OpenLLM, and everything about running LLMs in production

BentoML, 2023

aarnphm@bentoml.com

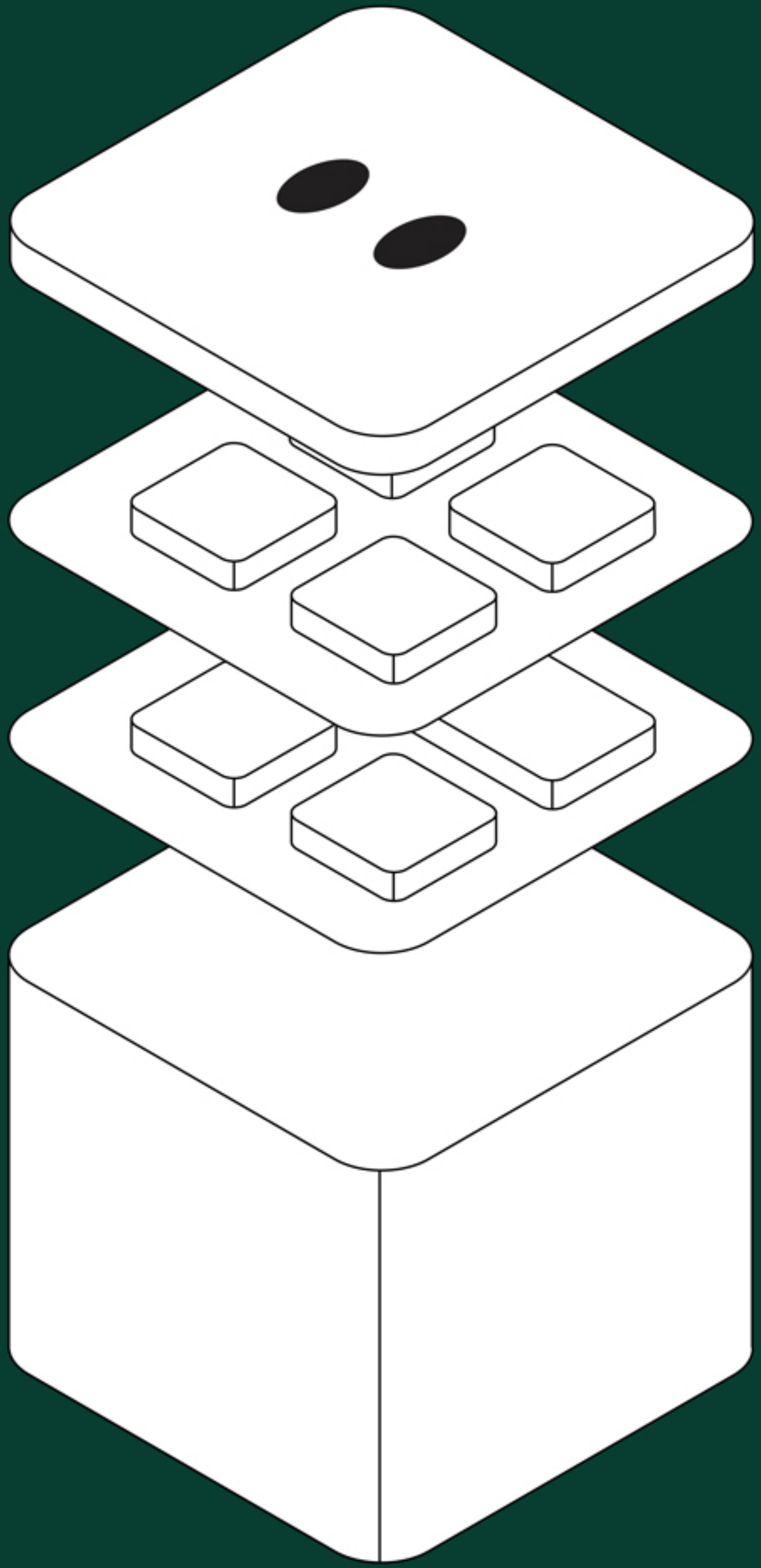
Introduction

Aaron Pham

at aarnphm[_] everywhere

Hobbies: Rock climbing and reading

BentoML

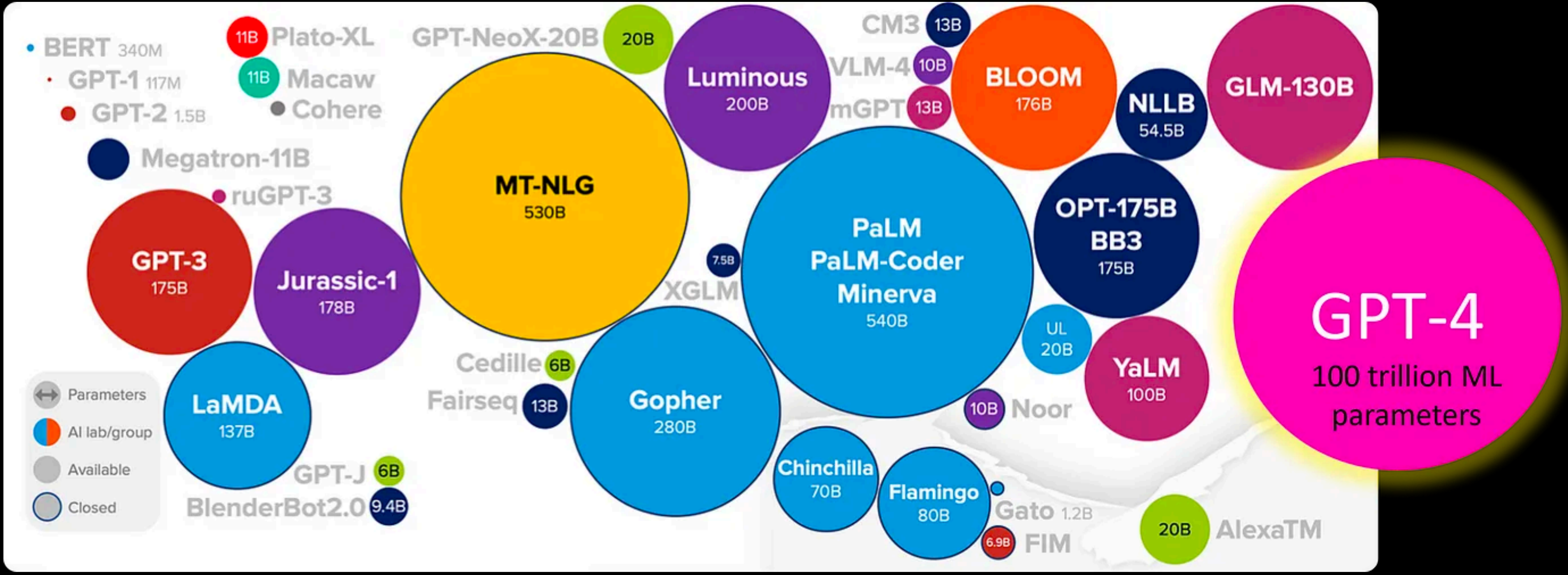


BentoML

www.bentoml.com

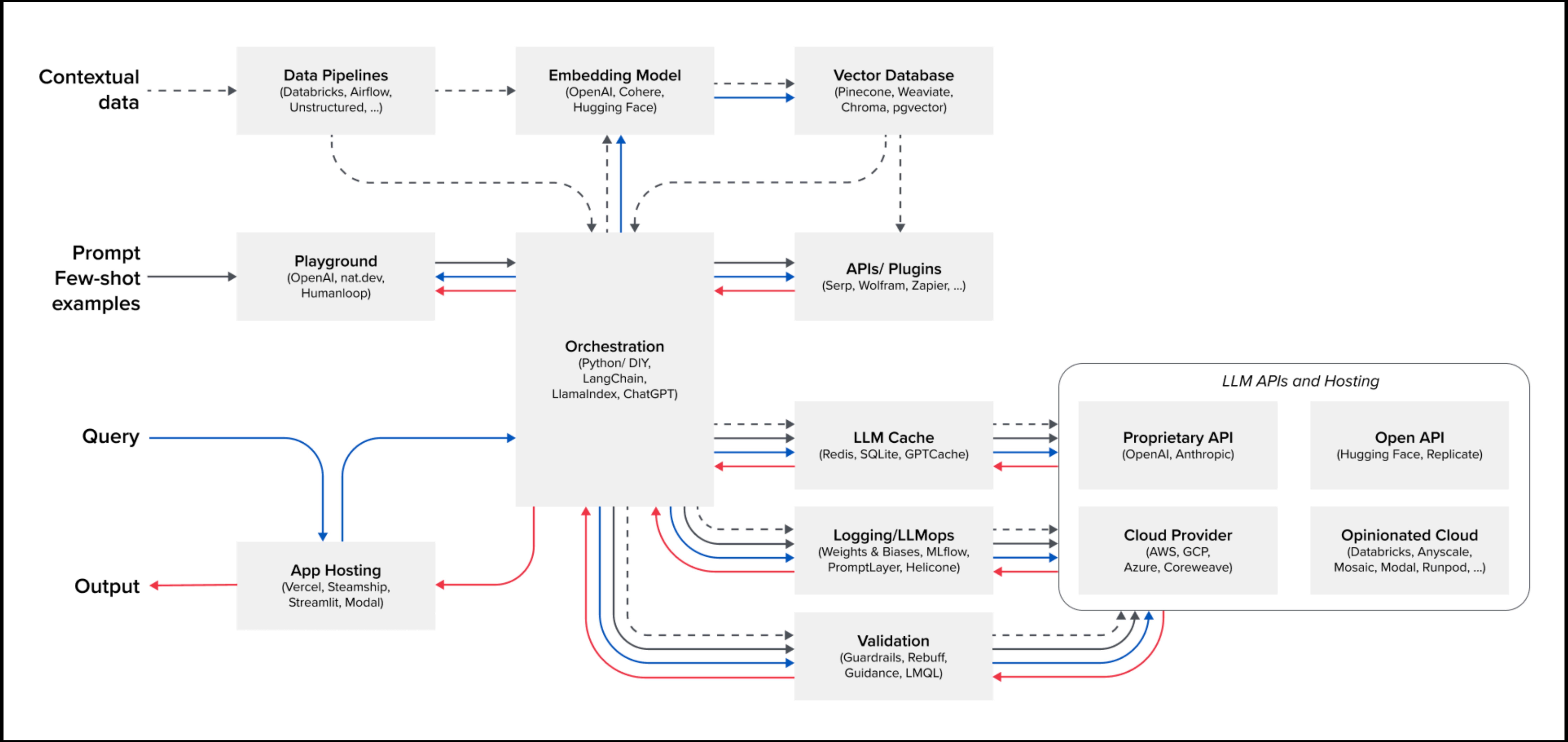
Freedom
to build.

AIGC BOOM!



BERT 340M
GPT-1 117M
GPT-2 1.5B
11B Plato-XL
11B Macaw
Cohere
Megatron-11B
ruGPT-3
GPT-NeoX-20B 20B
MT-NLG 530B
Luminous 200B
CM3 13B
VLM-4 10B
mGPT 13B
BLOOM 176B
NLLB 54.5B
GLM-130B
GPT-3 175B
Jurassic-1 178B
7.5B XGLM
PaLM PaLM-Coder Minerva 540B
OPT-175B BB3 175B
UL 20B
YaLM 100B
10B Noor
Cedille 6B
Fairseq 13B
Gopher 280B
LaMDA 137B
GPT-J 6B
BlenderBot2.0 9.4B
Chinchilla 70B
Flamingo 80B
Gato 1.2B
6.9B FIM
20B AlexaTM
Parameters
AI lab/group
Available
Closed
GPT-4
100 trillion ML parameters

Emerging LLMs stack (cred. a16z)

