Fast and scalable Inference for **GenAI workloads**

On YOUR Azure cloud. Or OURS.

Powered by Simplismart

Contact
soumyaa@simplismart.tech

Date
August 2025

MLOps Orchestration

with Simplismart

→

Fine-tune

Deploy

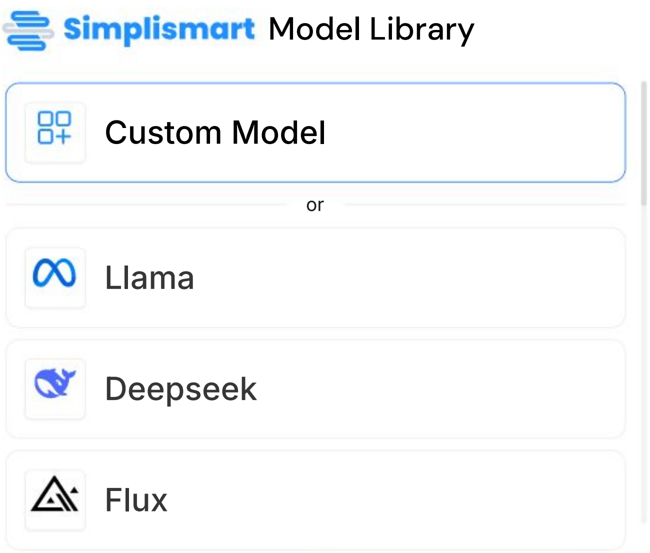
Inference

Monitoring

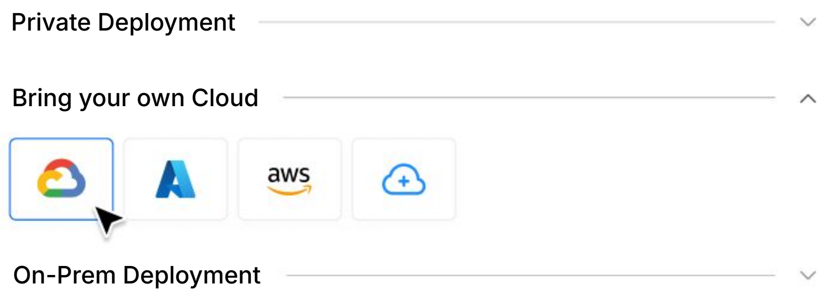
Benchmarking

Deploy Any Model via UI/SDK

Choose a model from our extensive model library



Choose any Cloud, yours or ours



Fastest Inference out of the box!

