

Sous la surface d'ILaaS Deepdive dans l'infrastructure technique

Journée technique IA ESUP 25/09/2025





Le projet ILaaS et les services utilisateur

Inférence LLM

Rappel de nos objectifs

Mutualiser des infrastructures de calcul pour l'inférence LLM

- Consolider l'hébergement dans des DC labellisés
- Centraliser les achats pour les nouvelles ressources
- Pouvoir utiliser les ressources pré-existantes

Mutualiser des usages

- Des usages interactifs ou en mode batch
- Des usages d'inférence différents: plusieurs LLM généralistes ou spécialisés, STT, ...

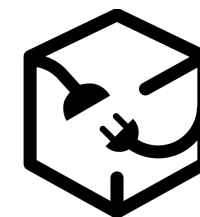
Conséquences techniques

- Compatibilité avec différentes technologies d'inférence (vLLM, Ollama, LMDeploy)
- Architecture multi-DC avec répartition de charge (disponibilité d'un même modèle)
- Gestion de priorités: entre usages différents, entre clients différents

API proposées pour construire des services

2 interfaces API proposées dès aujourd'hui

- Inférence LLM
- Service de transcription asynchrone (STT)

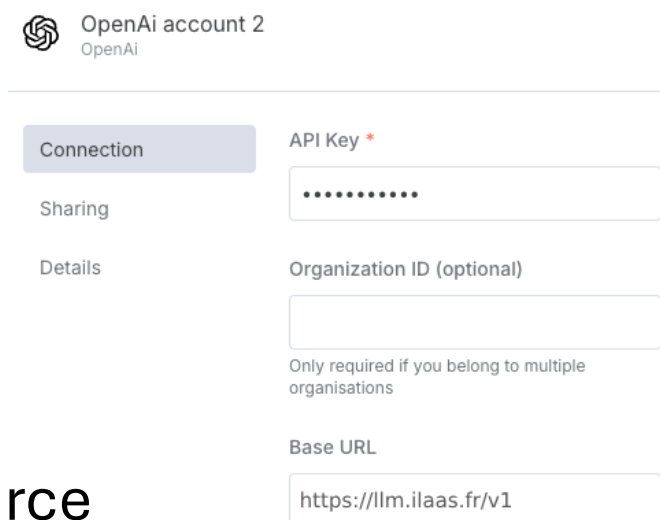


A venir (prochainement)

- Embedding
- Base de données Vectorielle
- OCR

Inférence LLM

- 2 endpoints disponibles:
/models, /chat/completions
- Une URL: <https://llm.ilaas.fr/v1>
- Une clé d'API pour y accéder



OpenAi account 2
OpenAi

Connection

API Key *

.....

Sharing

Details

Organization ID (optional)

Only required if you belong to multiple organisations

Base URL

<https://llm.ilaas.fr/v1>

```
import openai

openai.api_base = "https://llm.ilaas.fr/v1"
openai.api_key = "secret-key"

models_list = openai.Model.list()
model_name = models_list.data[0].id

completion = openai.ChatCompletion.create(
    model=model_name,
    messages=[
        {"role": "user",
         "content": "Qu'est-ce qu'ILaaS ?"}
    ]
)

print(completion.choices[0].message.content)
```

- Opensource
<https://github.com/CentraleSupelec/aristote-dispatcher>

Transcription

- Une API dédiée pour accéder au service
 - URL de base
 - Client_id et Secret
- Comment l'utiliser
 - Configuration dans un logiciel compatible (Ubicast Nudgis, ESUP-Pod)
 - Implémentation d'un client
- Opensource

** champs requis*

URL de l'API :

Identifiant du client d'API :

Clé d'API :

☒ Éditer la valeur

Langues autorisées pour les transcriptions : ☒ Anglais ☒ Français
Valeur par défaut : ['eng', 'fre']

Langues autorisées pour les traductions : ☒ Anglais ☒ Français
Valeur par défaut : ['eng', 'fre']

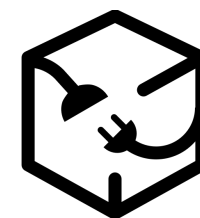
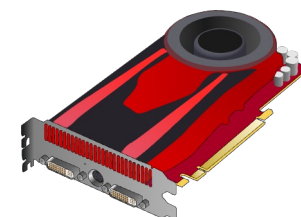


<https://pod.esup-portail.org/>

https://github.com/CentraleSupelec/aristote-api/blob/main/docs/AristoteAPI_Client_guide.md

Au-delà des API ?

- Nous disposons d'infrastructures matérielles et réseau capables d'opérer des services d'IA
 - DC labellisés et établissements : 5 en phase 1 ; **plus de 35 en phase 2** avec le **soutien de la DGRI**
- Nous disposons des briques essentielles sous forme d'API pour propulser des services se basant sur une infrastructure souveraine mutualisée
- Nous souhaitons proposer sur ILaaS une **marketplace** d'outils souverains (tchatbots, RAG, outils pédagogiques, ...)
 - En production: TIPS aunege
 - En cours de raccordement: Capytale
 - Et d'autres à venir...



