



# MASTERING SPEECH AI FOR MULTILINGUAL MULTIMEDIA TRANSFORMATION



NVIDIA  
**GTC**

# INTRODUCTION

The image shows a laptop screen displaying a video translator application. The title "Video translator" is at the top. Below it, the text "Transcribe your video:" is followed by a URL input field containing "https://www.youtube.com/...". There are two language selection buttons: "French" (with a French flag icon) and "English" (with an American flag icon). Below these is a dropdown menu showing "Male-en-US" with a male icon and a checkmark. To the right of the dropdown are two buttons: ".mp4" with a video camera icon and a dropdown arrow, and "Download" with a download icon. At the bottom left are two radio button options: "Subtitles" and "Voice".

Video translator

Transcribe your video:

[https://www.youtube.com/...](https://www.youtube.com/)

French English

Male-en-US

.mp4 Download

Subtitles Voice

# PRES

Eléa Petton

Machine Learning Engineer

OVHcloud

AI Solutions Team



# OVHcloud

A global presence to support you wherever you are



04 |

**37** datacenters  
in 17 locations,  
across 9 countries

**4** continents connected  
through OVHcloud

**48** points of presence<sup>1</sup>  
on a 87 Tbps<sup>2</sup>  
Bandwidth Network

# ABOUT OVHcloud

**OVHcloud** is a global cloud provider that specializes in delivering industry leading performance and cost-effective solutions to better manage, secure and scale data



# TABLE OF CONTENT

- Objectives

07



# TABLE OF CONTENT

- **Objectives**
- **Define** your needs

07  
08



# TABLE OF CONTENT

- **Objectives** 07
- **Define** your needs 08
- **Select** the models 11



# TABLE OF CONTENT

- **Objectives** 07
- **Define** your needs 08
- **Select** the models 11
- **Set up** RIVA models 14

# TABLE OF CONTENT

• <b>Objectives</b>	07
• <b>Define</b> your needs	08
• <b>Select</b> the models	11
• <b>Set up</b> RIVA models	14
• <b>Deploy</b> RIVA server	27

# TABLE OF CONTENT

• <b>Objectives</b>	07
• <b>Define</b> your needs	08
• <b>Select</b> the models	11
• <b>Set up</b> RIVA models	14
• <b>Deploy</b> RIVA server	27
• <b>Customize</b> RIVA vocabulary	32

# TABLE OF CONTENT

• <b>Objectives</b>	07
• <b>Define</b> your needs	08
• <b>Select</b> the models	11
• <b>Set up</b> RIVA models	14
• <b>Deploy</b> RIVA server	27
• <b>Customize</b> RIVA vocabulary	32
• <b>Develop</b> RIVA client app	37

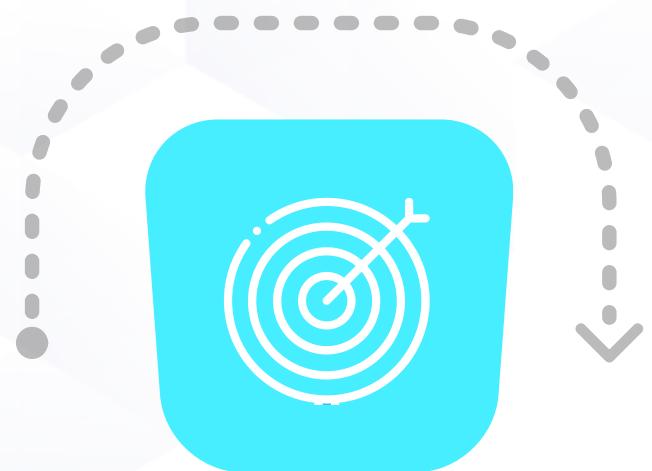
# TABLE OF CONTENT

• <b>Objectives</b>	07
• <b>Define</b> your needs	08
• <b>Select</b> the models	11
• <b>Set up</b> RIVA models	14
• <b>Deploy</b> RIVA server	27
• <b>Customize</b> RIVA vocabulary	32
• <b>Develop</b> RIVA client app	37
• <b>Deploy</b> the end to end app	47

# TABLE OF CONTENT

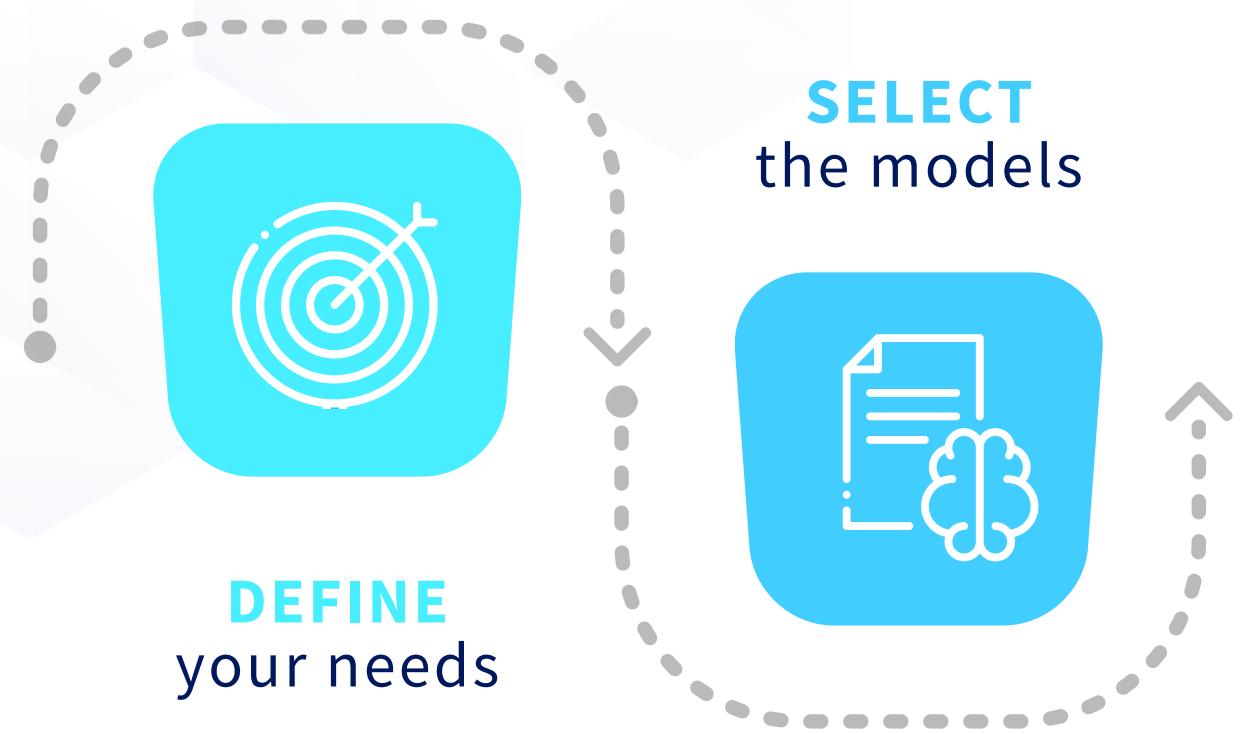
• <b>Objectives</b>	07
• <b>Define</b> your needs	08
• <b>Select</b> the models	11
• <b>Set up</b> RIVA models	14
• <b>Deploy</b> RIVA server	27
• <b>Customize</b> RIVA vocabulary	32
• <b>Develop</b> RIVA client app	37
• <b>Deploy</b> the end to end app	47
• <b>Conclusion</b>	50