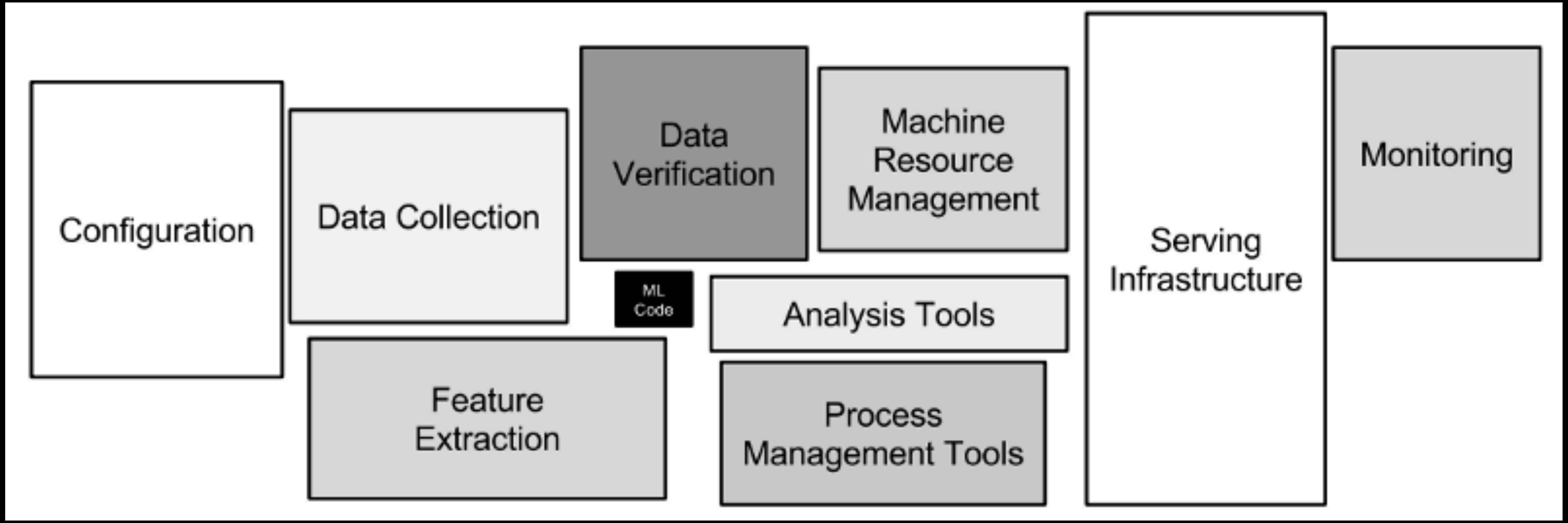

Contextual data
Data Pipelines (Databricks, Airflow, Unstructured, ...)
Embedding Model (OpenAI, Cohere, Hugging Face)
Vector Database (Pinecone, Weaviate, Chroma, pgvector)
Prompt Few-shot examples
Playground (OpenAI, nat.dev, Humanloop)
Orchestration (Python/ DIY, LangChain, LlamaIndex, ChatGPT)
APIs/ Plugins (Serp, Wolfram, Zapier, ...)
Query
Output
App Hosting (Vercel, Steamship, Streamlit, Modal)
LLM Cache (Redis, SQLite, GPTCache)
Logging/LLMops (Weights & Biases, MLflow, PromptLayer, Helicone)
Validation (Guardrails, Rebuff, Guidance, LMQL)
LLM APIs and Hosting
Proprietary API (OpenAI, Anthropic)
Open API (Hugging Face, Replicate)
Cloud Provider (AWS, GCP, Azure, Coreweave)
Opinionated Cloud (Databricks, Anyscale, Mosaic, Modal, Runpod, ...)

Motivation

Running ML in production is... hard

Running LLM in production is even harder!

- Consistency/Hallucination
- Infrastructure Complexity
- Security/Compliance
- Bleeding-edge ecosystem
- Maintainability
- And more



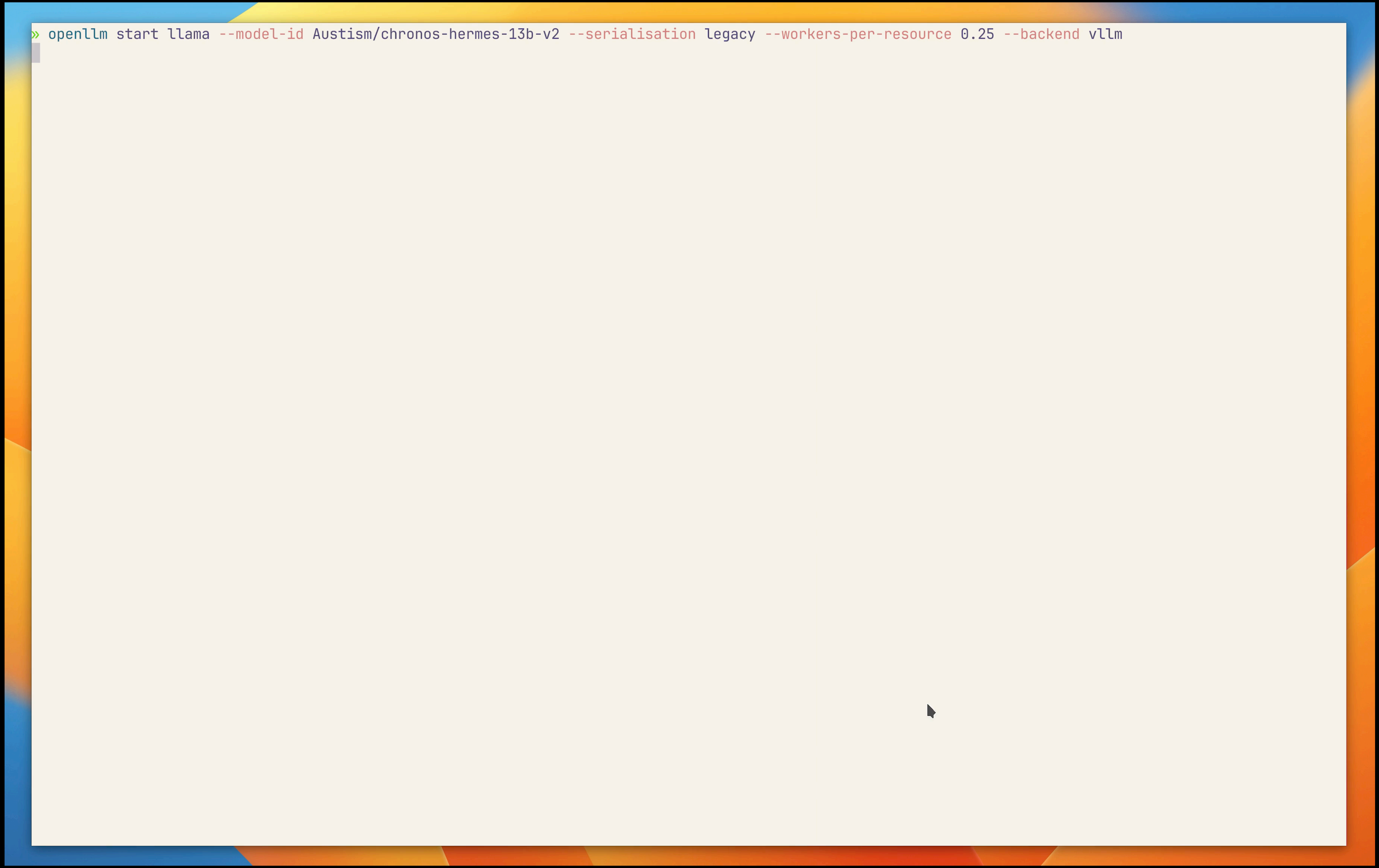
https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