Pick Any Model or Import Custom Weights

Large Language Models

- Deepseek
- Llama
- Mistral

Diffusion Model

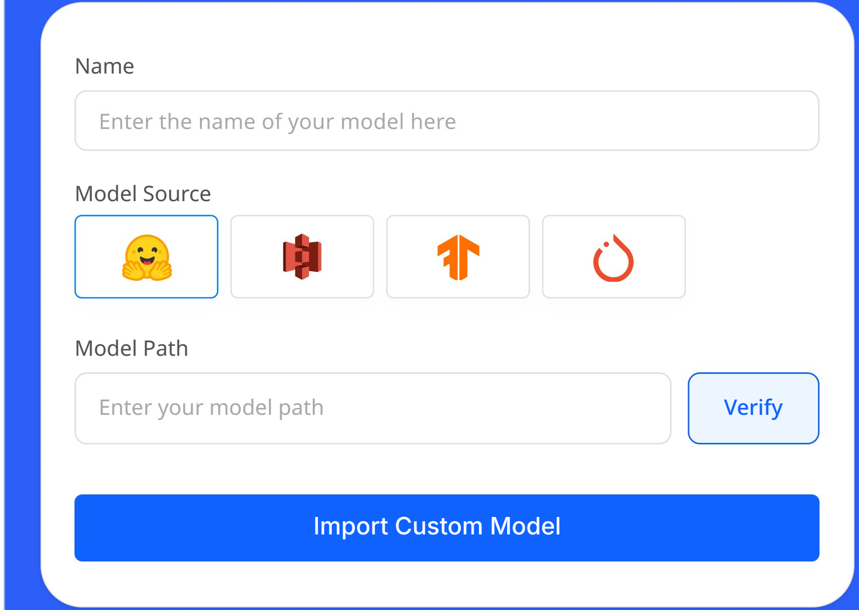
- FLUX
- SDXL
- SD

Speech-to-Text

- Whisper V3
- Whisper V2

Text-to-Speech

- MeloTTS
- Tortoise
- XTTS



Name

Enter the name of your model here

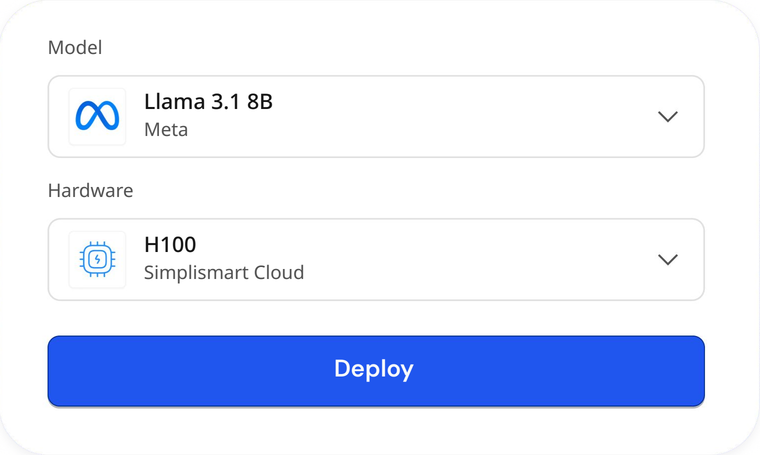
Model Source

Model Path

Enter your model path

Customize a deployment as you need

Deploy right out of the box



Model

Llama 3.1 8B
Meta

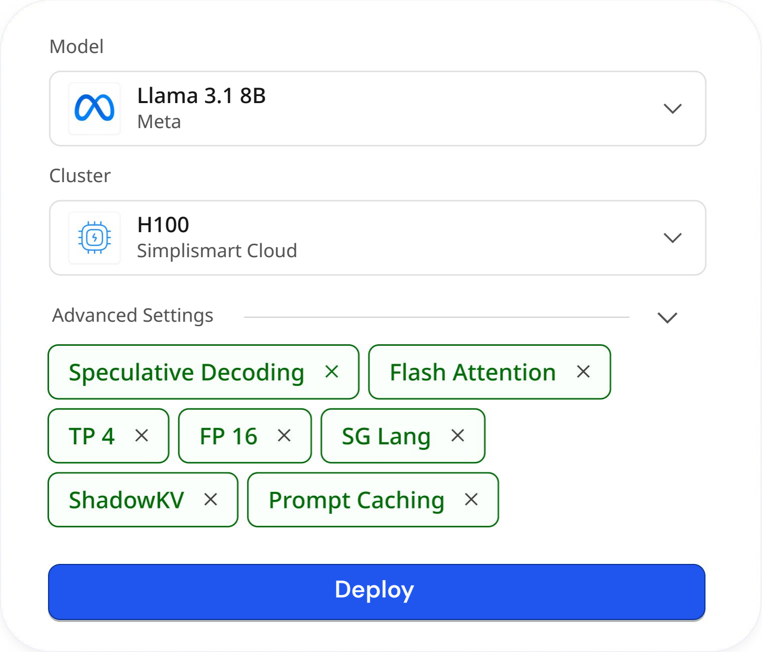
Hardware

H100
Simplismart Cloud