<https://tips.aunege.net/presentation.php>
<https://capytale2.ac-paris.fr/>





Sous le capot du service d'inférence LLM

[LLM.ilaas.fr](https://llm.ilaas.fr)

Choisir ses modèles d'inférence

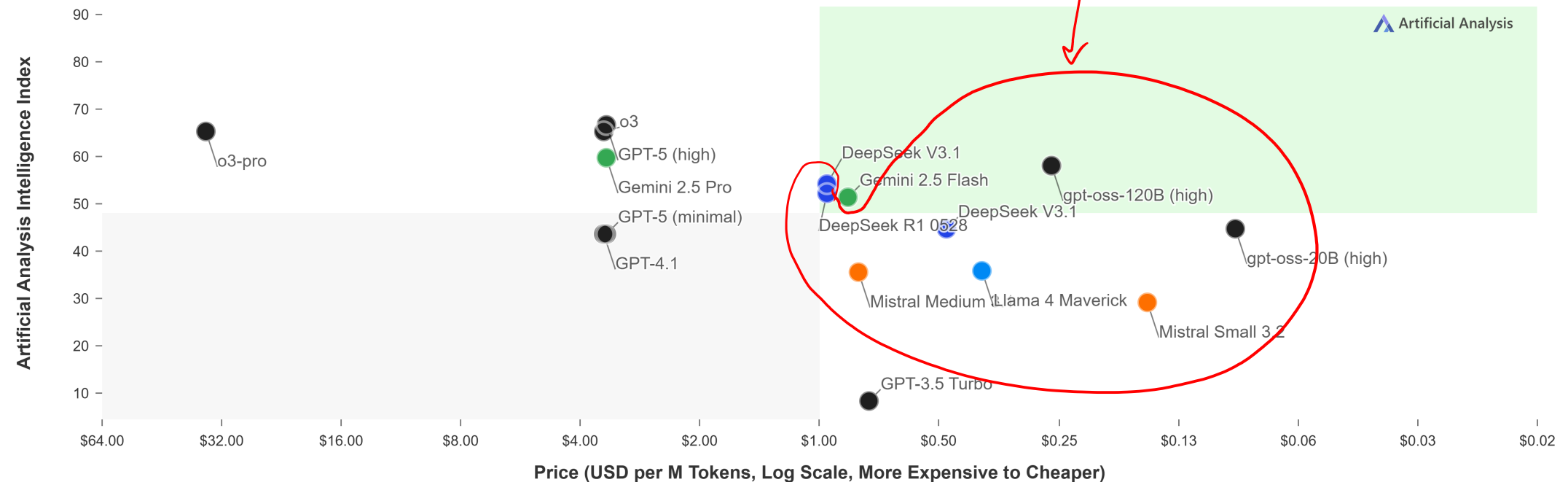
Des modèles suffisamment performants pour un impact raisonnable (budgétaire, environnemental)

Intelligence vs. Price (Log Scale)

Artificial Analysis Intelligence Index; Price: USD per 1M Tokens; Inspired by prior analysis by Swyx

