



DEVOX™ France



TRANSCENDEZ LES FRONTIÈRES LINGUISTIQUES

avec des APIs de Machine Learning sur mesure



 OVHcloud



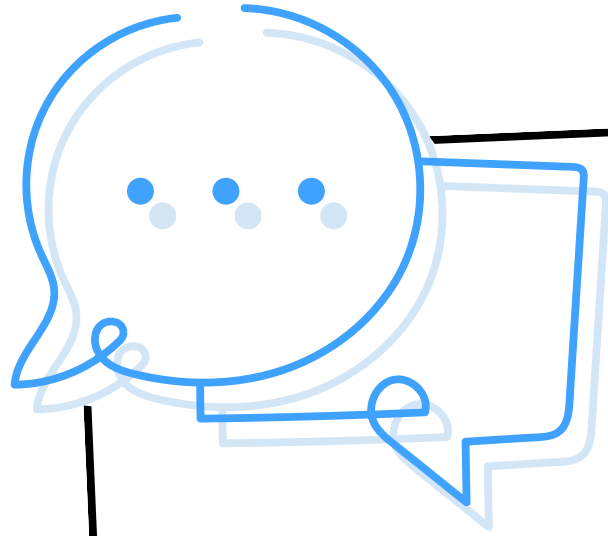
PRÉSENTATION

- Machine Learning Engineer
- OVHcloud
- AI Solutions Team



ÉLÉA PETTON





ÇA VOUS DIT ?

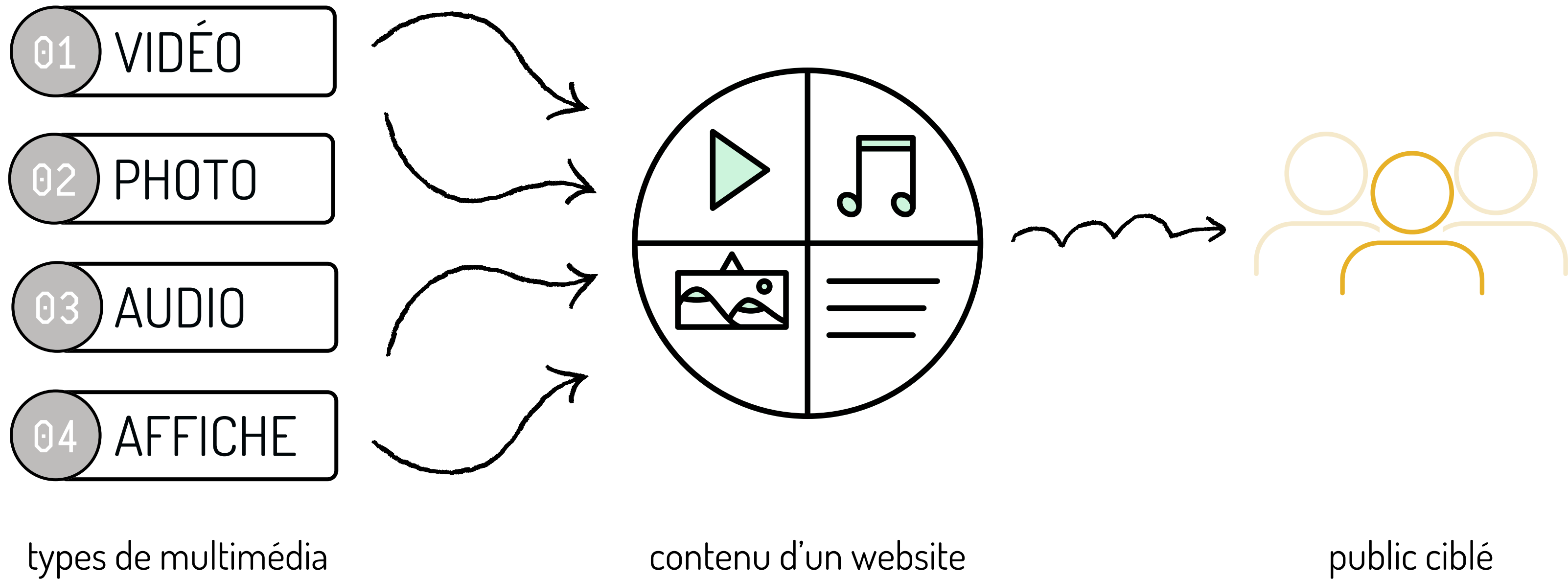
“ Embarquez dans le développement d’une solution de transcription temps réel de vos contenus multimédia...” ”



INTRODUCTION

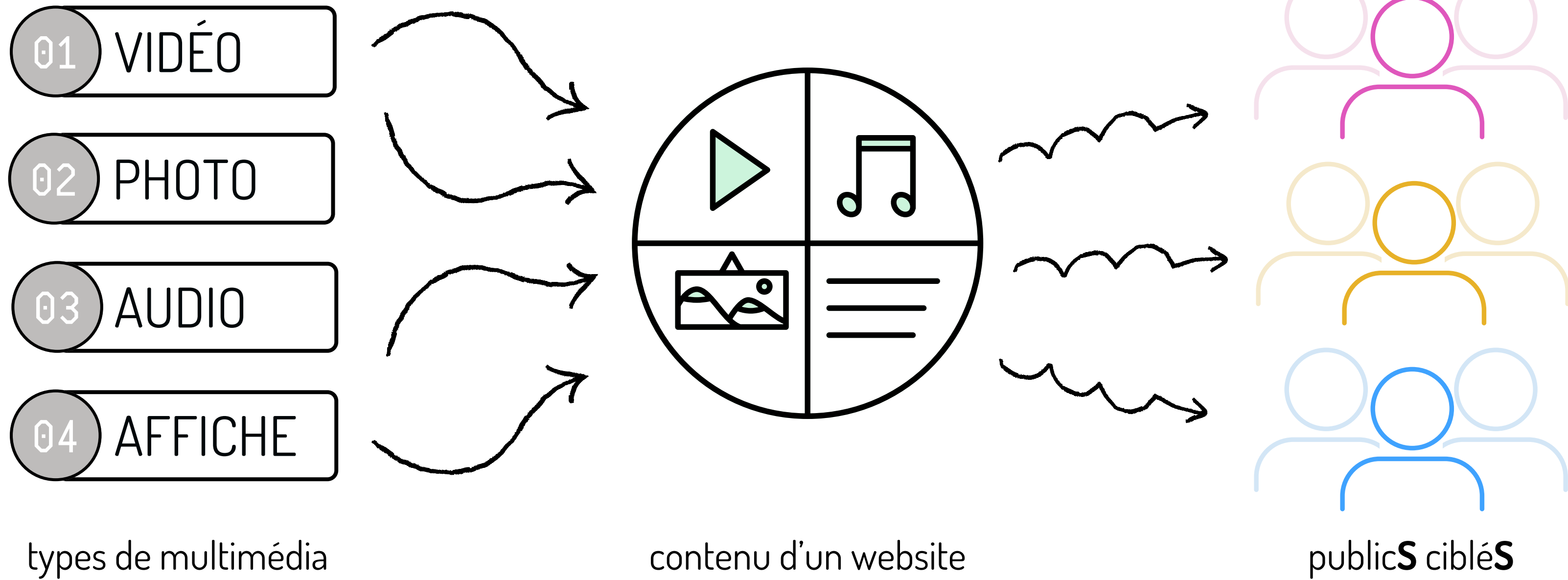
INTRODUCTION

“ Quels sont les différents types de contenus multimédia et leur(s) public(s) ? ”



INTRODUCTION

“ Quels sont les différents types de contenus multimédia et leur(s) public(s) ? ”



TRANSFORMER SES CONTENUS MULTIMEDIA, QU'EST-CE QUE ÇA VEUT DIRE ?

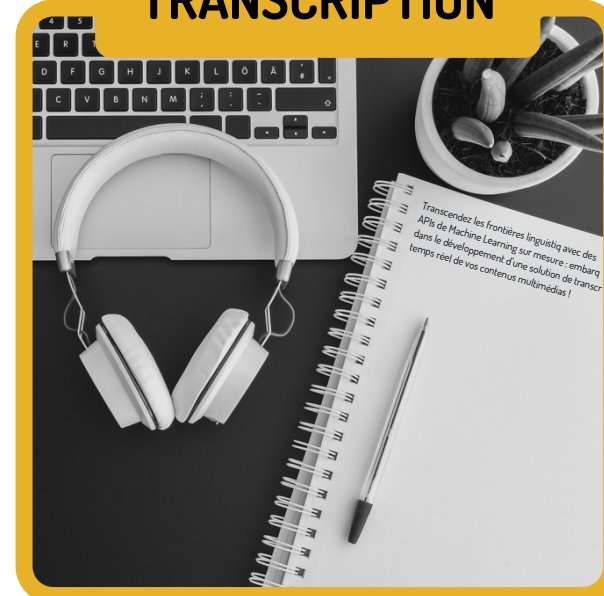
TRADUCTION



Changer la langue de...

- sa page web
- son post Twitter, LinkedIn, ...
- ses slides

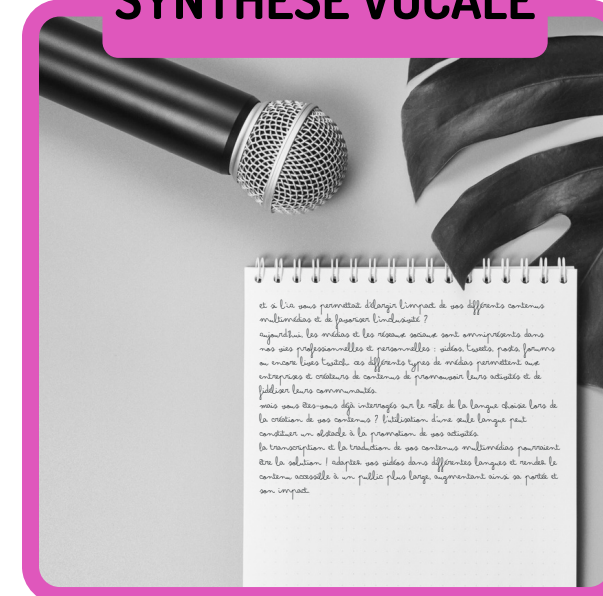
TRANSCRIPTION



Passer de l'oral à l'écrit pour...

- sous-titrer des vidéos, podcasts
- garder le contenu d'une réunion

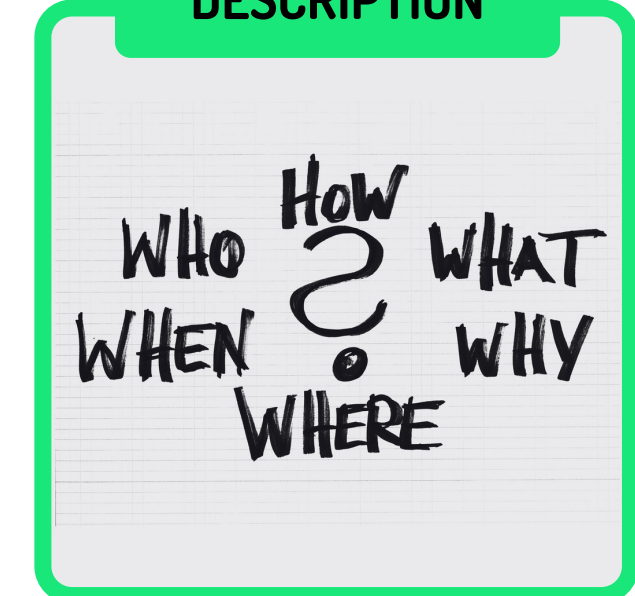
SYNTHÈSE VOCALE



Passer de l'écrit à l'oral pour...

- favoriser l'accessibilité
- doubler les voix

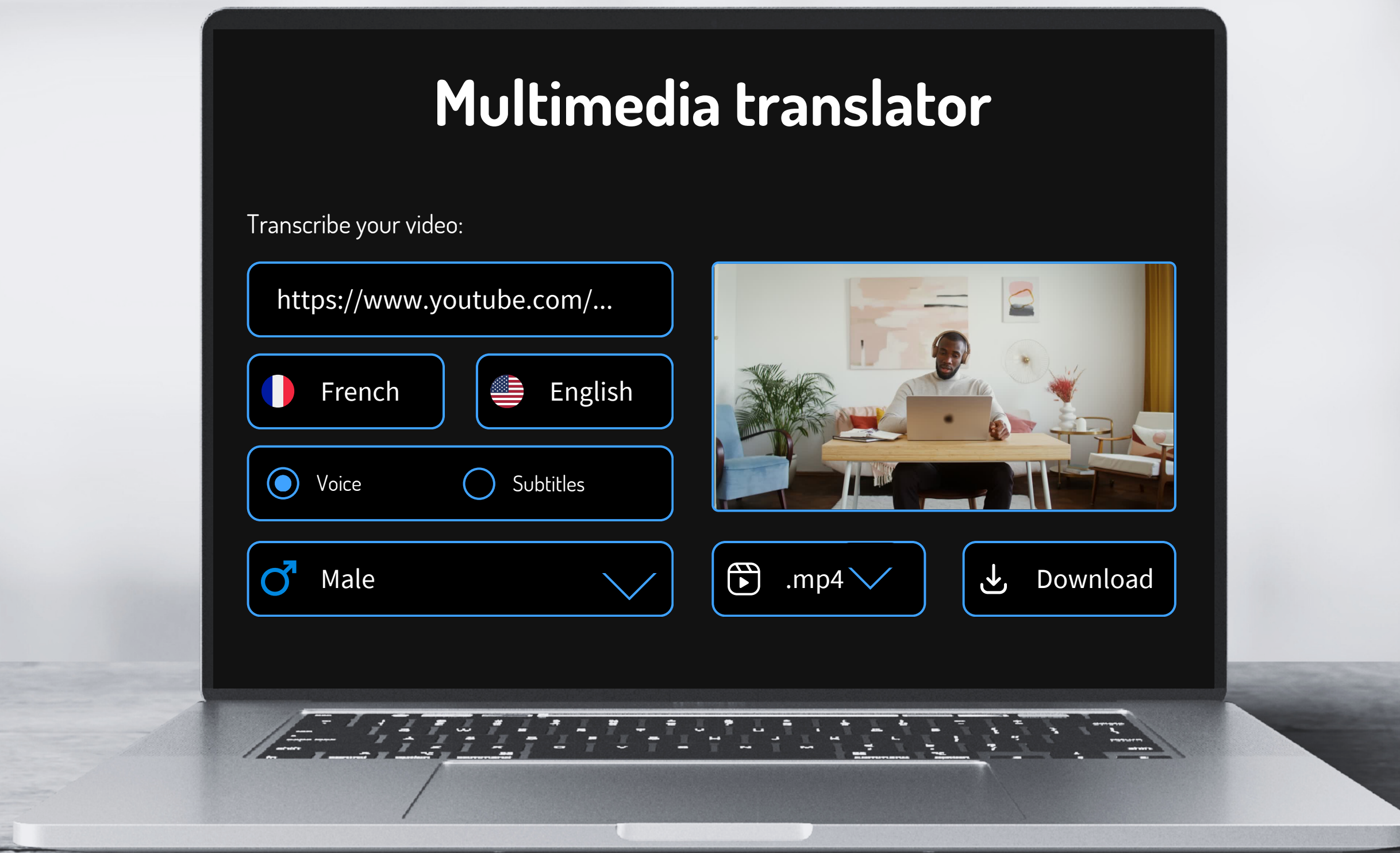
DESCRIPTION



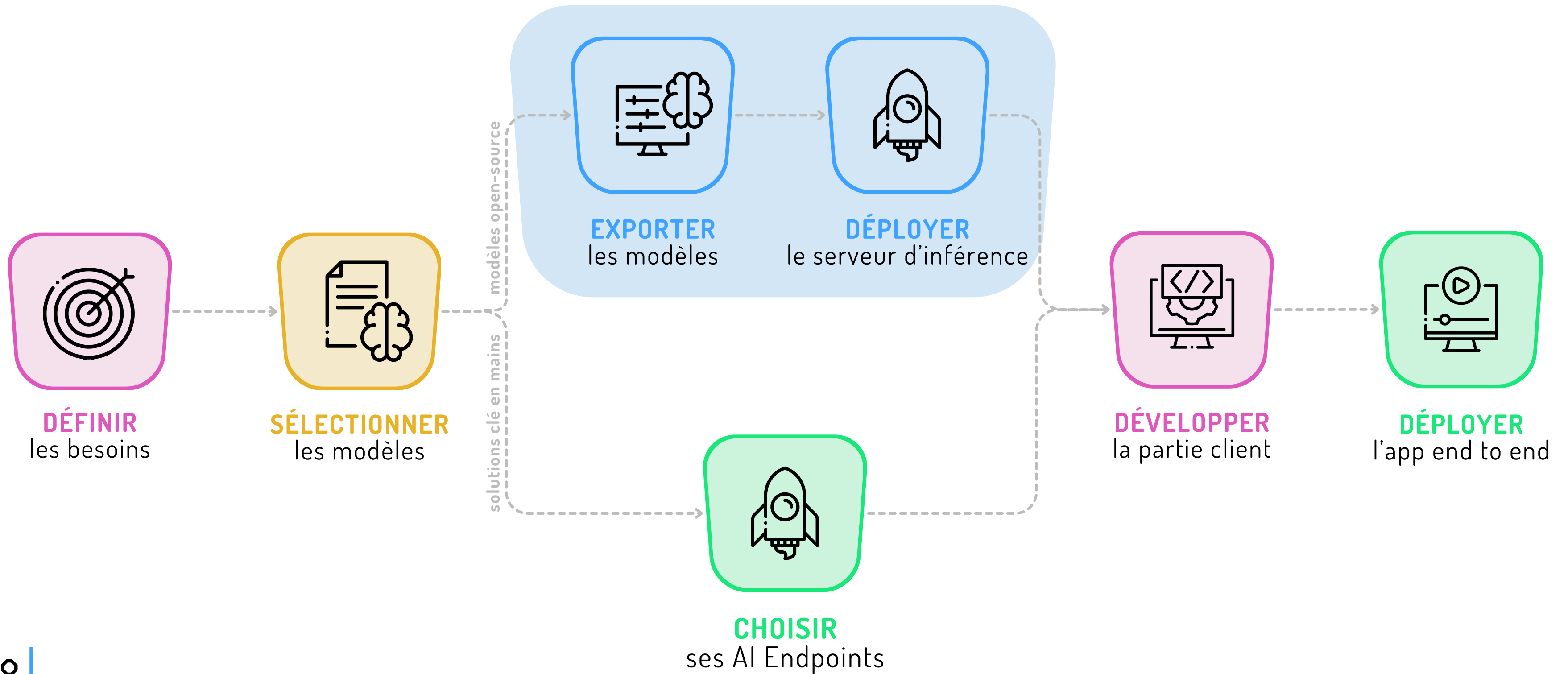
Décrire ou résumer...