BentoML

Hosting your own LLMs?

- Customisation/Flexibility
- Security
- Accuracy improvement
- Cost efficiency
- Offline access

Introducing, OpenLLM



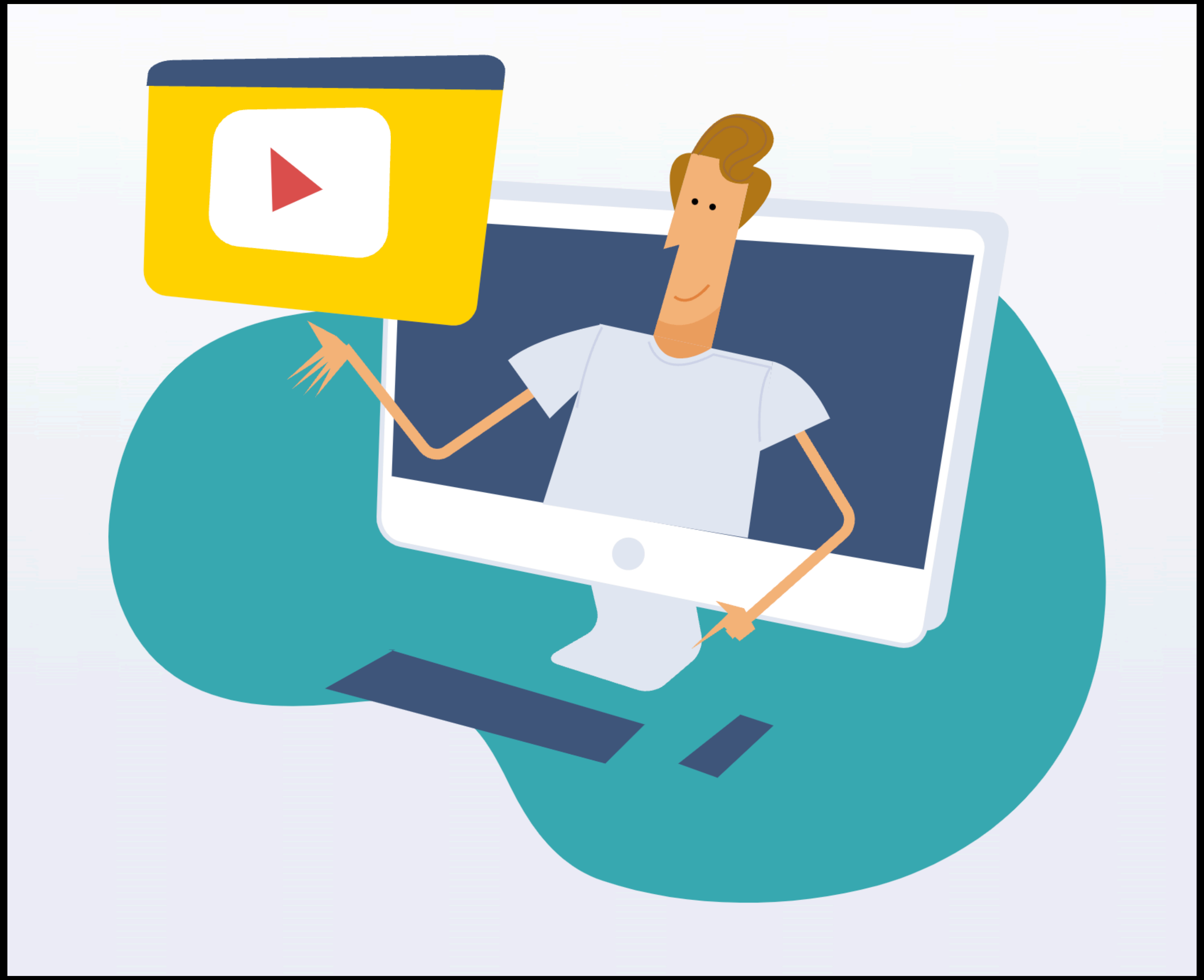
```
» openllm start llama --model-id Austism/chronos-hermes-13b-v2 --serialisation legacy --workers-per-resource 0.25 --backend vllm
```

BentoML

Demo time!