# OBJECTIVES

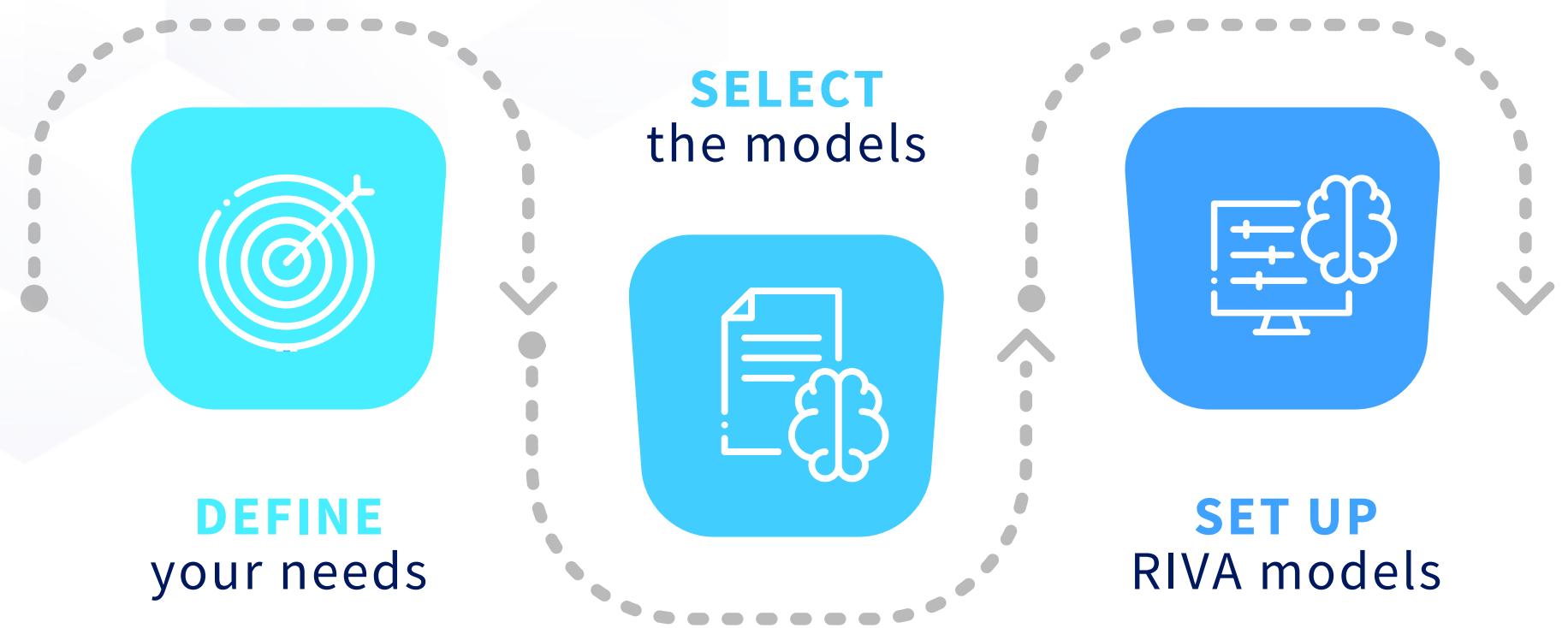


**DEFINE**  
your needs

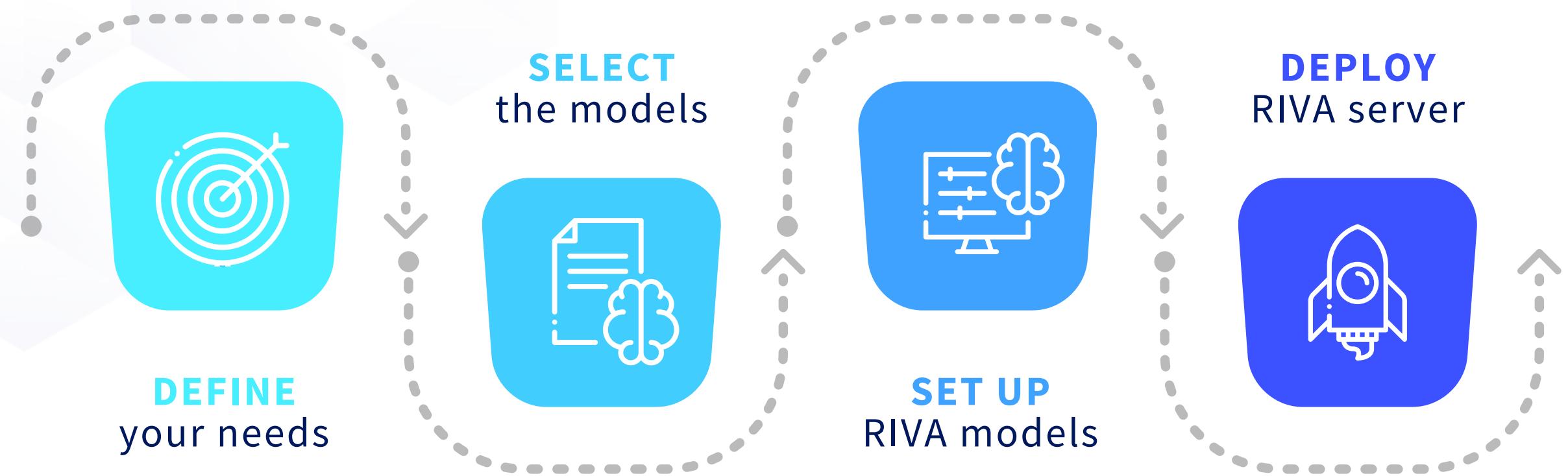
# OBJECTIVES



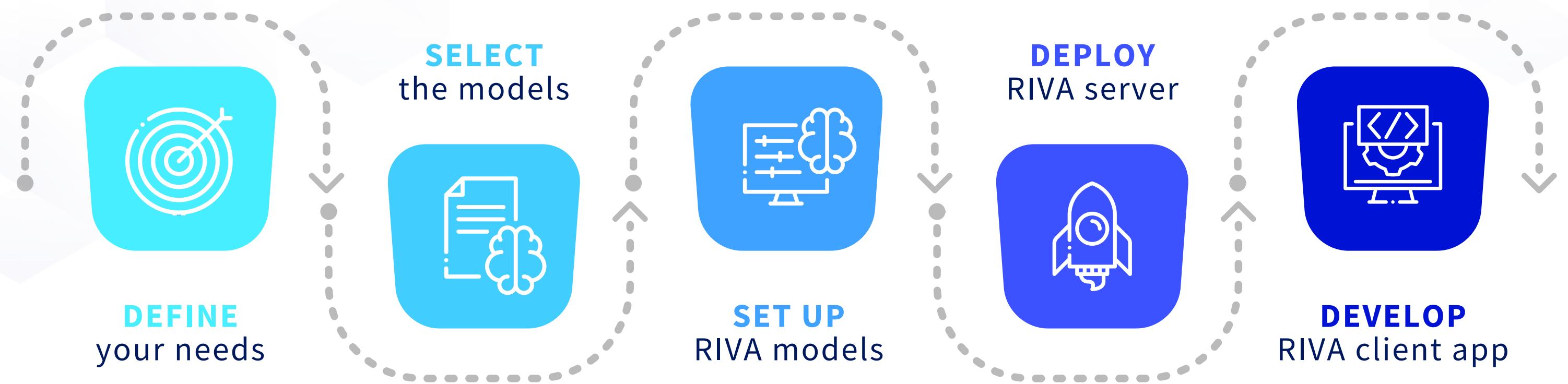
# OBJECTIVES



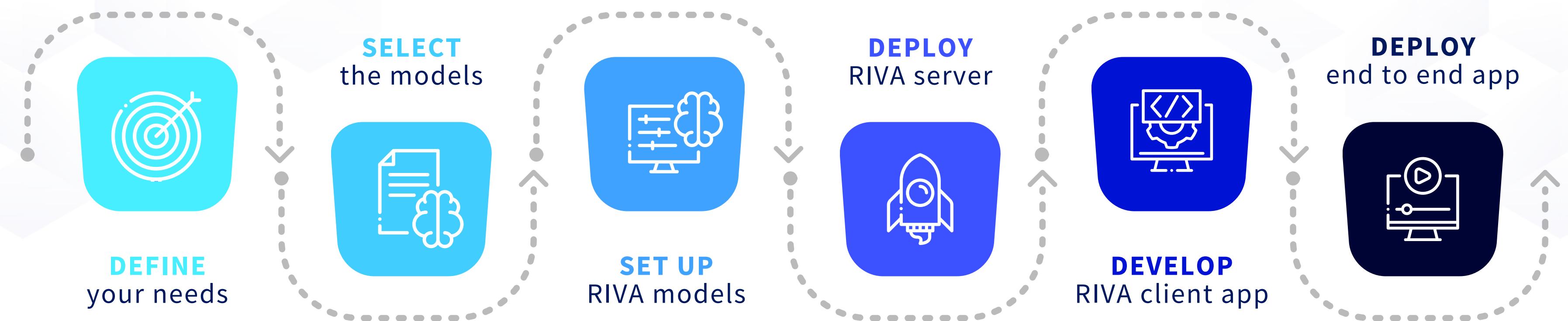
# OBJECTIVES



# OBJECTIVES



# OBJECTIVES





**DEFINE**  
your needs

# DEFINE YOUR NEEDS

## Transcribe meetings



- **Write down** my meetings
- **Transcribe** my meetings into other languages
- **Get summaries** of meetings

# DEFINE YOUR NEEDS



## International growth

- **Translate multimedia content** for multilingual usage
- **Adapt videos** for a global audience

## Transcribe meetings



- **Write down** my meetings
- **Transcribe** my meetings into other languages
- **Get summaries** of meetings

# DEFINE YOUR NEEDS



## International growth

- **Translate multimedia content** for multilingual usage
- **Adapt videos** for a global audience

## Transcribe meetings



- **Write down** my meetings
- **Transcribe** my meetings into other languages
- **Get summaries** of meetings

## Be more inclusive!



- **Subtitle my videos** for the hearing impaired
- **Include people** who don't speak my language