- une vidéo, un podcast
- le contenu d'une réunion
- une documentation

ET EN PRATIQUE, ÇA DONNE QUOI ?



OBJECTIFS





DÉFINIR

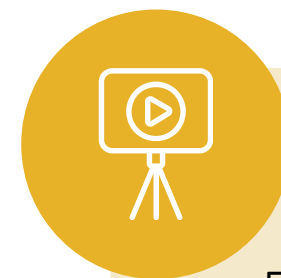
les besoins

QUELS SONT NOS PERSONNAGES ?



Se développer à l'international

- Traduire du contenu multimedia pour un usage multilingue
- Adapter les vidéos pour un public mondial



Élargir son public...

- En tant que créateur de contenu, j'aimerais **élargir ma communauté**
- Créer et publier un **contenu plus attrayant**



Transcrire les meetings

- **Retranscrire** les meetings à l'écrit
- **Transcrire** les meetings dans une autre langue
- **Résumer** les meetings



Être plus inclusif !

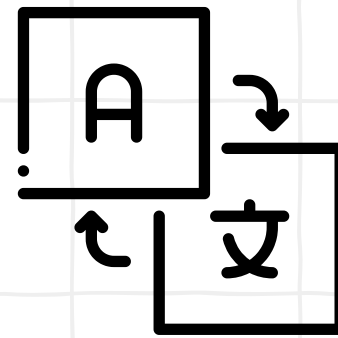
- **Sous-titrer les vidéos** pour les personnes malentendantes
- **Inclure les personnes** qui ne parlent pas la même langue

3 TÂCHES CIBLES

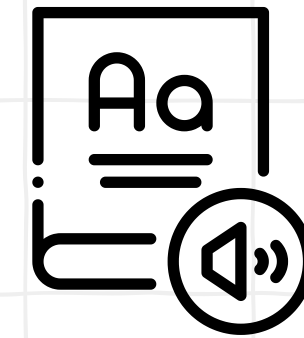
AUDIO
recognition



TEXT
translation



SPEECH
synthesis

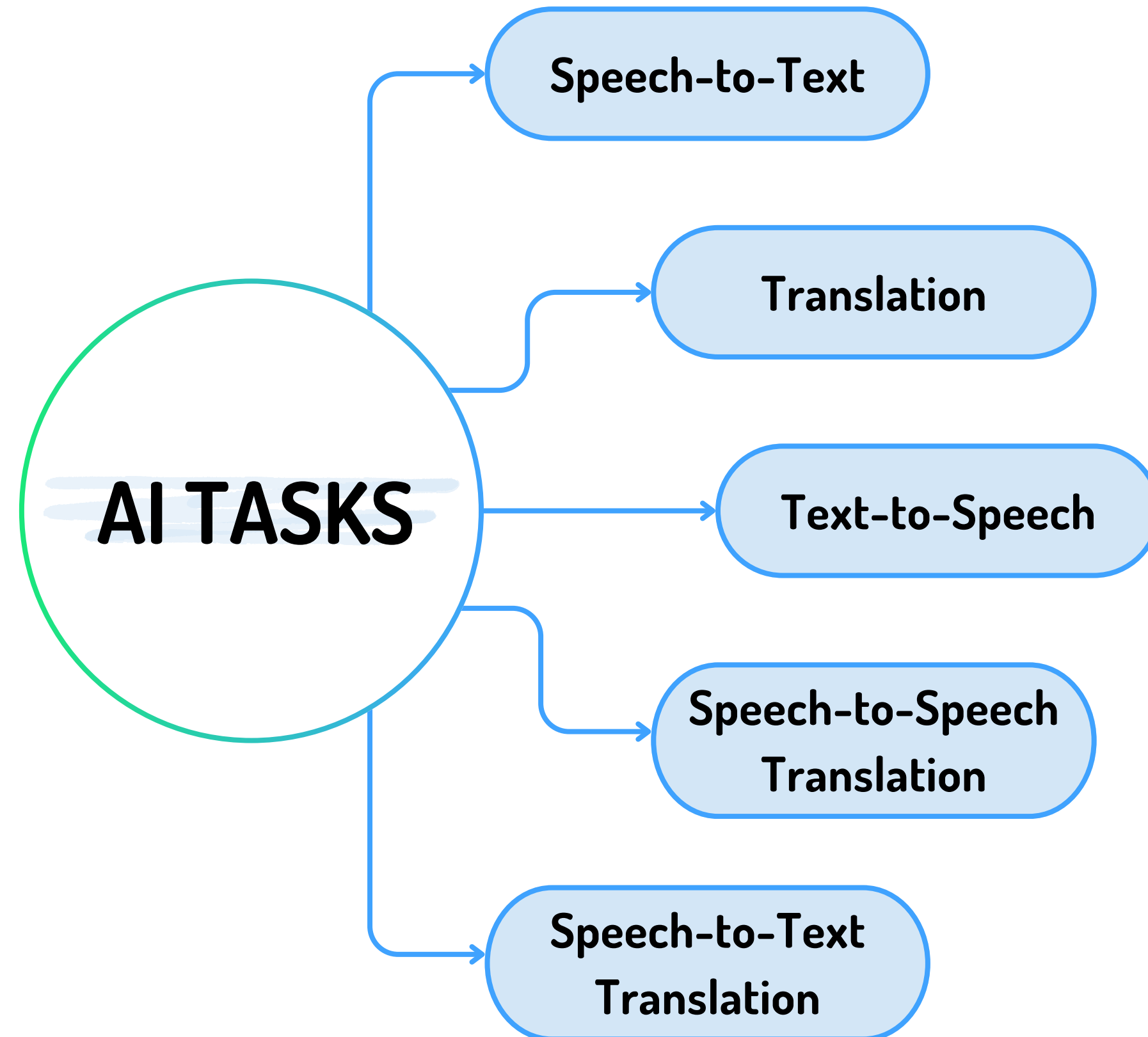




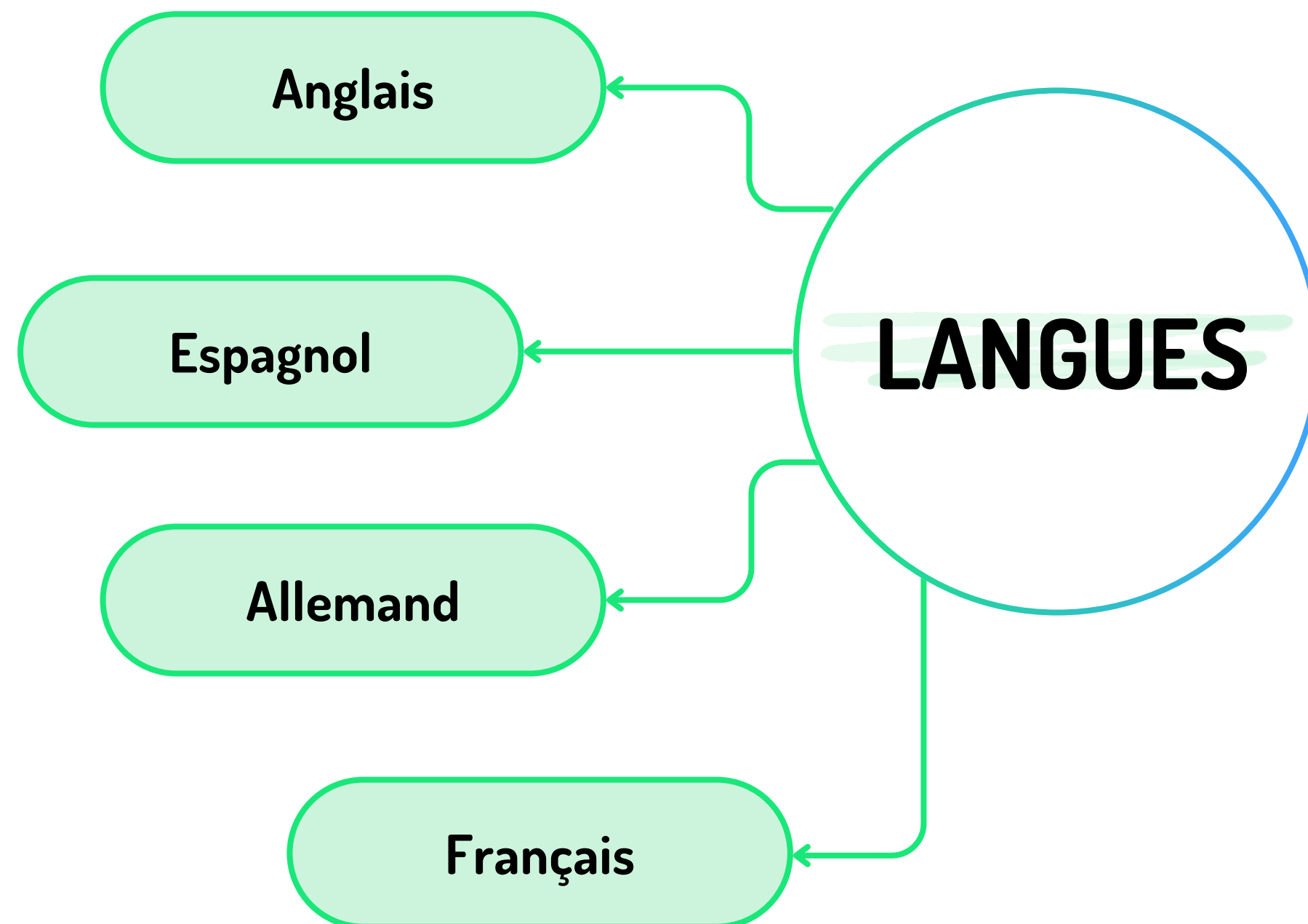
SÉLECTIONNER

les modèles

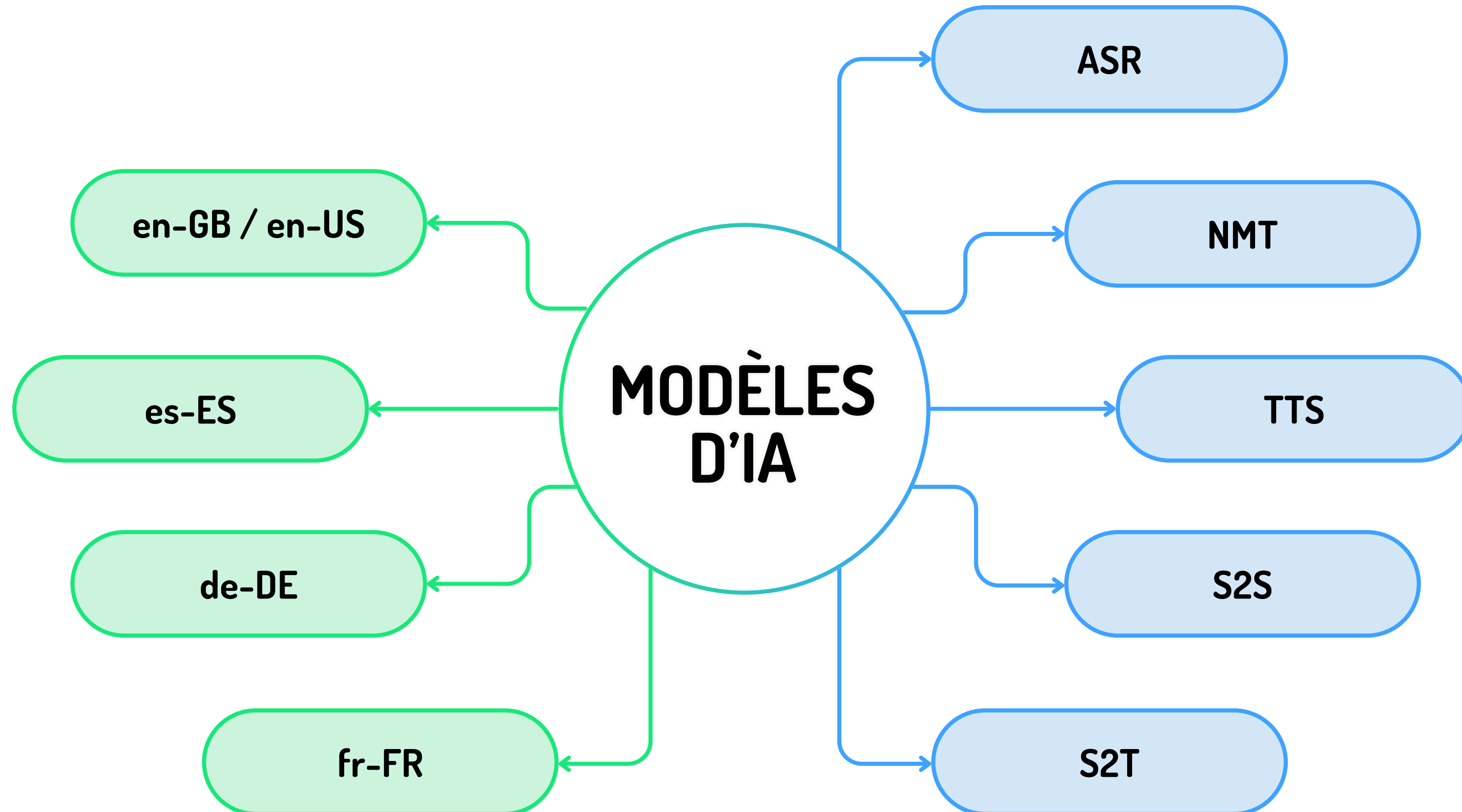
SÉLECTIONNER LES MODÈLES



SÉLECTIONNER LES MODÈLES



SÉLECTIONNER LES MODÈLES



COMMENT ACCÉDER À CES MODÈLES ?

Définir le but de ces modèles

“ Le besoin ici n'est pas d'entraîner les modèles mais bien d'en faire l'**inférence**. Cela implique déployer les modèles de manière optimisée : coûts minimum et meilleur ratio précision / latence ”

Prendre des modèles Open Source

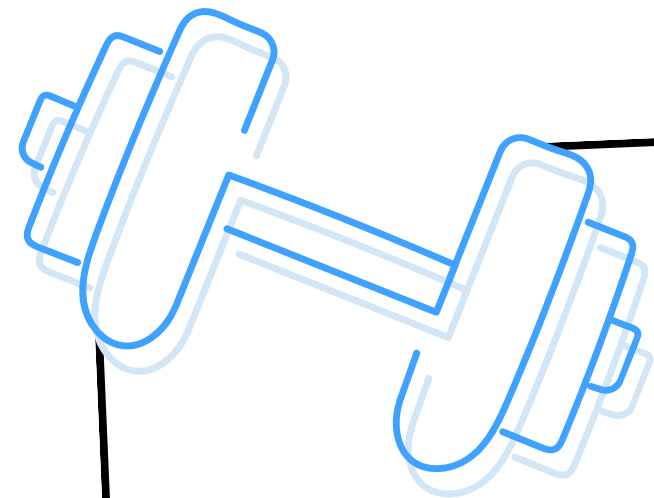
- ⊕ Accès à de nombreux modèles
- ⊕ Possibilité de “personnalisation”
- ⊕ Prédicibilité des coûts
- ⊖ Complexité du déploiement
- ⊖ Gestion du RUN
- ⊖ Licences à non usage commercial

Sélectionner des solutions clés en main

- ⊕ Endpoints API à disposition
- ⊕ Simplicité d'utilisation (call APIs)
- ⊕ Performances optimales
- ⊕ Pas de gestion du RUN
- ⊖ Coût variable - solution “Pay per call”
- ⊖ Peu de possibilité de personnalisation

- ✓ Connaissances en ML
- ✓ Notion d'optimisation de modèles
- ✓ Des GPUs à disposition
-
- ✓ & du temps !