Deploy via Simplismart's proprietary engine with recommended settings

or

Customise as per your needs



Model

Llama 3.1 8B
Meta

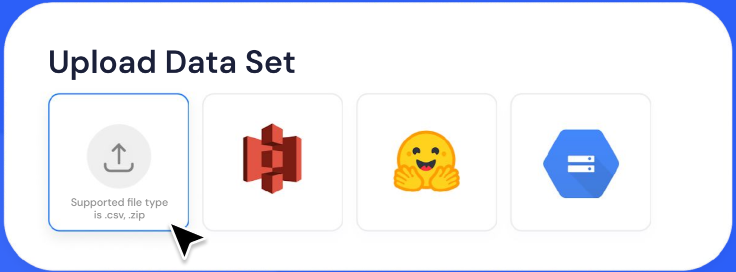
Cluster

H100
Simplismart Cloud

Advanced Settings

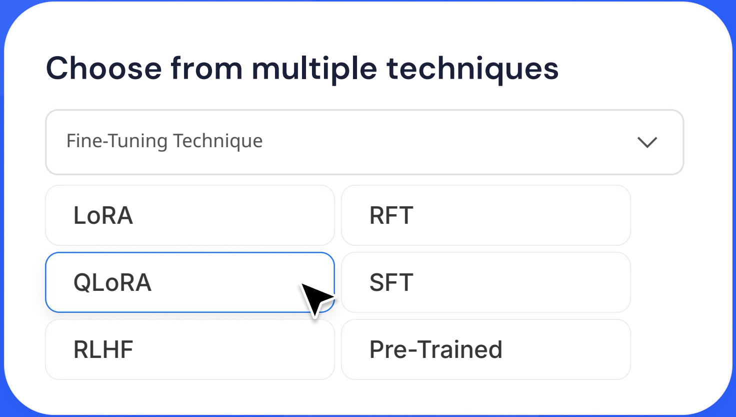
Speculative Decoding ×
Flash Attention ×
TP 4 ×
FP 16 ×
SG Lang ×
ShadowKV ×
Prompt Caching ×

Fine-Tuning



Upload Data Set

Supported file type is .csv, .zip



Choose from multiple techniques

Fine-Tuning Technique

LoRA

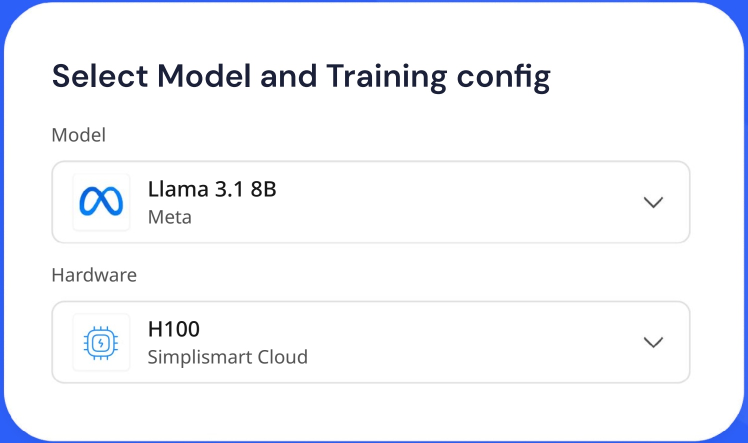
RFT

[✓ QLoRA]

