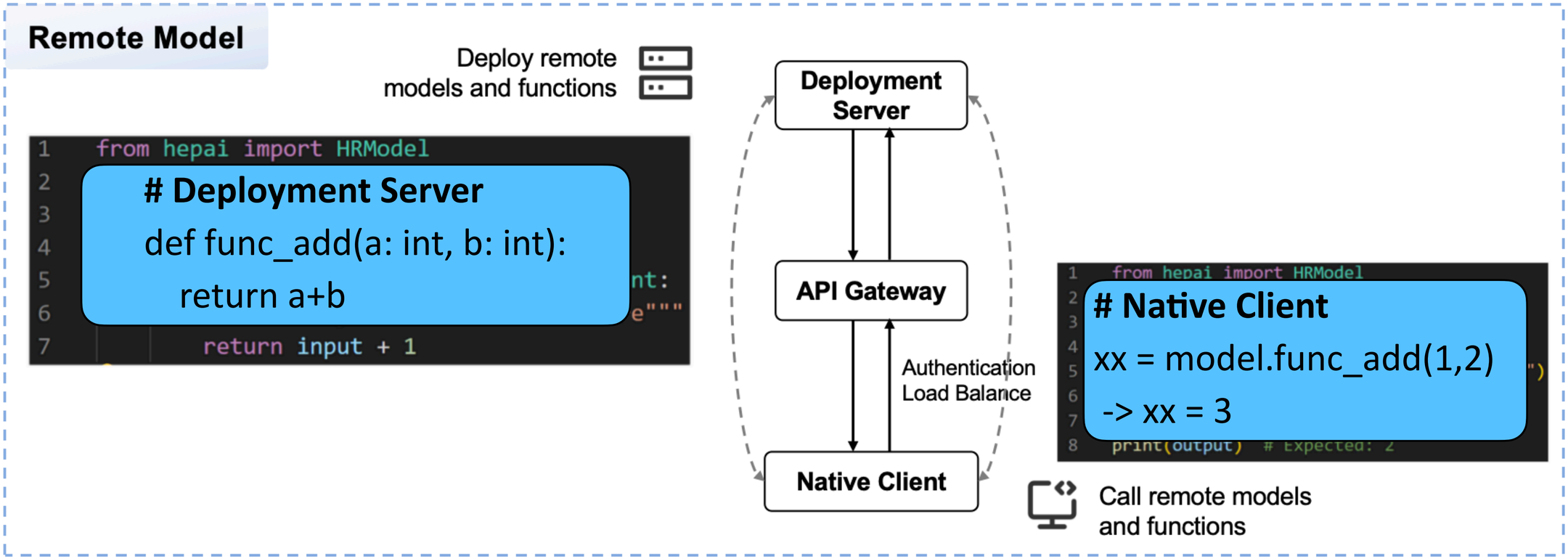


Remote Model is one of the unique features of HepAI frameworks

It enables low latency, distributed **calling of remote models** and **other software programs** by **deploying them to cloud servers** through **worker** and HepAI clients

