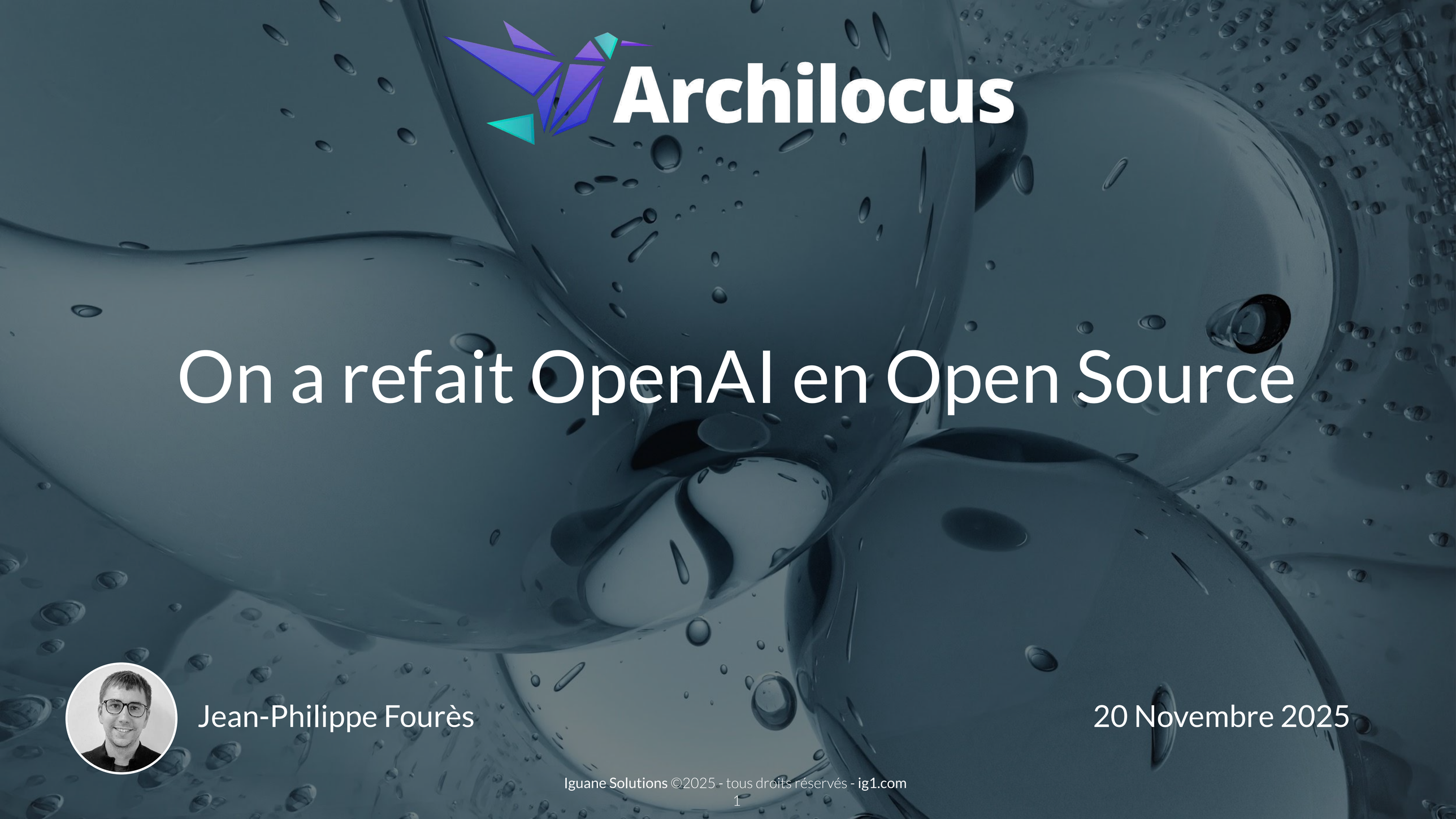


On a refait OpenAI en Open Source

Jean-Philippe Fourès

20 Novembre 2025

Disclaimer

- Pas Data Scientist
- Pas Machine Learning Engineer
- Pas éditeur de modèle
- Notre usage de l'IA (côte infra)

Introduction

Pourquoi faire sa propre stack IA ?