SFT

RLHF

Pre-Trained



Select Model and Training config

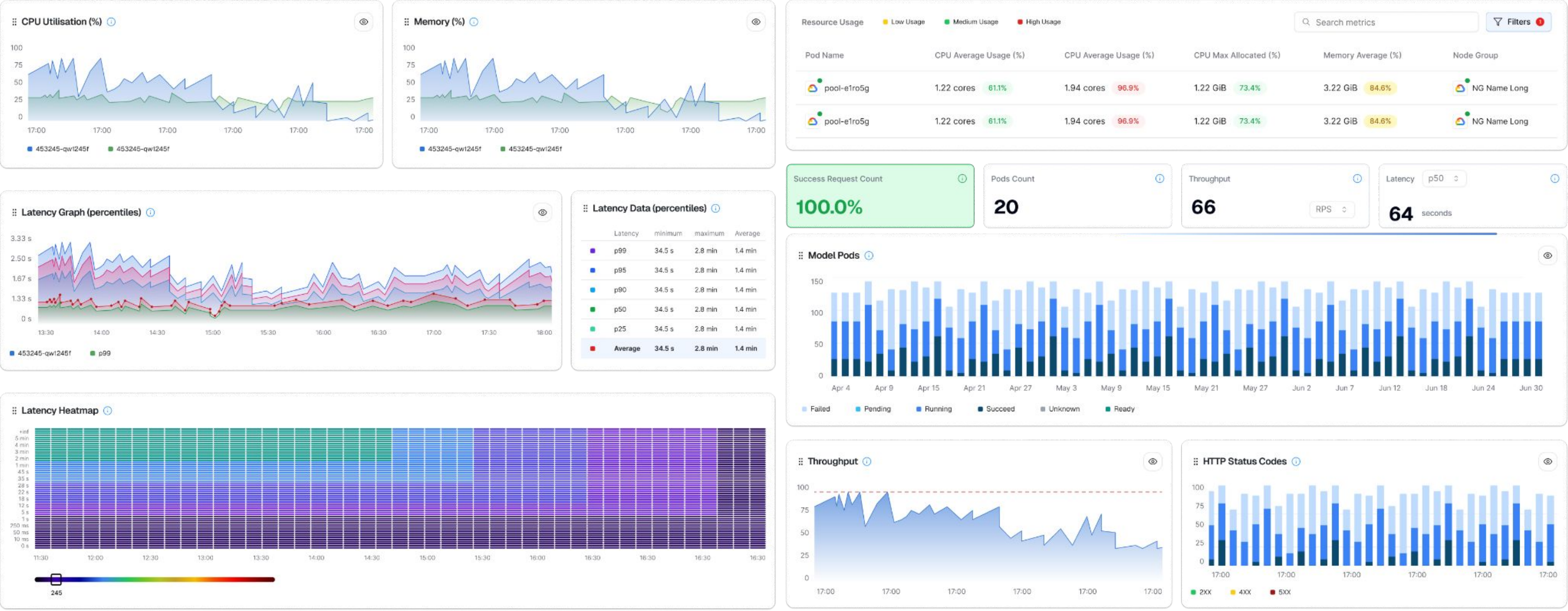
Model: Llama 3.1 8B, Meta

Hardware: H100, Simplismart Cloud

Distributed Training out of the box!

Built-in observability for SLA monitoring

Find real-time model-level and cluster health metrics



Latency Data (percentiles)

	Latency	minimum	maximum	Average
🟪	p99	34.5 s	2.8 min	1.4 min
🟦	p95	34.5 s	2.8 min	1.4 min
🟦	p90	34.5 s	2.8 min	1.4 min
🟩	p50	34.5 s	2.8 min	1.4 min
🟩	p25	34.5 s	2.8 min	1.4 min
🟥	**Average**	**34.5 s**	**2.8 min**	**1.4 min**

Resource Usage 🟨 Low Usage 🟩 Medium Usage 🟥 High Usage

Search metrics

Filters 1

Pod Name	CPU Average Usage (%)	CPU Average Usage (%)	CPU Max Allocated (%)	Memory Average (%)	Node Group
pool-e1ro5g	1.22 cores 61.1%	1.94 cores 96.9%	1.22 GiB 73.4%	3.22 GiB 84.6%	NG Name Long
pool-e1ro5g	1.22 cores 61.1%	1.94 cores 96.9%	1.22 GiB 73.4%	3.22 GiB 84.6%	NG Name Long