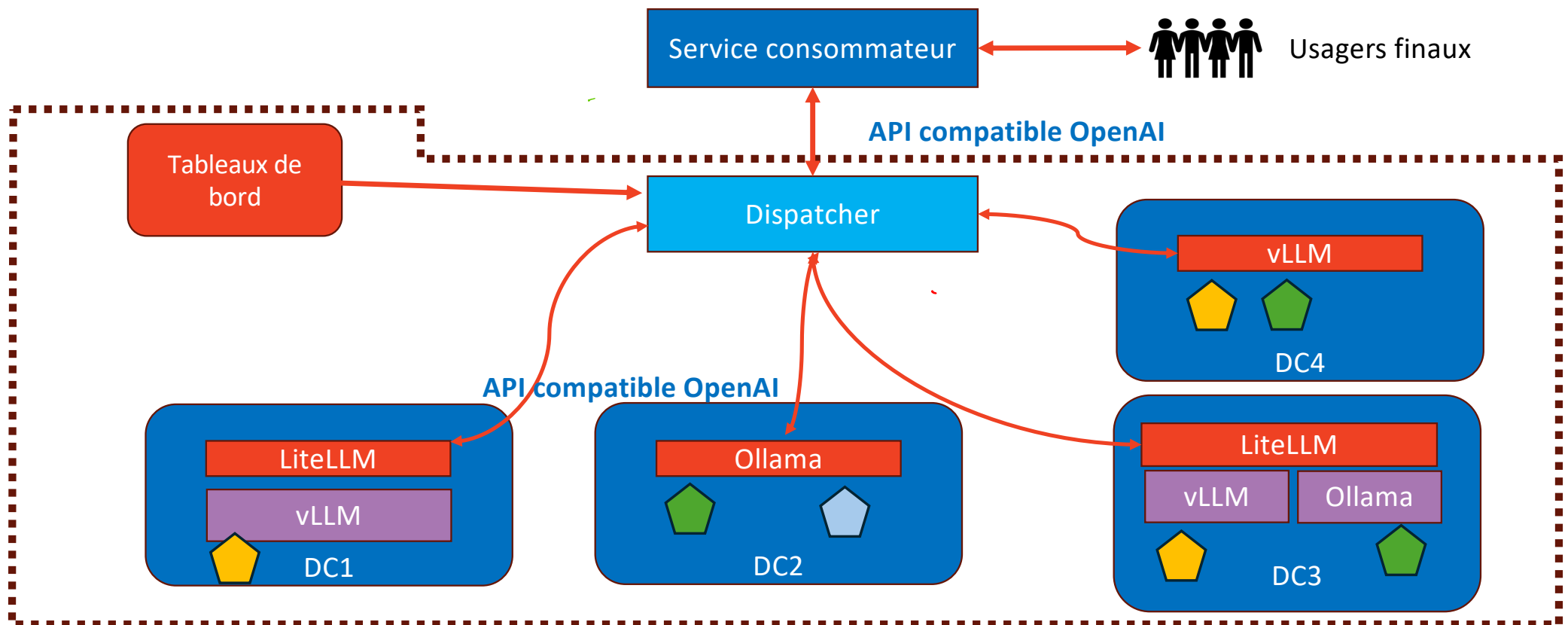
Most attractive quadrant

DeepSeek Google Meta Mistral OpenAI



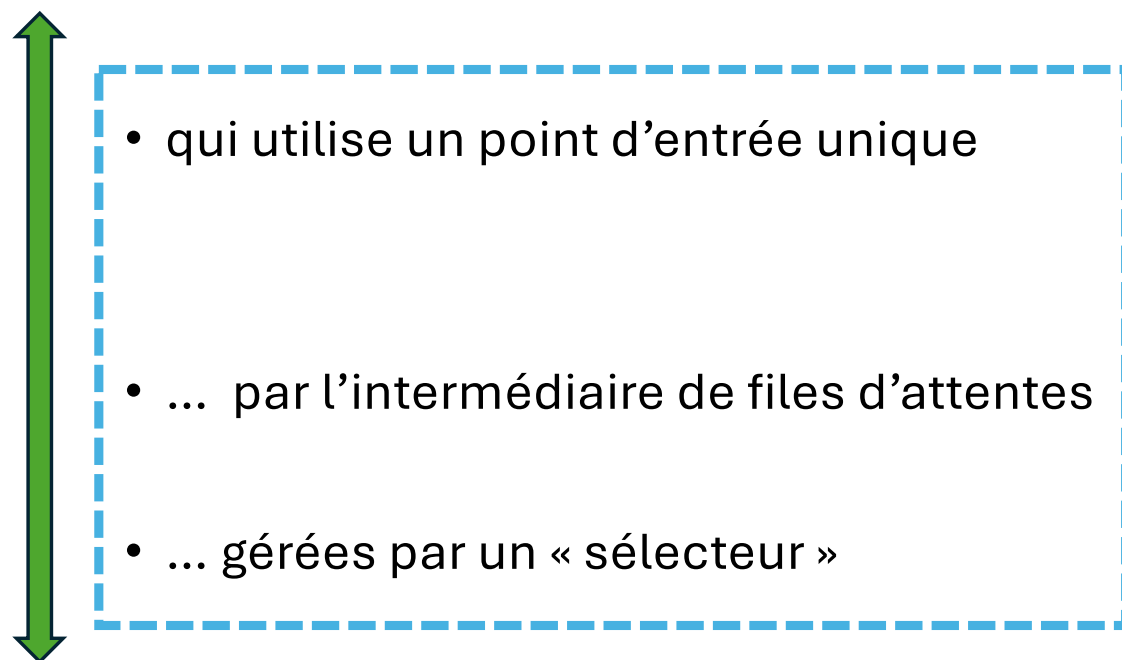
ILaaS unifie l'accès à plusieurs modèles hébergés sur plusieurs Datacentres