Autonomie

Liberté dans le choix
- des modèles
- de l'infra

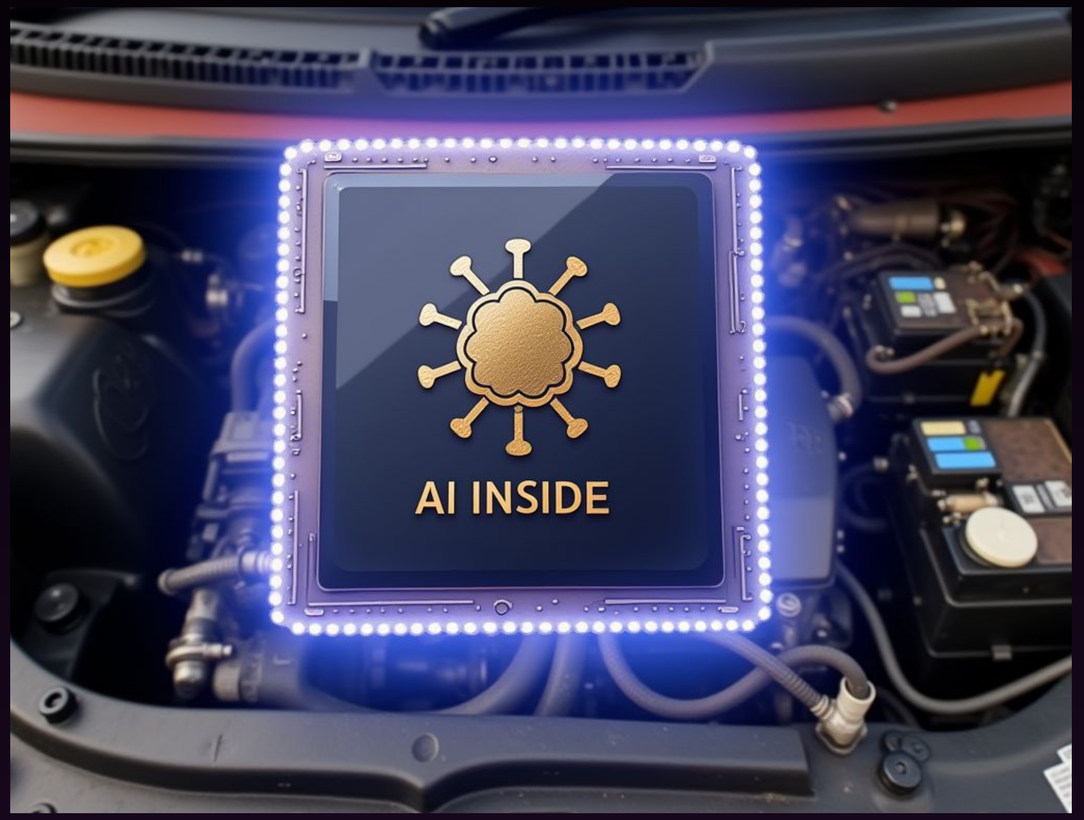
Gouvernance

Contrôle des données
Maîtrise du périmètre

Expertise

Maîtrise du savoir-faire
Augmentation compétences

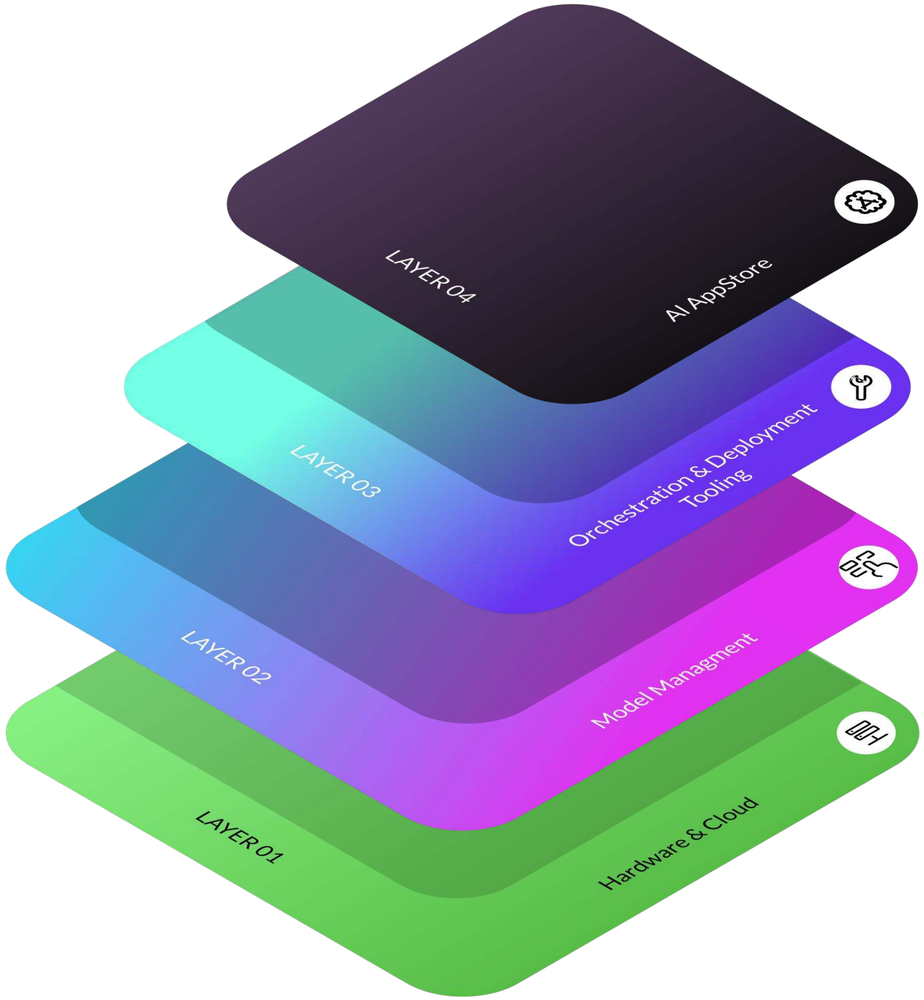
Le dessous des IA génératives



Point commun

Partagent le même concept de plateforme

Point commun



Mise en pratique

Construisons cette stack IA ensemble

- Quelle infra minimale ?
- Quel Modèle LLM ? open weight de préférence
- Quelle performance possible pour usage API ?
- Quelques Use Cases

INFRA

Infrastructure

Il faut du GPU !

GPU vs CPU: Parallèle vs Séquentiel

VRAM : Plus rapide que RAM

GPU plus performant pour l’IA que le CPU.

Infrastructure

Importance du GPU pour l’inférence

Performance Maximale: Inférence sur GPU seul

Important: Modèle + Contexte + Cache en VRAM

➡️ VRAM GPU: paramètre dimensionnant GPU

Infrastructure

Choix du modèle - Quelques exemples

Nom du modèle	Paramètres	Taille (format)
DeepSeek-V3-0324	685B	~700 Go (FP8)
Qwen3-235B-A22B-Thinking-2507	235B	~570 Go (BF16)
Qwen3-235B-A22B-Thinking-2507-FP8	235B	~240 Go (FP8)
Qwen3-Next-80B-A3B-Instruct	80B	~165 Go (BF16)
Qwen3-Next-80B-A3B-Instruct-FP8	80B	~82 Go (FP8)
Qwen3-Coder-30B-A3B-Instruct-FP8	30B	~30 Go (FP8)
Mistral-Large-Instruct-2407	123B	~250 Go (BF16)
Kimi-K2-Instruct-0905	1T (Trillion)	>1000 Go (FP8)

Infrastructure

Quelle Infra pour mon IA ?

Quelques GPUs:

Nvidia “Ada” L40s	48GB VRAM
Nvidia “Hopper” H100	80 GB VRAM
Nvidia “Hopper” H200	141 GB VRAM
Nvidia “Blackwell” RTX 5090	32 GB VRAM
Nvidia “Blackwell” RTX 6000 Pro	96 GB VRAM
AMD MI300x	192 GB VRAM

Infrastructure

Quel GPU pour mon modèle ?

Prenons Qwen3 Next 80B

QWEN 3 Next 80B A3B Instruct BF16	164 GB
QWEN 3 Next 80B A3B Instruct FP8	**82** GB

ET aussi besoin de RAM pour : Contexte et KV cache:
~6 GB pour 262k tokens

Infrastructure

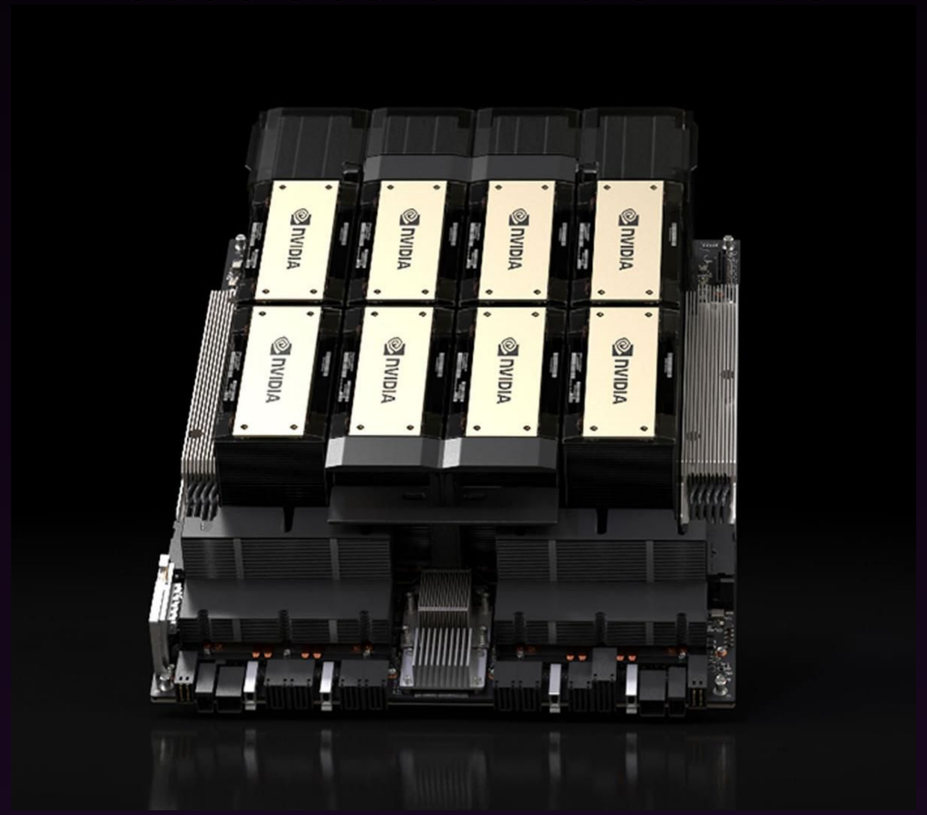
Quelle Infra pour mon IA ?