Success Request Count: 100.0%

Pods Count: 20

Throughput: 66 RPS

Latency p50: 64 seconds

Optimised Infra Control

with Simplismart

Rapid Auto-scaler

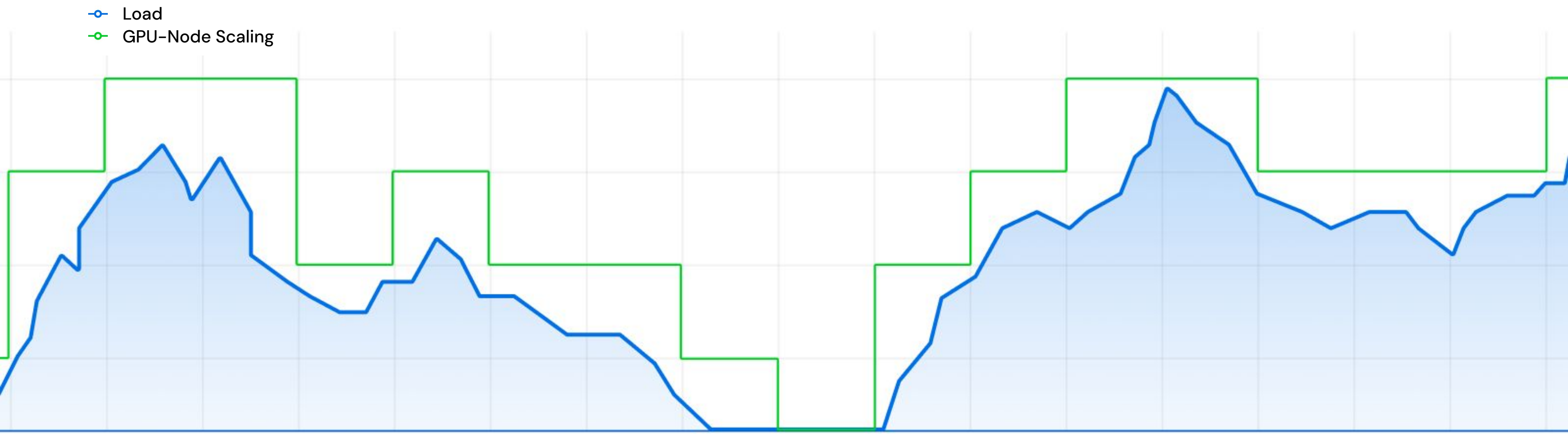
On-premises

Custom Scaling

SLA maintenance

Minimise Infra Overheads to the Last Bit

Scale down to zero in case of zero load
Cold-start time as low as 50s
Scale up to multiple GPUs at once, without doing it sequentially



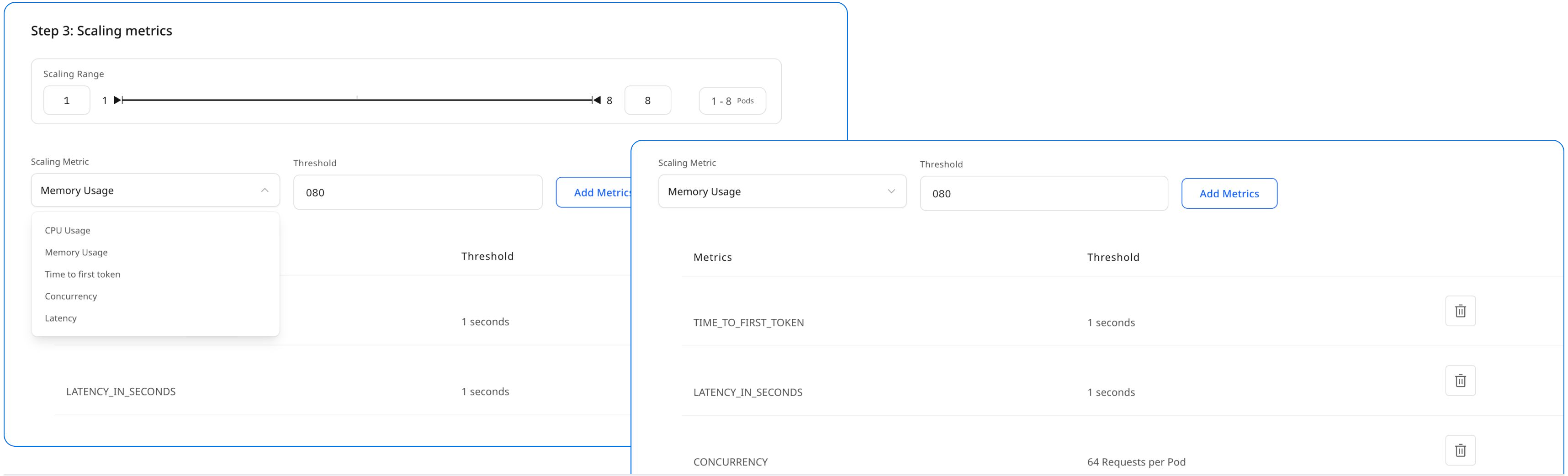
Simplismart enables application-level SLA enforcement.