- Serveurs d'inférence compatibles OpenAI

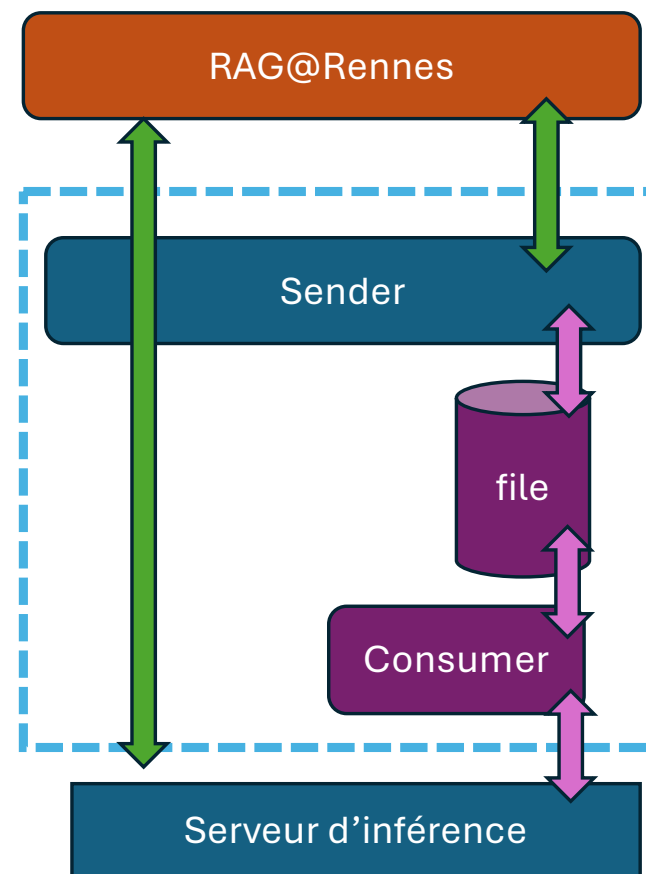


Le dispatcher est un service de mise en relation

- entre un « utilisateur »