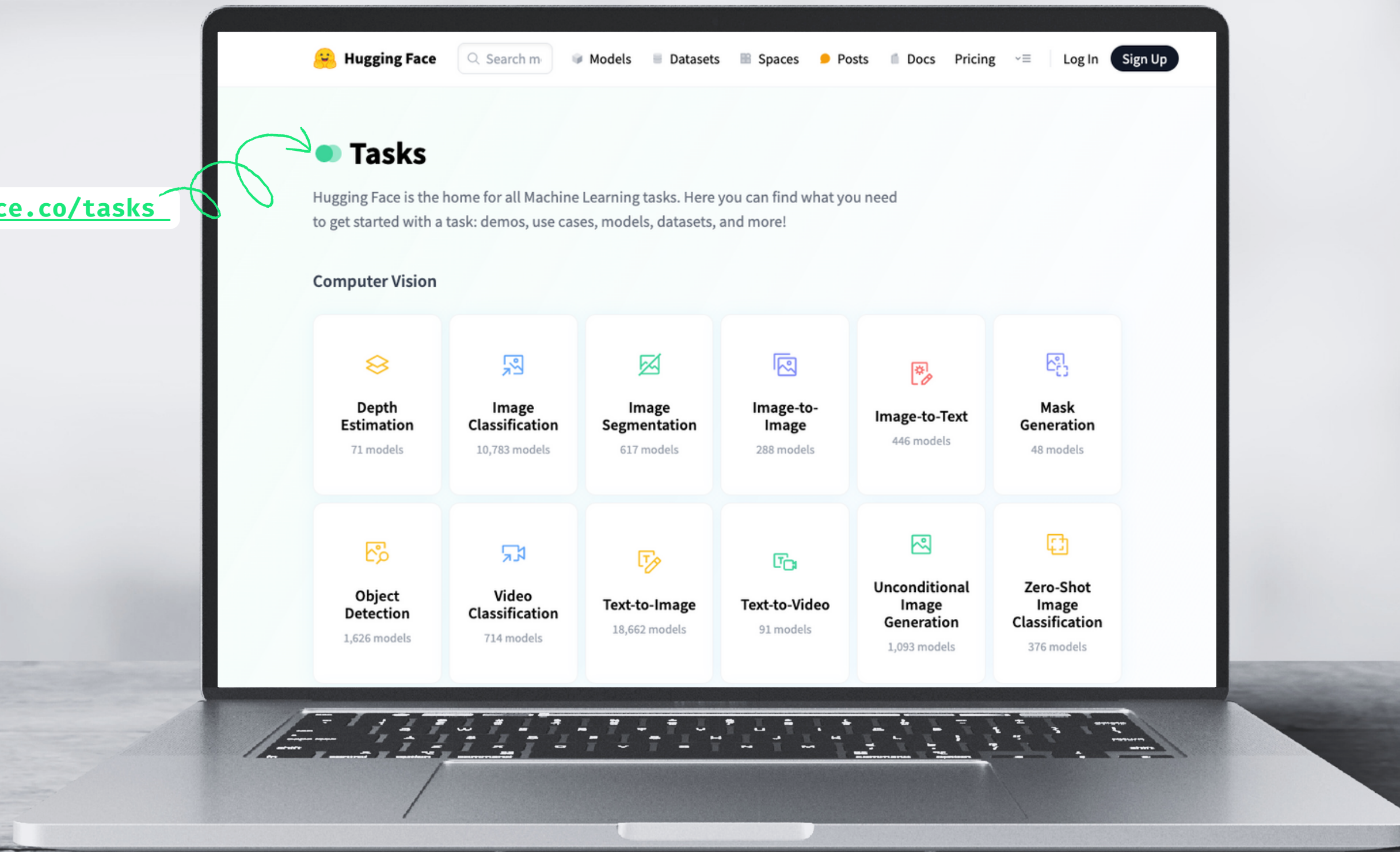
- ✓ Compréhension des paramètres d'entrée du modèle
- ✓ Appropriation des sorties du modèle
-
- ✓ & des connaissances en Puzzles !



UTILISER l'Open Source

COMMENT TROUVER CES MODÈLES ?

<https://huggingface.co/tasks>



HUGGING FACE TASKS



**Automatic
Speech
Recognition**

15,482 models

ASR



Translation

3,441 models

NMT

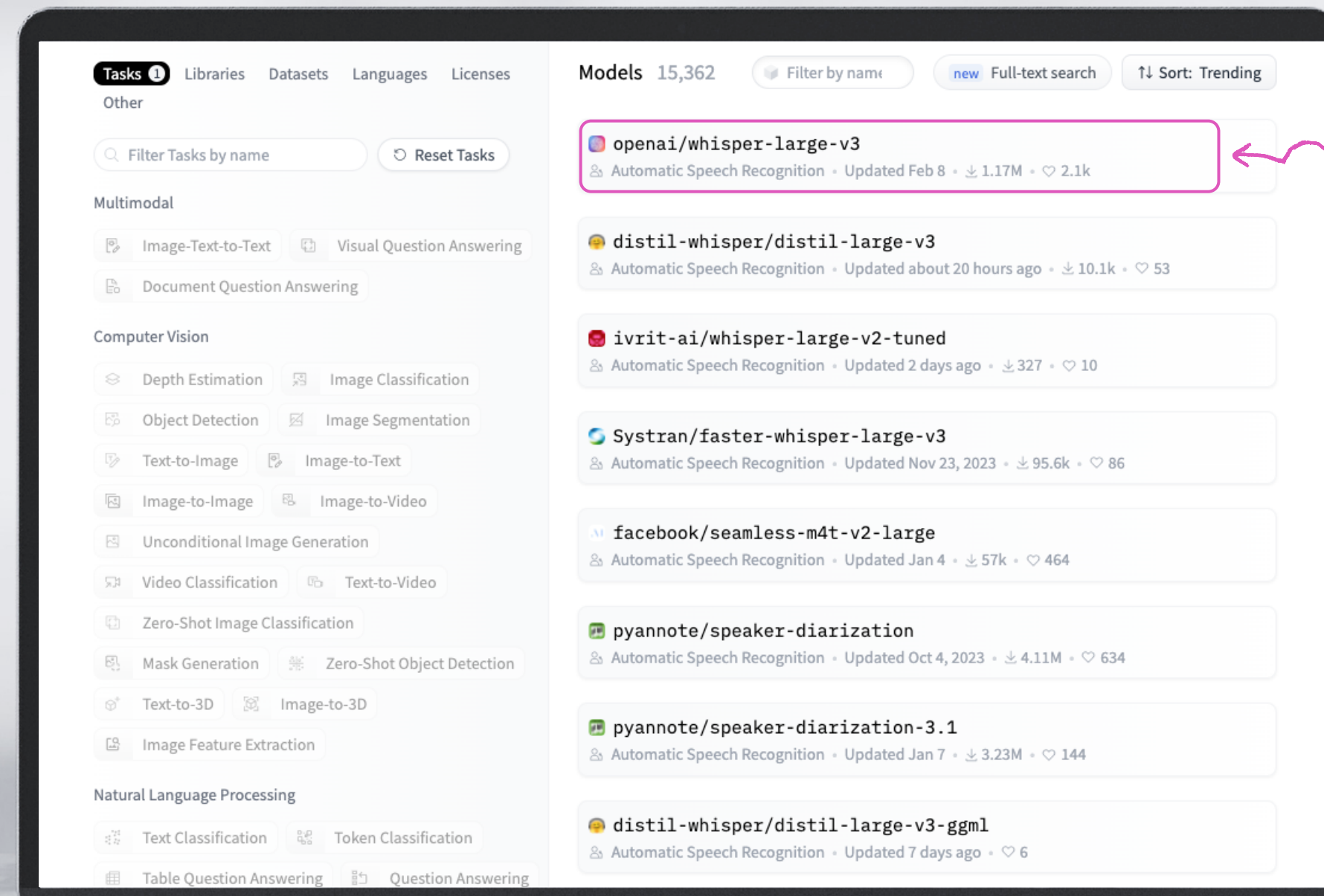


Text-to-Speech

1,849 models

TTS

AUTOMATIC SPEECH RECOGNITION



The screenshot shows the Hugging Face Models page for the task "Automatic Speech Recognition". The left sidebar lists various tasks, and the main content area displays a list of models. The model "openai/whisper-large-v3" is highlighted with a pink box and a callout arrow.

Tasks 1 Libraries Datasets Languages Licenses

Other

Filter Tasks by name Reset Tasks

Multimodal

- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Text-to-Image
- Image-to-Text
- Image-to-Image
- Image-to-Video
- Unconditional Image Generation
- Video Classification
- Text-to-Video
- Zero-Shot Image Classification
- Mask Generation
- Zero-Shot Object Detection
- Text-to-3D
- Image-to-3D
- Image Feature Extraction

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering

Models 15,362 Filter by name new Full-text search Sort: Trending

- openai/whisper-large-v3**
Automatic Speech Recognition · Updated Feb 8 · ↓ 1.17M · ♥ 2.1k
- distil-whisper/distil-large-v3
Automatic Speech Recognition · Updated about 20 hours ago · ↓ 10.1k · ♥ 53
- ivrit-ai/whisper-large-v2-tuned
Automatic Speech Recognition · Updated 2 days ago · ↓ 327 · ♥ 10
- Systran/faster-whisper-large-v3
Automatic Speech Recognition · Updated Nov 23, 2023 · ↓ 95.6k · ♥ 86
- facebook/seamless-m4t-v2-large
Automatic Speech Recognition · Updated Jan 4 · ↓ 57k · ♥ 464
- pyannote/speaker-diarization
Automatic Speech Recognition · Updated Oct 4, 2023 · ↓ 4.11M · ♥ 634
- pyannote/speaker-diarization-3.1
Automatic Speech Recognition · Updated Jan 7 · ↓ 3.23M · ♥ 144
- distil-whisper/distil-large-v3-ggml
Automatic Speech Recognition · Updated 7 days ago · ♥ 6

openai/whisper-large-v3

TRANSLATION

The screenshot shows the Hugging Face Models page for translation tasks. The left sidebar contains navigation tabs: Tasks (1), Libraries, Datasets, Languages, Licenses, and Other. Below these are search filters and task categories: Multimodal (Image-Text-to-Text, Visual Question Answering, Document Question Answering), Computer Vision (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Text-to-Image, Image-to-Text, Image-to-Image, Image-to-Video, Unconditional Image Generation, Video Classification, Text-to-Video, Zero-Shot Image Classification, Mask Generation, Zero-Shot Object Detection, Text-to-3D, Image-to-3D, Image Feature Extraction), and Natural Language Processing (Text Classification, Token Classification, Table Question Answering, Question Answering). The main content area shows a list of models under the 'Models 3,436' heading, with filters for 'Filter by name', 'new Full-text search', and 'Sort: Trending'. The models listed are:

- google-t5/t5-small (Translation, Updated Jun 30, 2023, 4.04M, 232)
- google-t5/t5-base** (Translation, Updated Feb 14, 2.83M, 423) - highlighted with a yellow box and arrow
- facebook/nllb-200-distilled-600M (Translation, Updated Feb 14, 466k, 333)
- Helsinki-NLP/opus-mt-zh-en (Translation, Updated Aug 16, 2023, 2.24M, 353)
- utrobinmv/t5_translate_en_ru_zh_large_1024 (Translation, Updated 15 days ago, 32.1k, 27)
- webbigdata/C3TR-Adapter (Translation, Updated 3 days ago, 561, 13)
- Helsinki-NLP/opus-mt-en-zh (Translation, Updated Aug 16, 2023, 98k, 271)
- facebook/nllb-200-3.3B (Translation, Updated Feb 11, 2023, 20.6k, 172)

google-t5/t5-base

TEXT TO SPEECH

The screenshot shows the Hugging Face Models page for Text-to-Speech models. The left sidebar contains navigation tabs for Tasks, Libraries, Datasets, Languages, and Licenses. The main content area displays a list of models, with the top model, **coqui/XTTS-v2**, highlighted by a blue box and a callout bubble. The callout bubble contains the text **coqui/XTTS-v2**. Below the highlighted model, other models are listed, including **suno/bark**, **microsoft/speecht5_tts**, **metavoicelo/metavoicelo-1B-v0.1**, **facebook/mms-tts**, **facebook/mms-tts-eng**, **collabora/whisperspeech**, and **facebook/mms-tts-tam**. The right sidebar shows filters for Models (1,844), Filter by name, Full-text search, and Sort: Trending.

Tasks 1 Libraries Datasets Languages Licenses

Other

Filter Tasks by name Reset Tasks

Multimodal

Image-Text-to-Text Visual Question Answering

Document Question Answering

Computer Vision

Depth Estimation Image Classification

Object Detection Image Segmentation

Text-to-Image Image-to-Text

Image-to-Image Image-to-Video

Unconditional Image Generation

Video Classification Text-to-Video

Zero-Shot Image Classification

Mask Generation Zero-Shot Object Detection

Text-to-3D Image-to-3D

Image Feature Extraction

Natural Language Processing

Text Classification Token Classification

Table Question Answering Question Answering

Models 1,844 Filter by name new Full-text search Sort: Trending

coqui/XTTS-v2
Text-to-Speech · Updated Dec 11, 2023 · ↓ 413k · ♥ 965

suno/bark
Text-to-Speech · Updated Oct 4, 2023 · ↓ 74.3k · ♥ 811

microsoft/speecht5_tts
Text-to-Speech · Updated Nov 8, 2023 · ↓ 304k · ♥ 480

metavoicelo/metavoicelo-1B-v0.1
Text-to-Speech · Updated 10 days ago · ↓ 2.35k · ♥ 654

facebook/mms-tts
Text-to-Speech · Updated Jul 25, 2023 · ♥ 91

facebook/mms-tts-eng
Text-to-Speech · Updated Sep 6, 2023 · ↓ 673k · ♥ 93

collabora/whisperspeech
Text-to-Speech · Updated 6 days ago · ♥ 97

facebook/mms-tts-tam
Text-to-Speech · Updated Feb 19 · ↓ 720 · ♥ 5

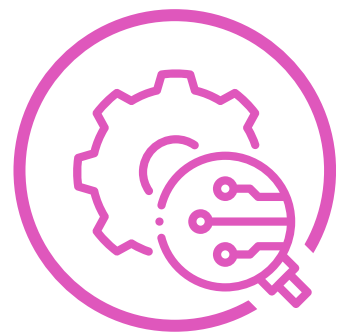
coqui/XTTS-v2

COMMENT DÉPLOYER CES MODÈLES ?



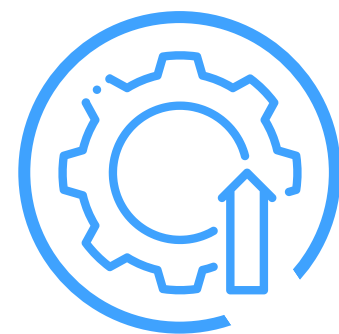
ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



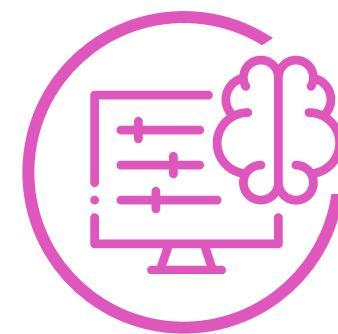
EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



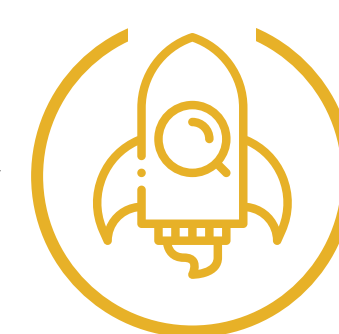
FORMATER

le code et le repo
de modèles



STOCKER

les modèles dans
un bucket S3



DÉPLOYER

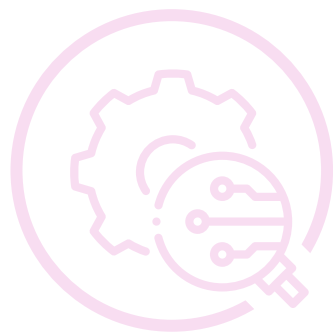
le serveur d'inférence
dans le cloud

COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



FORMATER

le code et le repo
de modèles



STOCKER

les modèles dans
un bucket S3



DÉPLOYER

le serveur d'inférence
dans le cloud

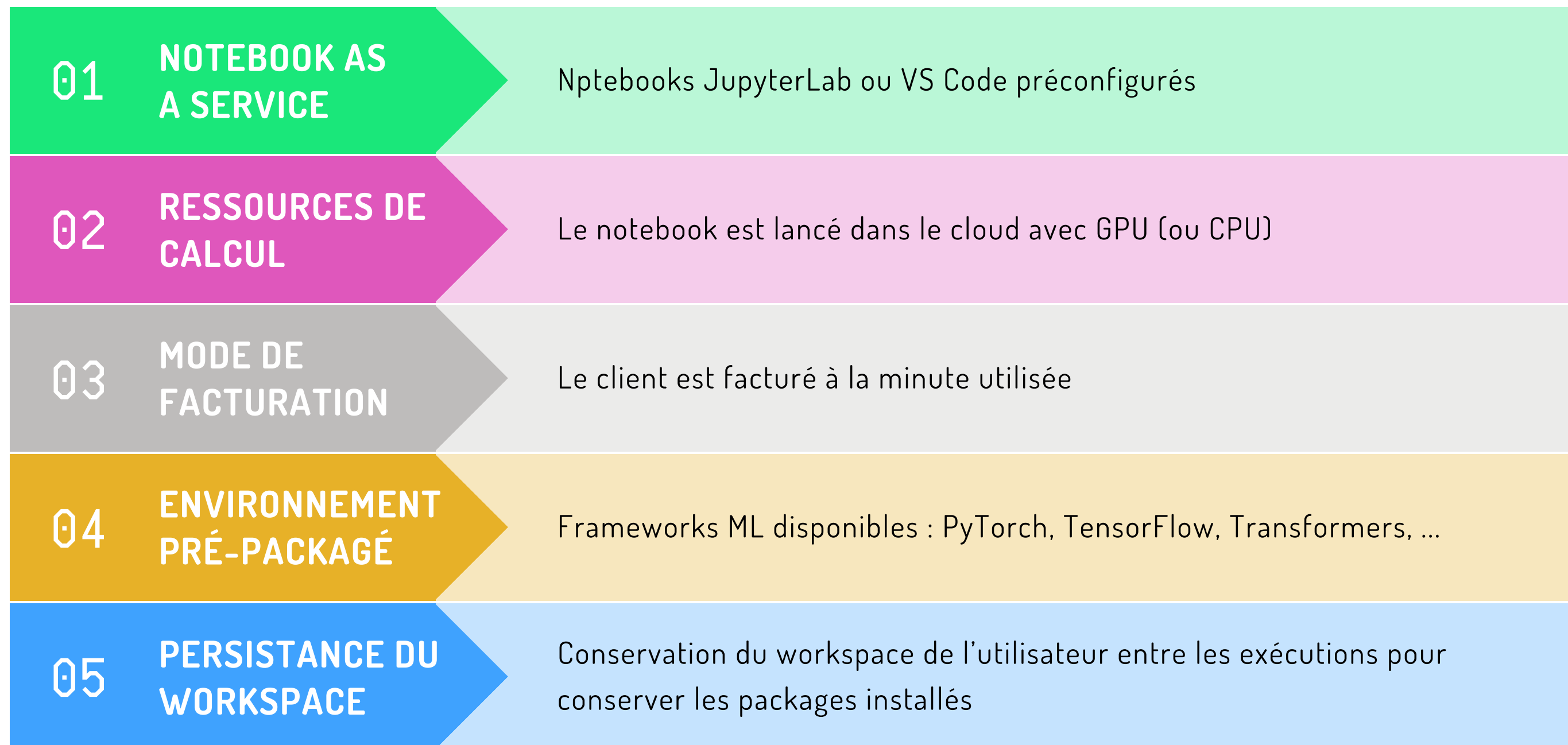


ACCÉDER à un notebook

ACCÉDER À UN NOTEBOOK AVEC GPU_s



OVHcloud - AI NOTEBOOKS

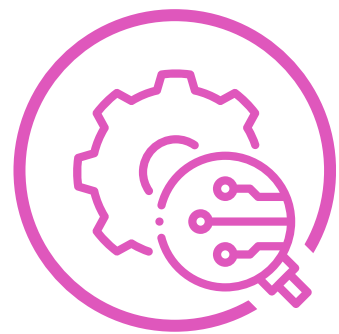


COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



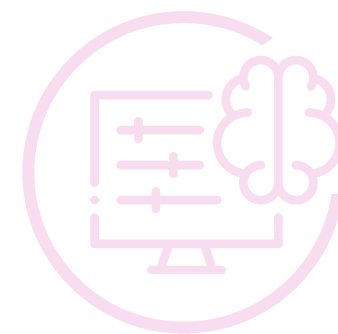
EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