App Metrics: TTFT, latency, concurrency

Infra Metrics: CPU, memory usage

Multi-Metric Logic: Combine triggers (e.g., latency + memory)

Custom Range: Define min–max pods (e.g., 1–8)



Step 3: Scaling metrics

Scaling Range: 1 – 8 (1 - 8 Pods)

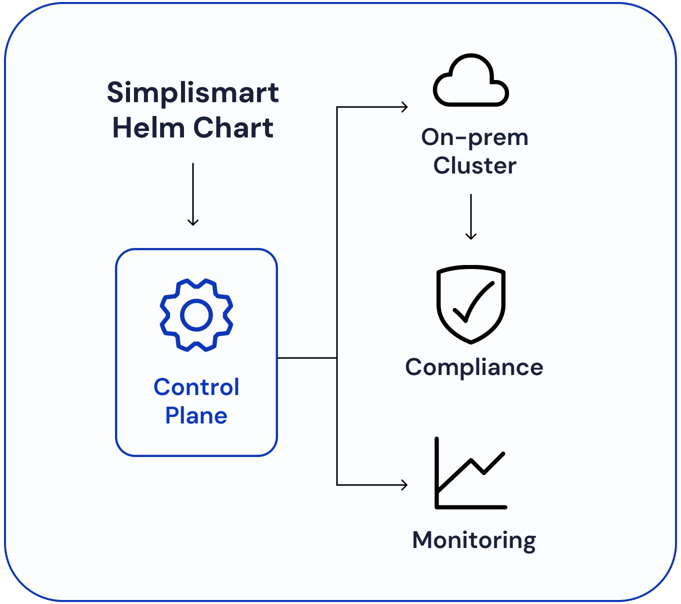
Scaling Metric: Memory Usage

Threshold: 080

Metrics	Threshold
TIME_TO_FIRST_TOKEN	1 seconds
LATENCY_IN_SECONDS	1 seconds
CONCURRENCY	64 Requests per Pod

Zero-Friction On-Prem & Edge GPU Deployment

Deploy seamlessly into air-gapped, hybrid, or edge clusters — with built-in observability and governance.



Why Infra Teams Love Our On-Prem Model"

01 **Single Helm Chart**
Same config deploys in AWS, Azure, or air-gapped metal.

02 **No External Calls**
Fully self-contained control plane, deploys without internet.