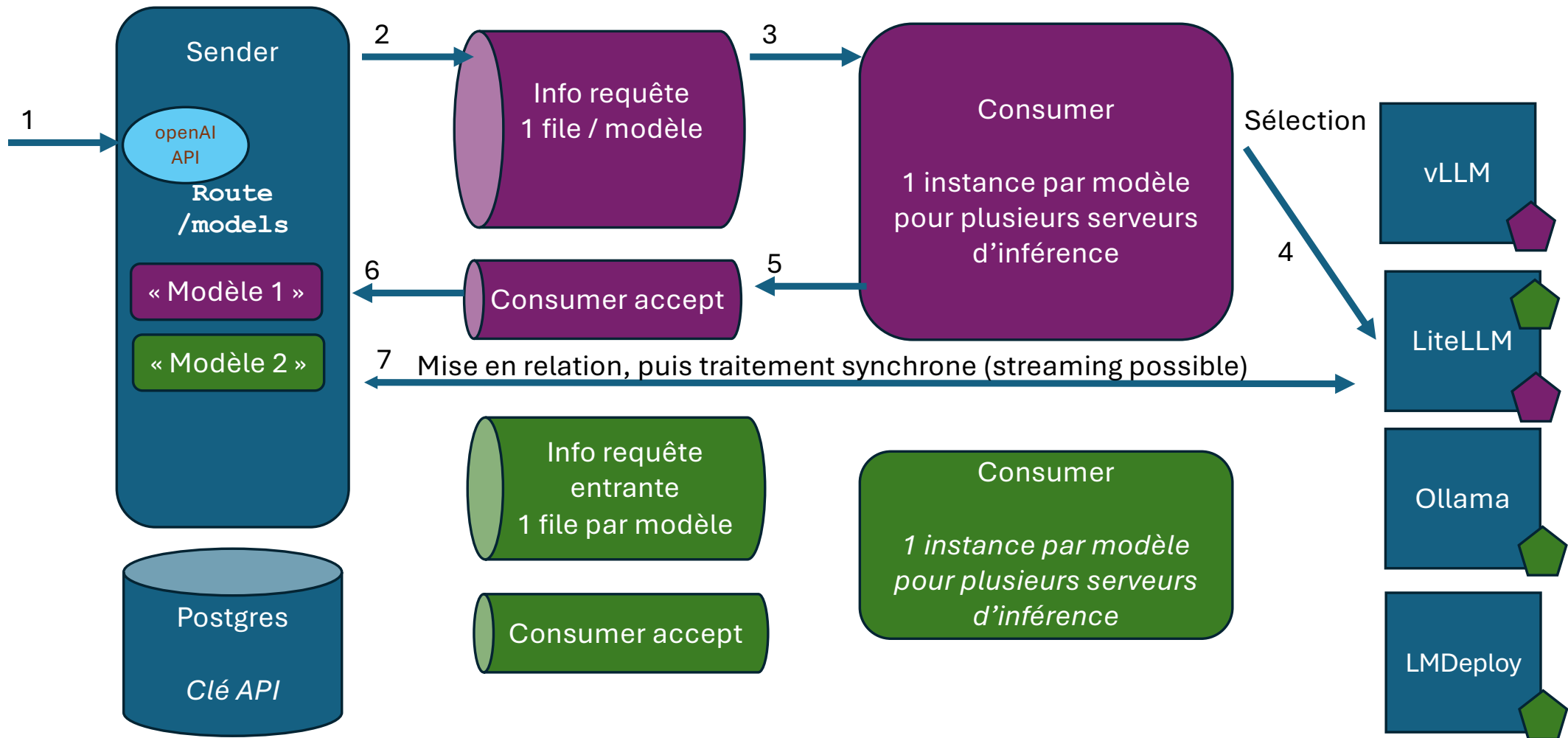


- et un serveur d'inférence



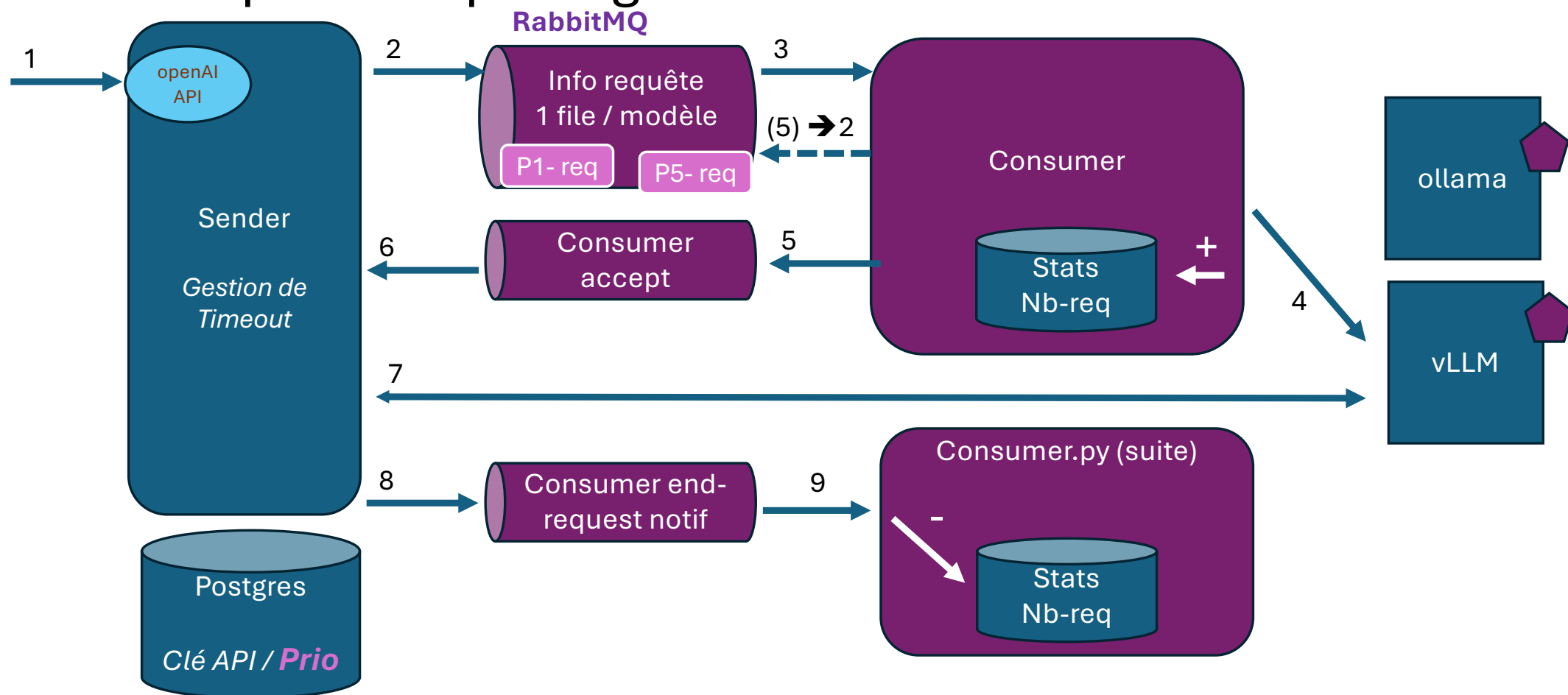
Traitement simplifié d'une requête

Stratégie round-robin sans queuing



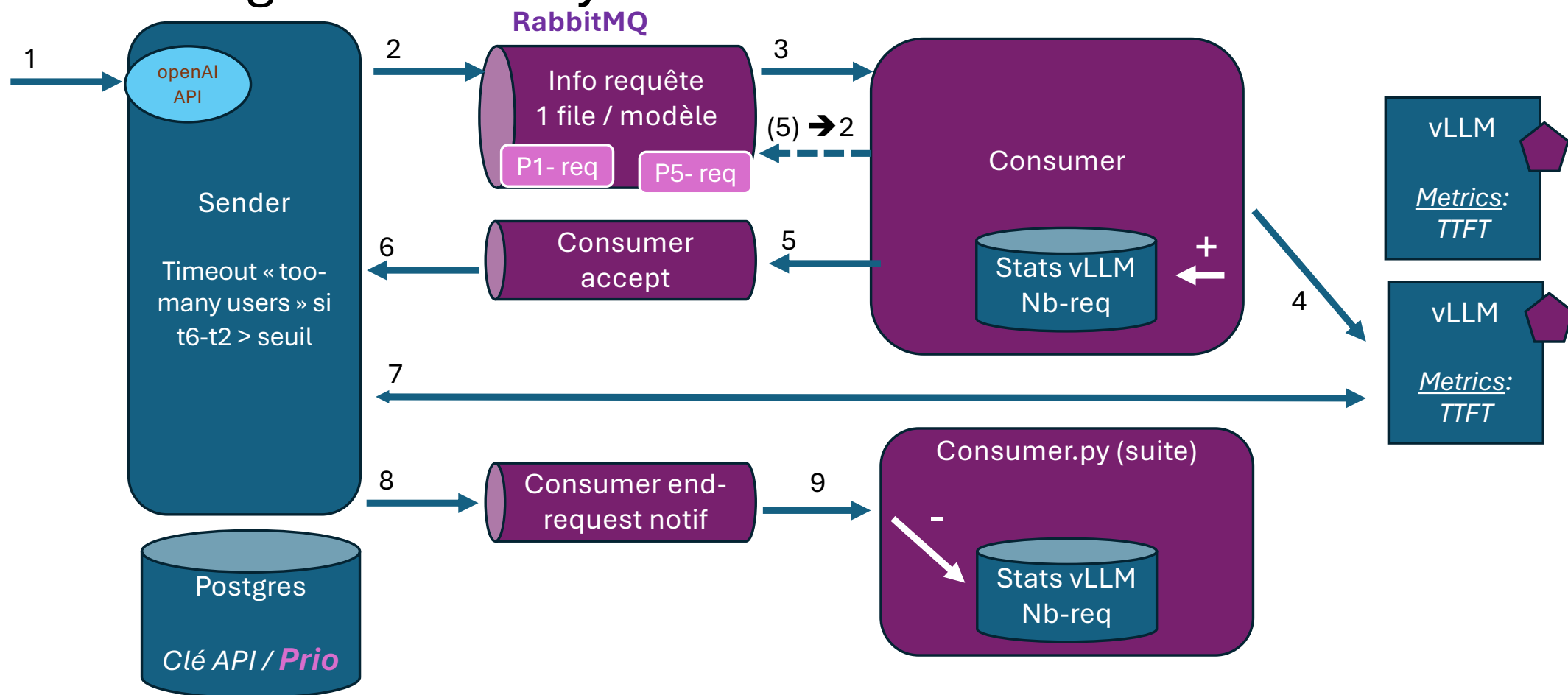
Gestion de priorités

Politique de queuing



Optimisation du routage

Stratégie Least-Busy



Stratégies et Politiques de requête

Fonctionnement	Stratégie de sélection	Politique de requête
Round Robin simple Gestion de la disponibilité Répartition de charge simple	Round Robin	Sans queueing
Round Robin Gestion de la disponibilité Répartition de charge simple Gestion de priorités	Round Robin	Basé sur le nombre de requêtes simultanées
Least busy (vLLM) Gestion de la disponibilité Répartition de charge optimisée Gestion de priorités Garantie de performances dans certaines limites	Basé sur statistiques vLLM	Basé sur statistiques vLLM et sur le nombre de requêtes simultanées