# DEFINE YOUR NEEDS



# KEY TARGETS



**AUDIO**  
recognition

# KEY TARGETS



**AUDIO**  
recognition



**TEXT**  
translation

# KEY TARGETS



**AUDIO**  
recognition



**TEXT**  
translation

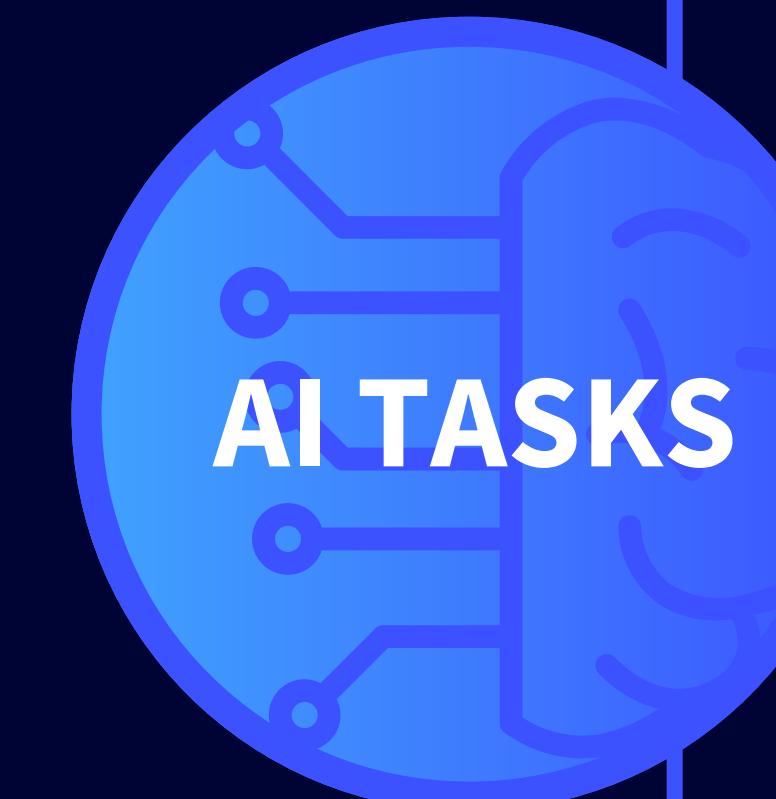


**SPEECH**  
synthesis



# SELECT the models

# SELECT THE MODELS



Speech-to-Text

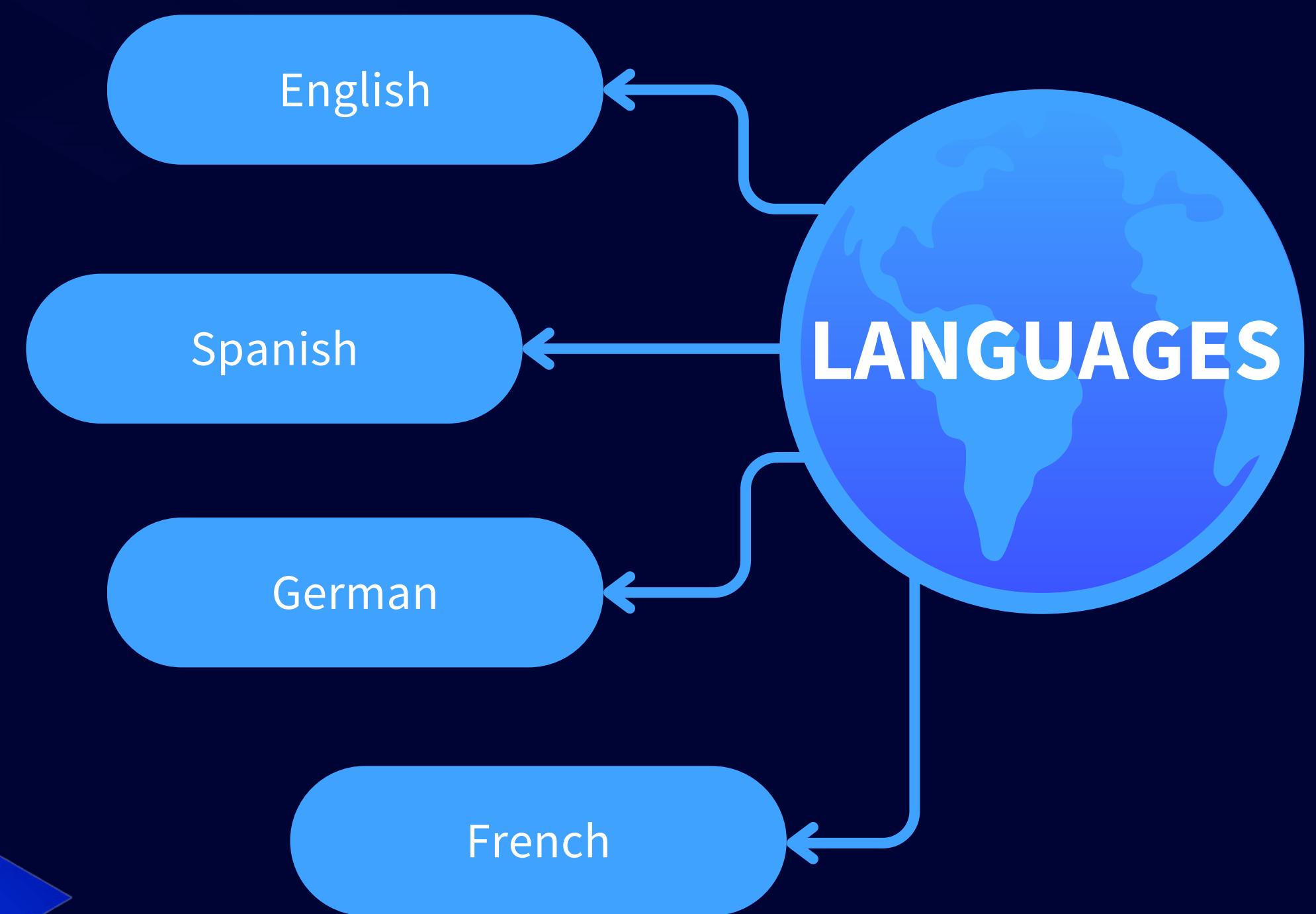
Translation

Text-to-Speech

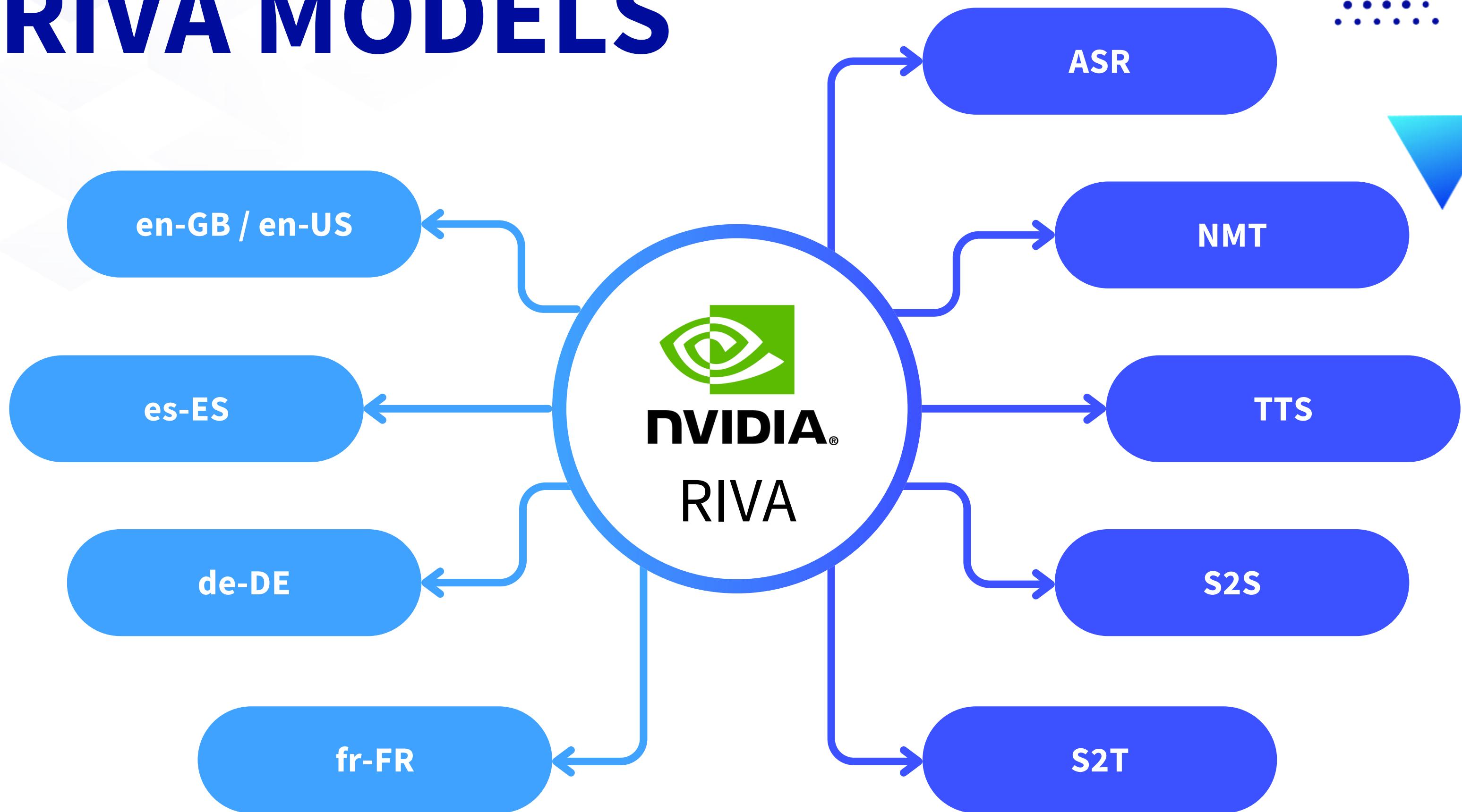
Speech-to-Speech  
Translation

Speech-to-Text  
Translation

# SELECT THE MODELS



# KEY RIVA MODELS





# SET UP RIVA models

# SET UP RIVA MODELS



**CONFIGURE  
NGC CLI**

# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



DOWNLOAD  
RIVA resources

# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file

# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models

# SET UP RIVA MODELS



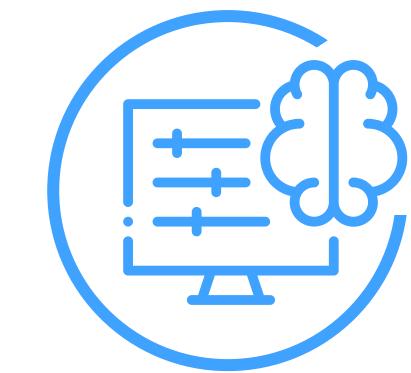
CONFIGURE  
NGC CLI



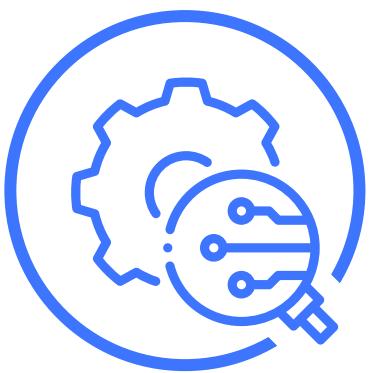
DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models



TEST  
it locally

# SET UP RIVA MODELS



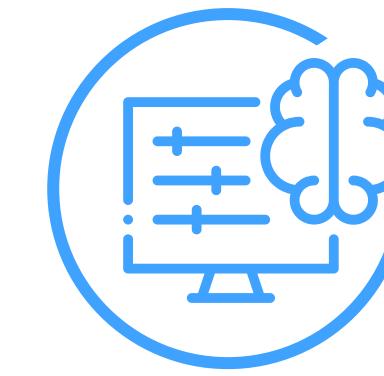
CONFIGURE  
NGC CLI



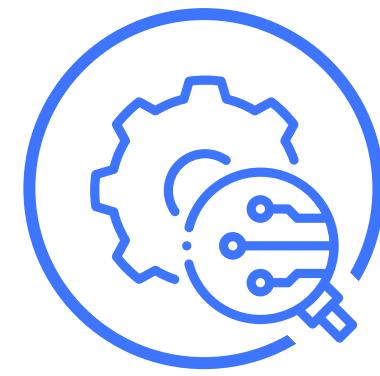
DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models



TEST  
it locally



STORE  
RIVA models

# SET UP RIVA MODELS



**CONFIGURE**  
**NGC CLI**



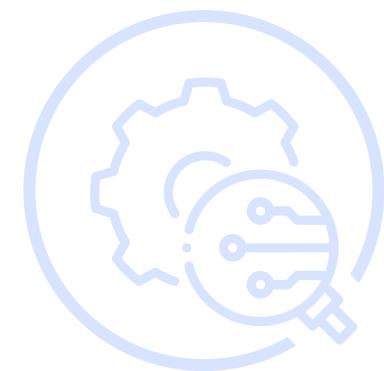
**DOWNLOAD**  
RIVA resources



**CUSTOMIZE**  
configuration file



**INITIALIZE**  
RIVA models



**TEST**  
it locally



**STORE**  
RIVA models

# CONFIGURE NGC CLI

The terminal window shows the following command sequence:

```
[ubuntu@nvidia-models-building:~$ docker login nvcr.io
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ubuntu@nvidia-models-building:~$ ngc registry resource download-version nvidia/riva/riva_quickstart:2.14.0
Getting files to download...
----- • 147.8/147.8     • Remaining: 0:00:00 • 165.3 kB/s • Elapsed: 0:00:02 • Total: 26 - Completed: 26 - Failed: 0
          KiB

-----
Download status: COMPLETED
Downloaded local path resource: /home/ubuntu/riva_quickstart_v2.14.0
Total files downloaded: 26
Total transferred: 147.85 KB
Started at: 2024-01-19 16:33:34
Completed at: 2024-01-19 16:33:37
Duration taken: 2s
-----
[ubuntu@nvidia-models-building:~$ ls riva_quickstart_v2.14.0/
asr_lm_tools      examples           protos        riva_clean.sh  riva_start.sh    riva_stop.sh
config.sh         nemo2riva-2.14.0-py3-none-any.whl  riva-model-repo  riva_init.sh   riva_start_client.sh
ubuntu@nvidia-models-building:~$ ]
```

# SET UP RIVA MODELS



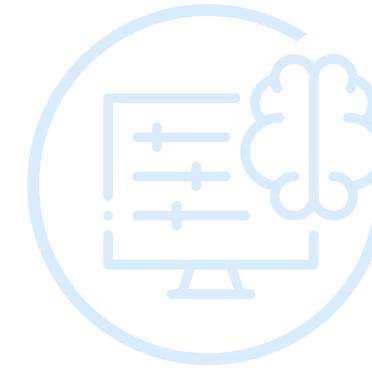
CONFIGURE  
NGC CLI



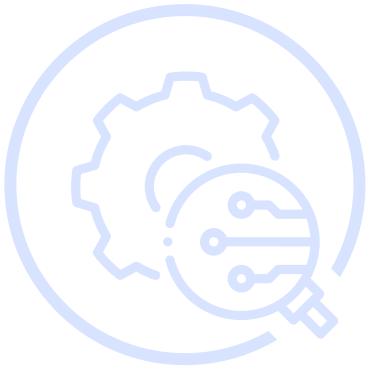
**DOWNLOAD**  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models

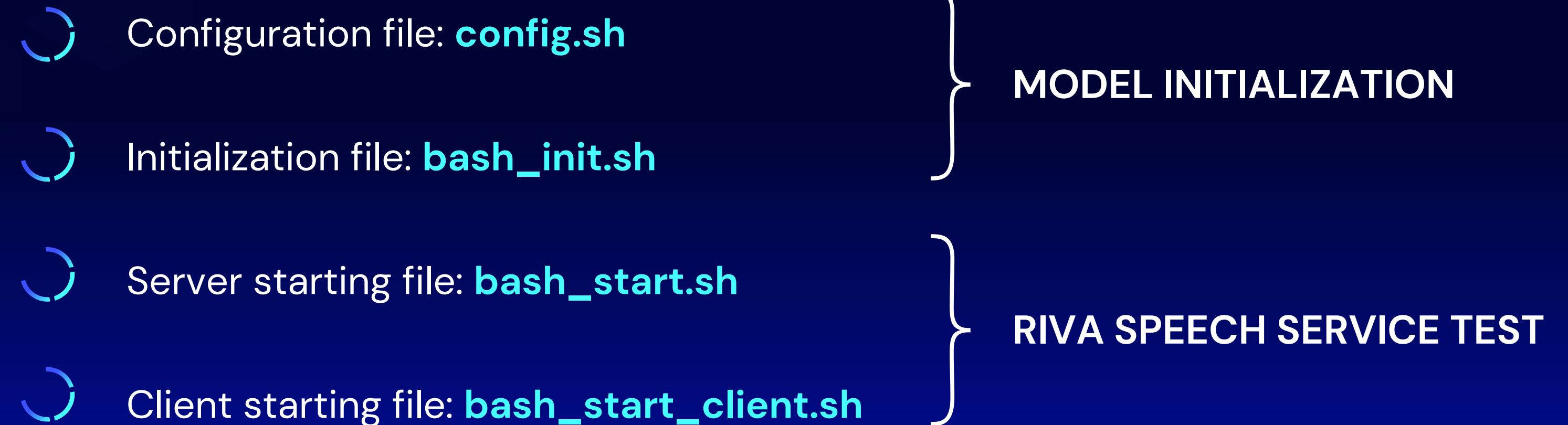


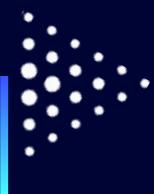
TEST  
it locally



STORE  
RIVA models

# DOWNLOAD RIVA RESOURCES

- ⟳ Configuration file: **config.sh**
  - ⟳ Initialization file: **bash\_init.sh**
  - ⟳ Server starting file: **bash\_start.sh**
  - ⟳ Client starting file: **bash\_start\_client.sh**
- 
- The diagram uses curly braces to group the resources. The first two items are grouped under the label "MODEL INITIALIZATION". The last two items are grouped under the label "RIVA SPEECH SERVICE TEST".



# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



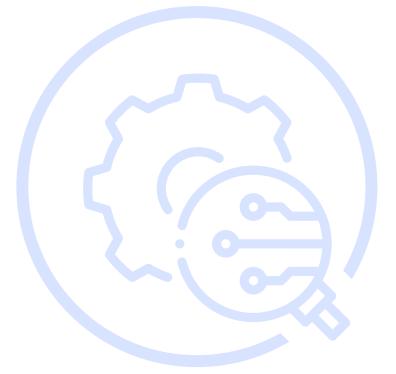
DOWNLOAD  
RIVA resources



**CUSTOMIZE**  
**configuration file**



INITIALIZE  
RIVA models



TEST  
it locally



STORE  
RIVA models

# CUSTOMIZE CONFIGURATION FILE



- ↳ Set ASR service at True
- ↳ Select languages (en, de, es, fr)
- ↳ Allow punctuation
- ↳ Active diarization

# CUSTOMIZE CONFIGURATION FILE



- ↳ Set ASR service at True
- ↳ Select languages (en, de, es, fr)
- ↳ Allow punctuation
- ↳ Active diarization

- ↳ Set TTS service at True
- ↳ Select languages (en, de, es)

# CUSTOMIZE CONFIGURATION FILE

## ASR models

- ↳ Set ASR service at True
- ↳ Select languages (en, de, es, fr)
- ↳ Allow punctuation
- ↳ Active diarization

## TTS models

- ↳ Set TTS service at True
- ↳ Select languages (en, de, es)

## NMT models

- ↳ Set NMT service at True
- ↳ Active bilingual models for the following languages: en, es, de and fr

# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



**INITIALIZE  
RIVA models**



TEST  
it locally



STORE  
RIVA models

# INITIALIZE RIVA MODELS



**LAUNCH**  
initialization script

# INITIALIZE RIVA MODELS



**LAUNCH**  
initialization script



**CHECK**  
model directory

# INITIALIZE RIVA MODELS



**LAUNCH**  
initialization script



**CHECK**  
model directory



**CHANGE**  
user rights

# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



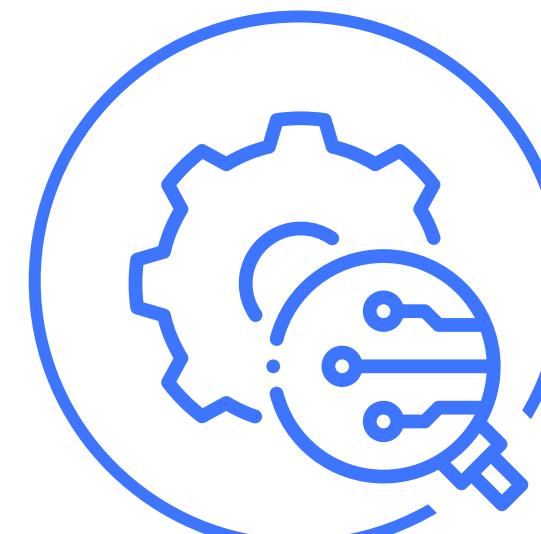
DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models