Remote Model in HaiDDF

Actuators



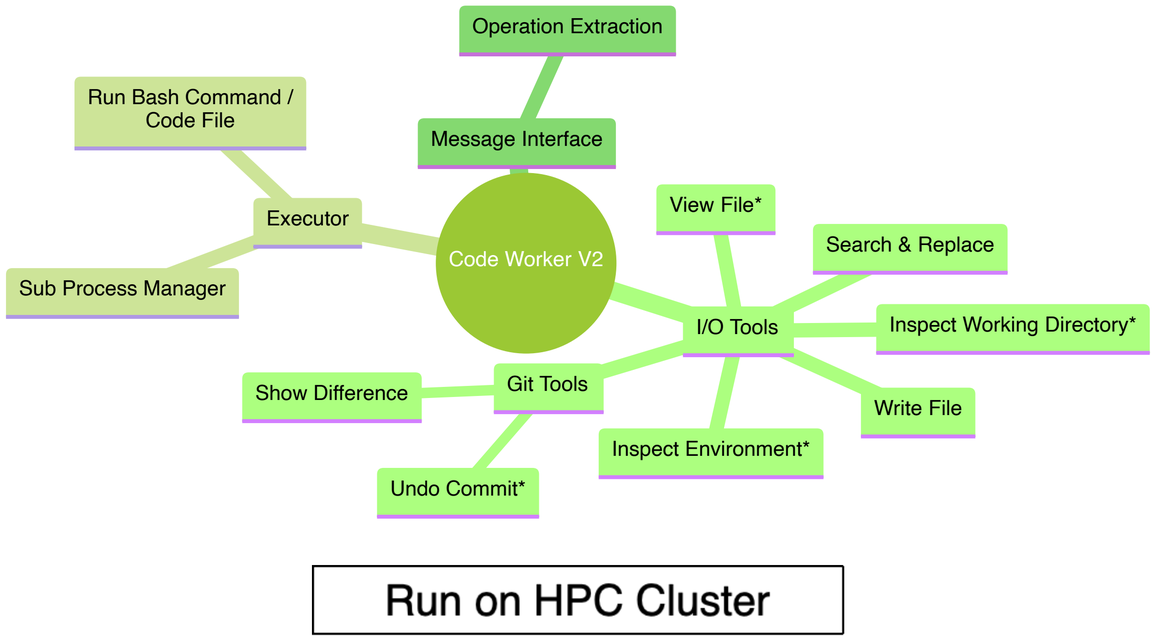
Remote Model is one of the unique features of HepAI frameworks

It enables low latency, distributed **calling of remote models** and **other software programs** by **deploying them to cloud servers** through **worker** and HepAI clients

BESIII code worker

- The subprocess management of the 1st worker version is developed with **Mingrun Li (李明润)**
- The initial interface design of the 2nd worker version is developed with **Xuliang Zhu (朱栩量)**

BESIII code worker



HaiDDF

Dr.Sai

- I/O Tools (files and data)
- Message interface
- Executor and Job management

From **Q&A chatbot** to **expert assistant**

BESIII code worker with no request

```
[2025-07-04 23:19:32,411] [hepai/code-worker-v2-BOSS-8] [INFO]: Starting loop...
[2025-07-04 23:19:32,516] [worker_app.py] [INFO]: Worker register successfully: `wk-40398a31-5b6`
WorkerInfo(id='wk-40398a31-5b6', type='common', network_info=WorkerNetworkInfo(host='0.0.0.0', port=42900, route_prefix='/apiv2',
host_name='aiboss001.ihep.ac.cn', worker_address='http://202.122.33.201:42900/apiv2'), resource_info=[ModelResourceInfo(model_name
='hepai/code-worker-v2-BOSS-8', model_type='common', model_version='1.0', model_description='This is a demo worker of HEP AI frame
work (HepAI)', model_author=None, model_onwer=['liaoyp@ihep.ac.cn'], model_groups=['default'], model_users=['liaoyp@ihep.ac.cn'],
model_functions=['__call__', 'get_dict', 'get_float', 'get_int', 'get_list', 'get_stream', 'hello_world', 'inspect_environment', '
inspect_system', 'interface', 'list_callable_functions', 'print_function_args', 'run_command', 'search_replace', 'write_code'])],
status_info=WorkerStatusInfo(speed=1, queue_length=0, status='ready'), check_heartbeat=True, last_heartbeat=None, vserion='2.0', m
etadata={})
INFO:     Started server process [1235764]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:42900 (Press CTRL+C to quit)
[2025-07-04 23:20:32,523] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^@[2025-07-04 23:21:32,531] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^@[2025-07-04 23:22:32,539] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^@[2025-07-04 23:23:32,545] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^@[2025-07-04 23:24:32,553] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^@[2025-07-04 23:25:32,560] [worker_app.py] [INFO]: Heartbeat sent successfully: `wk-40398a31-5b6`
^CINFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [1235764]
Stop worker successful. res: {"id":"wk-40398a31-5b6","stopped":true,"message":"Worker: `wk-40398a31-5b6` stopped.","shutdown":fals
e}
```