GPU possibles:

Nvidia “Ada” L40s	48GB VRAM
Nvidia “Hopper” H100	80 GB VRAM
Nvidia “Hopper” H200	141 GB VRAM
Nvidia “Blackwell” RTX 5090	32 GB VRAM
Nvidia “Blackwell” RTX 6000 Pro	96 GB VRAM
AMD MI300x	192 GB VRAM

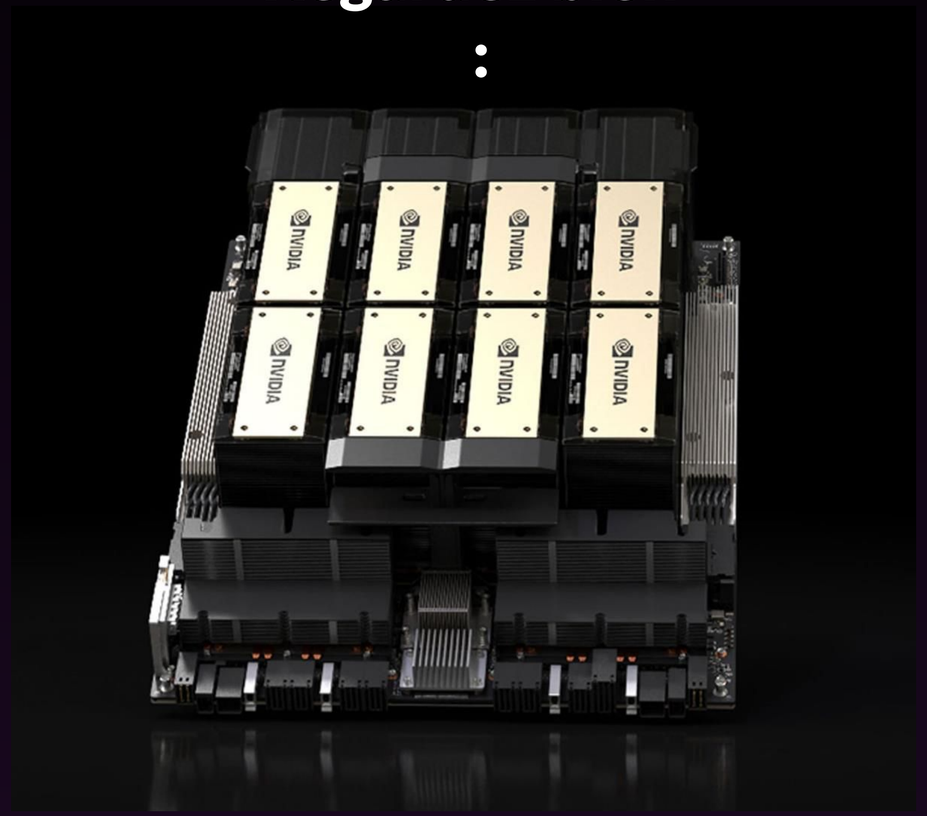
Infrastructure

Tout d’abord Nvidia H200



Infrastructure

Regardez bien :



Infrastructure

HGX H200 SXM

HGX: Serveur avec 8 H200 SXM5

NVLink entre GPUs

Comment isoler les GPU pour usage dédié ?

Infrastructure

HGX H200 SXM

La Virtualisation

- OpenNebula 6.10
- Serveur GPU: hyperviseur (KVM)
- VMs avec 1, 2 ou 4 GPUs selon besoin

Infrastructure

HGX H200 SXM

Notre objectif :

- Passthrough 1 (ou plus) GPU(s) à 1 VM
- Autoriser NVLink entre GPUs sur une même VM
- Interdire NVLink entre GPUs sur VMs différentes

Infrastructure

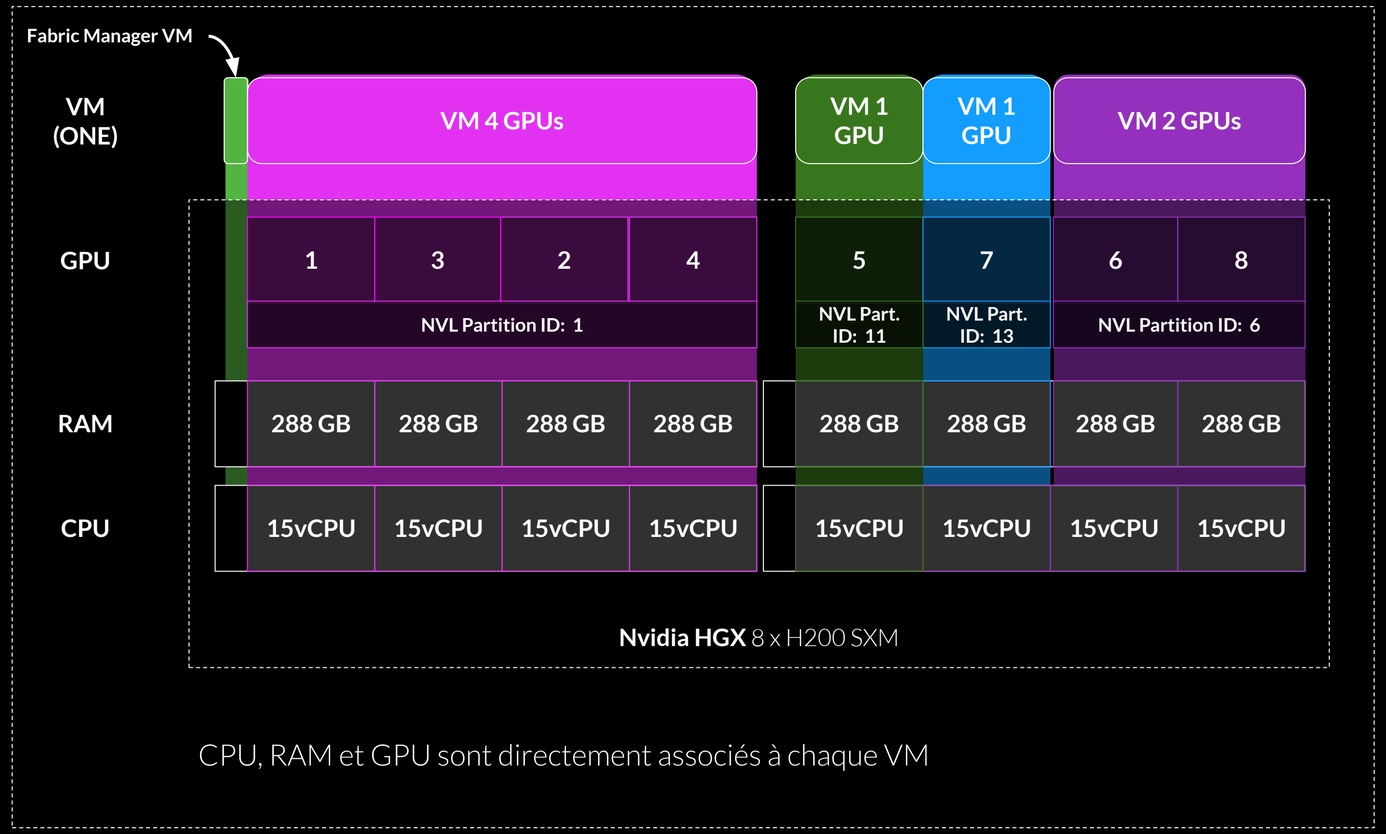
HGX H200 SXM

Partitionnement GPU NVLink

- Shared NVSwitch Virtualization Model
- VM Fabric Manager
- Passthrough GPU(s) sur VMs

Infrastructure

HGX H200 SXM - Exemple de configuration



CPU, RAM et GPU sont directement associés à chaque VM

Infrastructure

Une infra modulaire

Pourquoi plusieurs GPUs ?