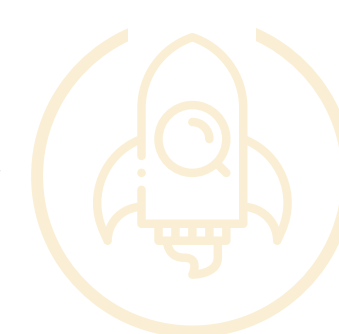
FORMATER

le code et le repo
de modèles



STOCKER

les modèles dans
un bucket S3



DÉPLOYER

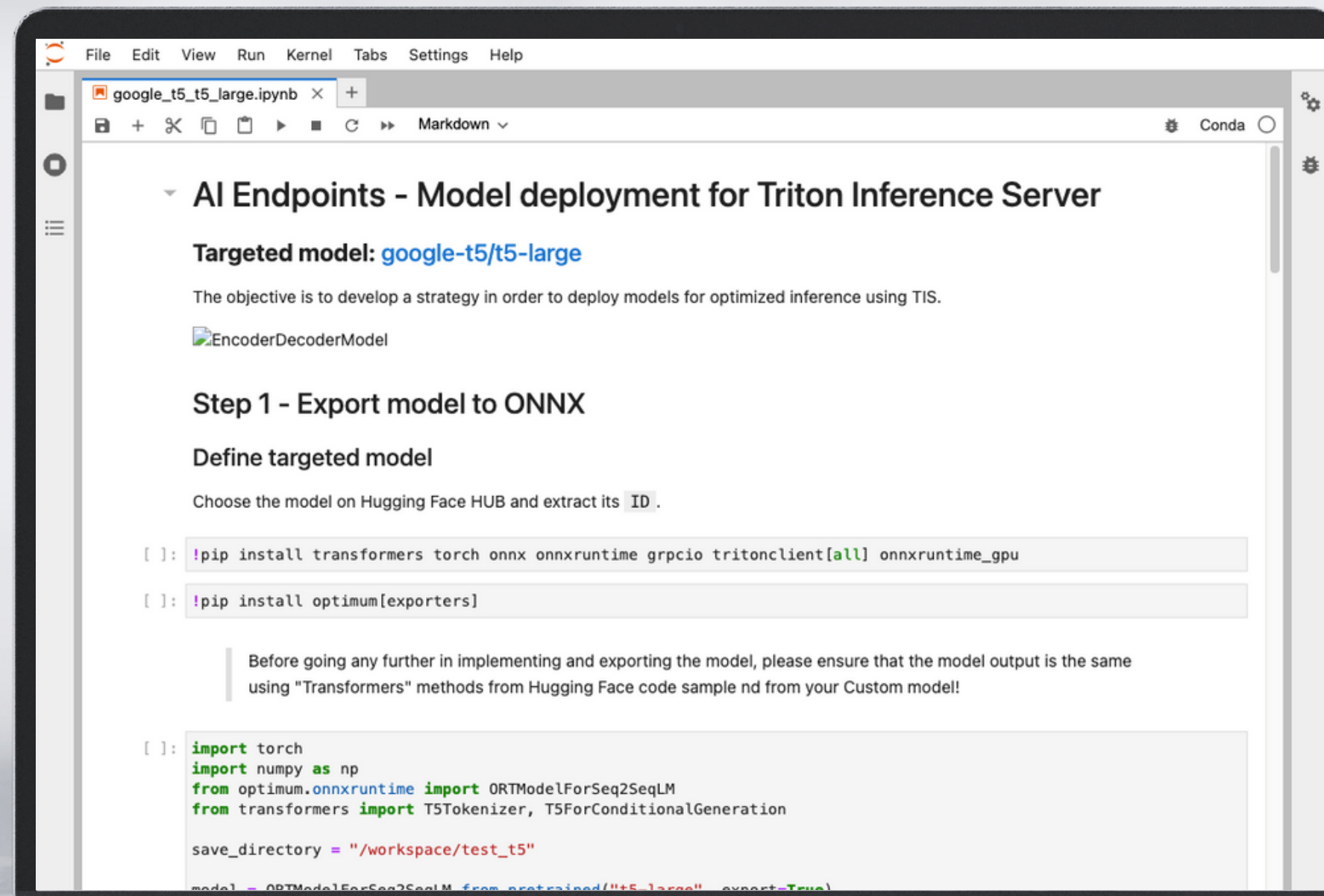
le serveur d'inférence
dans le cloud



TESTER

les modèles

TESTER LES MODÈLES LOCALEMENT




The image shows a laptop screen displaying a Jupyter Notebook. The notebook is titled "google_t5_t5_large.ipynb" and contains the following content:

```
File Edit View Run Kernel Tabs Settings Help
google_t5_t5_large.ipynb x +
+ ✂ 📄 ▶ ■ ↺ ⏪ ⏩ Markdown v Conda ○
```

AI Endpoints - Model deployment for Triton Inference Server

Targeted model: [google-t5/t5-large](#)

The objective is to develop a strategy in order to deploy models for optimized inference using TIS.

 EncoderDecoderModel

Step 1 - Export model to ONNX

Define targeted model

Choose the model on Hugging Face HUB and extract its ID.

```
[ ]: !pip install transformers torch onnx onnxruntime grpcio tritonclient[all] onnxruntime_gpu
```

```
[ ]: !pip install optimum[exporters]
```

Before going any further in implementing and exporting the model, please ensure that the model output is the same using "Transformers" methods from Hugging Face code sample and from your Custom model!

```
[ ]: import torch
import numpy as np
from optimum.onnxruntime import ORTModelForSeq2SeqLM
from transformers import T5Tokenizer, T5ForConditionalGeneration

save_directory = "/workspace/test_t5"

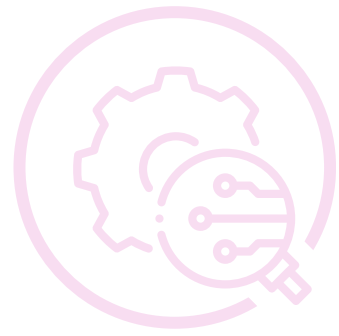
model = ORTModelForSeq2SeqLM.from_pretrained("t5-large", export=True)
```

COMMENT DÉPLOYER CES MODÈLES ?



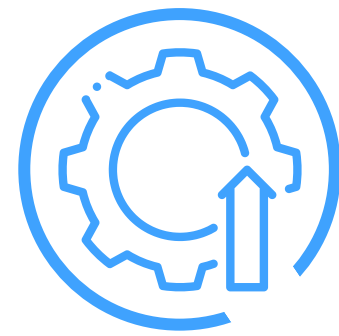
ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



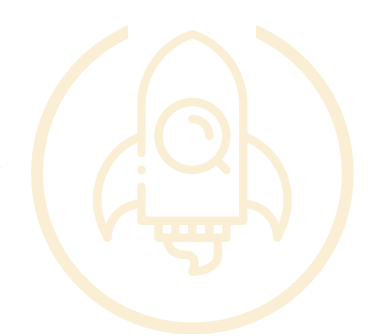
FORMATER

le code et le repo
de modèles



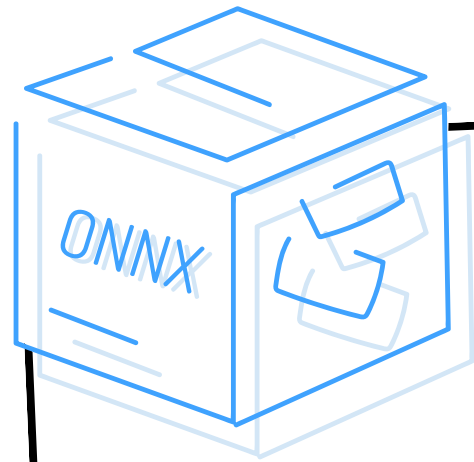
STOCKER

les modèles dans
un bucket S3



DÉPLOYER

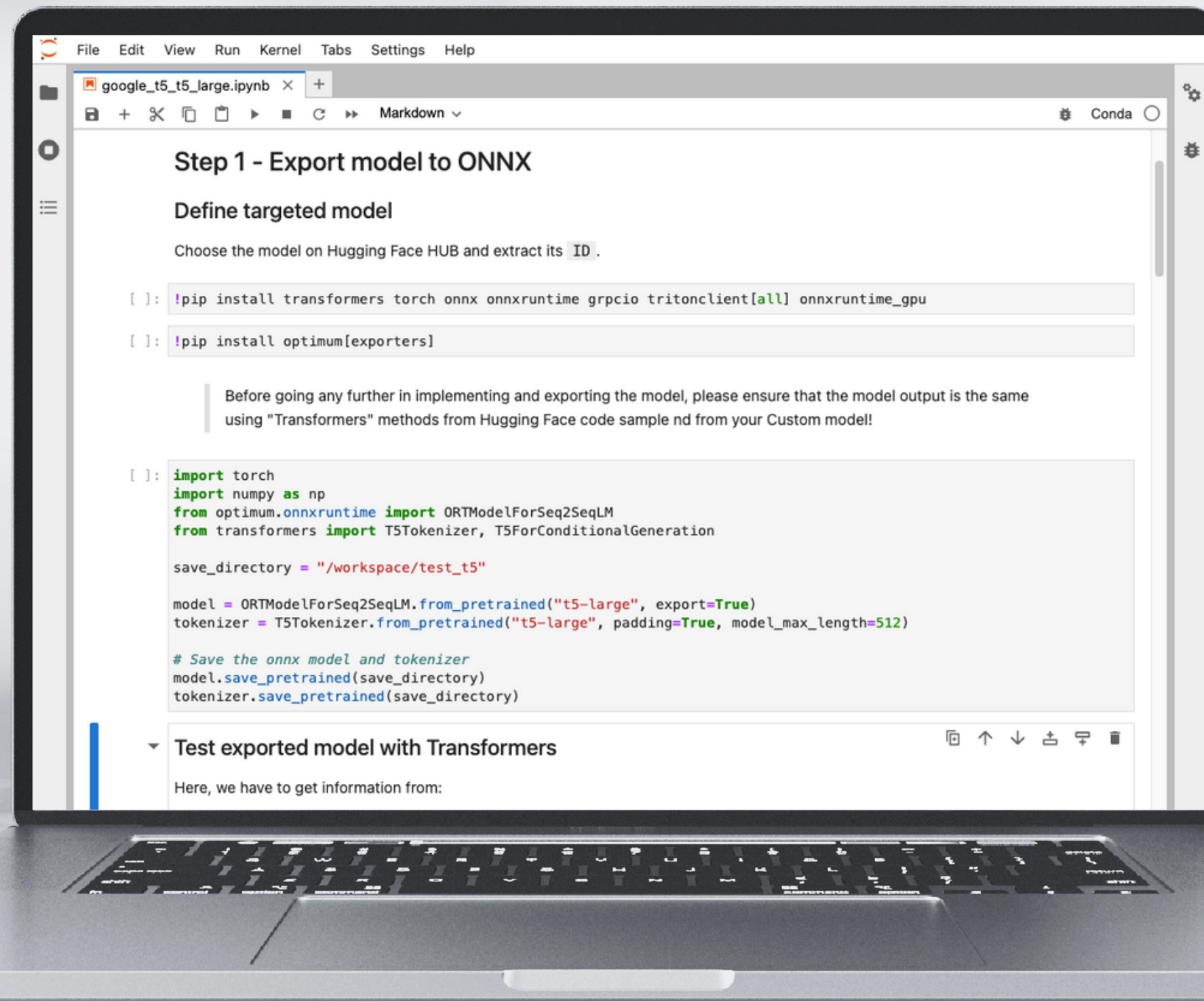
le serveur d'inférence
dans le cloud



EXPORTER

les modèles

EXPORTER LES MODÈLES AU FORMAT ONNX



The image shows a laptop screen displaying a Jupyter Notebook titled "google_t5_t5_large.ipynb". The notebook content is as follows:

```
File Edit View Run Kernel Tabs Settings Help
google_t5_t5_large.ipynb x +
+ ✂ 📄 ▶ ⏪ ⏩ ↻ ⏴ ⏵ Markdown v Conda ○
```

Step 1 - Export model to ONNX

Define targeted model

Choose the model on Hugging Face HUB and extract its ID.

```
[ ]: !pip install transformers torch onnx onnxruntime grpcio tritonclient[all] onnxruntime_gpu
```

```
[ ]: !pip install optimum[exporters]
```

Before going any further in implementing and exporting the model, please ensure that the model output is the same using "Transformers" methods from Hugging Face code sample and from your Custom model!

```
[ ]: import torch
import numpy as np
from optimum.onnxruntime import ORTModelForSeq2SeqLM
from transformers import T5Tokenizer, T5ForConditionalGeneration

save_directory = "/workspace/test_t5"

model = ORTModelForSeq2SeqLM.from_pretrained("t5-large", export=True)
tokenizer = T5Tokenizer.from_pretrained("t5-large", padding=True, model_max_length=512)

# Save the onnx model and tokenizer
model.save_pretrained(save_directory)
tokenizer.save_pretrained(save_directory)
```

Test exported model with Transformers

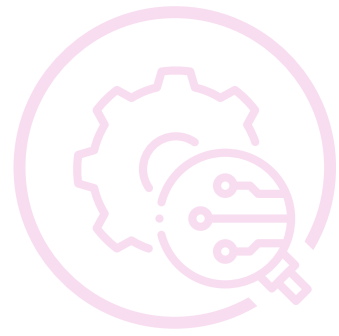
Here, we have to get information from:

COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



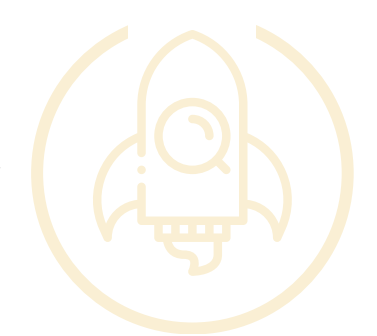
FORMATER

le code et le repo
de modèles



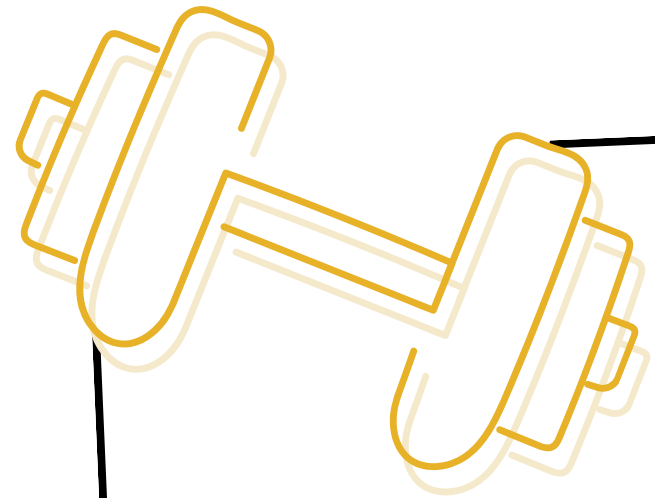
STOCKER

les modèles dans
un bucket S3



DÉPLOYER

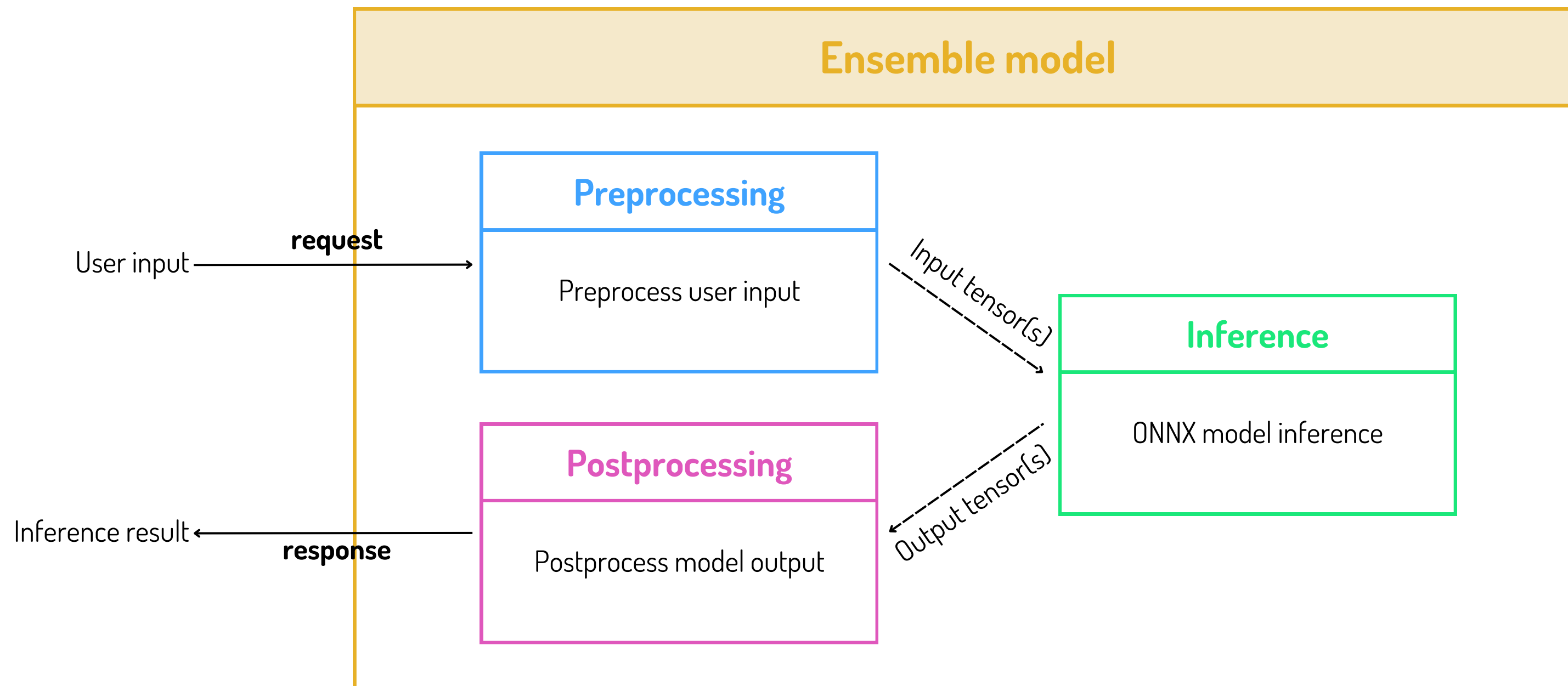
le serveur d'inférence
dans le cloud



DÉVELOPPER

les fonctions pre/post

DÉVELOPPER LES FONCTIONS DE PRÉ/POST PROCESSING

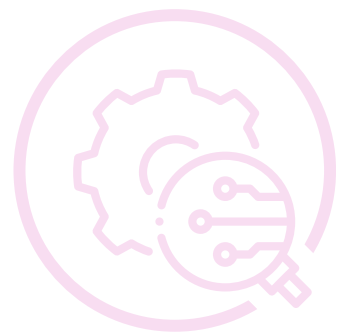


COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



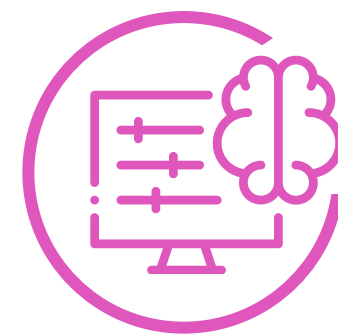
EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



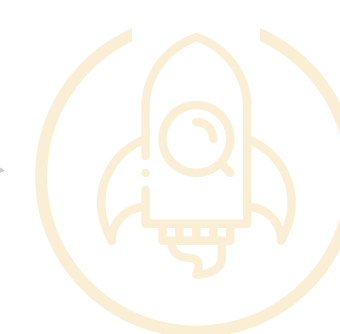
FORMATER

le code et le repo
de modèles



STOCKER

les modèles dans
un bucket S3



DÉPLOYER

le serveur d'inférence
dans le cloud



FORMATER

le code et le repo

FORMATER LE CODE ET LE REPO DE MODÈLES

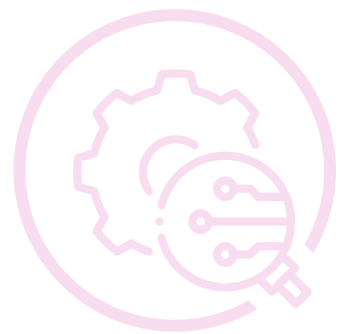


COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



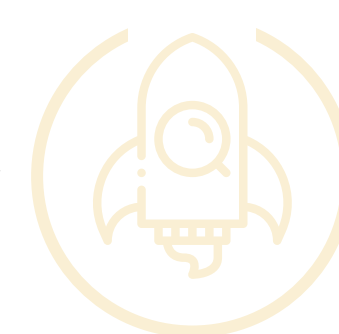
FORMATER

le code et le repo
de modèles



STOCKER

les modèles dans
un bucket S3



DÉPLOYER

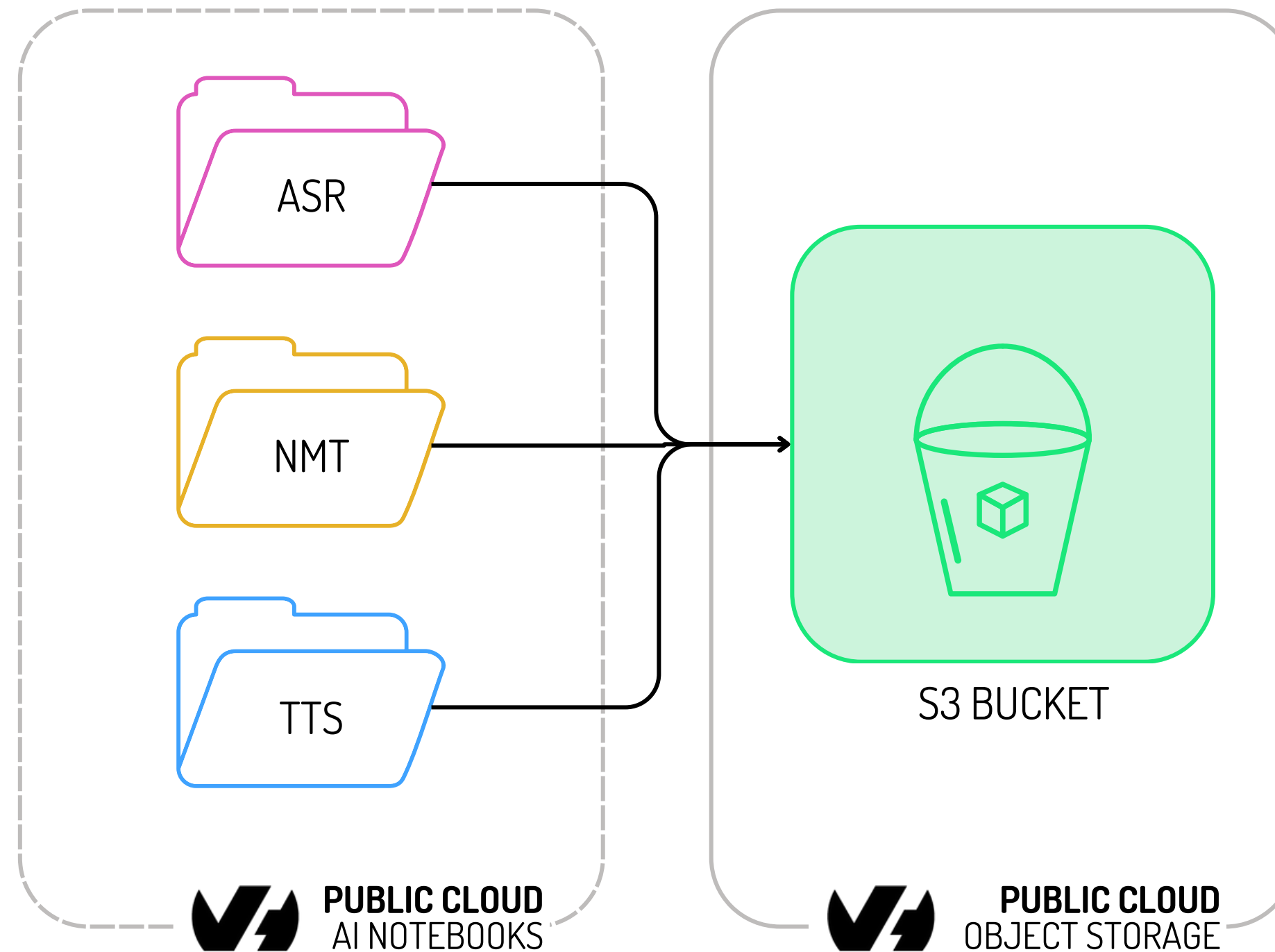
le serveur d'inférence
dans le cloud



STOCKER

les modèles

STOCKER LES MODÈLES DANS UN BUCKET S3

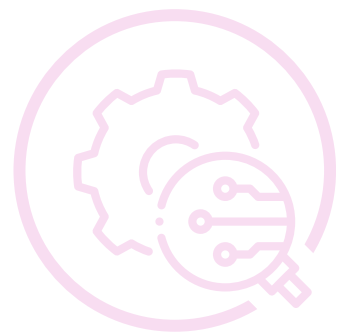


COMMENT DÉPLOYER CES MODÈLES ?



ACCÉDER

à un notebook
avec GPUs



TESTER

le(s) modèle(s)
localement



EXPORTER

les modèles au
format ONNX



DÉVELOPPER

les fonctions de
pré/post processing



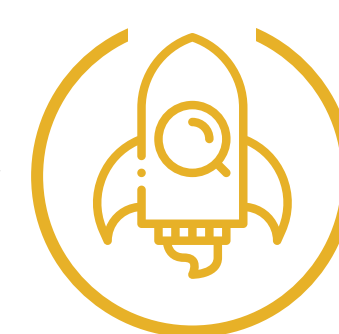
FORMATER

le code et le repo
de modèles



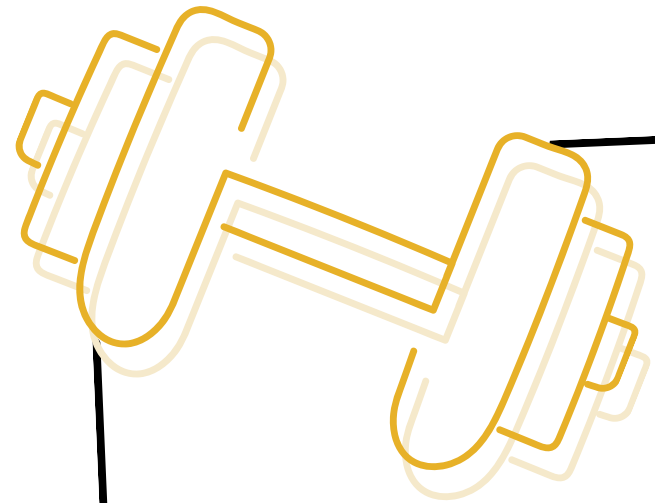
STOCKER

les modèles dans
un bucket S3



DÉPLOYER

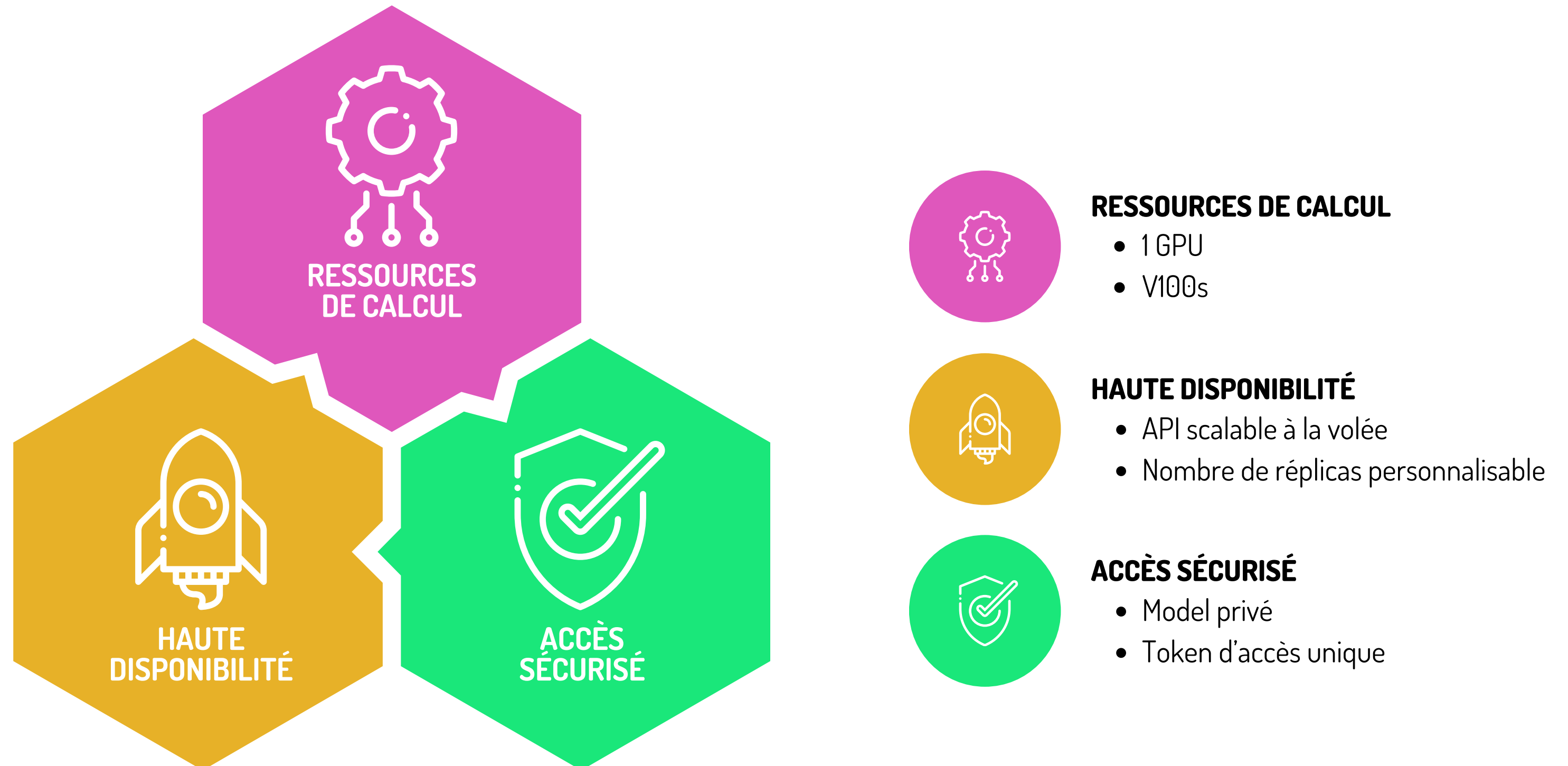
le serveur d'inférence
dans le cloud



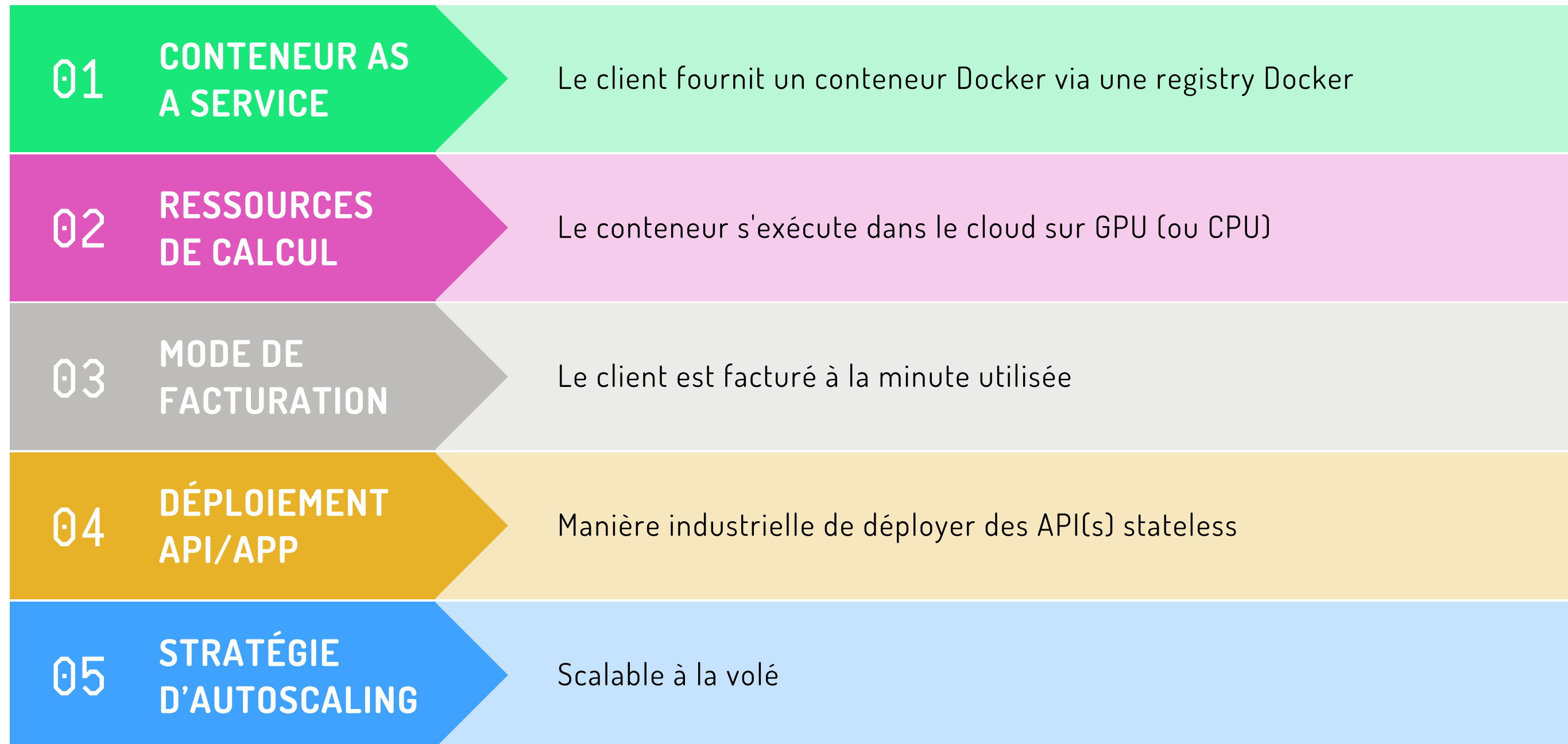
DÉPLOYER

le serveur d'inférence

DÉPLOYER LE SERVEUR D'INFÉRENCE DANS LE CLOUD



OVHcloud - AI DEPLOY



INFERENCE SERVER AVEC AI DEPLOY

Select the flavor and the number of GPUs

```
OVHcloud instance
ubuntu@nvidiainstance:~$ ovhai app run --name nmt_model \
>   --flavor ai1-1-gpu \
>   --gpu 1 \
>   --default-http-port 8000 \
>   --grpc-port 8001 \
>   -v my_model_repository@S3GRA/google_t5_t5_large/://data:ro \
>   -v standalone:/workspace:rw \
>   nvcv.io/nvidia/tritonserver:22.01-py3 \
>   -- bash -c 'pip install transformers torch &&
>               tritonserver --model-store /data'
```