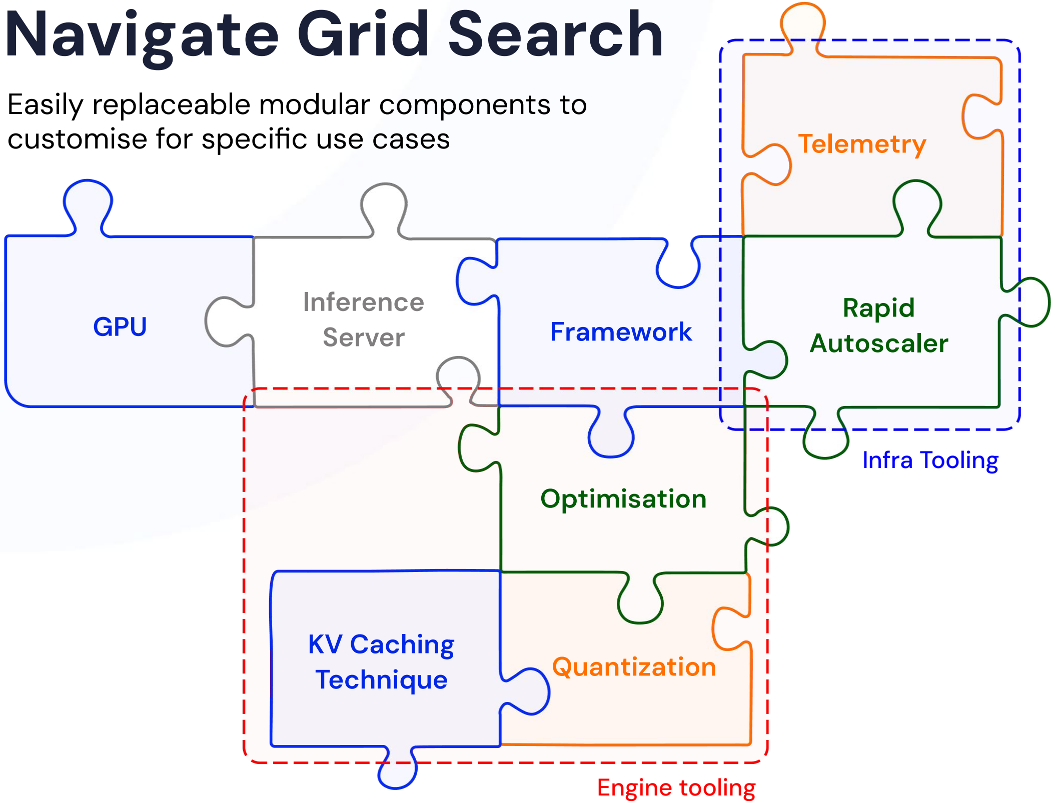
03 **Compliant by Design**
Supports FIPS 140-2, HIPAA, SOC2 via Qualys integration.

04 **No data storage by Simplismart**
On-prem deployment ensures no data leakages, 100% privacy

Modular Architecture

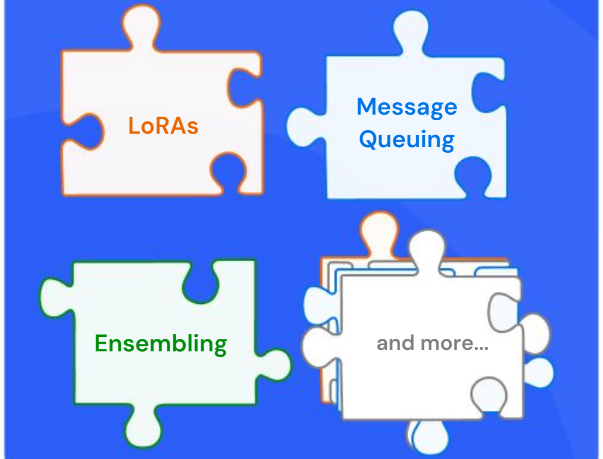
Modular Architecture to Navigate Grid Search