- Le modèle est trop grand
- Contexte important
- Besoin de puissance de calcul (usage très intensif)

Infrastructure

RTX 6000 Pro

Infrastructure

Serveur avec RTX 6000 Pro

Points forts vs H200

- Plus de Coeur CUDA (24000 vs 16000 H200)
- Support Format NVFP4 et MXFP4
- Prix

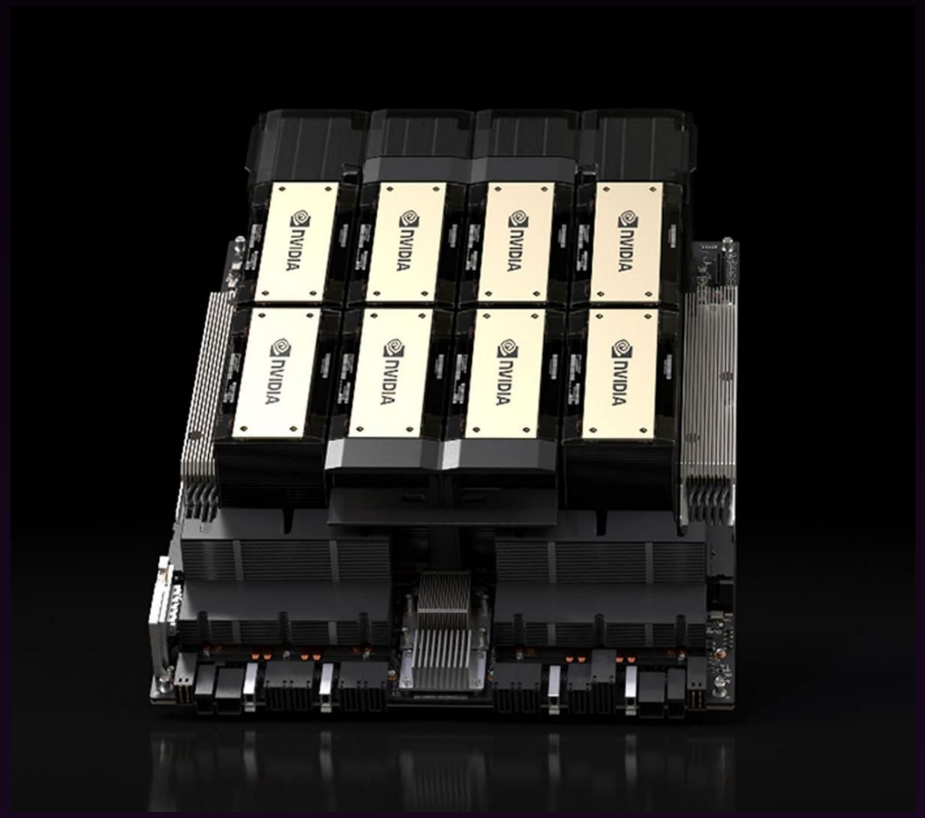
Points Faibles vs H200

- VRAM 96GB vs 141 GB H200
- Bande passante VRAM < H200
- Pas de NVLink

Pas besoin de faire de l’isolation

Infrastructure

H200 vs RTX 6000 Pro



Infrastructure

Scenario A -Small IN / Small Out (500 ±75 input tokens - 500 ±75 generated tokens)

H200	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	**117,71**	**199,89**	**358,4**	**584,78**	**937,37**	**1410,41**	**1984,28**	**2638,02**
Time To First Token (ms)	112,15	114,45	123,3	127,45	126,07	140	164,1	**211,83**
Inter Token Latency (ms)	**8,29**	**11,79**	**10,28**	**12,59**	**15,91**	**21,36**	**30,55**	**46,31**
Output Token Throughput Per User (tokens/sec/user)	**120,62**	**107,72**	**97,35**	**79,67**	**63,14**	**47,14**	**33,14**	**22,02**

RTX 6000 Pro	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	76,82	130,17	227,13	380,82	645,98	1013,15	1518,17	2041,43
Time To First Token (ms)	**75,28**	**87,16**	**88,41**	**98,17**	**98,05**	**118,87**	**161,58**	243,21
Inter Token Latency (ms)	12,89	16,04	16,46	19,55	23,14	29,7	39,71	59,58
Output Token Throughput Per User (tokens/sec/user)	77,56	69,31	60,83	51,26	43,32	33,76	25,31	16,97

Infrastructure

Scenario B - Heavy IN / Small Out (10 000 ±1 000 input tokens - 500 ±75 generated tokens)

H200	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	110,44	182,49	307,21	462,92	652,28	860,88	1054,53	1169,5
Time To First Token (ms)	335,53	343,44	369,22	403,7	454,21	586,11	851,46	1399,54
Inter Token Latency (ms)	8,41	11,84	11,6	15,57	22,63	34,73	57,29	104,26
Output Token Throughput Per User (tokens/sec/user)	118,88	102,37	86,53	64,73	44,82	29,35	17,97	10,04

RTX 6000 Pro	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	72,13	118,15	195,73	295,86	425,12	551,75	463,49	435,68
Time To First Token (ms)	426,53	450,91	470,89	526,91	640,56	847,27	19203	93689,48
Inter Token Latency (ms)	13,05	20,06	18,47	24,63	34,68	54,27	97,41	104,88
Output Token Throughput Per User (tokens/sec/user)	76,63	65,65	54,33	40,92	29,16	18,74	10,67	9,8

Infrastructure

Scenario C - Heavy In / Heavy Out (10 000 ±1 000 input tokens - 10 000 ±1 000 generated tokens)

H200	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	**116,48**	**201,84**	**346,75**	**578,23**	**866,03**	**1253,6**	**1785,05**	**2271,2**
Time To First Token (ms)	**336,3**	**346,62**	**357,09**	**386,08**	**439,49**	**547,52**	**780,15**	**1270,85**
Inter Token Latency (ms)	**8,43**	**11,97**	**10,45**	**12,9**	**16,79**	**23,22**	**33,85**	**53,68**
Output Token Throughput Per User (tokens/sec/user)	**118,67**	**105,87**	**95,82**	**77,72**	**59,92**	**43,46**	**29,84**	**18,85**

RTX 6000 Pro	1	2	4	8	16	32	64	128
Output Token Throughput (tokens/sec)	75,15	129,84	217,49	344,01	512,6	762,49	574,11	565,44
Time To First Token (ms)	429,15	444,74	470,29	507,21	603,73	803,03	**76704,12**	**313741,75**
Inter Token Latency (ms)	13,08	16,79	17,28	21,31	27,31	38,71	73,18	75,31
Output Token Throughput Per User (tokens/sec/user)	76,46	67,02	57,91	47,08	36,95	26,03	14,09	13,56

Infrastructure

Benchmarks - Results

Vainqueur : H200

Explications : H200 plus de KV Cache possible -> parallélisme

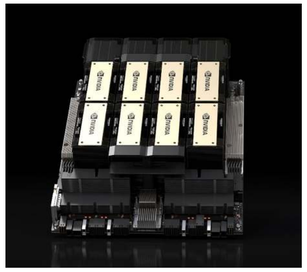
Cependant, RTX 6000 pro reste intéressante car:

- coût sensiblement inférieur
- support de NVFP4: même modèle plus “petit”

Notre collection NVFP4 sur Hugging Face :

Infrastructure

Bilan



H200 idéale :

Grand Modèles (NVLink pour lier GPUs)