INFERENCE SERVER AVEC AI DEPLOY

Precise HTTP
and gRPC port

```
OVHcloud instance
ubuntu@nvidiainstance:~$ ovhai app run --name nmt_model \
> --flavor ai1-1-gpu \
> --gpu 1 \
> --default-http-port 8000 \
> --grpc-port 8001 \
> -v my_model_repository@S3GRA/google_t5_t5_large/://data:ro \
> -v standalone:/workspace:rw \
> nvcv.io/nvidia/tritonserver:22.01-py3 \
> -- bash -c 'pip install transformers torch &&
> tritonserver --model-store /data'
```

INFERENCE SERVER AVEC AI DEPLOY

Attach your S3 buckets
with your NMT models

```
OVHcloud instance
ubuntu@nvidiainstance:~$ ovhai app run --name nmt_model \
> --flavor ai1-1-gpu \
> --gpu 1 \
> --default-http-port 8000 \
> --grpc-port 8001 \
> -v my_model_repository@S3GRA/google_t5_t5_large/./data:ro \
> -v standalone:/workspace:rw \
> nvcv.io/nvidia/tritonserver:22.01-py3 \
> -- bash -c 'pip install transformers torch &&
> tritonserver --model-store /data'
```

INFERENCE SERVER AVEC AI DEPLOY

Select Triton Server
docker image ←

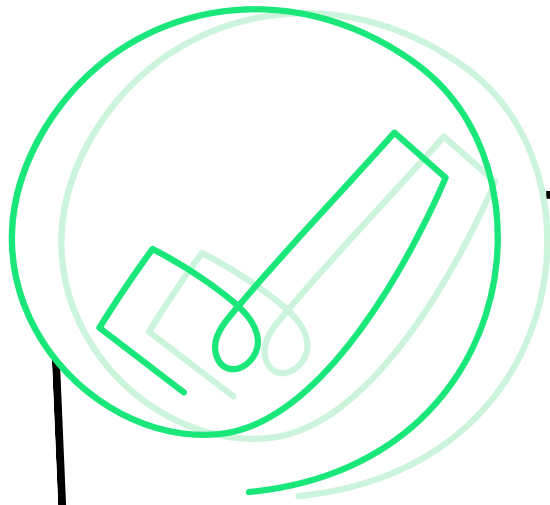
```
OVHcloud instance
ubuntu@nvidiainstance:~$ ovhai app run --name nmt_model \
> --flavor ai1-1-gpu \
> --gpu 1 \
> --default-http-port 8000 \
> --grpc-port 8001 \
> -v my_model_repository@S3GRA/google_t5_t5_large/://data:ro \
> -v standalone:/workspace:rw \
> nvcr.io/nvidia/tritonserver:22.01-py3 \
> -- bash -c 'pip install transformers torch &&
> tritonserver --model-store /data'
```

INFERENCE SERVER AVEC AI DEPLOY

Start Triton Inference
Server for NMT

```
OVHcloud instance
ubuntu@nvidiainstance:~$ ovhai app run --name nmt_model \
> --flavor ai1-1-gpu \
> --gpu 1 \
> --default-http-port 8000 \
> --grpc-port 8001 \
> -v my_model_repository@S3GRA/google_t5_t5_large/://data:ro \
> -v standalone:/workspace:rw \
> nvc.io/nvidia/tritonserver:22.01-py3 \
> -- bash -c 'pip install transformers torch &&
> tritonserver --model-store /data'
```

**BESOIN D'UNE SOLUTION
CLÉ EN MAIN ?**



CHOISIR ses AI Endpoints

OVHcloud - AI ENDPOINTS

AI Endpoints
Unlock the Future: Seamless AI with Uncompromised Privacy

Our platform, currently in an exciting alpha phase, is designed with you—the developer—at its core. Here, you'll discover a seamless integration of world-renowned AI models alongside a handpicked selection of Nvidia's optimized models, to generate text, classify images, and much more. All tailored to catapult your projects into the next dimension of intelligence and efficiency.

[Get your free token +](#)

Available Now

Explore our inaugural APIs

Mistral-7B-Instruct-v0.2
Mistral-7B-Instruct-v0.2 Large Language Model (LLM) is a pretrained generative text model with 7 billion parameters.

Mixtral-8x7B-Instruct-v0.1
Mixtral-8x7B-Instruct-v0.1 Large Language Model (LLM) is a pretrained generative Sparse Mixture of Experts.

Coming soon

Keep updated on our [Discord](#) server for all newly available APIs

CodeLlama-13b-Instruct-hf
Code Llama is a pretrained and fine-tuned generative text models with 13 billion parameters. This model is designed for general code synthesis and understanding.

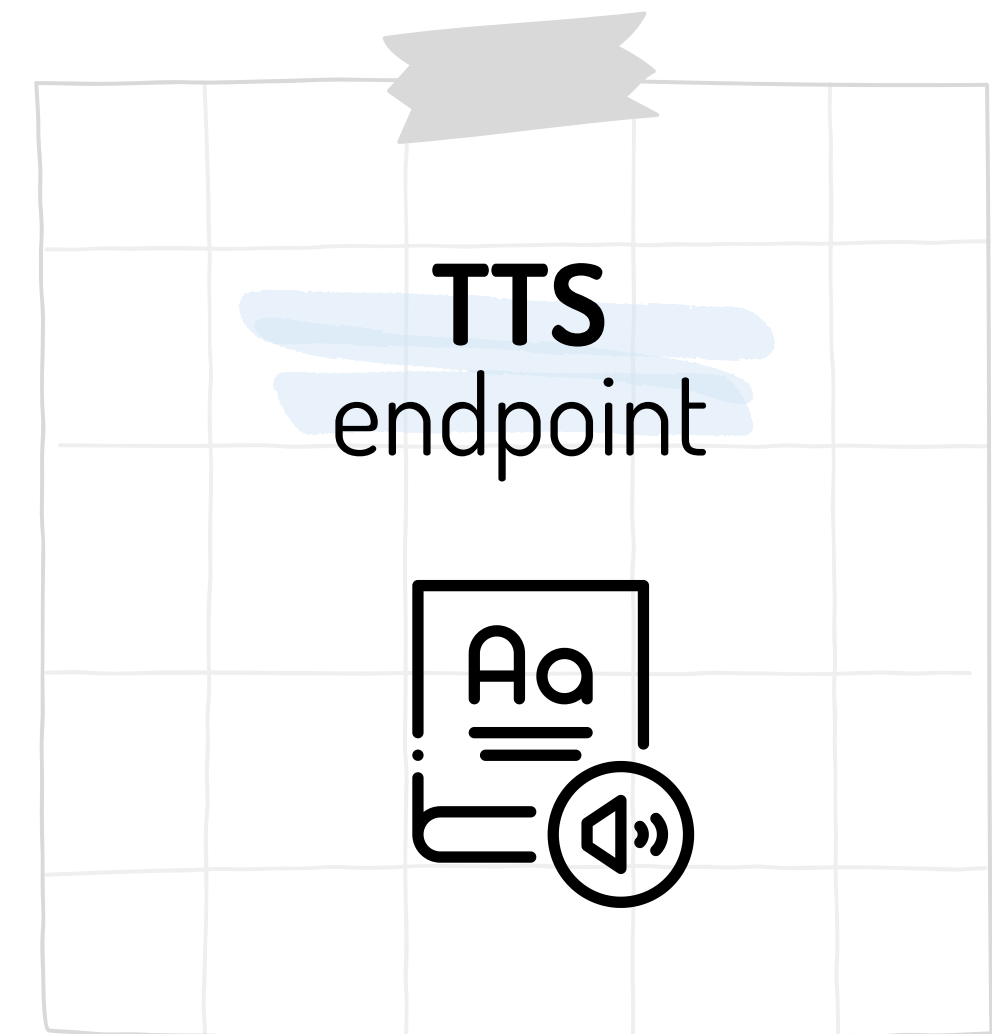
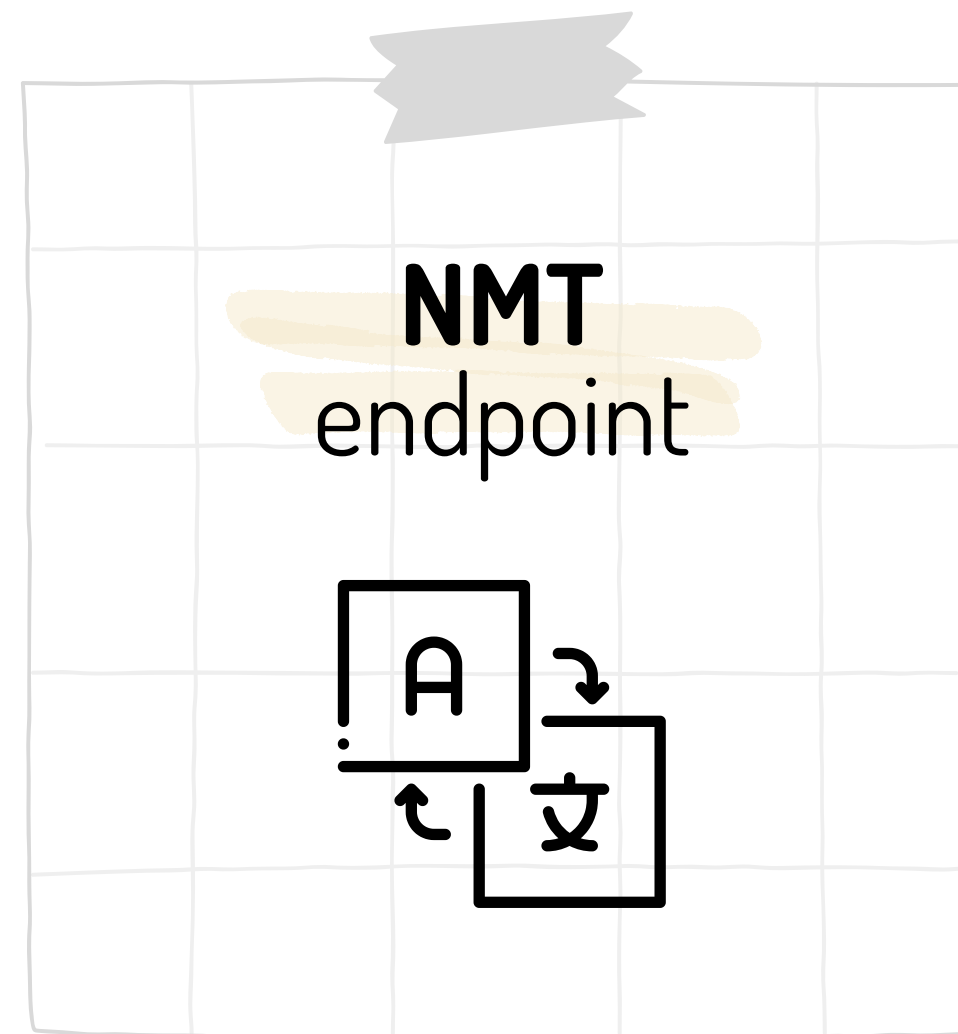
Llama-2-13b-chat-hf
Llama 2 is a pretrained and fine-tuned generative text model of 13 billion parameters. This "chat" version is optimized for dialogue use cases.

Mixtral-8x7B-Instruct-v0.1

Mixtral-8x7B-Instruct-v0.1 Large Language Model (LLM) is a pretrained generative Sparse Mixture of Experts.

OVHcloud - AI ENDPOINTS

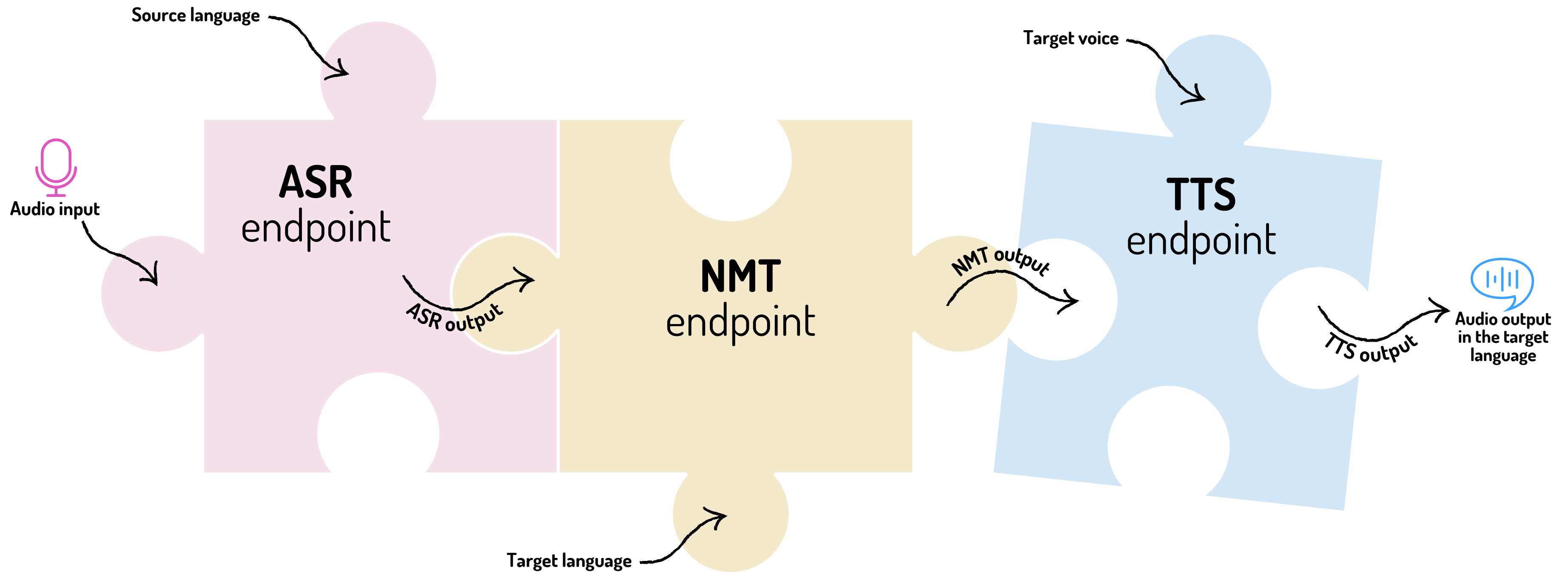
“ Choisissez les endpoints API qui correspondent à vos besoins ! ”

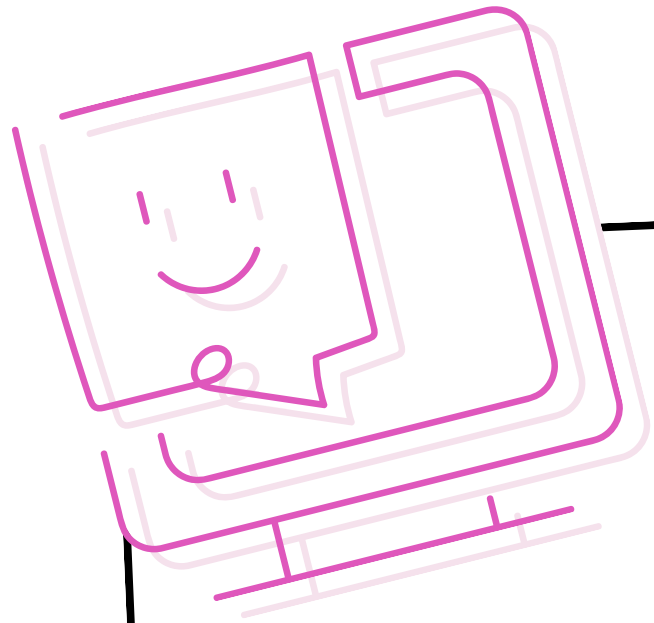


**COMMENT CONNECTER CES
AI ENDPOINTS ENTRE EUX ?**

OVHcloud - AI ENDPOINTS

“Connectez vos AI Endpoints selon vos besoins !”