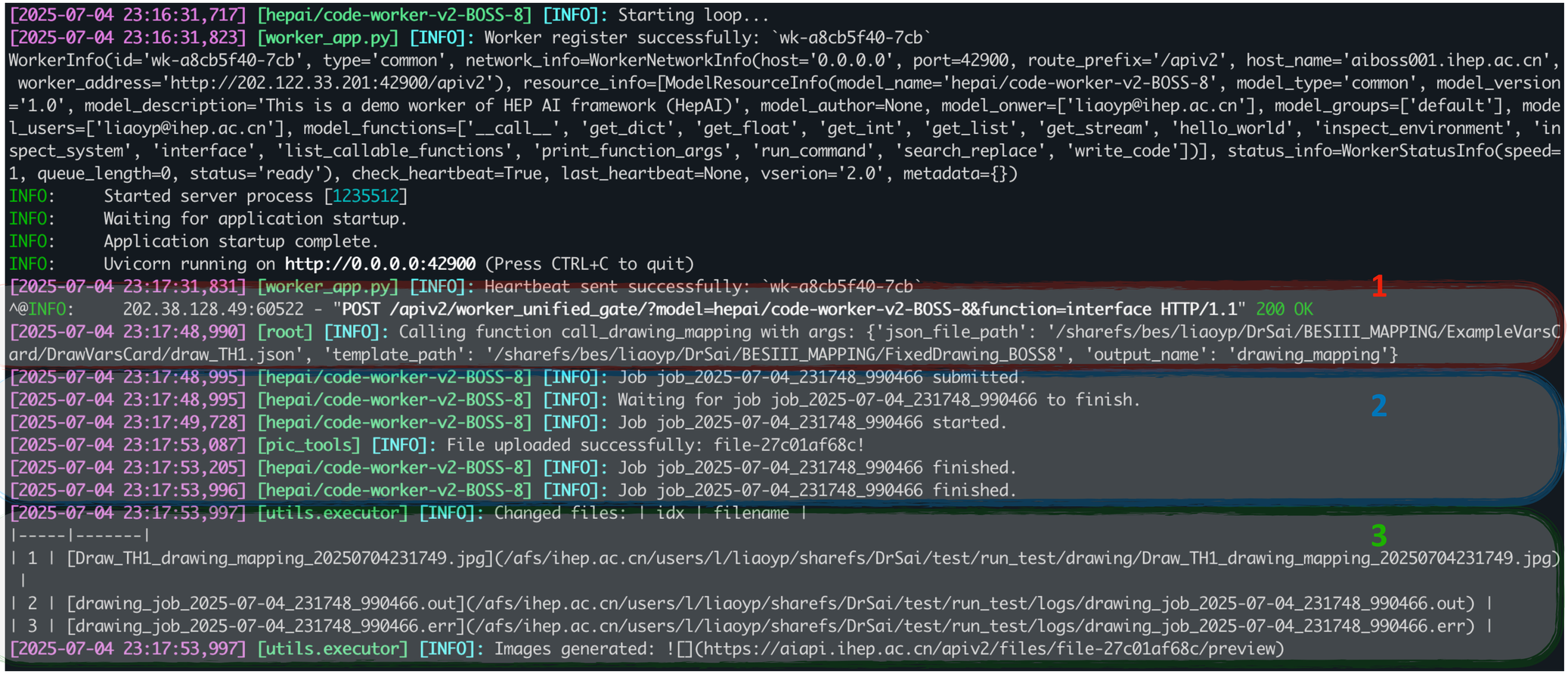
1 Worker config

2 Server information

3 Heartbeat for process maintenance

4 Close Worker

BESIII code worker with a request



Request

Job process

Output and file changes

Remote callable functions in BESIII code worker

```
Available functions by interface in hepai/code-worker-v2-BOSS-8:
- view_file
- write_file
- search_replace
- inspect_environment
- inspect_working_directory
- run_command
- execute_code_file
- call_algorithm_mapping_BOSS7
- call_algorithm_mapping
- call_joboption_mapping
- call_drawing_mapping
- call_getting_branch_name
- call_optimizer_fom_mapping
- call_optimizer_tmva_mapping
- call_boss_jobs_query
- call_merge_and_print_files
- call_kill_boss_job

You can also call the following functions directly:
- inspect_system
- inspect_environment
- write_code
- search_replace
- run_command
- get_stream
- list_callable_functions()
- print_function_args(function: str)
```