Meilleur sur requêtes à grand contexte

Besoin de parallélisme: grande échelle

RTX 6000 Pro idéale :

Modèles <= 80B

Support format NVFP4 (gain RAM)

Coût très inférieur à H200

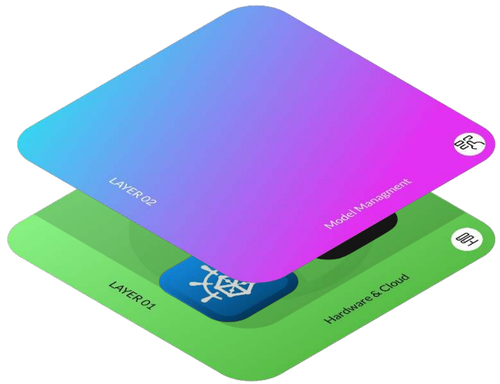
Infrastructure

On a l’infra pour notre IA

Modèles

Modèles Open ~~Source~~ Weight

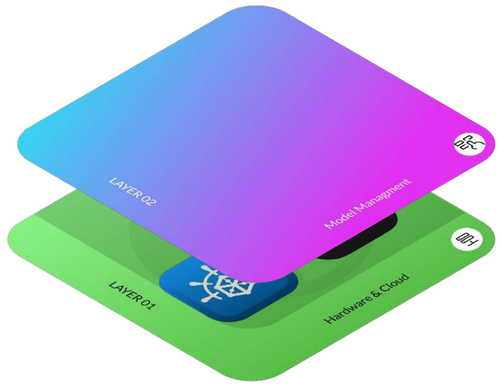
Un catalogue “illimité”

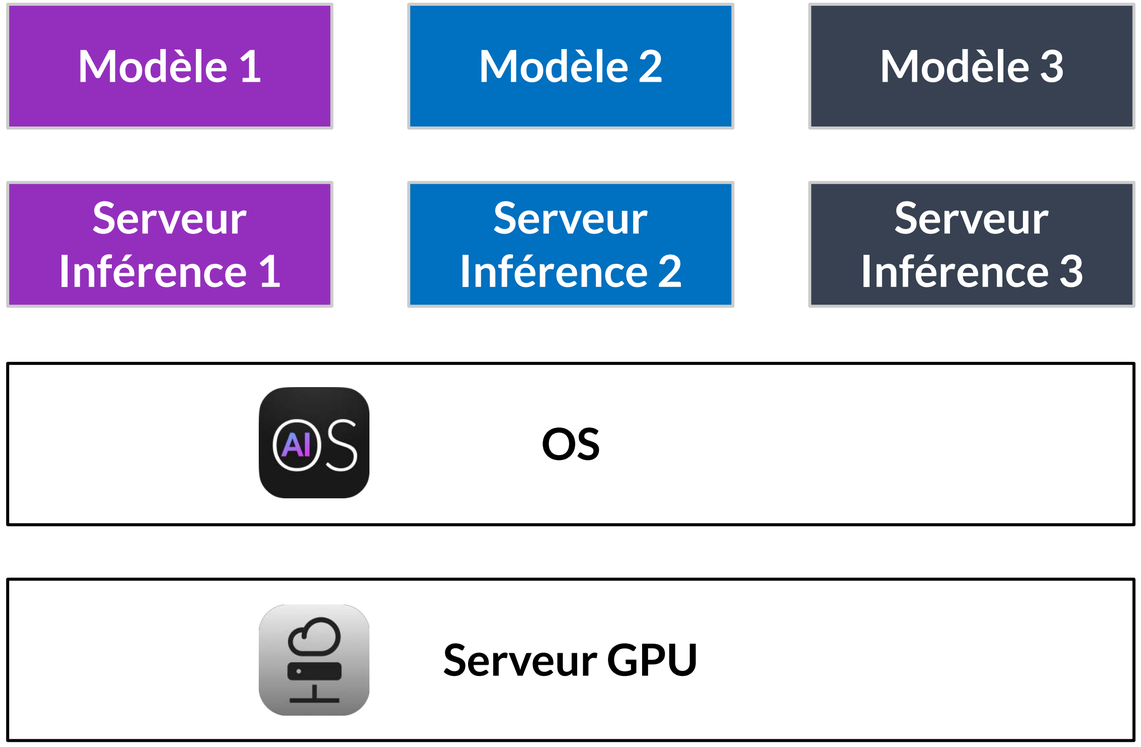


- Hugging Face: La référence
- Large choix de modèles
- Modèles adaptés à son marché

Modèles Open ~~Source~~ Weight

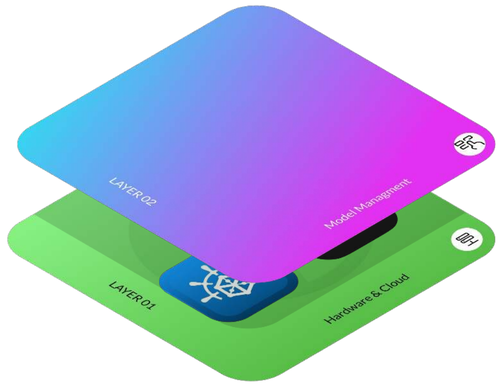
Comment déployer un modèle





Modèles Open Source

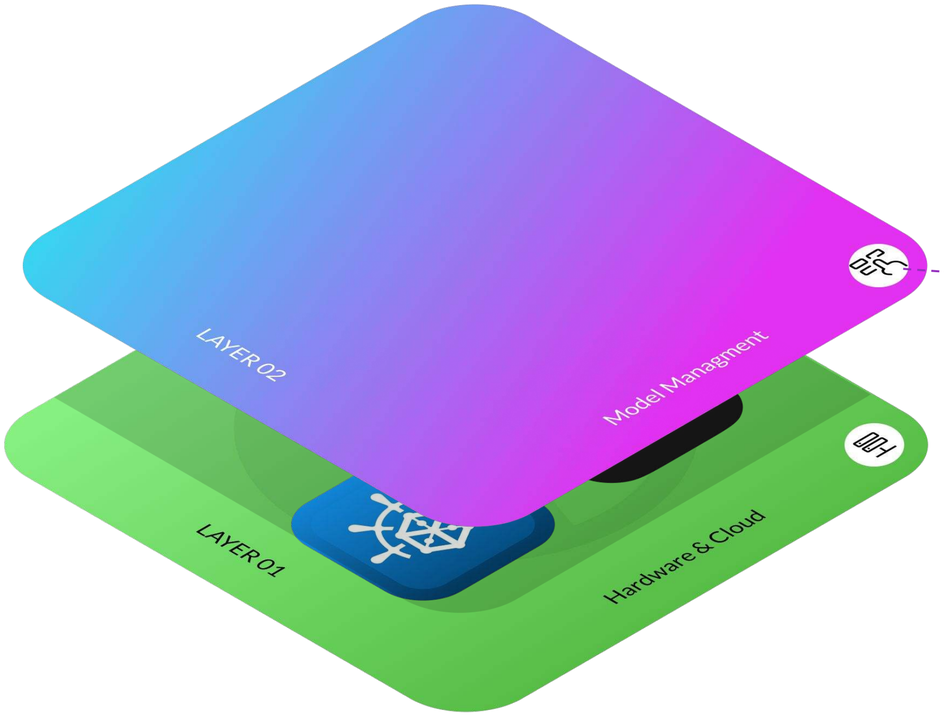
OS



- Base: Ubuntu 24.04 (LTS)
- Drivers Nvidia (CUDA, container-toolkit, smi)
- Dépendances : Docker, Conda, …
- Security: firewall rules, VPN