DÉVELOPPER

la partie client

DÉVELOPPER LA PARTIE CLIENT

-  **Entrer** un lien de vidéo YouTube
-  **Transcrire** la partie audio de la vidéo en texte
-  **Sous-titrer** la vidéo dans n'importe quelle langue
-  **Doubler** la voix de la vidéo dans une autre langue
-  **Choisir** le genre de la voix de doublage
-  **Télécharger** la vidéo résultante

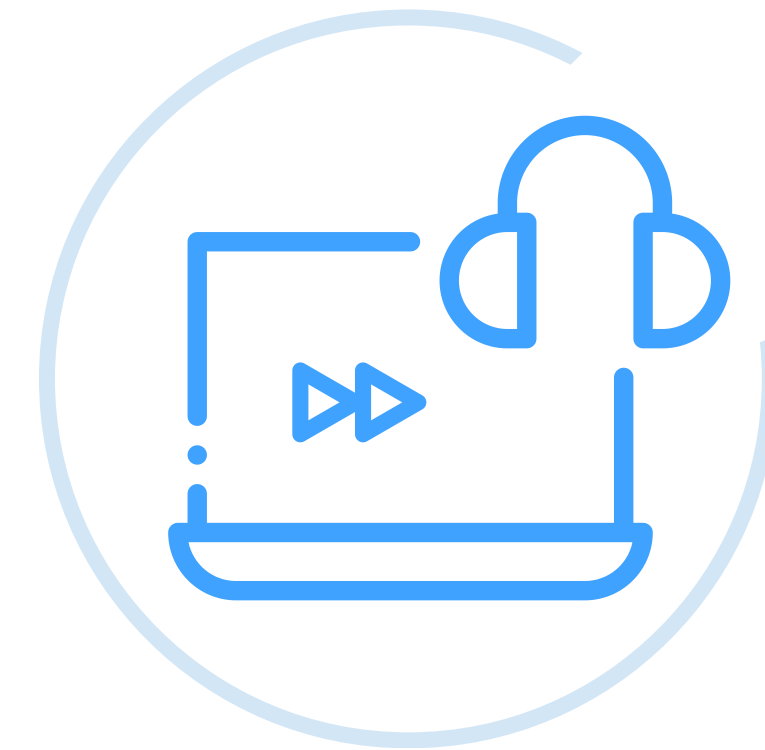
DÉVELOPPER LA PARTIE CLIENT



GÉNÉRER
un fichier SRT de
sous-titres



CONSERVER
les silences pendant
la traduction



DOUBLER
l'audio d'une vidéo
dans une autre langue

DÉVELOPPER LA PARTIE CLIENT



GÉNÉRER
un fichier SRT de
sous-titres

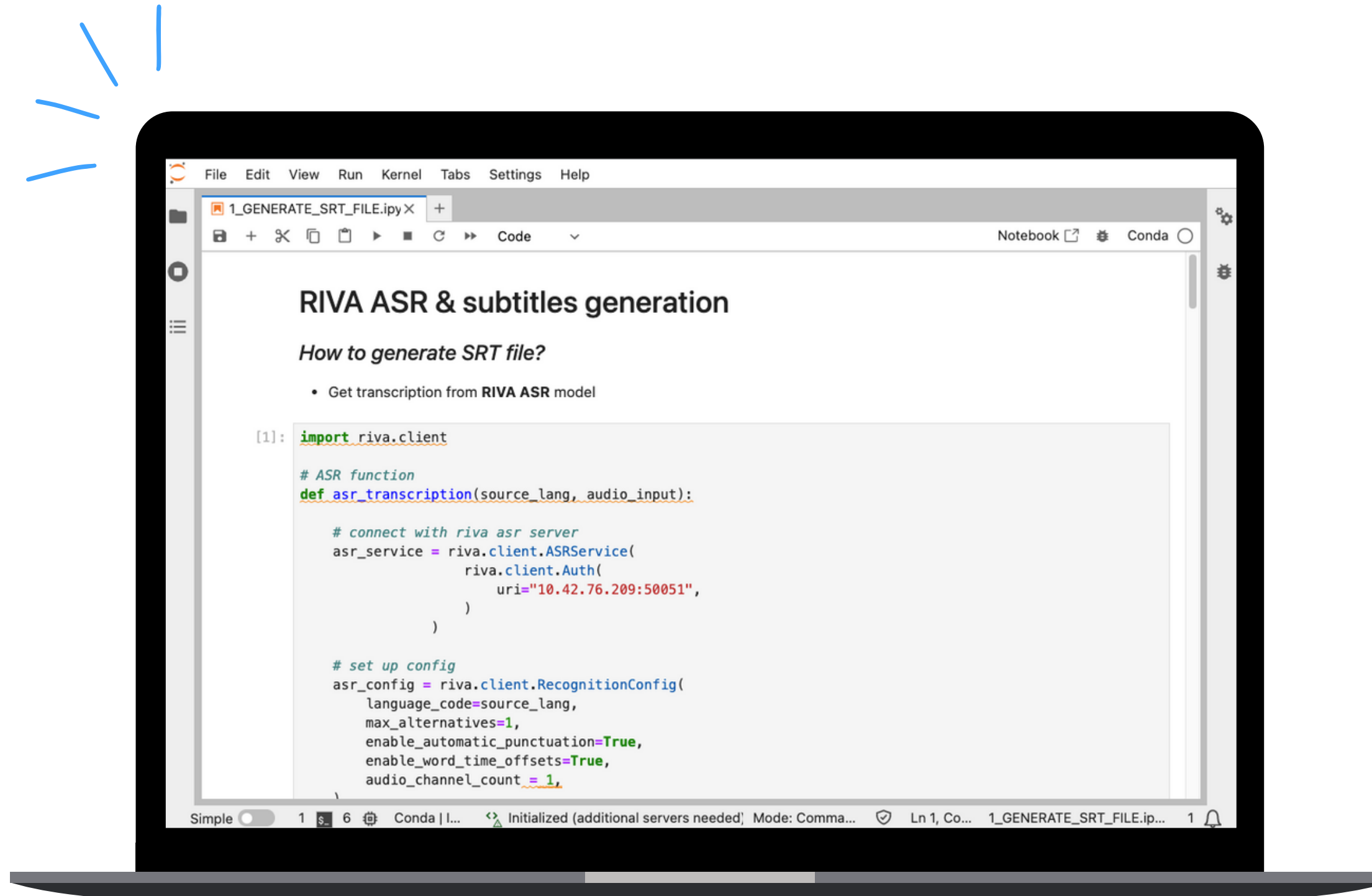


CONSERVER
les silences pendant
la traduction



DOUBLER
l'audio d'une vidéo
dans une autre langue

GÉNÉRER UN FICHER SRT



```
File Edit View Run Kernel Tabs Settings Help
1_GENERATE_SRT_FILE.ipynk
+
+ ✂ 📄 📄 ▶ ■ ↻ ▶▶ Code ▾
Notebook 📄 ⚙️ Conda ○ ⚙️
```

RIVA ASR & subtitles generation

How to generate SRT file?

- Get transcription from **RIVA ASR** model

```
[1]: import riva.client

# ASR function
def asr_transcription(source_lang, audio_input):

    # connect with riva asr server
    asr_service = riva.client.ASRService(
        riva.client.Auth(
            uri="10.42.76.209:50051",
        )
    )

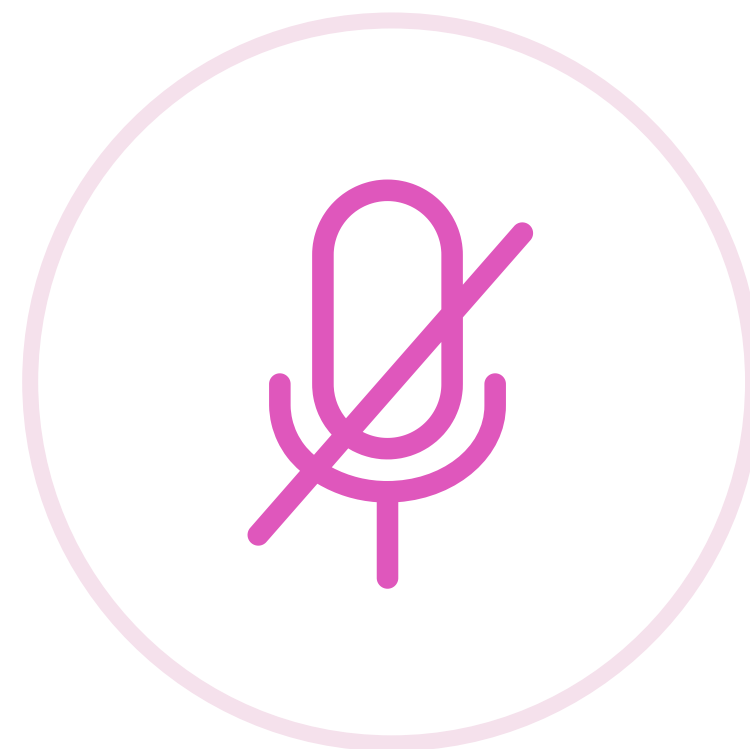
    # set up config
    asr_config = riva.client.RecognitionConfig(
        language_code=source_lang,
        max_alternatives=1,
        enable_automatic_punctuation=True,
        enable_word_time_offsets=True,
        audio_channel_count = 1,
```

Simple 1 6 Conda | l... Initialized (additional servers needed) Mode: Comma... Ln 1, Co... 1_GENERATE_SRT_FILE.ip... 1

DÉVELOPPER LA PARTIE CLIENT



GÉNÉRER
un fichier SRT de
sous-titres

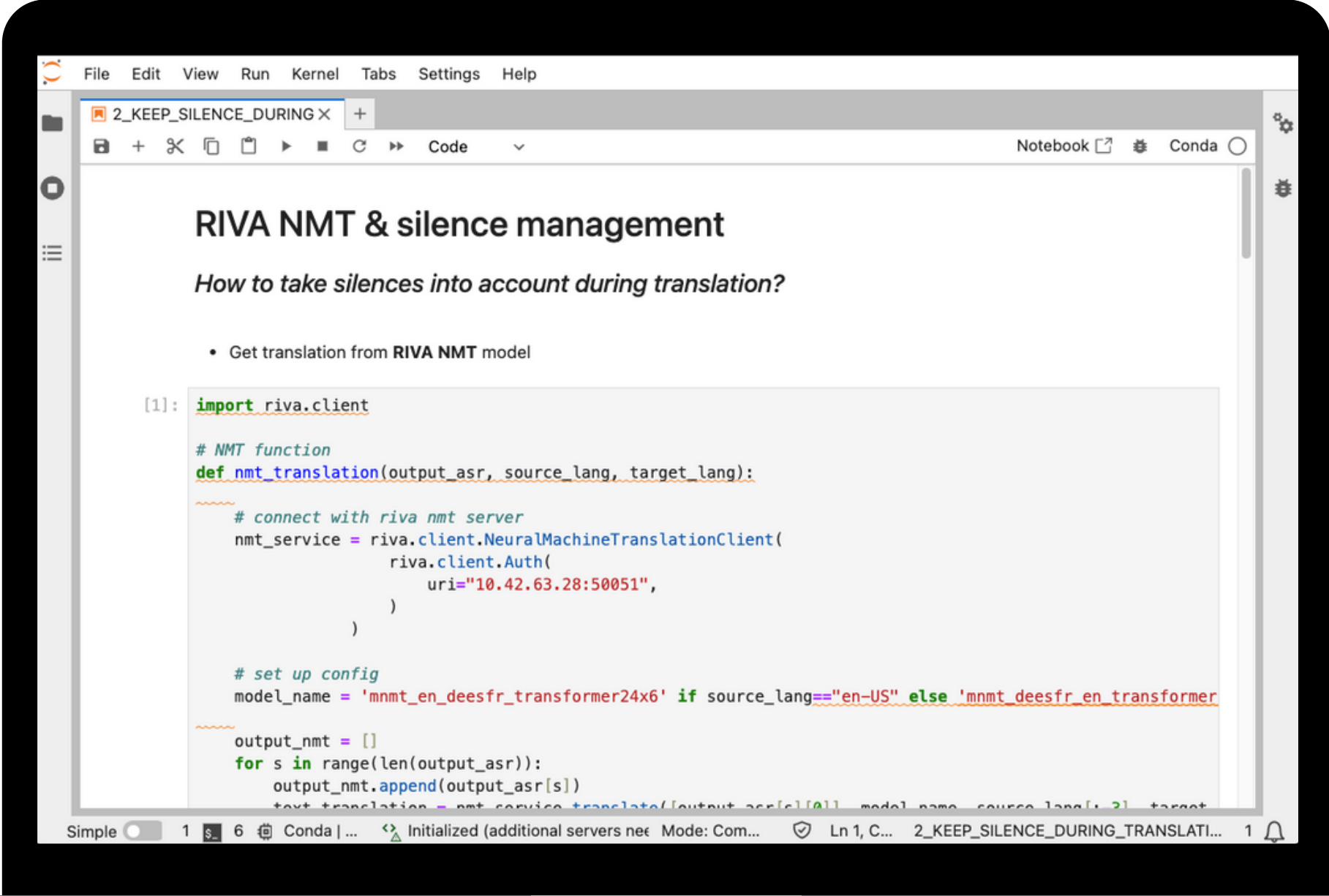


CONSERVER
les silences pendant
la traduction



DOUBLER
l'audio d'une vidéo
dans une autre langue

CONSERVER LES SILENCES



```
File Edit View Run Kernel Tabs Settings Help
2_KEEP_SILENCE_DURING x +
+ ✂ 📄 ▶ ⏪ ⏩ Code Notebook Conda
RIVA NMT & silence management
How to take silences into account during translation?
• Get translation from RIVA NMT model
[1]: import riva.client
# NMT function
def nmt_translation(output_asr, source_lang, target_lang):
    # connect with riva nmt server
    nmt_service = riva.client.NeuralMachineTranslationClient(
        riva.client.Auth(
            uri="10.42.63.28:50051",
        )
    )
    # set up config
    model_name = 'mnmt_en_deesfr_transformer24x6' if source_lang=="en-US" else 'mnmt_deesfr_en_transformer'
    output_nmt = []
    for s in range(len(output_asr)):
        output_nmt.append(output_asr[s])
        text_translation = nmt_service.translate([output_asr[s][0]], model_name, source_lang, target_lang)
```