Common functions with IO

Special functions for BESIII analysis automation

- Now support BOSS-8.0.0-pre2 for nominal running
- Currently, please do not use or easily trust any physical results
- Thanks to Mingrun for technical support

HTCondor job management

At present, BESIII code worker provides a series of **common** and **BESIII-specific** remote callable functions

Quick start of BESIII code worker

The BESIII code worker heavily relies on the IHEP cluster and the BESIII Offline Software System (BOSS); therefore, a unified approach to maintain, update, and start the worker is highly needed.

Feature	Description
One-Click startup	Compile with **PyInstaller** or **Docker** to eliminate environment issues
Upgrade	Template files → modify without restarting worker Core logic → recompile, replace executable, restart to activate
Permission	User control via HepAI group permissions; grant access to other users as required

Tool calling or Skills?

Two ways to enhance the domain ability of agents: **Tool-calling** and **Skills**

Tool-calling

- Provide tool descriptions to Agent (brief intro, input args, output format...)
- LLM organizes original output to form necessary args
- **Better control over output** 😄
- Need to design the functions manually 😕
- Lack of Proactivity & Scalability 😢

OpenClaw Skills

- Knowledge documents (Markdown) & instructions
- LLM reads guide, decides actions
- **Agent auto-searches, installs & learns new skills** 🤩
- **Only skill names loaded; full content on-demand** 😀
- Question with Security & hallucinations ❓

In the end

- **Dr.Sai-BESIII** has many components, and **BESIII code worker** serves as the main **actuator** of Dr.Sai-BESIII
  - Remote callable functions service based on **HepAI-DDF (HaiDDF)**
- When design an agent or MAS, you may need to build your own worker to deal with the experiment-related problems
- Tool calling or Skills, you should make a balance!
- For more details, please find **tutorials** of HepAI
  - Open-Dr.Sai docs: https://docs-drsai.ihep.ac.cn/
  - HepAI API service: https://note.ihep.ac.cn/s/cZptfF8r9

Hands-On


docs-drsai.ihep.ac.cn
1
Open Dr. Sai 文档
OpenDrSai智能体/多智能体系统开发指南
HepAI AI平台开发指南
2
搜索文档...
模型/工具上云
模型云服务化
3
访问云模型
在线文件系统
简体中文
如何将AI模型部署成可通过API访问的云服务?
如何将AI模型部署成可通过API访问的云服务?
一、HepAI分布式部署框架如何工作?
二、如何实现远程模型?
1 快速开始
2 完整代码示例
3 Worker监控和API接口
4 Q&A:
HepAI平台提出了无限函数（IF, Infinite Function）协议，并基于此协议构建了自研的分布式部署框架（HepAI-DDF）。该框架实现了异构算力的统一部署与智能负载均衡，为AI模型、科学工具、科学数据、智能体等资源云化提供核心技术支撑。
页面导航
如何将AI模型部署成可通过API访问的云服务?
一、HepAI分布式部署框架如何工作?
二、如何实现远程模型?
1 快速开始
2 完整代码示例
3 Worker监控和API接口
4 Q&A:
一、HepAI分布式部署框架如何工作?
应用层
HaiChat 聊天机器人
"赛博士" AI智能体
天文警报信息汇集平台
...
HepAI-DDF 客户端/Http请求
pip install hepai

Hands-On

See scripts in: https://code.ihep.ac.cn/liaoypkirk/worker-handson

Git clone git@code.ihep.ac.cn:liaoypkirk/worker-handson.git

Thank you for listening!

Yipu Liao (廖一朴)
IHEP, CAS, Beijing
liaoyp@ihep.ac.cn