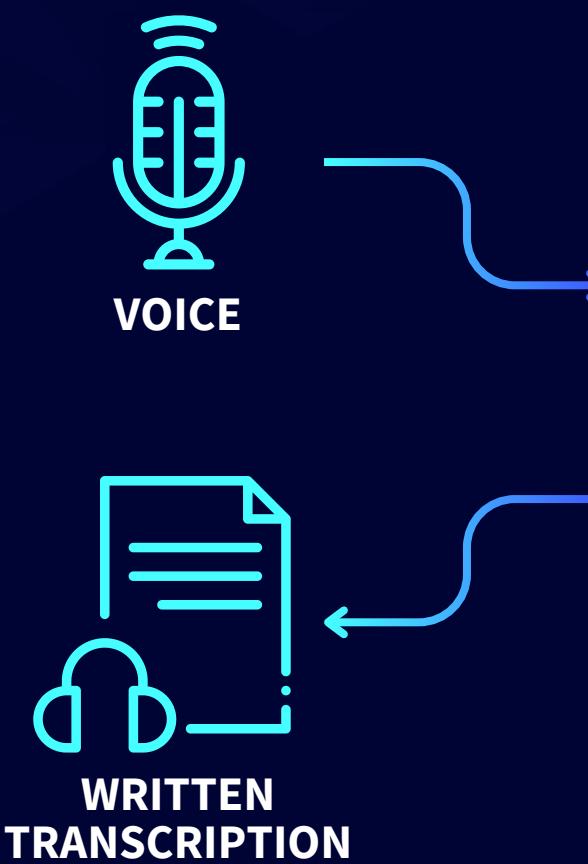


**TEST  
it locally**

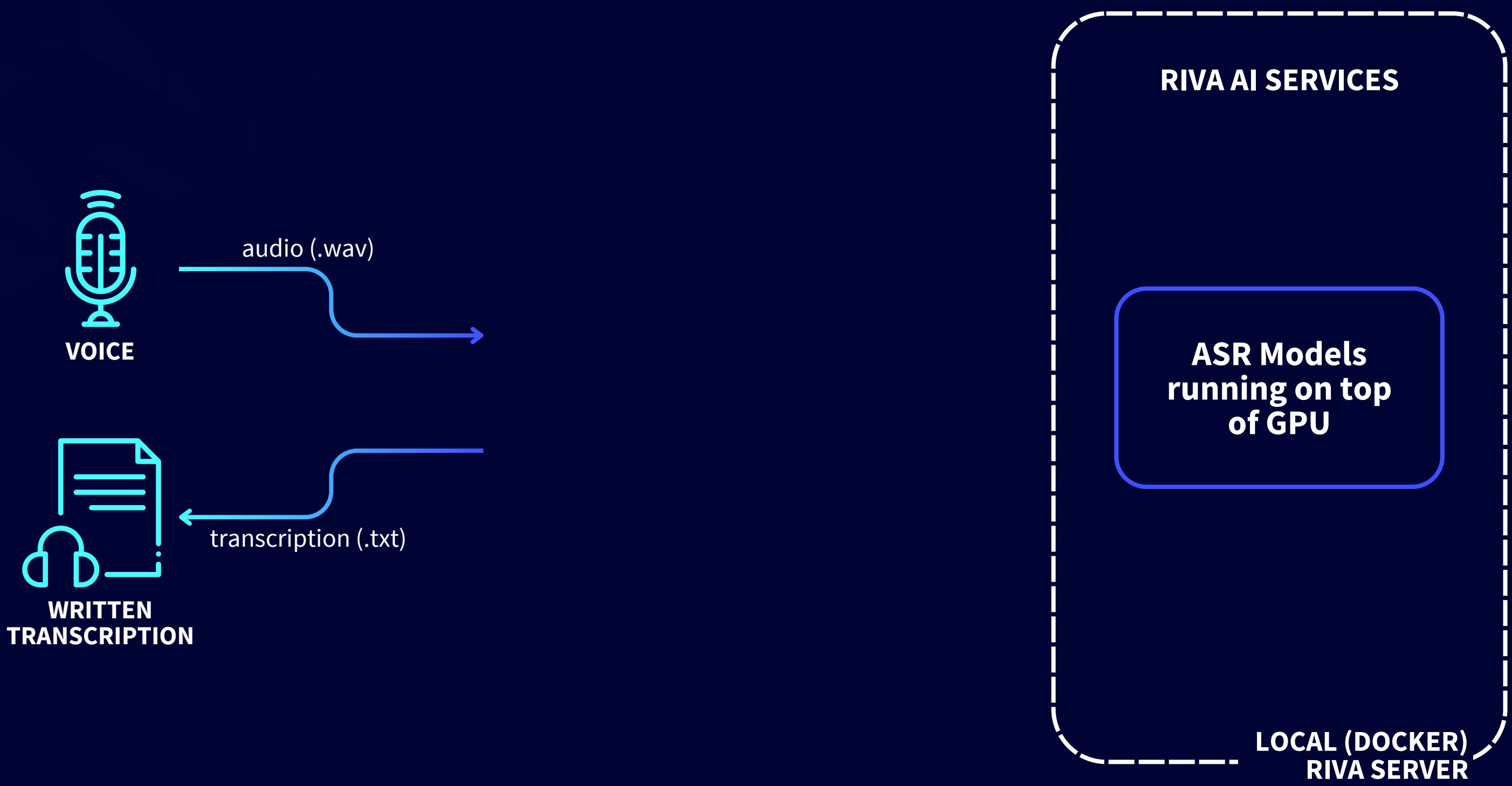


STORE  
RIVA models

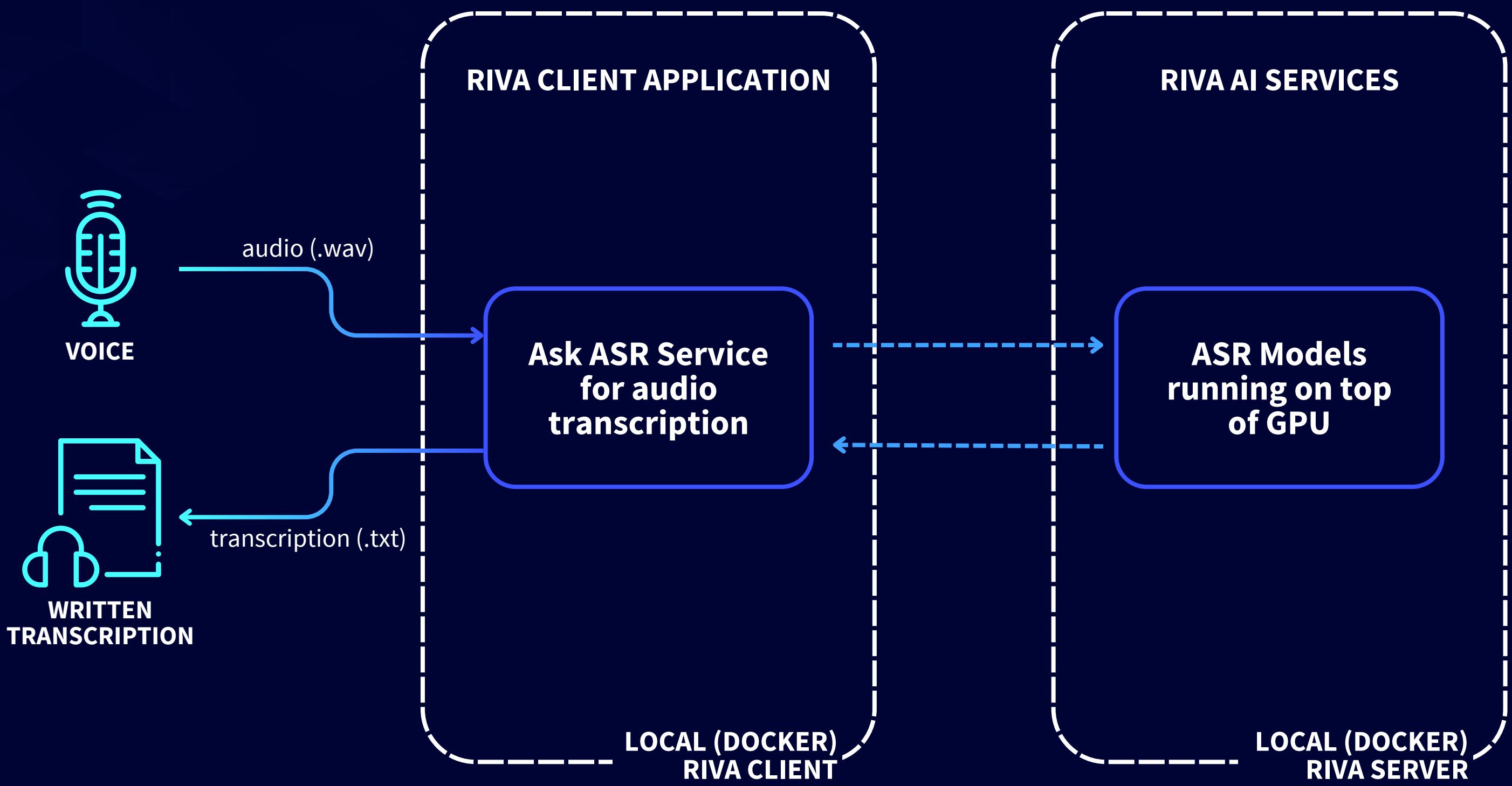
# TEST IT LOCALLY



# TEST IT LOCALLY



# TEST IT LOCALLY



# SET UP RIVA MODELS



CONFIGURE  
NGC CLI



DOWNLOAD  
RIVA resources



CUSTOMIZE  
configuration file



INITIALIZE  
RIVA models

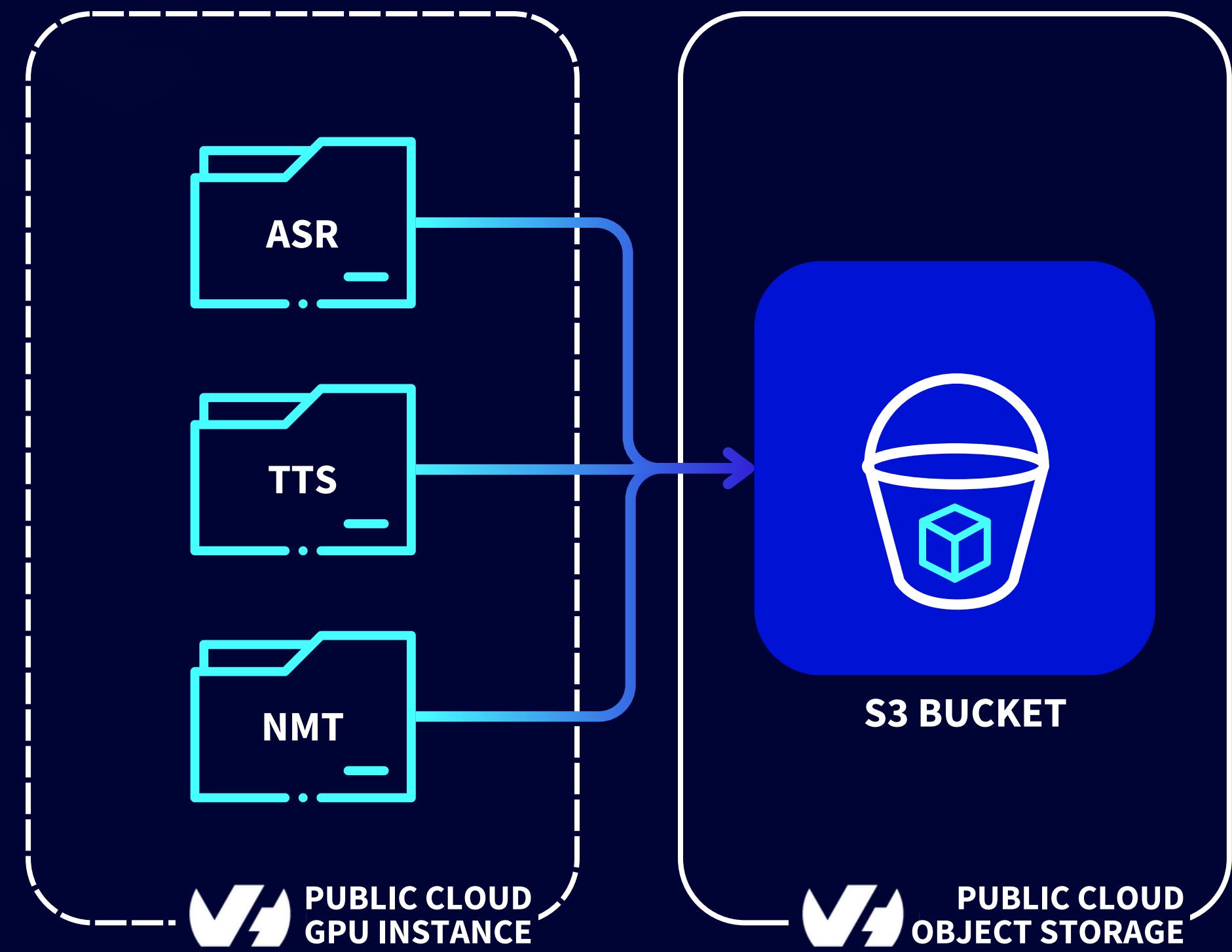


TEST  
it locally



**STORE**  
RIVA models

# STORE RIVA MODELS



# DEPLOY<sup>⚡</sup>

## RIVA server

# DEPLOY RIVA SERVER



## COMPUTE RESOURCES

- 1 GPU
- H100

## HIGH AVAILABILITY

- API scalable on the fly
- Custom number of replicas

## SECURE ACCESS

- Private mode
- Personal token access

# OVHcloud AI DEPLOY

01

**CONTAINER  
AS A SERVICE**

Customer provides a docker container through Docker registry

# OVHcloud AI DEPLOY

01 **CONTAINER  
AS A SERVICE**

Customer provides a docker container through Docker registry

02 **COMPUTE  
RESOURCES**

Container is running in the cloud over GPU (or CPU)

# OVHcloud AI DEPLOY

01 **CONTAINER AS A SERVICE**

Customer provides a docker container through Docker registry

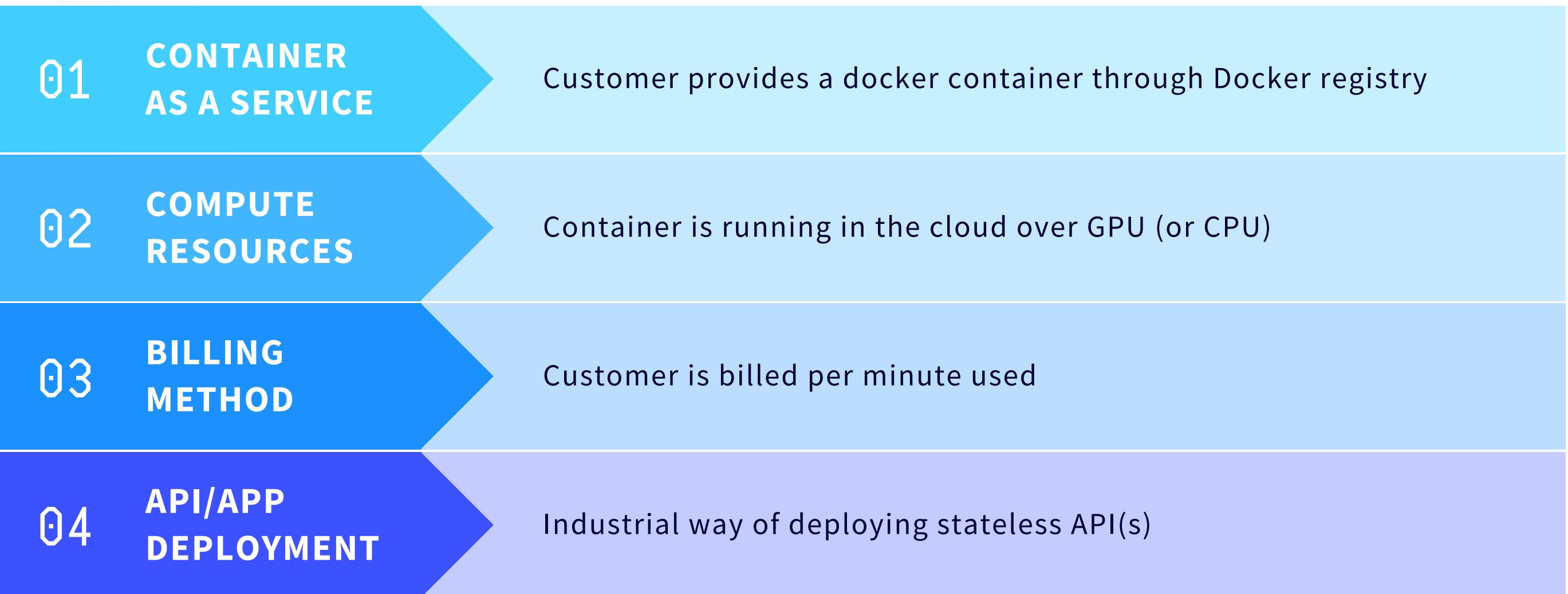
02 **COMPUTE RESOURCES**

Container is running in the cloud over GPU (or CPU)