- LLM Inference

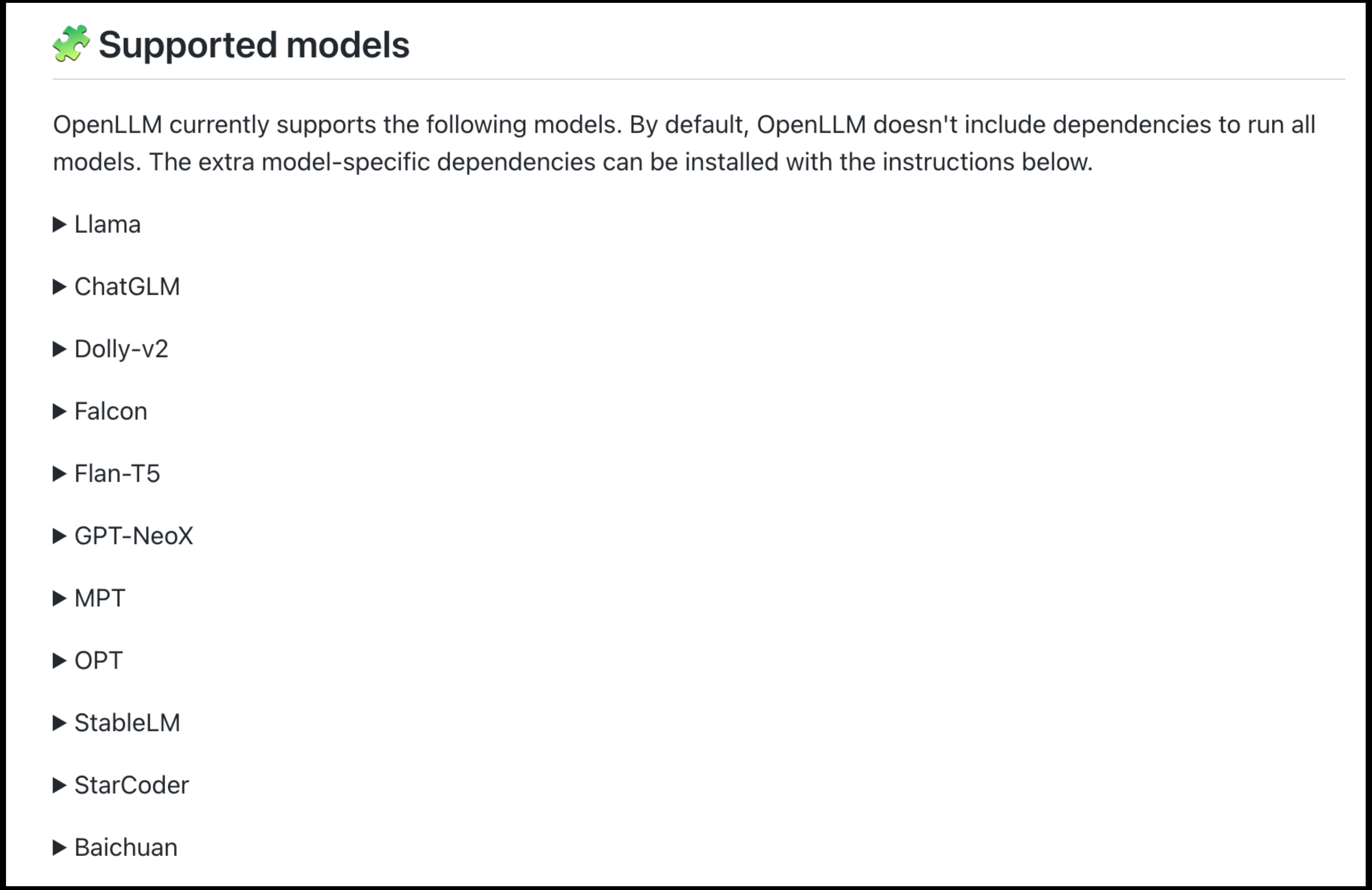
- Fine tuning and serve Llama with QLoRA



BentoML

Open-source LLMs

Supports a wide range of architectures and runtime, but not limited to Llama, StableLM, ChatGLM, StarCoder, and more.



🧩 Supported models

OpenLLM currently supports the following models. By default, OpenLLM doesn't include dependencies to run all models. The extra model-specific dependencies can be installed with the instructions below.