Modèles Open Source

Serveur Inférence - Plusieurs projets selon l'usage



2

Layer 02: Model Management

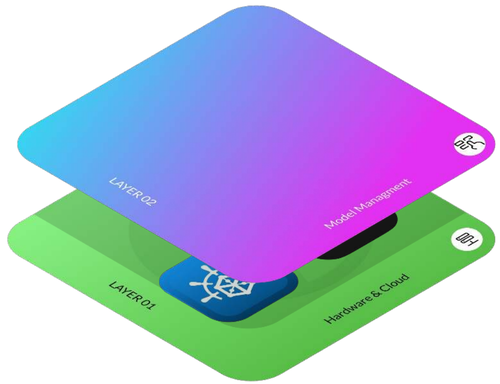
Text Gen. & Tools	Code	RAG
LLM with tools support	Code Gen. &Chat	embed
		rerank
vLLM	vLLM	vLLM

Multimodal	Image Gen.	Reasoning
Vision LM	Image Gen.	Reasoning LM
vLLM	ComfyUI	vLLM

Speech to Text	Text to Speech	Video Generation
Speech to Text	Text to Speech	Video Generation
Faster whisper	Kokoro	Comfy UI

Modèles Open Source

Partage du GPU - Plusieurs Modèles



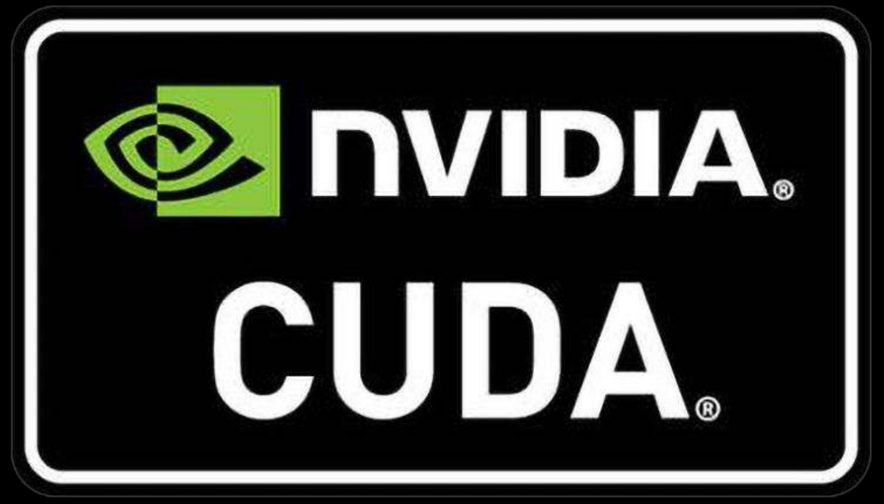
Dans le cas de plusieurs Modèles:

- Partage du temps d’accès au GPU
- Perception “saccadée” ou ralentie du traitement

MPS

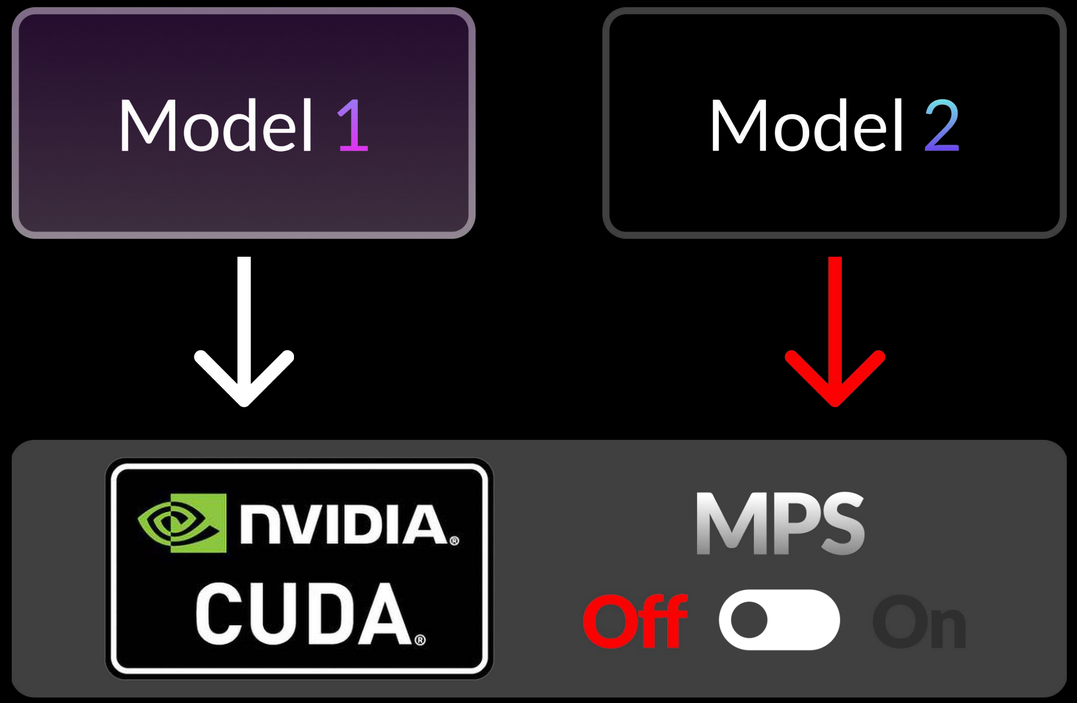
Multi-Process Service

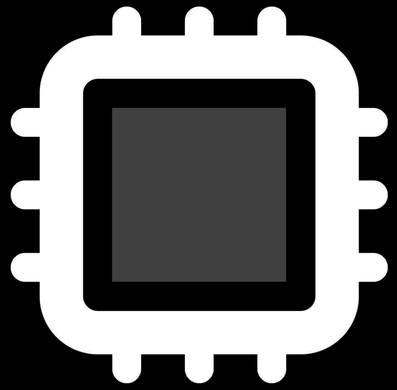
https://docs.nvidia.com/deploy/mps/index.html