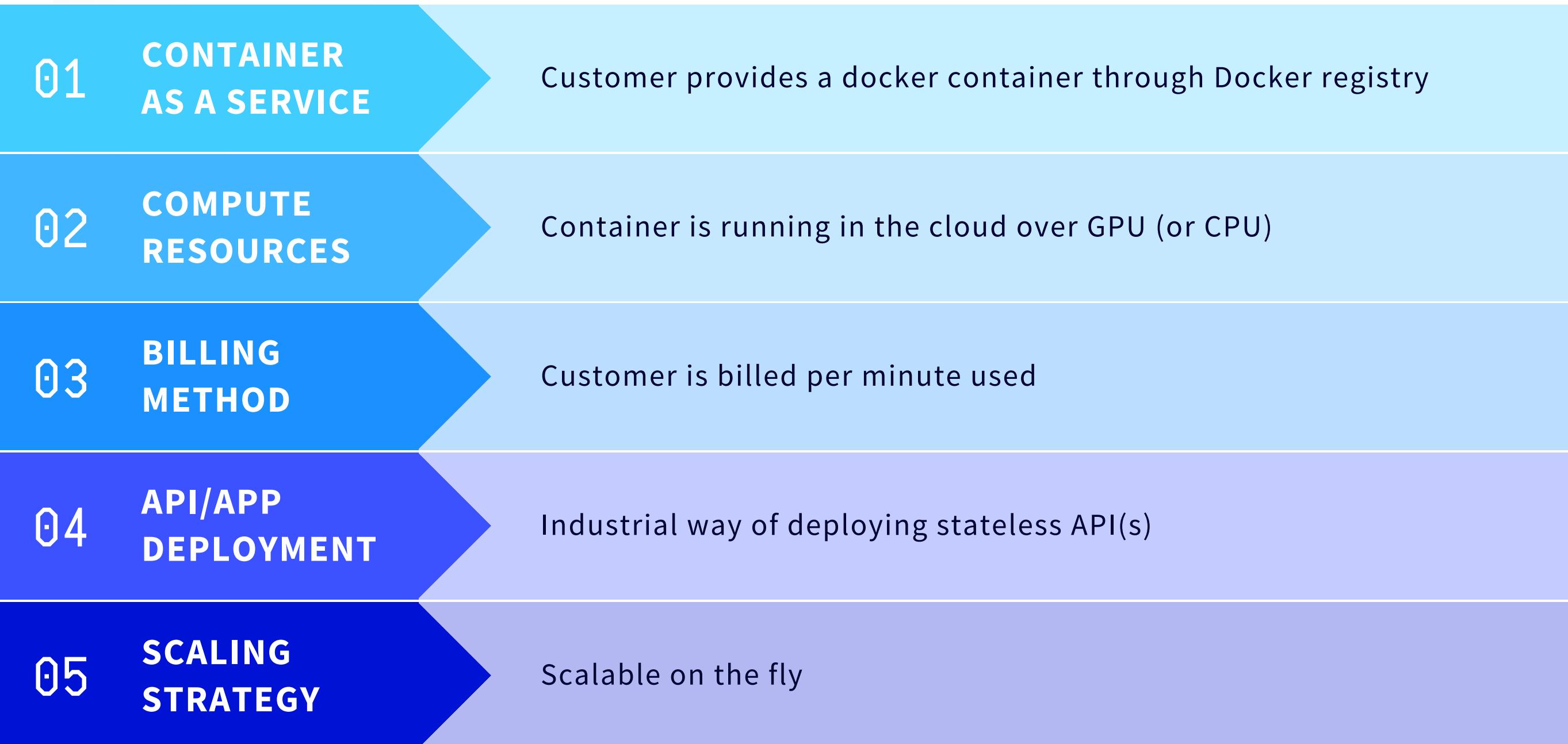
03 **BILLING METHOD**

Customer is billed per minute used

# OVHcloud AI DEPLOY

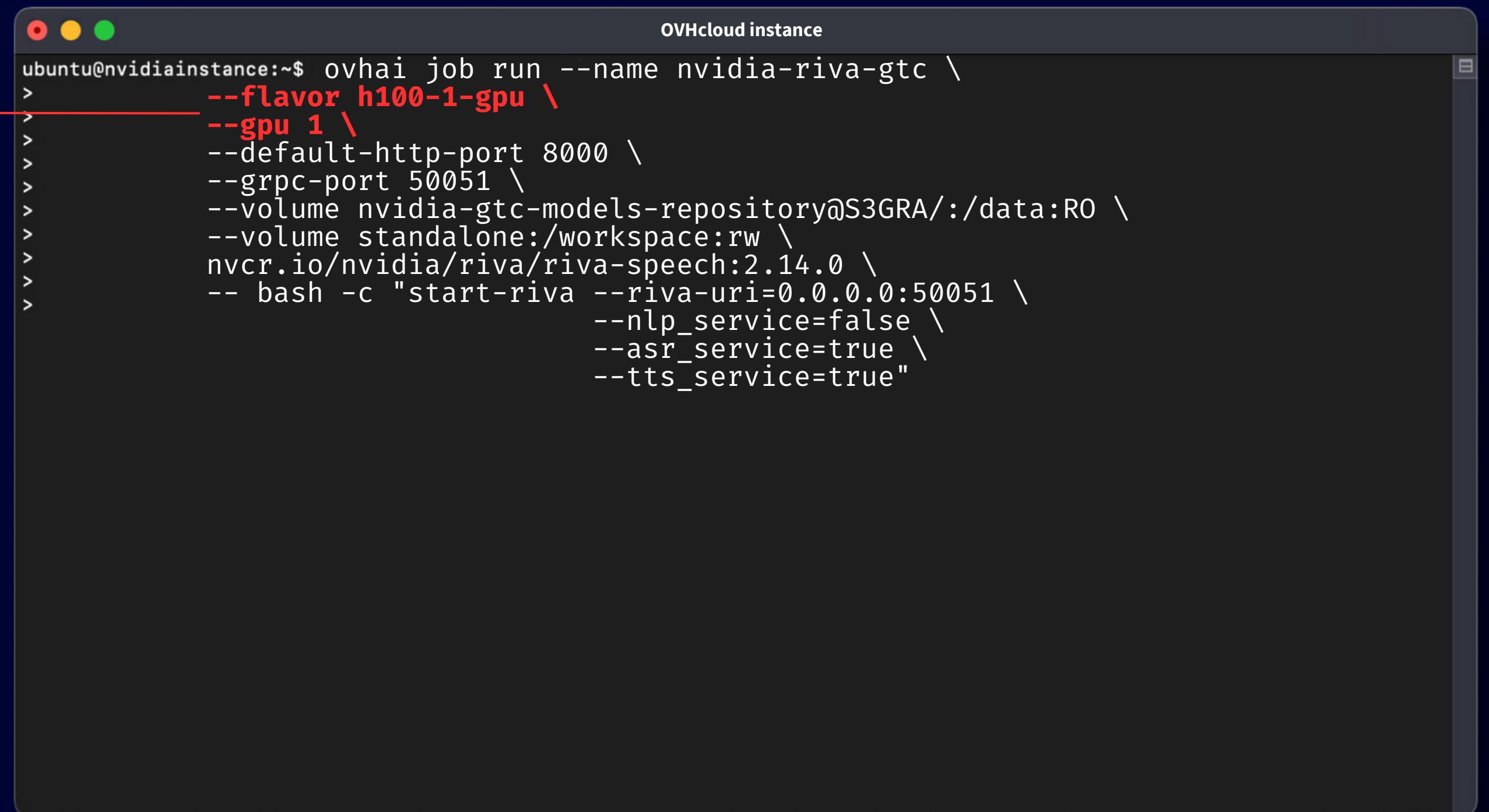


# OVHcloud AI DEPLOY



# RIVA SERVER x AI DEPLOYMENT

Select the flavor and  
the number of GPUs



A terminal window titled "OVHcloud instance" showing a command being run. The command is:

```
ubuntu@nvidiainstance:~$ ovhai job run --name nvidia-riva-gtc \
>   --flavor h100-1-gpu \
>   --gpu 1 \
>   --default-http-port 8000 \
>   --grpc-port 50051 \
>   --volume nvidia-gtc-models-repository@S3GRA/:/data:RO \
>   --volume standalone:/workspace:rw \
>   nvcr.io/nvidia/riva/riva-speech:2.14.0 \
>   -- bash -c "start-riva --riva-uri=0.0.0.0:50051 \
>             --nlp_service=false \
>             --asr_service=true \
>             --tts_service=true"
```

The text "Select the flavor and the number of GPUs" is highlighted in red and has a red arrow pointing to the "--flavor" and "--gpu" parameters in the terminal command.

# RIVA SERVER x AI DEPLOYMENT

Precise HTTP  
and gRPC port

```
ubuntu@nvidiainstance:~$ ovhai job run --name nvidia-riva-gtc \
>           --flavor h100-1-gpu \
>           --gpu 1 \
>           --default-http-port 8000 \
>           --grpc-port 50051 \
>           --volume nvidia-gtc-models-repository@S3GRA/:/data:RO \
>           --volume standalone:/workspace:rw \
>           nvcr.io/nvidia/riva/riva-speech:2.14.0 \
>           -- bash -c "start-riva --riva-uri=0.0.0.0:50051 \
>                         --nlp_service=false \
>                         --asr_service=true \
>                         --tts_service=true"
```

# RIVA SERVER x AI DEPLOYMENT

Attach your S3 buckets  
with RIVA models

```
ubuntu@nvidiainstance:~$ ovhai job run --name nvidia-riva-gtc \
>   --flavor h100-1-gpu \
>   --gpu 1 \
>   --default-http-port 8000 \
>   --grpc-port 50051 \
>   --volume nvidia-gtc-models-repository@S3GRA:/data:RO \
>   --volume standalone:/workspace:rw \
>   nvcr.io/nvidia/riva/riva-speech:2.14.0 \
>   -- bash -c "start-riva --riva-uri=0.0.0.0:50051 \
>   --nlp_service=false \
>   --asr_service=true \
>   --tts_service=true"
```

# RIVA SERVER x AI DEPLOY

Select RIVA Speech  
docker image

```
ubuntu@nvidiainstance:~$ ovhai job run --name nvidia-riva-gtc \
>   --flavor h100-1-gpu \
>   --gpu 1 \
>   --default-http-port 8000 \
>   --grpc-port 50051 \
>   --volume nvidia-gtc-models-repository@S3GRA/:/data:RO \
>   --volume standalone:/workspace:rw \
>   nvcr.io/nvidia/riva/riva-speech:2.14.0 \
>   -- bash -c "start-riva --riva-uri=0.0.0.0:50051 \
>   --nlp_service=false \
>   --asr_service=true \
>   --tts_service=true"
```

# RIVA SERVER x AI DEPLOYMENT

Start RIVA server  
for ASR / TTS / NMT

```
ubuntu@nvidiainstance:~$ ovhai job run --name nvidia-riva-gtc \
>   --flavor h100-1-gpu \
>   --gpu 1 \
>   --default-http-port 8000 \
>   --grpc-port 50051 \
>   --volume nvidia-gtc-models-repository@S3GRA/:/data:RO \
>   --volume standalone:/workspace:rw \
>   nvcr.io/nvidia/riva/riva-speech:2.14.0 \
>   -- bash -c "start-riva --riva-uri=0.0.0.0:50051 \
>   --nlp_service=false \
>   --asr_service=true \
>   --tts_service=true"
```

# CUSTOMIZE RIVA vocabulary

# WHAT CAN BE CUSTOMIZED?



↳ increased chances of recognition  
for a provided list of keywords

# WHAT CAN BE CUSTOMIZED?

 **Boost words** 

↳ increased chances of recognition  
for a provided list of keywords

 **Extend the vocabulary** 

↳ non-vocabulary words  
↳ terminologies  
↳ abbreviations

# WHAT CAN BE CUSTOMIZED?

 **Boost words** 

↳ increased chances of recognition  
for a provided list of keywords

 **Extend the vocabulary** 

↳ non-vocabulary words  
↳ terminologies  
↳ abbreviations

 **Customize the lexicon** 

↳ explicit pronunciation(s), in the  
form of tokenized sequences

# TEST RIVA ON OVHcloud AUDIO

```
audio_input = "audio_samples/audio_ovhcloud_en_1.wav"

# open and read audio file
with open(audio_input, 'rb') as fh:
    audio = fh.read()
```

→ Read audio file

```
import riva.client

uri = "10.42.76.209:50051"
auth = riva.client.Auth(uri=uri)
asr_service = riva.client.ASRService(auth)
config = riva.client.RecognitionConfig(
    language_code="en-US",
    max_alternatives=1,
    enable_automatic_punctuation=True,
    enable_word_time_offsets=True,
    audio_channel_count = 1,
)
```

→ Transcribe audio thanks to **RIVA ASR** model

```
#riva.client.add_audio_file_specs_to_config(config, audio)
resp = asr_service.offline_recognize(audio, config)
```

```
print("ASR result:", resp.results[0].alternatives[0].transcript)
```

→ Display result

ASR result: Is your domain name currently with another registrar and you'd like to transfer it to Vh Cloud? You can do this via a transfer procedure.

→ **ERROR → Vh Cloud**

# APPLY WORD BOOSTING

```
import riva.client

uri = "10.42.76.209:50051"
auth = riva.client.Auth(uri=uri)
asr_service = riva.client.ASRService(auth)
config = riva.client.RecognitionConfig(
    language_code="en-US",
    max_alternatives=1,
    enable_automatic_punctuation=True,
    enable_word_time_offsets=True,
    audio_channel_count = 1,
)

boosted_lm_words = ["OVH", "cloud"]
boosted_lm_score = 20.0
riva.client.add_audio_file_specs_to_config(config, audio)
riva.client.add_word_boosting_to_config(config, boosted_lm_words, boosted_lm_score)

resp = asr_service.offline_recognize(audio, config)

print("ASR result:", resp.results[0].alternatives[0].transcript)
```

ASR result: Is your domain name currently with another registrar and you'd like to transfer it to **Vh Cloud**? You can do this via a transfer procedure.

→ Apply Word Boosting



# CUSTOMIZE PRONUNCIATION

```
# tokenizer model path
model_path = "../data-asr/models/conformer-fr-FR-asr-offline-ctc-decoder"
```

Access **tokenizer** model

```
TOKEN="OVHcloud"
PRONUNCIATION="o v h cloud"

import sentencepiece as spm
s = spm.SentencePieceProcessor(model_file=model_path)
for n in range(5):
    print(TOKEN + '\t' + ''.join(
        s.encode(
            (
                PRONUNCIATION,
                out_type=str,
                enable_sampling=True,
                alpha=0.1,
                nbest_size=-1
            )
        )
    ))
```

Generate new **lexicon entries**

```
OVHcloud      - o _ v _ h _ c l ou d
OVHcloud      - o _ v _ h _ c l ou d
OVHcloud      - o _ v _ h _ c l ou d
OVHcloud      - o _ v _ h _ c l ou d
OVHcloud      - o _ v _ h _ c l ou d
```

```
echo -e "OVHcloud\t_ o _ v _ h _ c l ou d" >> lexicon.txt
echo -e "ovh\t_a u _ h" >>
```

Modify the **lexicon file**



# “DEVELOP” RIVA client app

# DEVELOP RIVA CLIENT APP



**Upload** my own video / **copy** a video link

# DEVELOP RIVA CLIENT APP

-  **Upload** my own video / **copy** a video link
-  **Transcribe** the audio part into text

# DEVELOP RIVA CLIENT APP

- ↻ **Upload** my own video / **copy** a video link
- ↻ **Transcribe** the audio part into text
- ↻ **Subtitle** video in any language

# DEVELOP RIVA CLIENT APP

- ↻ **Upload** my own video / **copy** a video link
- ↻ **Transcribe** the audio part into text
- ↻ **Subtitle** video in any language
- ↻ **Dub** the video in another language

# DEVELOP RIVA CLIENT APP

- ↻ **Upload** my own video / **copy** a video link
- ↻ **Transcribe** the audio part into text
- ↻ **Subtitle** video in any language
- ↻ **Dub** the video in another language
- ↻ **Choose** the gender of the dubbing voice

# DEVELOP RIVA CLIENT APP

- ⟳ **Upload** my own video / **copy** a video link
- ⟳ **Transcribe** the audio part into text
- ⟳ **Subtitle** video in any language
- ⟳ **Dub** the video in another language
- ⟳ **Choose** the gender of the dubbing voice
- ⟳ **Download** resulting video