Architecture simulation de la gestion des priorités

Exemple de test réalisé

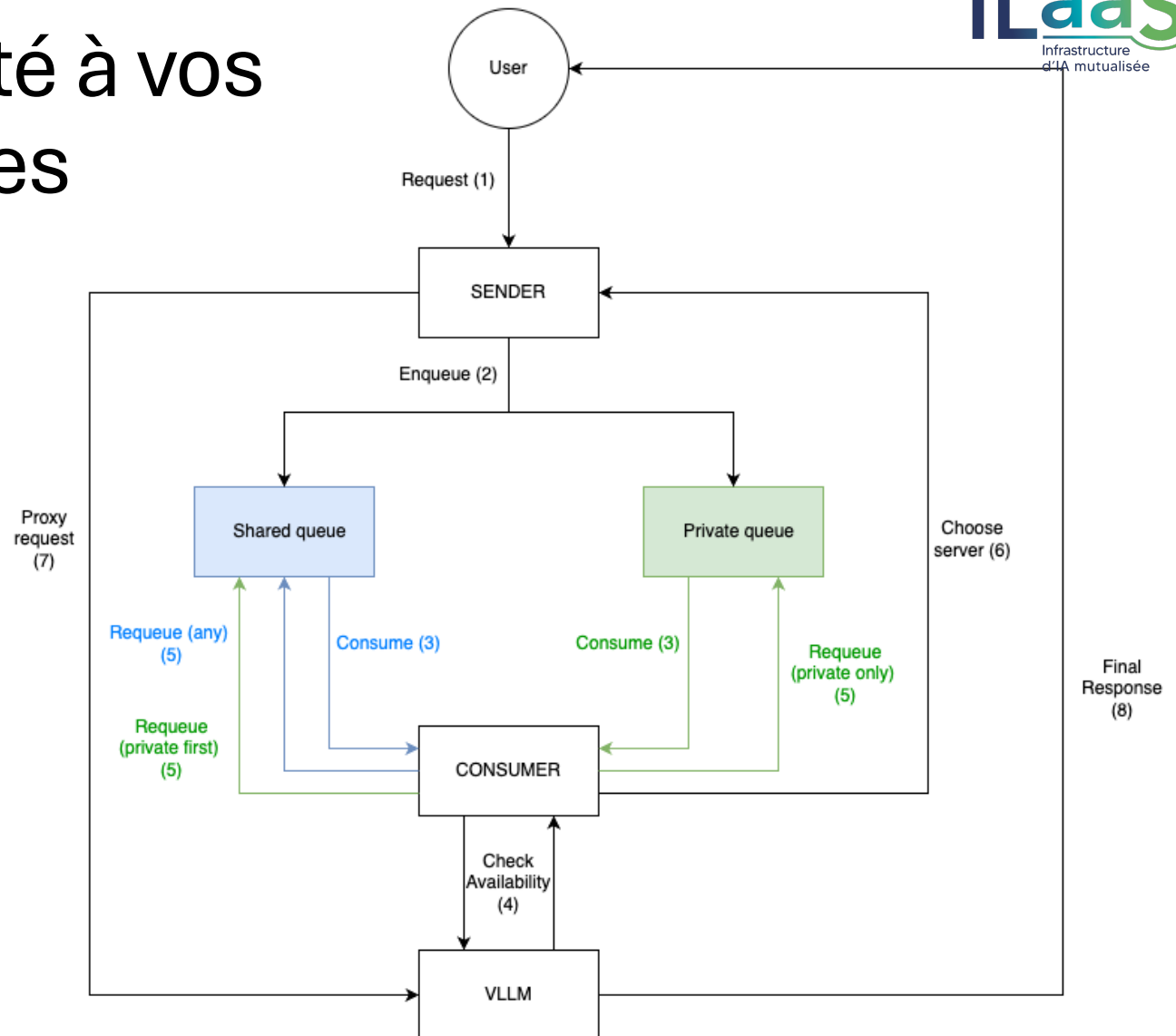
- Rampe de ~1mn pour passer de 0 à 50 utilisateurs simultanés
- Plateau à 50 utilisateurs actifs pendant ~3mn
 - connexion ouverte, nouvelle requête à chaque réponse obtenue
- Rampe finale diminuant le nombre total d'utilisateurs
 - Pas de nouvelle requête après la dernière réponse

➔ permet de combiner usages interactifs et batch



Accéder en priorité à vos propres ressources

- Vous fournissez de la puissance d'inférence...
 - Vous utilisez llm.ilaas.fr...
- ➔ Vous avez un coupe-file pour un accès privilégié à vos serveurs



Hébergement et ajout/suppression de modèle

- Les composants Dispatcher sont hébergés dans un cluster Kubernetes (actuellement OVH , hébergement ESR en cours d'étude)
 - Sender, Consumer, BdD, Dashboards
- Ressources de calcul sont hébergées dans les DC Labellisés ESR
 - Pour mutualiser un serveur d'inférence
 - Remplir un formulaire technique: type de serveur, nom du modèle, quantification, taille de la fenêtre, option de tools-calling, URL d'accès et clé d'accès
 - Ouverture de flux @IP des nœuds du cluster Kubernetes ILaaS vers le serveur
 - Déploiement de la nouvelle configuration via Helm



Hébergement et ajout/suppression de modèle

- Values de déploiement ILaaS (Kubernetes)

`models:`

```
- name: mistral31-24b
  model: "mistralai/Mistral-Small-3.1-24B-Instruct-2503"
  vllm_servers:
    - url: https://dc1.univ-xxx.fr/vllm
      token: <secret>
- url: https://dc1.univ-xxx.fr/vllm
  token: <secret>
  routing_strategy: least-busy
```



Least-Busy limité à
des serveurs vLLM
actuellement

```
- name: llama31-8b2
  model: "meta-llama/Llama-3.1-8B-Instruct"
  vllm_servers:
    - url: https://dc2.univ-yyy.fr
      token: <secret>
  routing_strategy: round-robin
```



Indépendant des
serveurs

Conséquence ajout/suppression dynamique de modèles

- Gouvernance nécessaire sur les modèles
 - Choix d'un ensemble de modèles pour couvrir les fonctionnalités
 - LLM généraliste
 - LLM multimodal
 - LLM de raisonnement
 - LLM de tool calling
 - Choisir les paramétrages en fonction des usages
 - Quantification,
 - Taille de la fenêtre de contexte, ...
 - Offrir une continuité de service
 - Disposer du même modèle sur plusieurs DC

Et bientôt...