Compute Unified Device Architecture

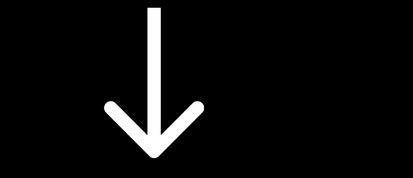

MPS


Model 1
Model 2
NVIDIA
CUDA
MPS
Off
On



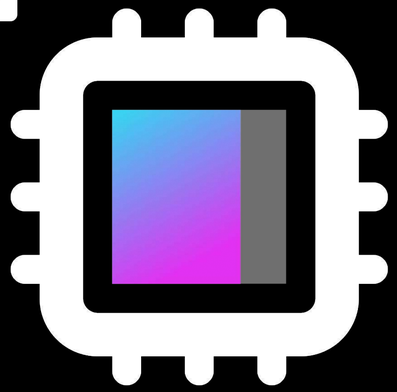
Model 1

Model 2



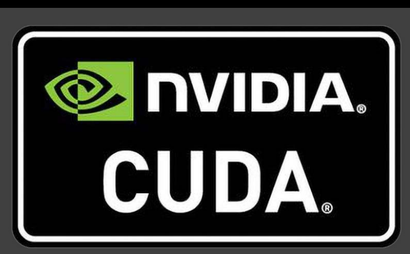
MPS
Off On

60%



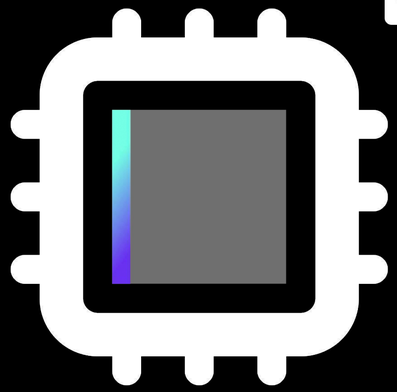
Model 1

Model 2

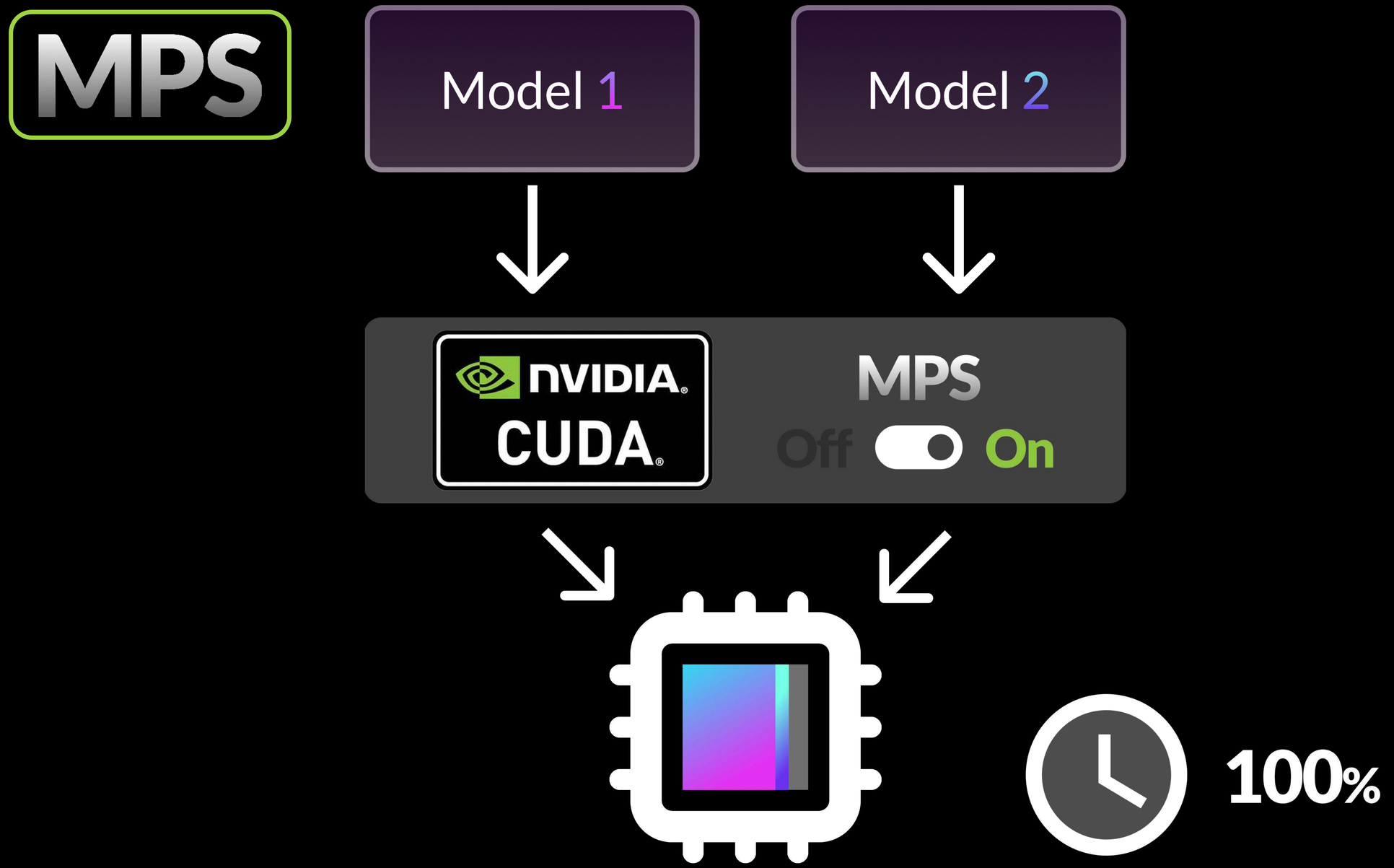


MPS

Off On

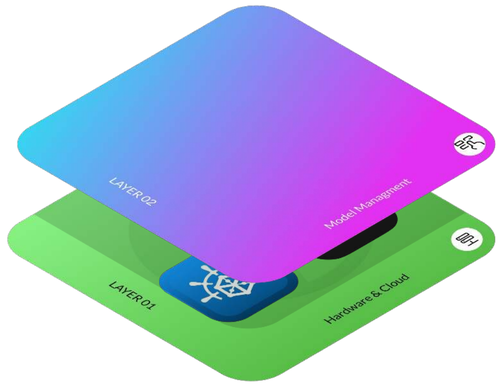


40%


MPS
Model 1
Model 2
NVIDIA
CUDA
MPS
Off
On
100%

Modèles Open Source

Partage du GPU - Plusieurs Modèles



Grâce au MPS:

- Optimisation du GPU
- Plus efficient en Multi-modèle
- Fluidité des traitements

Modèles Open Source

On a déployé nos modèles

Exploitation

Exploitation

AI stack Management operations

Orchestrateur (LiteLLM)

- Endpoint API “OpenAI”
- Gestion utilisateurs / Modèles
- Routage des modèles

Exploitation

On a notre équivalent “OpenAI”

…

Selon la vision utilisateur

Exploitation

Gestion de la plateforme IA

Metrologie (Prometheus / Grafana)

- Métriques HW et LLM
- Tracing Applications LLM
- Mesure Empreinte Carbone

Exploitation

Gestion de la plateforme IA

Outils Dev (Continue.dev, IG1)

- Serveur Configuration Copilot Dev
- Traducteur API Ollama < > OpenAI

Exploitation

Gestion de la plateforme IA

Outils Data

- Model Studio: Fine tuning, entraînement
- Outil d’évaluation des modèles et prompts

Exploitation

On a une plateforme IA privée et complète

Sympa ton API mais OpenAI ce n’est pas qu’une API.

Applications

Applications

Les App OpenAI

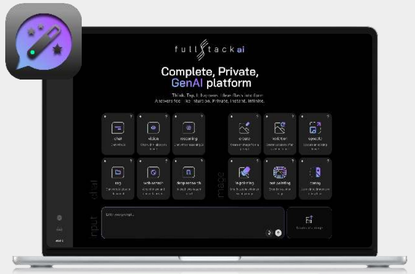
OpenAI propose plusieurs Applications :