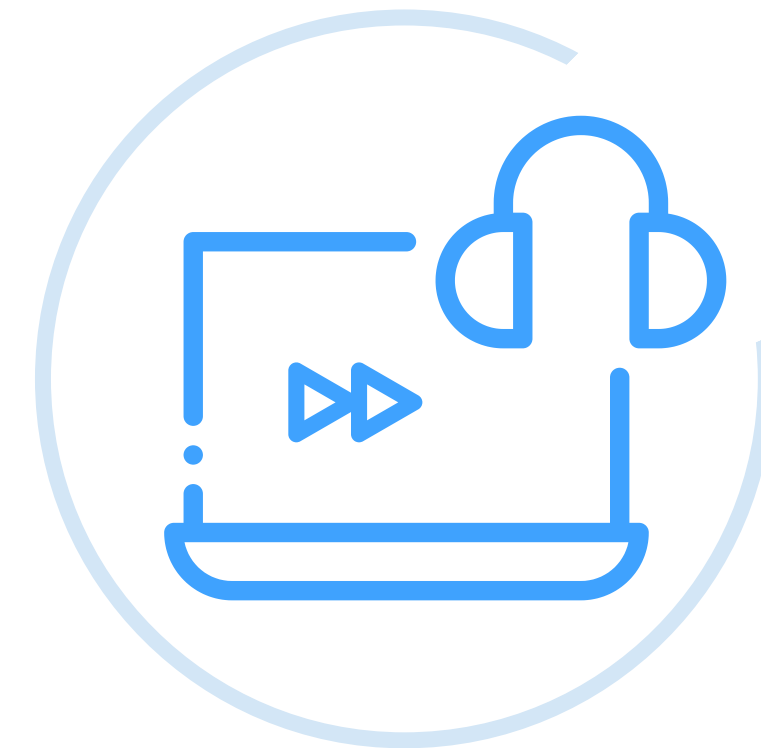
DÉVELOPPER LA PARTIE CLIENT



GÉNÉRER
un fichier SRT de
sous-titres

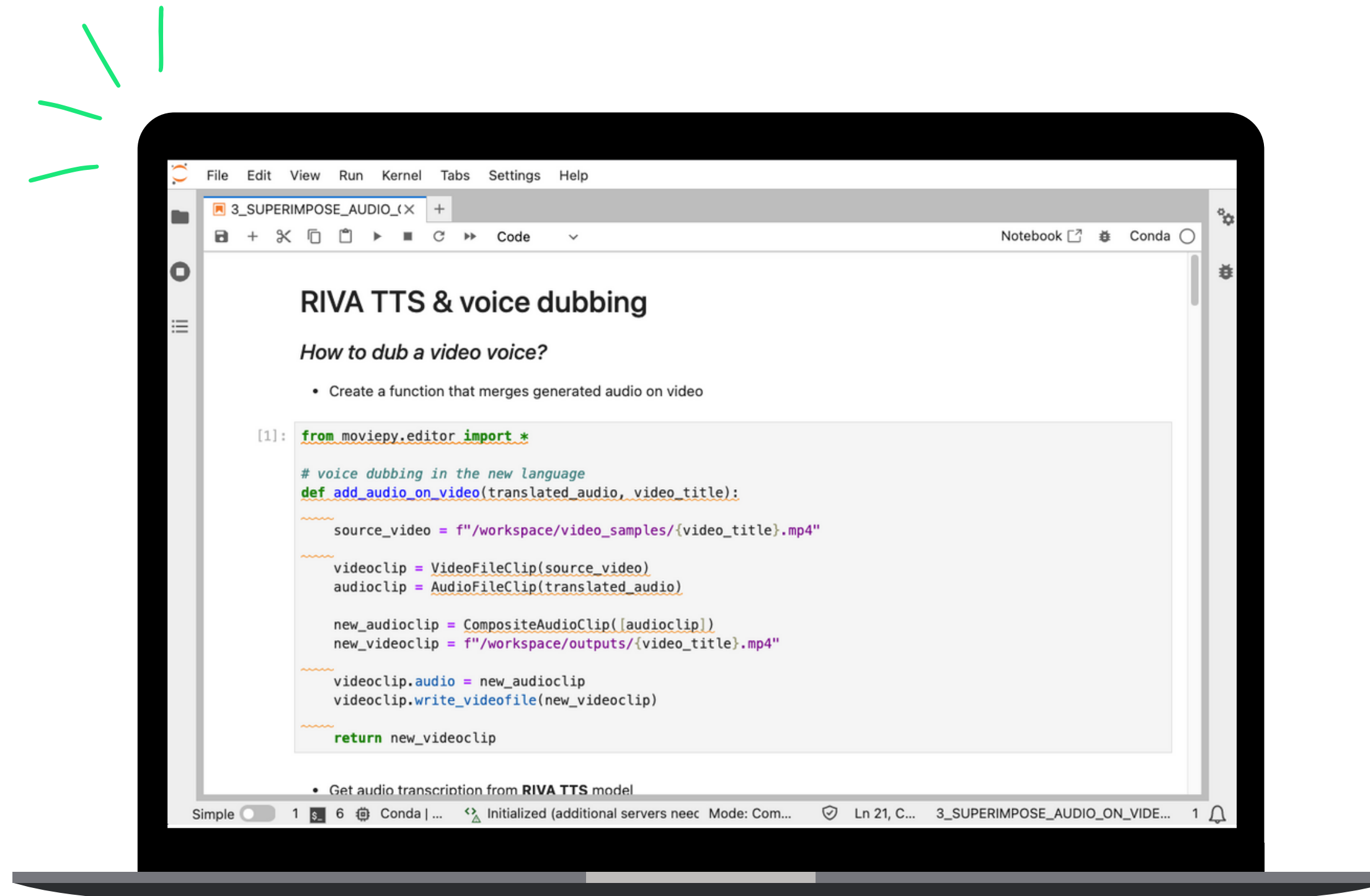


CONSERVER
les silences pendant
la traduction



DOUBLER
l'audio d'une vidéo
dans une autre langue

DOUBLER L'AUDIO D'UNE VIDÉO

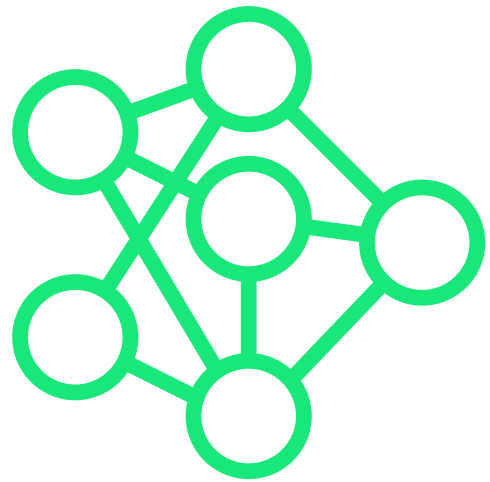




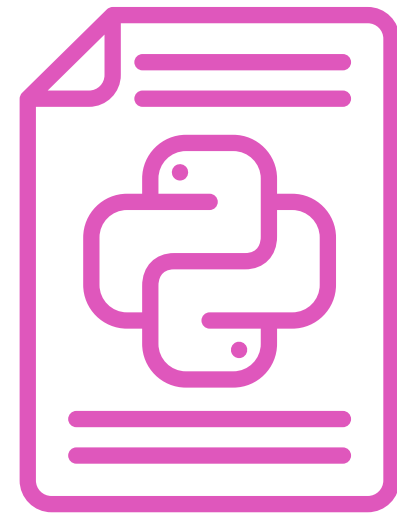
DÉPLOYER

l'app end-to-end

DÉPLOYER L'APP END-TO-END



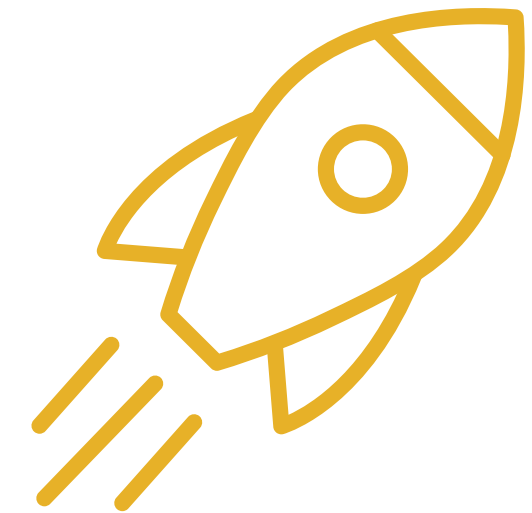
Inference
Servers



Custom client
code

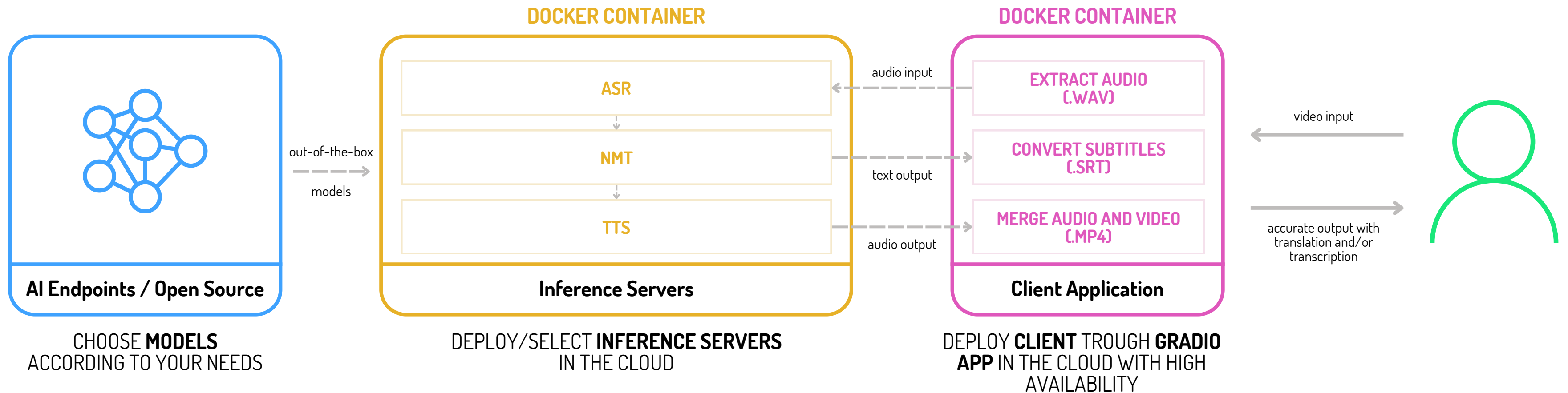


Docker
image



AI Deploy
app

DÉPLOYER L'APP END-TO-END

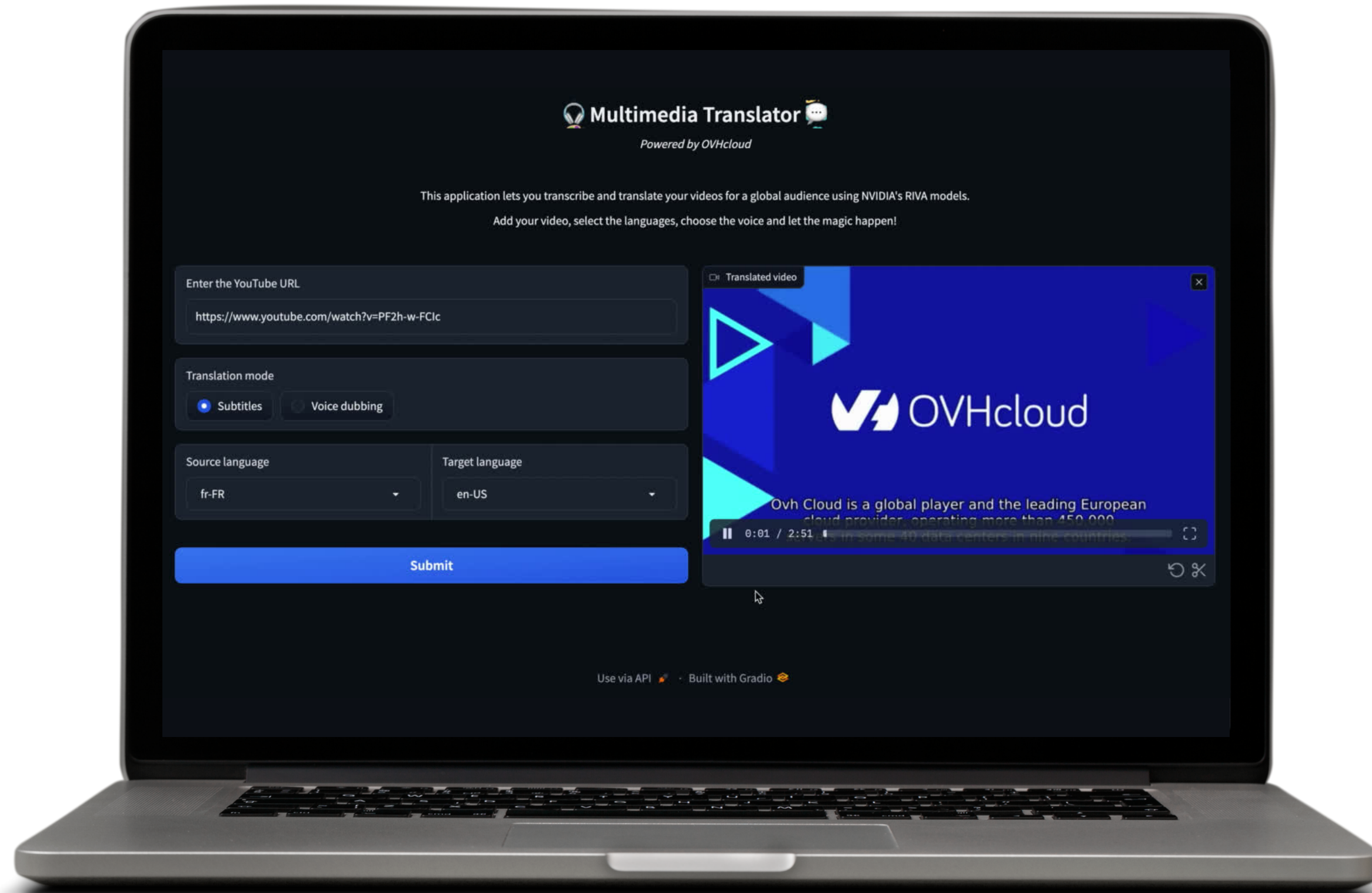




DÉMO

de l'app

DÉMO DE L'APP



DÉMO DE L'APP



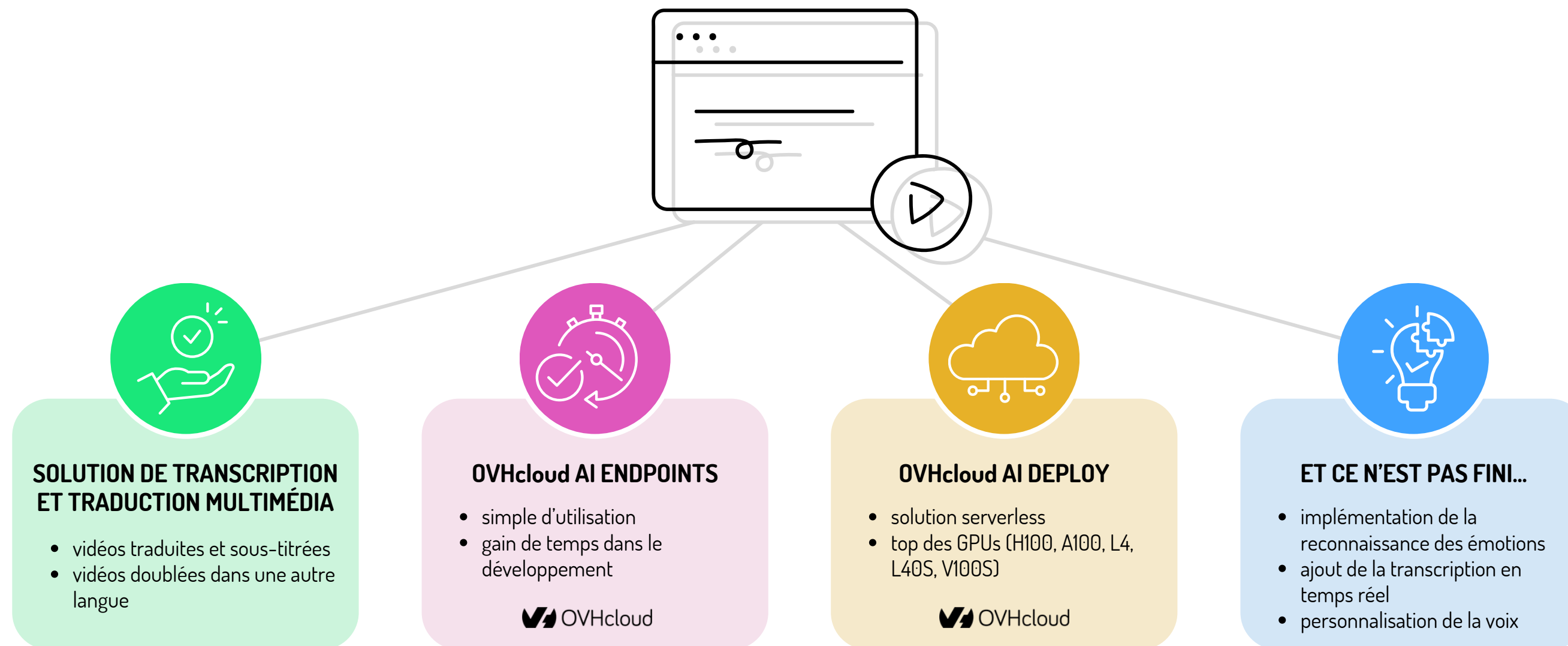
<https://bit.ly/multimedia-translator-devovx>



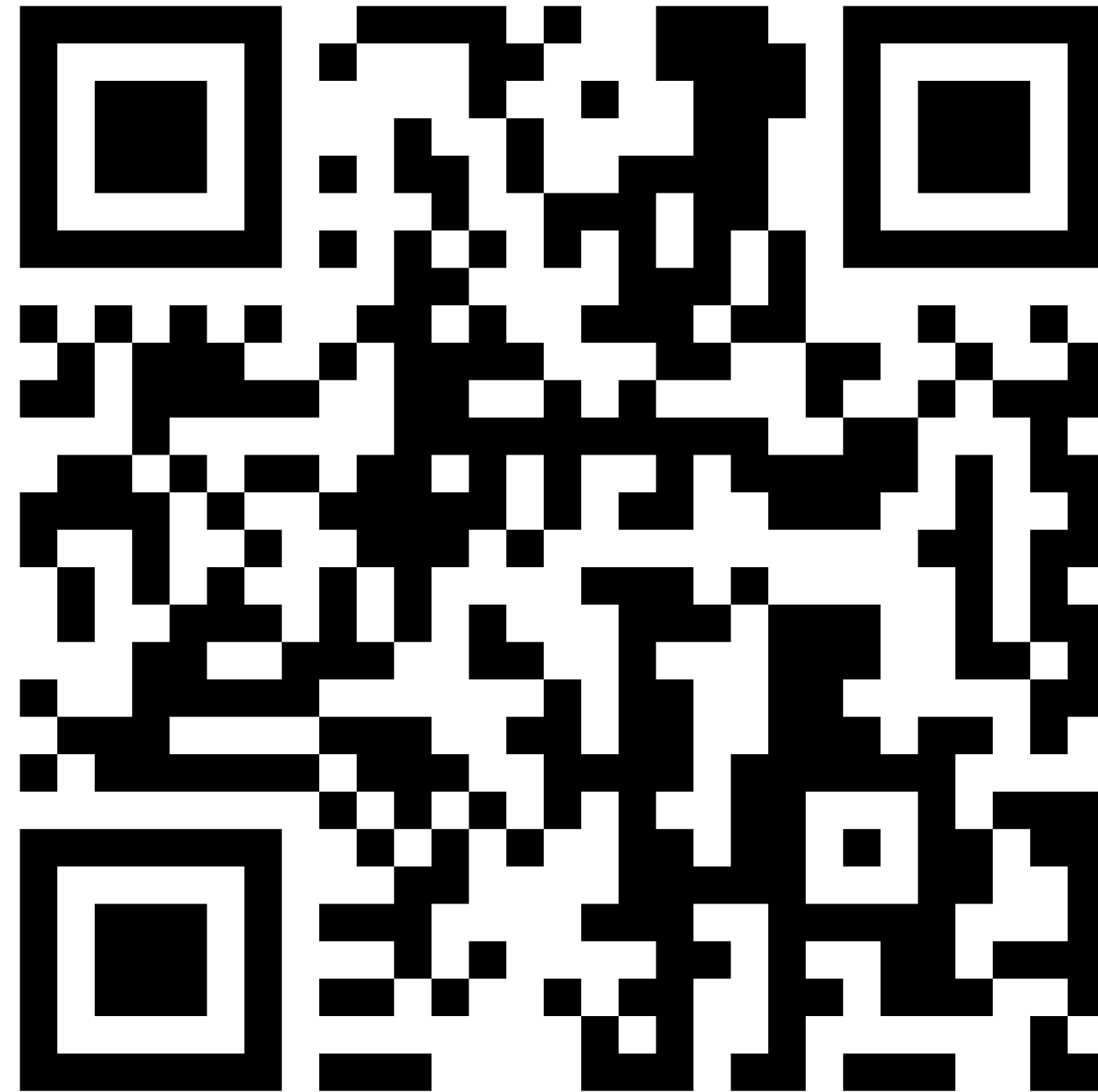
CONCLUSION

CONCLUSION

Le “**multimode multimedia translator**” en résumé...



OVHcloud AI ENDPOINTS



<https://endpoints.ai.cloud.ovh.net/>



MERCI!

À VOUS DE TESTER!



<https://bit.ly/multimedia-translator-devovx>