Easily replaceable modular components to customise for specific use cases



Add Ons

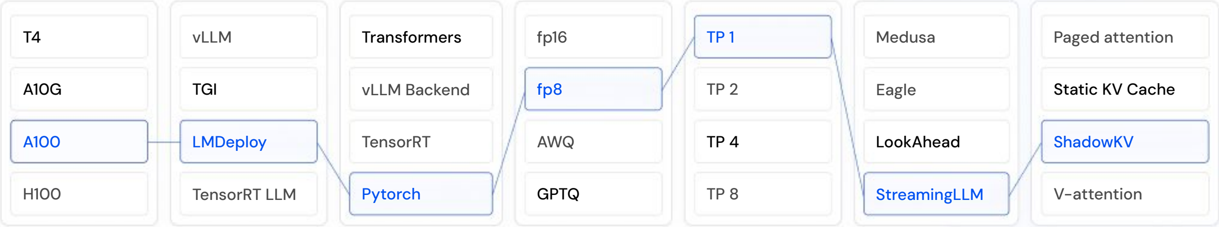
Multiple components to add all required functionality to ML Stack



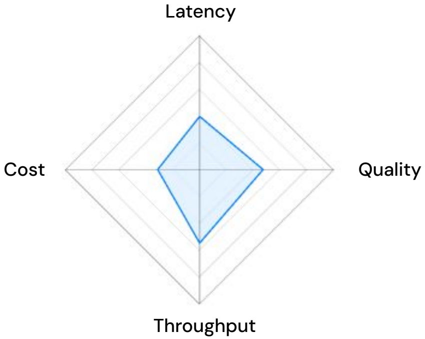
Different Configurations for Different Use Cases

Cost sensitive usecase

— Acceptable Latency ↓ Low Costs ↑ Acceptable Throughput ↑ High Quality

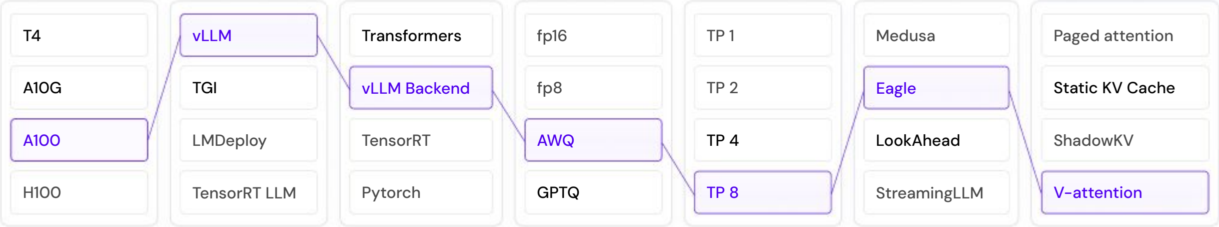


T4	vLLM	Transformers	fp16	**TP 1**	Medusa	Paged attention
A10G	TGI	vLLM Backend	**fp8**	TP 2	Eagle	Static KV Cache
A100	**LMDeploy**	TensorRT	AWQ	TP 4	LookAhead	**ShadowKV**
H100	TensorRT LLM	**Pytorch**	GPTQ	TP 8	**StreamingLLM**	V-attention

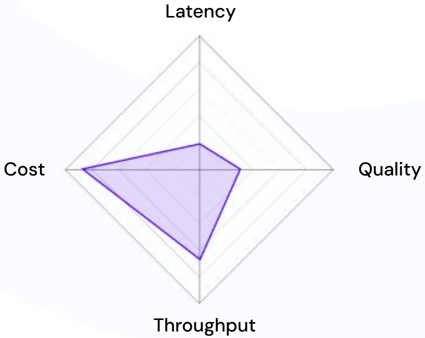


Latency sensitive usecase

↓ Low Latency — High Costs ↑ High Throughput ↑ Acceptable Quality



T4	**vLLM**	Transformers	fp16	TP 1	Medusa	Paged attention
A10G	TGI	**vLLM Backend**	fp8	TP 2	**Eagle**	Static KV Cache
A100	LMDeploy	TensorRT	**AWQ**	TP 4	LookAhead	ShadowKV
H100	TensorRT LLM	Pytorch	GPTQ	**TP 8**	StreamingLLM	**V-attention**


Latency
Cost
Quality
Throughput