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file



**KEEP SILENCE**  
during translation

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file



**KEEP SILENCE**  
during translation

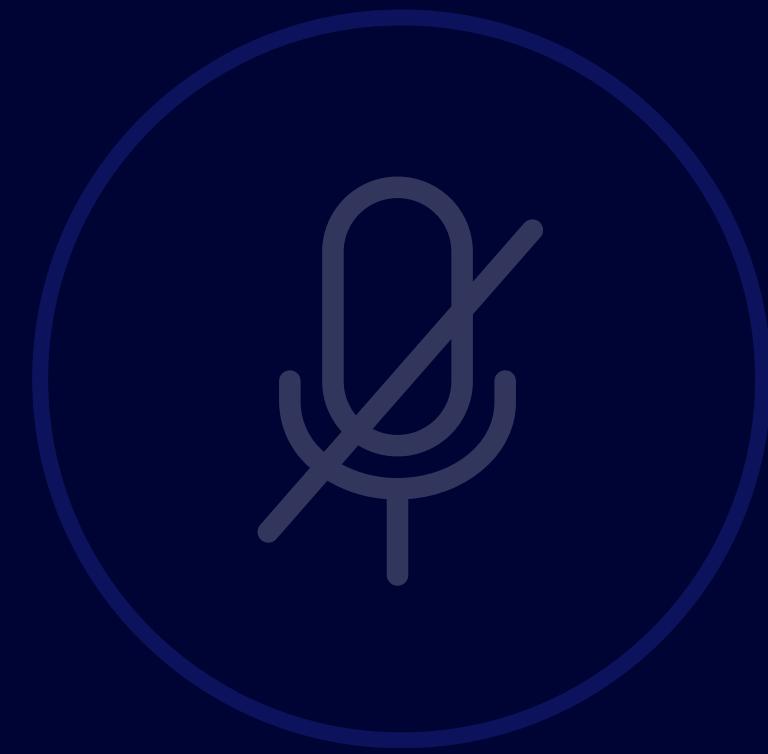


**SUPERIMPOSE**  
audio on video

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file



**KEEP SILENCE**  
during translation



**SUPERIMPOSE**  
audio on video

# GENERATE SRT FILE

The image shows a Jupyter Notebook interface with a dark theme. The title bar reads "1\_GENERATE\_SRT\_FILE.ipynb". The main content area has a heading "RIVA ASR & subtitles generation" and a sub-section "How to generate SRT file?". Below this, a bulleted list says "Get transcription from RIVA ASR model". A code cell [1:] contains the following Python code:

```
[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,
```

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file

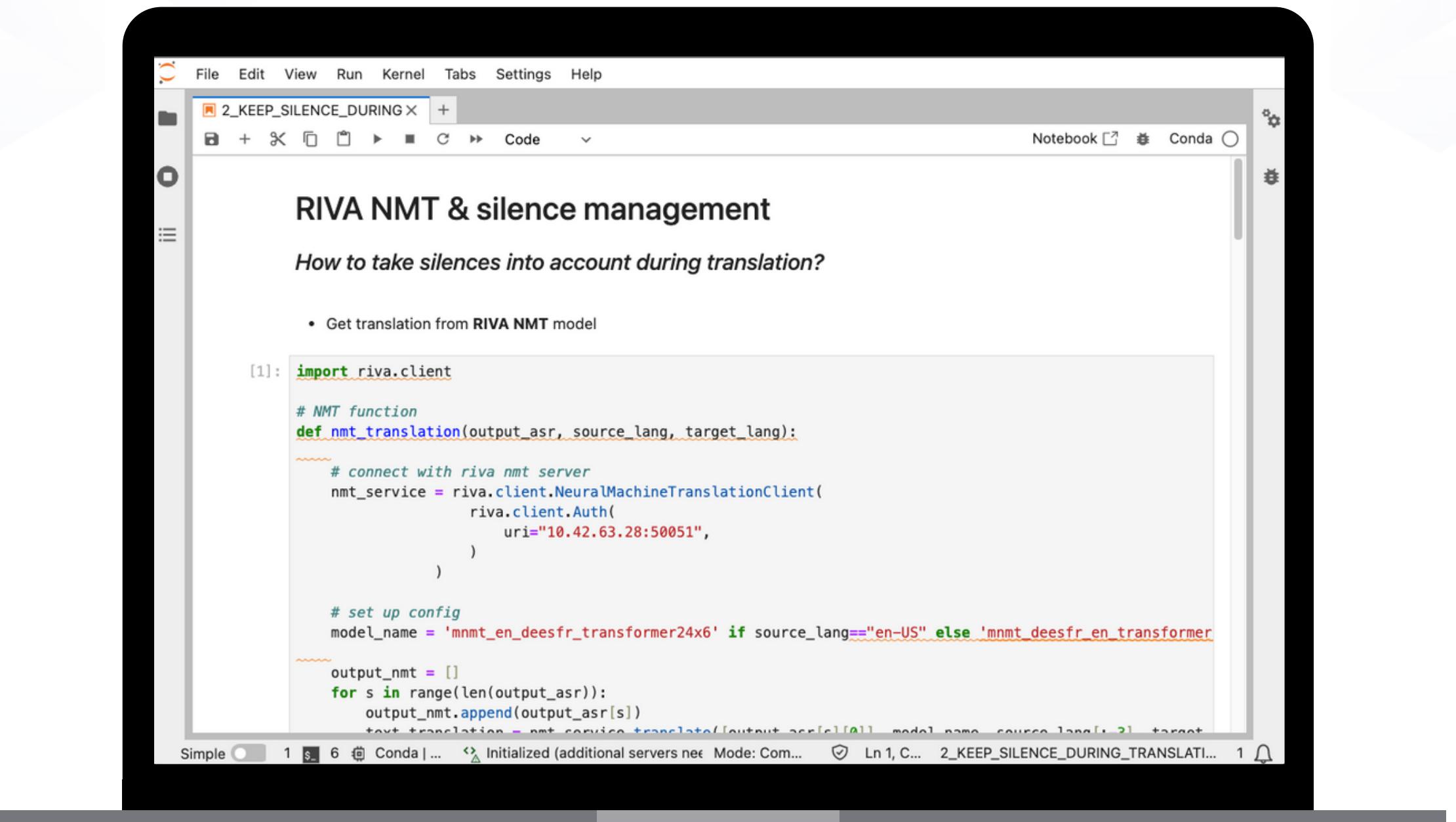


**KEEP SILENCE**  
during translation



**SUPERIMPOSE**  
audio on video

# KEEP SILENCE DURING TRANSLATION



The image shows a laptop screen displaying a Jupyter Notebook interface. The title of the notebook is "RIVA NMT & silence management". Below the title, there is a subtitle "How to take silences into account during translation?". A bulleted list under the subtitle contains one item: "Get translation from RIVA NMT model". The main content of the notebook is a Python code cell labeled [1]:

```
[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)
```

# FOCUS ON KEY FEATURES



**GENERATE**  
SRT file

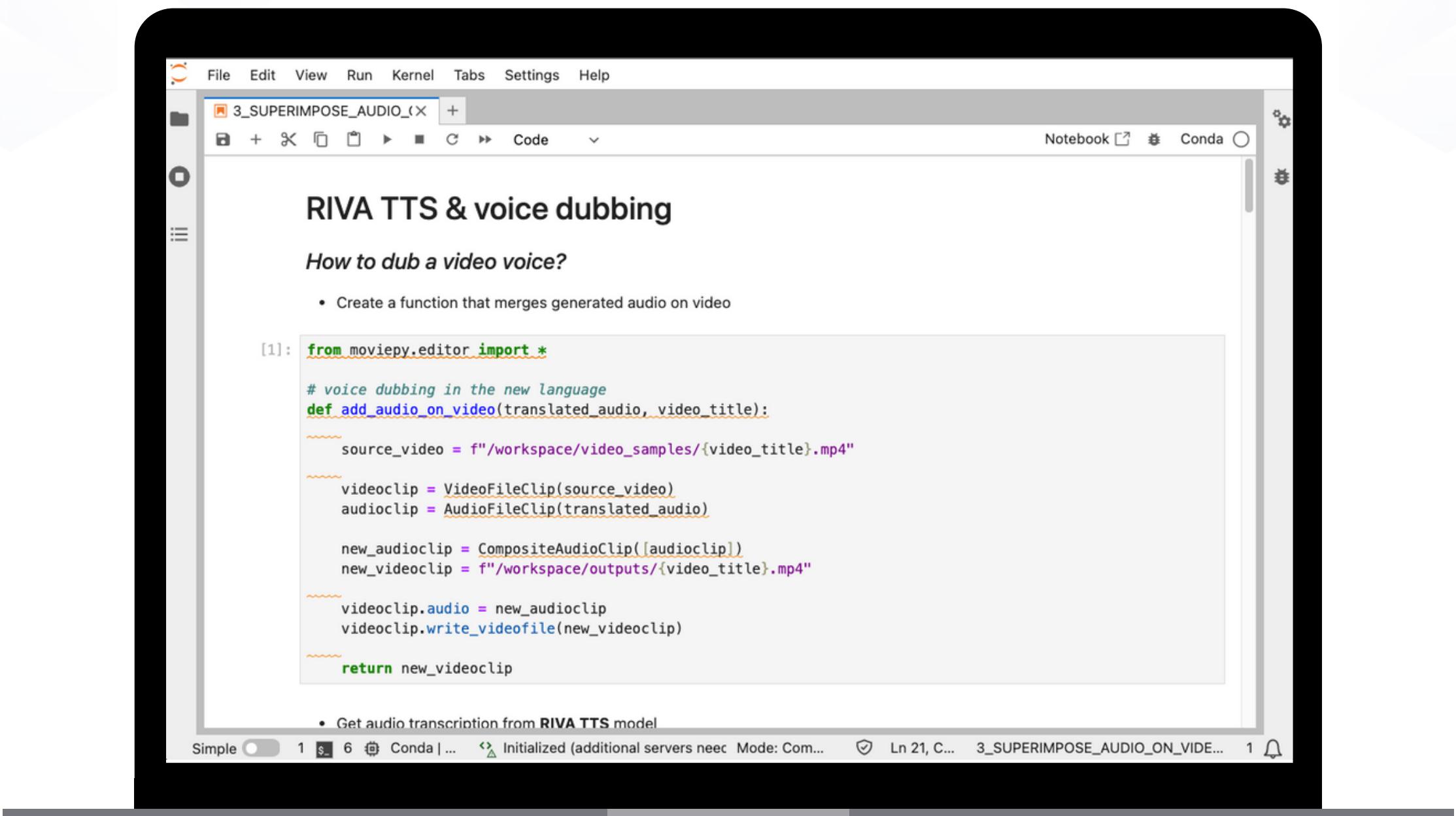


**KEEP SILENCE**  
during translation



**SUPERIMPOSE**  
audio on video

# SUPERIMPOSE AUDIO ON VIDEO



The image shows a laptop screen displaying a Jupyter Notebook titled "3\_SUPERIMPOSE\_AUDIO.ipynb". The notebook has a title section "RIVA TTS & voice dubbing" and a subtitle "How to dub a video voice?". Below the subtitle is a bulleted list: "• Create a function that merges generated audio on video". The code cell [1] contains the following Python code:

```
[1]: from moviepy.editor import *
# voice dubbing in the new language
def add_audio_on_video(translated_audio, video_title):
    source_video = f"/workspace/video_samples/{video_title}.mp4"
    videoclip = VideoFileClip(source_video)
    audioclip = AudioFileClip(translated_audio)

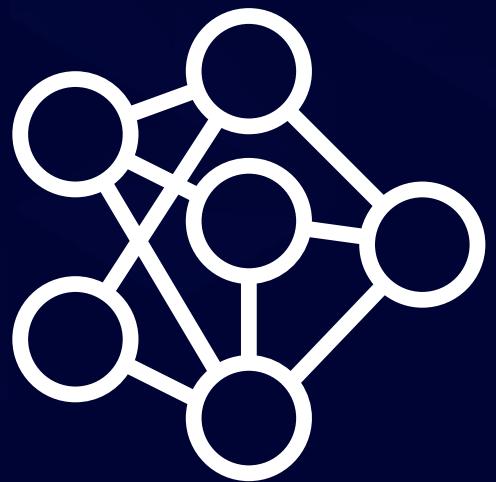
    new_audioclip = CompositeAudioClip([audioclip])
    new_videoclip = f"/workspace/outputs/{video_title}.mp4"

    videoclip.audio = new_audioclip
    videoclip.write_videofile(new_videoclip)

    return new_videoclip
```

At the bottom of the code cell, there is a note: "• Get audio transcription from RIVA TTS model". The status bar at the bottom of the notebook window shows "Simple" mode, "Ln 21, C..." and "3\_SUPERIMPOSE\_AUDIO\_ON\_VIDE...", and a bell icon.