▶ Llama

▶ ChatGLM

▶ Dolly-v2

▶ Falcon

▶ Flan-T5

▶ GPT-NeoX

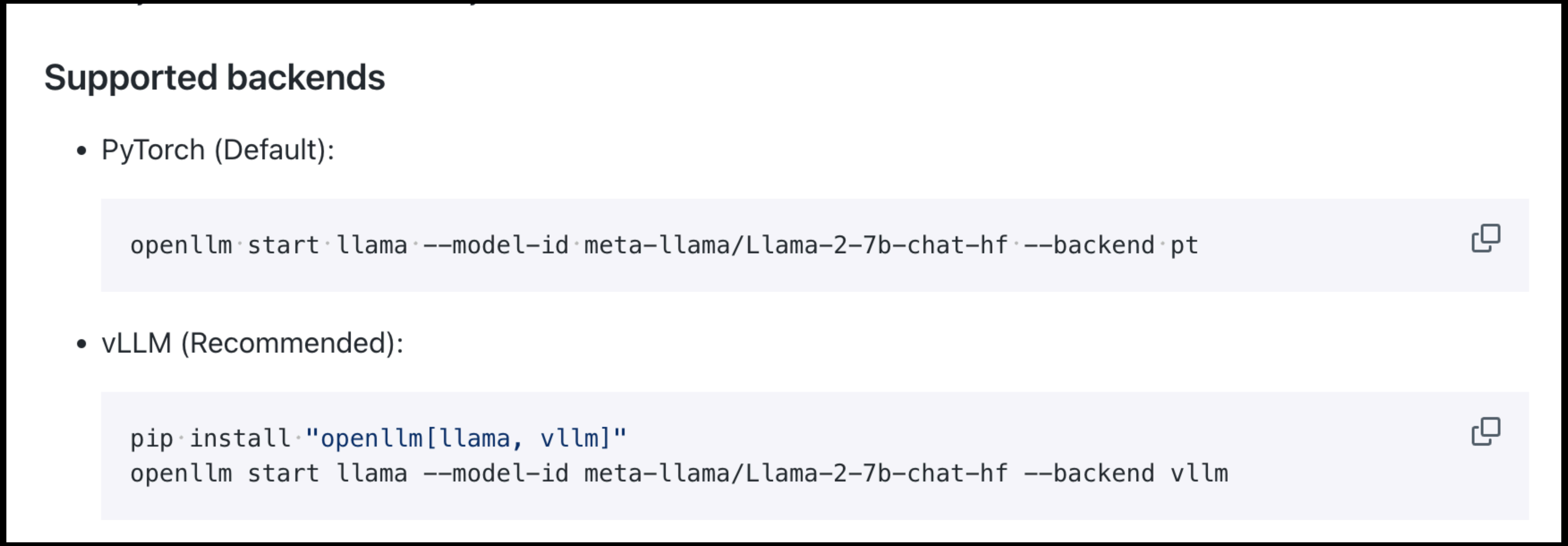
▶ MPT

▶ OPT

▶ StableLM

▶ StarCoder

▶ Baichuan



Supported backends

- PyTorch (Default):

```
openllm start llama --model-id meta-llama/Llama-2-7b-chat-hf --backend pt
```

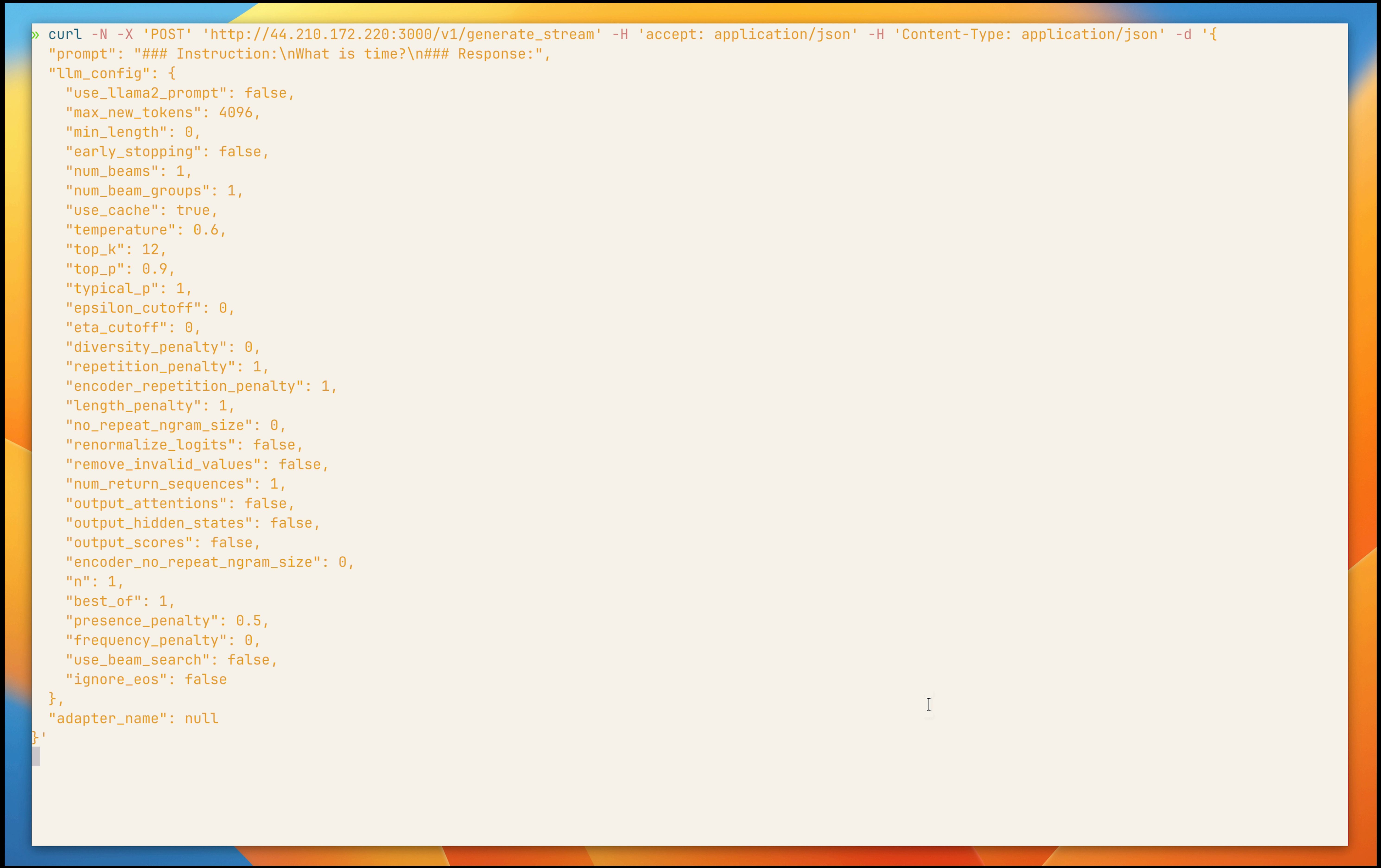
- vLLM (Recommended):

```
pip install "openllm[llama, vllm]"
openllm start llama --model-id meta-llama/Llama-2-7b-chat-hf --backend vllm
```