- ChatGPT
- Les GPTs,
- Operator
- Un SDK

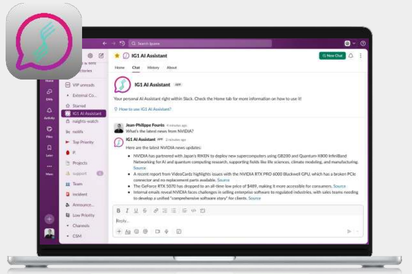
Applications

Layer 04: AI Applications

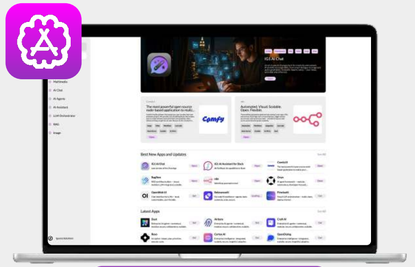
4



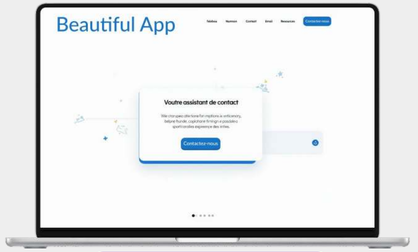
IG1 AI Chat



IG1 AI Assistant



AI AppStore



Any APP via API

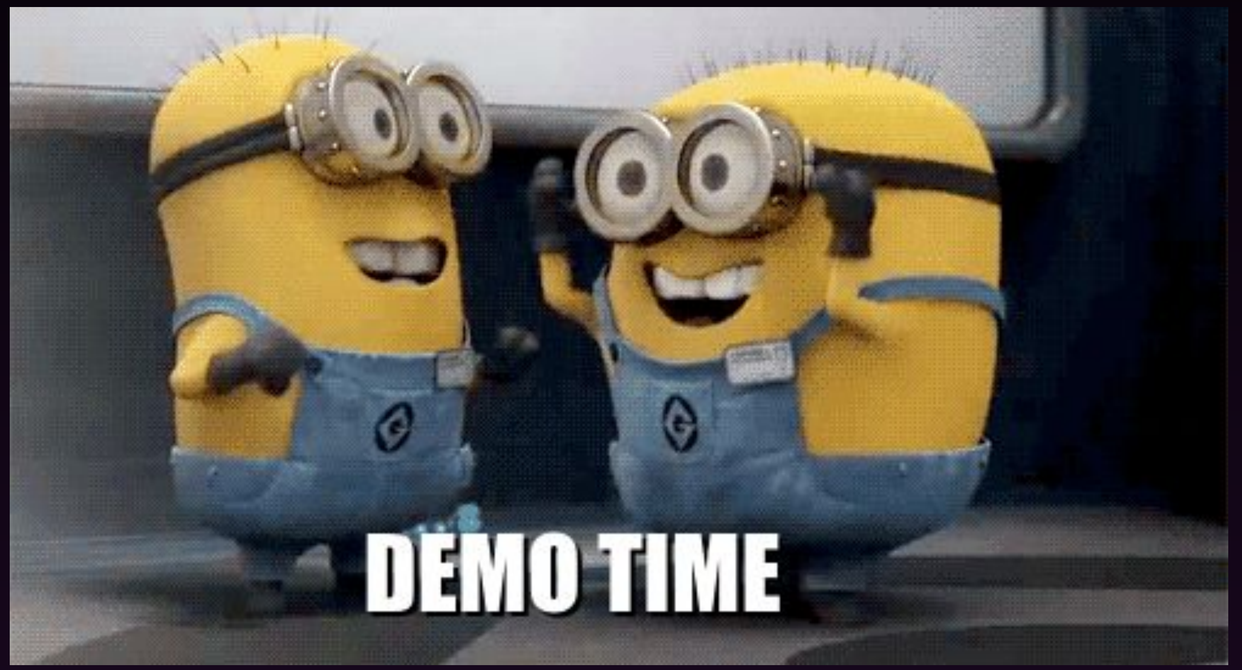
Applications

Quelques App Open Source pour sa Plateforme IA

Projets intéressants pour sa stack IA:

- **Chat** : OpenWebUI, IG1 AI Slack Assistant
- **RAG** : Onyx, Ragflow, Dify
- **Web Search** : Perplexica
- **Génération Image / vidéo**: ComfyUI
- **Agents IA**: N8N (no code), Crew AI

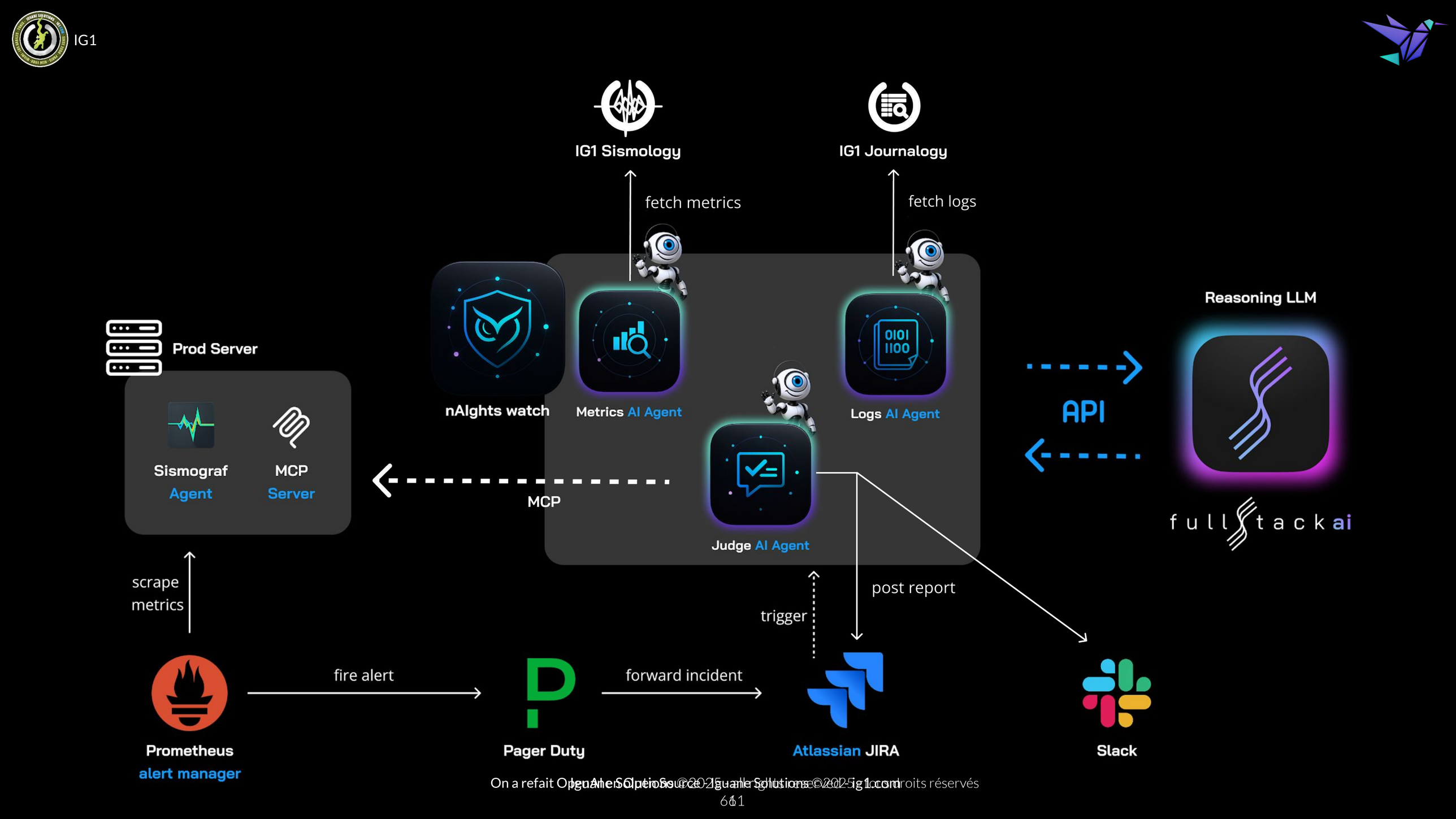
Mise en pratique



Mise en pratique

nAIghts watch

the **Multi-agents AI Incident Manager**




nAights watch

C’est simple non ?

MERCI !

Jean-Philippe Foures

VP Product

@jpfoures

@jaypif

Iguane Solutions

fullstackai


IGUANE SOLUTIONS - IG1.COM - SINCE 2000 - PARIS - NEW YORK - MIAMI - LOS ANGELES - TUNIS -