- Instanciation d'un nouveau portail d'API pour
 - L'embedding
 - La gestion de collections de documents (base de données Qdrant)
 - L'OCR
- Basé sur la technologie OpenGateLLM qui propulse AlbertAPI (DINUM)
<https://github.com/etalab-ia/OpenGateLLM>
- Et à terme, ajout/suppression dynamique de modèles
 - Possibilité d'ajouter un serveur d'inférence pour les membres du consortium authentifiés
 - Gestion des « modèles critiques » (service minimum)



Sous le capot du service d'inférence STT

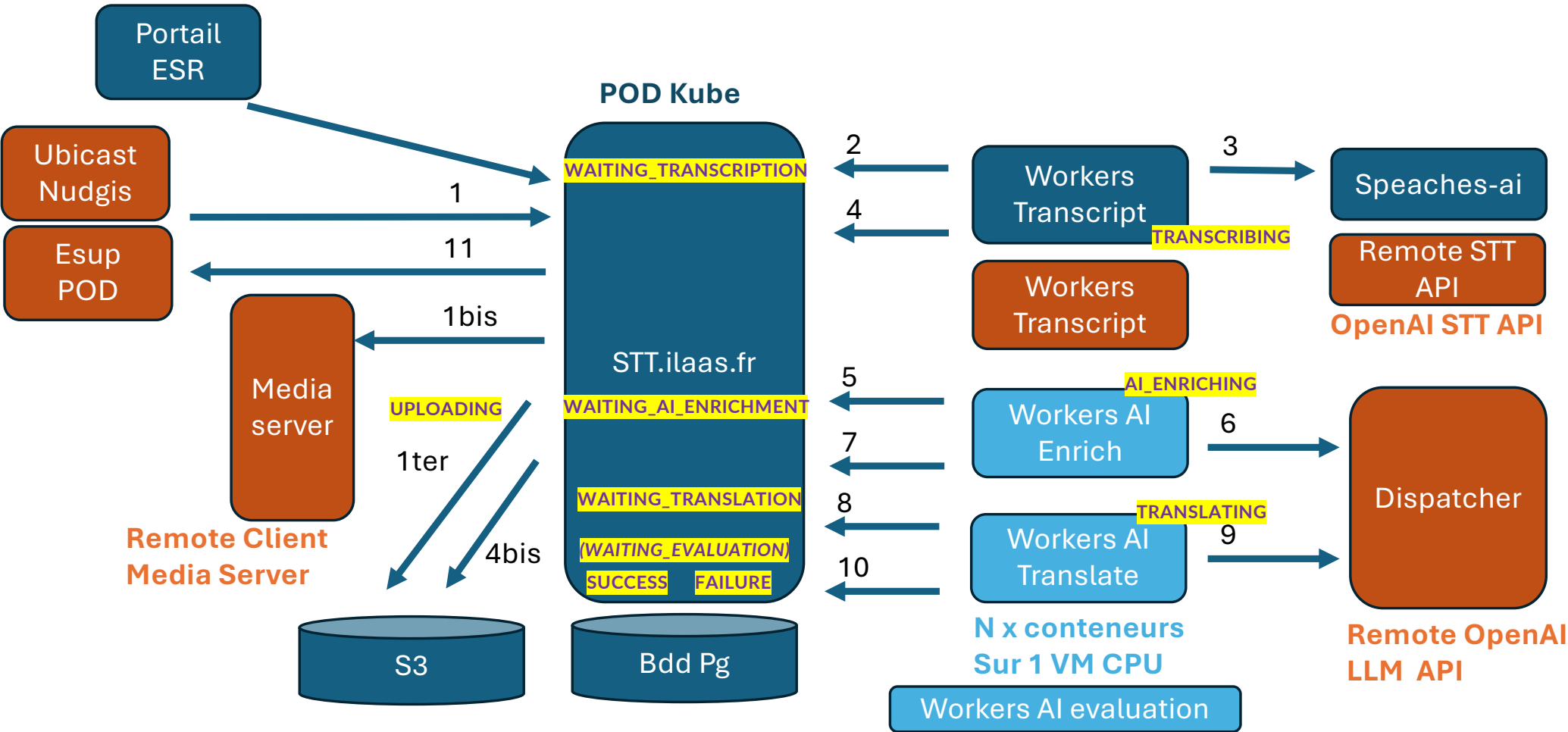
[STT.ilaas.fr](https://stt.ilaas.fr)

Retranscription audio/vidéo et plus

- Service de retranscription asynchrone: produit des transcripts et sous-titrages (VTT, SRT)
- Permet aussi:
 - La traduction
 - L'utilisation d'un LLM pour générer optionnellement
 - des métadonnées (titre, description, mots clés)
 - Des comptes rendus ou des quiz
- Basé sur la technologie Aristote
- Hébergé dans un cluster Kubernetes (actuellement chez OVH)



STT.ilaas.fr





Questions