BentoML

Built-in optimisation

Token streaming via SSE



```
» curl -N -X 'POST' 'http://44.210.172.220:3000/v1/generate_stream' -H 'accept: application/json' -H 'Content-Type: application/json' -d '{
  "prompt": "### Instruction:\nWhat is time?\n### Response:",
  "llm_config": {
    "use_llama2_prompt": false,
    "max_new_tokens": 4096,
    "min_length": 0,
    "early_stopping": false,
    "num_beams": 1,
    "num_beam_groups": 1,
    "use_cache": true,
    "temperature": 0.6,
    "top_k": 12,
    "top_p": 0.9,
    "typical_p": 1,
    "epsilon_cutoff": 0,
    "eta_cutoff": 0,
    "diversity_penalty": 0,
    "repetition_penalty": 1,
    "encoder_repetition_penalty": 1,
    "length_penalty": 1,
    "no_repeat_ngram_size": 0,
    "renormalize_logits": false,
    "remove_invalid_values": false,
    "num_return_sequences": 1,
    "output_attentions": false,
    "output_hidden_states": false,
    "output_scores": false,
    "encoder_no_repeat_ngram_size": 0,
    "n": 1,
    "best_of": 1,
    "presence_penalty": 0.5,
    "frequency_penalty": 0,
    "use_beam_search": false,
    "ignore_eos": false
  },
  "adapter_name": null
}'
```

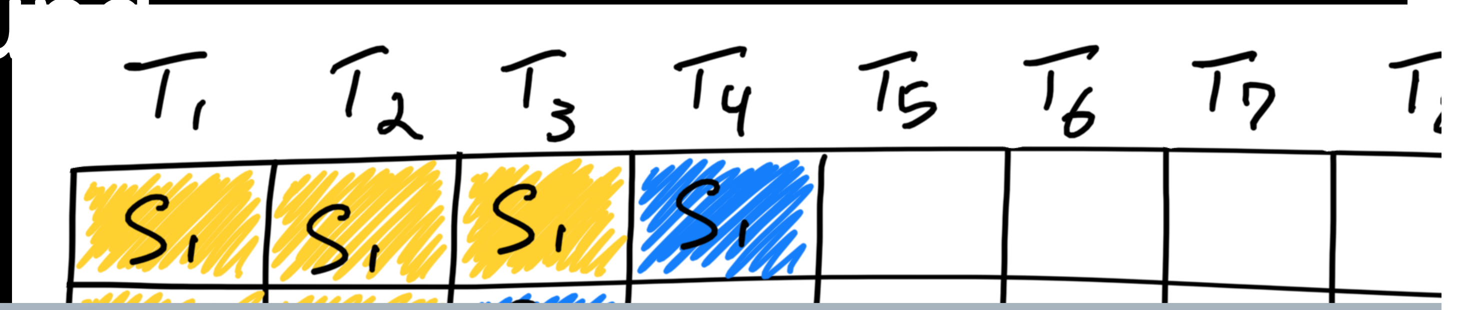
BentoML

Built-in optimisation

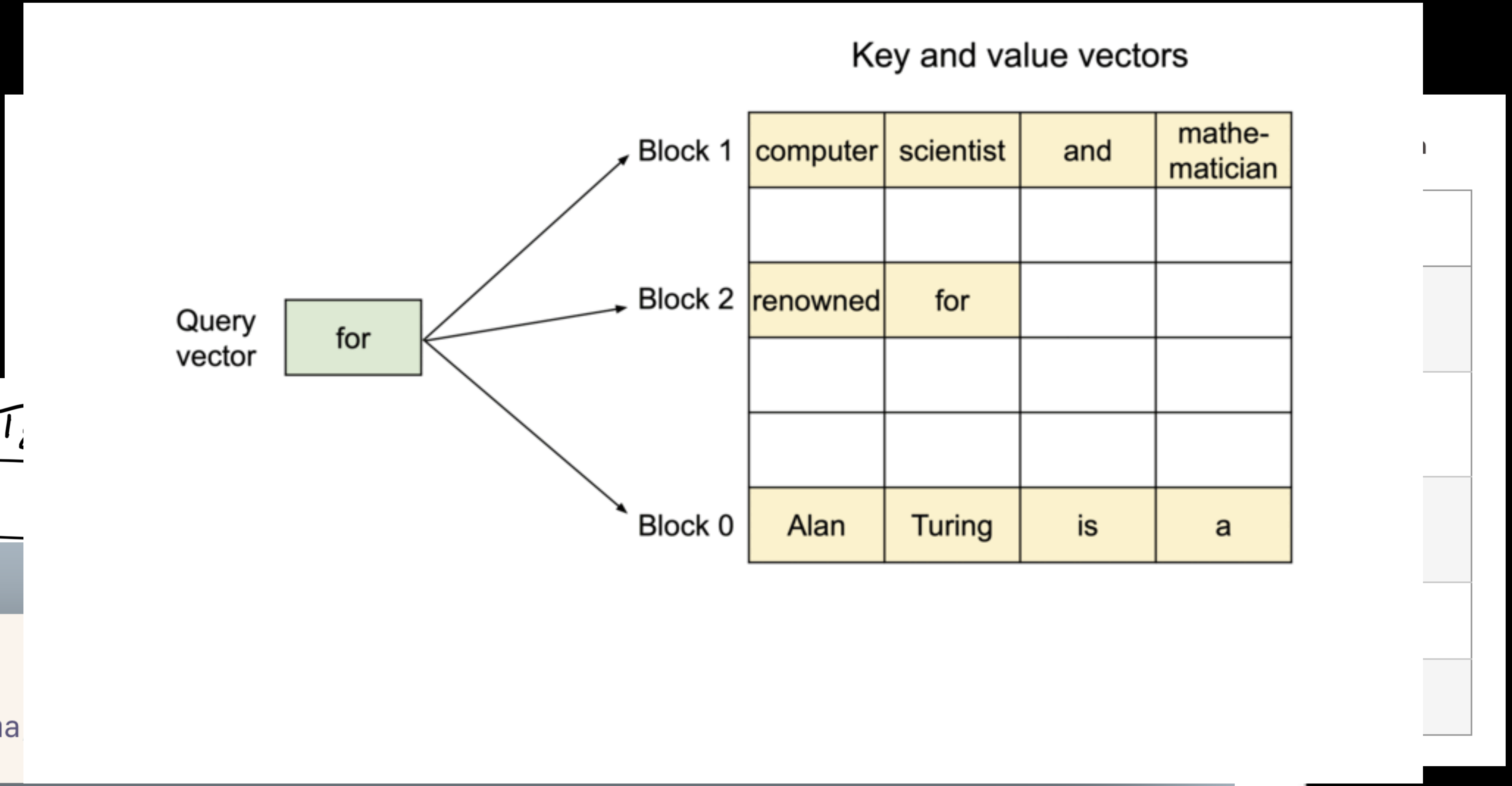
Continuous batching

- Naive batching
- LLM Inference
- Continuous batching
- Prefill ➡️ decode ➡️ past values
- PagedAttention
- IO Bou

Given the following prompt:
What is the capital of California:



```
openllm start llama --model-id meta-llama
```



https://docs.nvidia.com/deeplearning/performance/dl-pehttps://kipp.ly/t-background/index.html#understand-kv-perf-cache
https://hugginghttps://www.blog/anyscale/continuous-batching-llm-inference#sect2-present-first-10x-speedup

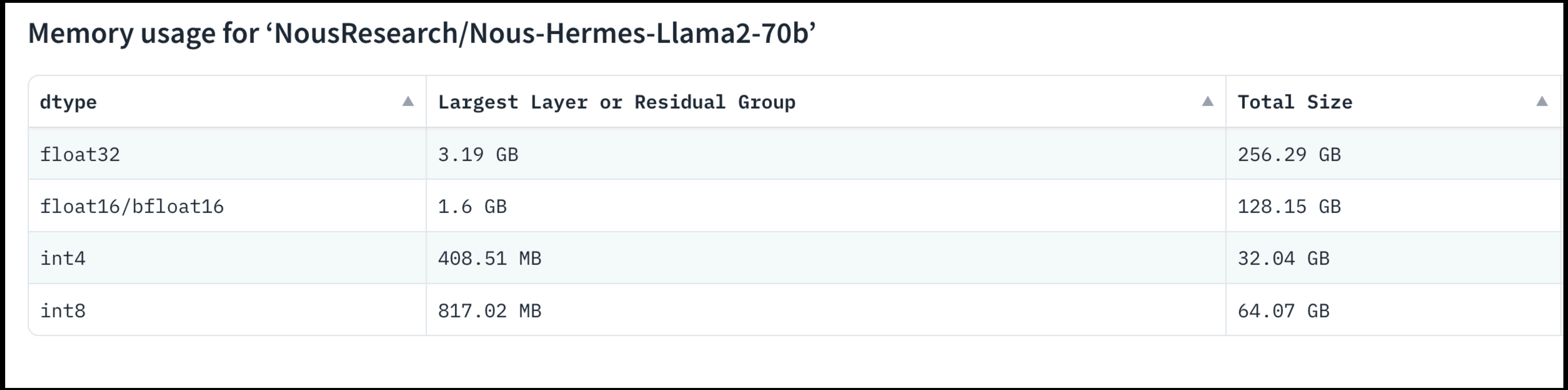
Built-in optimisation

Quantisation: GPTQ, kbit

1 MB: GPU Memory required for 1 token of output for 13B model (1 word ~= 1.3 tokens)

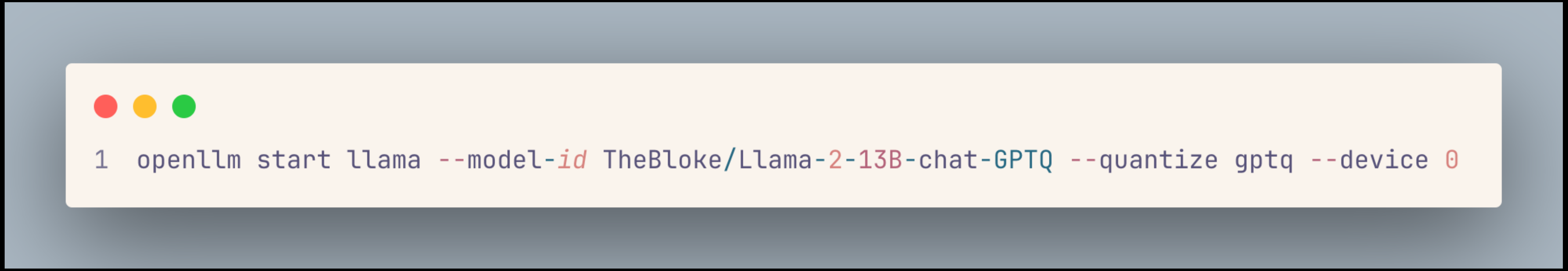
512 max new tokens = 512MB VRAM

- Reduce memory footprint
- Improve general throughput



Memory usage for 'NousResearch/Nous-Hermes-Llama2-70b'

dtype	Largest Layer or Residual Group	Total Size
float32	3.19 GB	256.29 GB
float16/bfloat16	1.6 GB	128.15 GB
int4	408.51 MB	32.04 GB
int8	817.02 MB	64.07 GB



```
openllm start llama --model-id TheBloke/Llama-2-13B-chat-GPTQ --quantize gptq --device 0
```

https://github.com/ray-project/llm-numbers?tab=readme-ov-file#gpu-memory

Why OpenLLM?

Specialities

- Built-in Inference Optimization with MQA, PagedAttention
- Quantization with GPTQ, k-bit
- Accelerators support with multi-GPUs deployments, TPUs
- Monitoring and evaluation
- Fine-tuning support with qLoRA and various tuning techniques
- Integration with AI tools like LangChain, HuggingFace Agents etc.

Powered by BentoML

- Package model files + dependencies + code into a Bento
- Bentos can be managed and versioned properly in a central place
- Automatically generate docker image for deployment
- Streamlined deployment process: batch inference, online-serving
- Flexible deployment strategy: Docker, Yatai + Kubernetes, bentoctl + Terraform, BentoCloud

BentoML

OpenLLM Roadmap

- System prompts
- Unified fine-tuning API for models
- Better CPU inference with GGUF
- Javascript/Typescript Client library
- OpenAI Compatible APIs
- Optimized modeling for Flash Attentions
- AWQ support, custom CUDA Kernels

BentoML

Thank you!

Q & A