# BUILD GRADIO APP



RIVA models  
server

# BUILD GRADIO APP



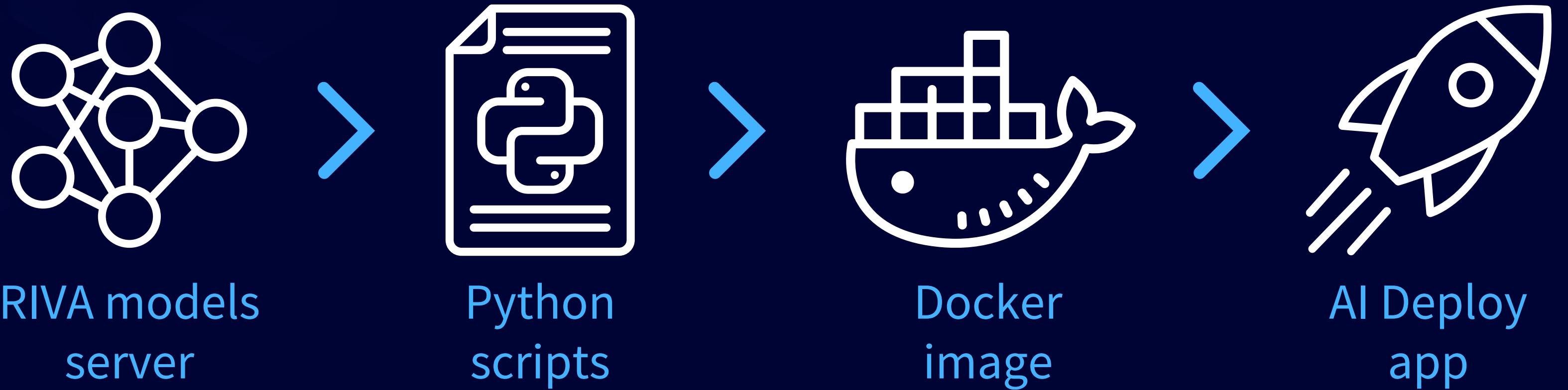
RIVA models  
server

Python  
scripts

# BUILD GRADIO APP



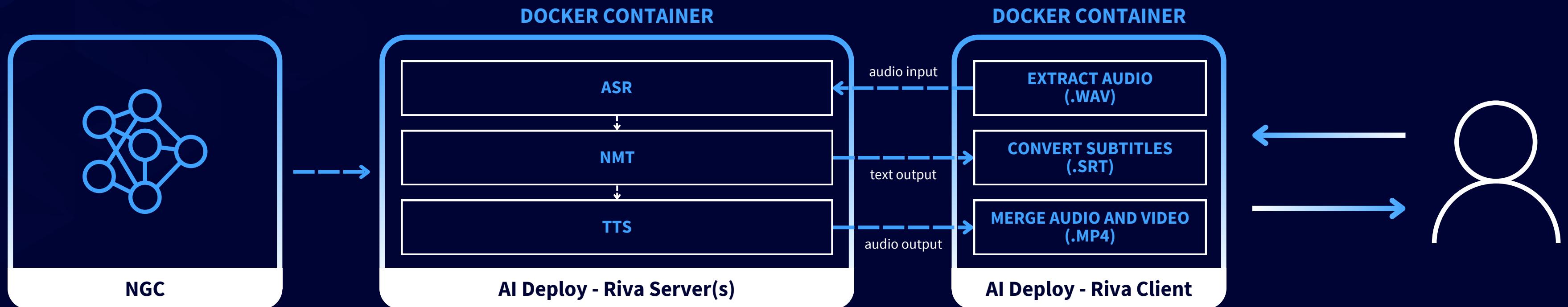
# BUILD GRADIO APP



# DEPLOY<sup>⚡</sup>

## end to end app

# RIVA CLIENT x AI DEPLOY

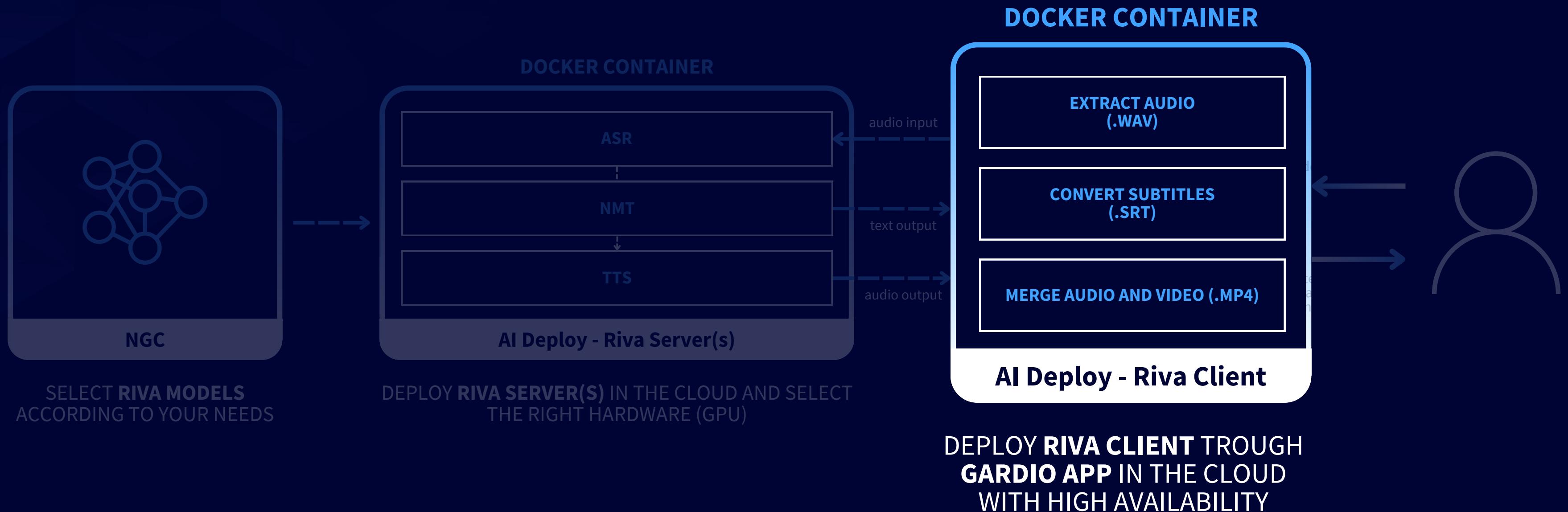


SELECT RIVA MODELS  
ACCORDING TO YOUR NEEDS

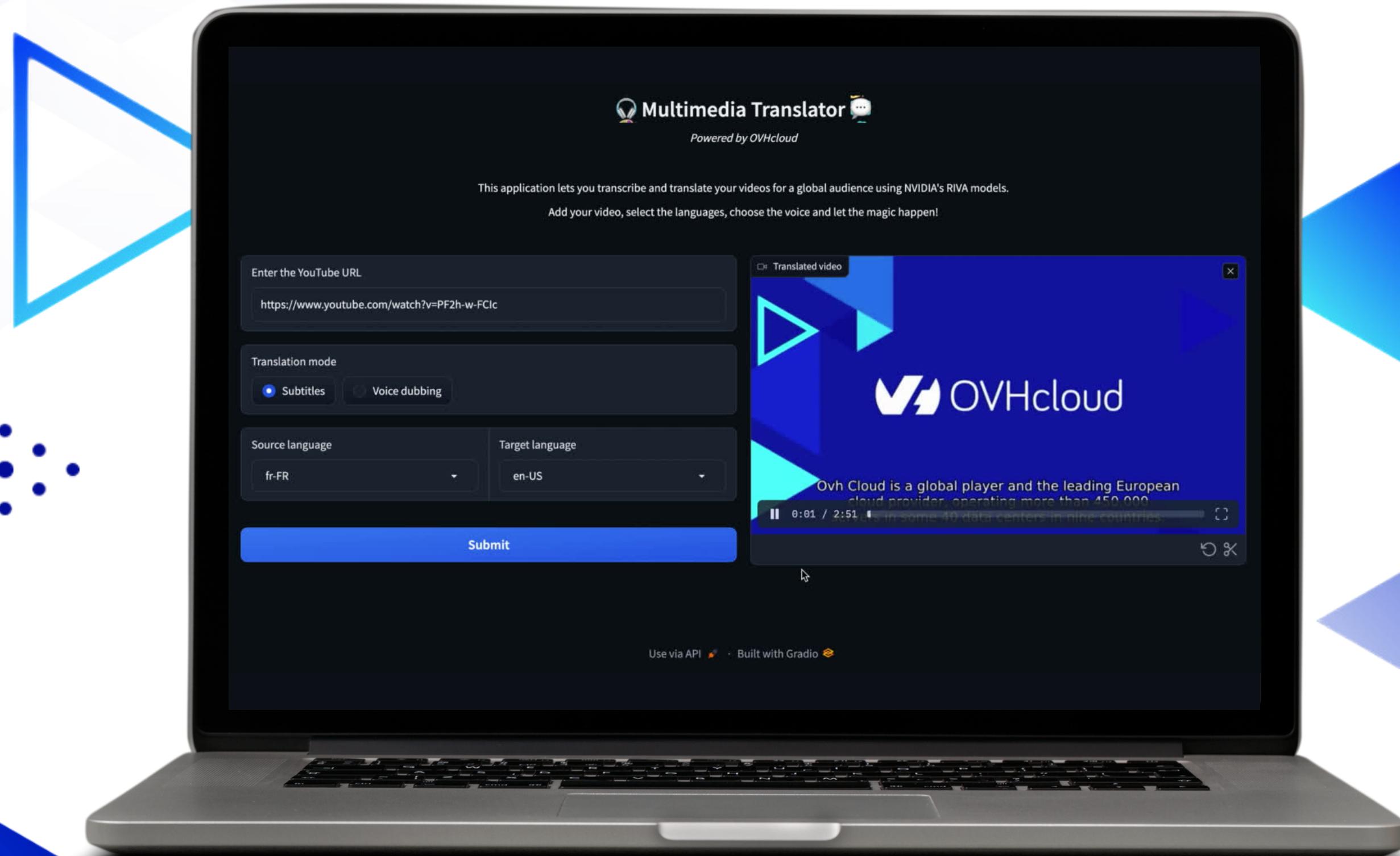
DEPLOY RIVA SERVER(S) IN THE CLOUD AND SELECT  
THE RIGHT HARDWARE (GPU)

DEPLOY RIVA CLIENT THROUGH  
GARDIO APP IN THE CLOUD  
WITH HIGH AVAILABILITY

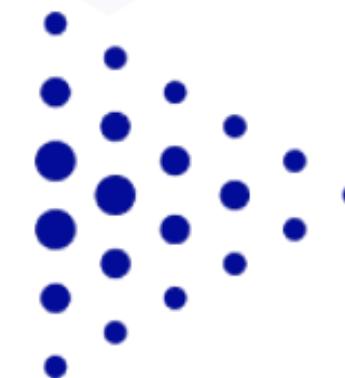
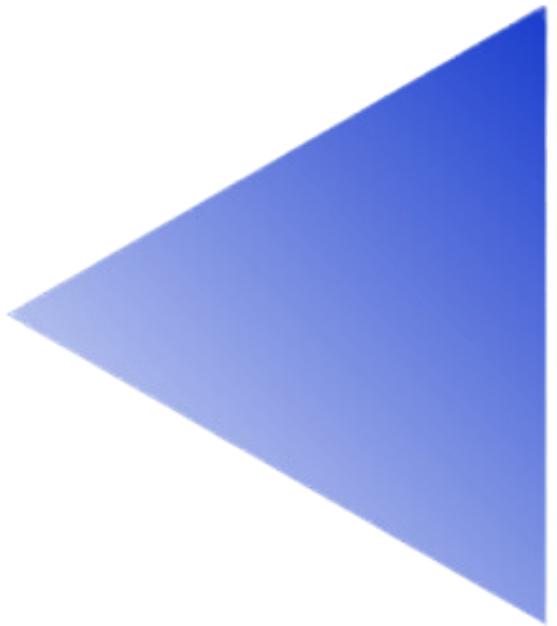
# RIVA CLIENT x AI DEPLOY



# DEMO END TO END APPLICATION



# DEMO END TO END APPLICATION



# CONCLUSION

# CONCLUSION

*The **multimode multimedia translator** in a nutshell!*



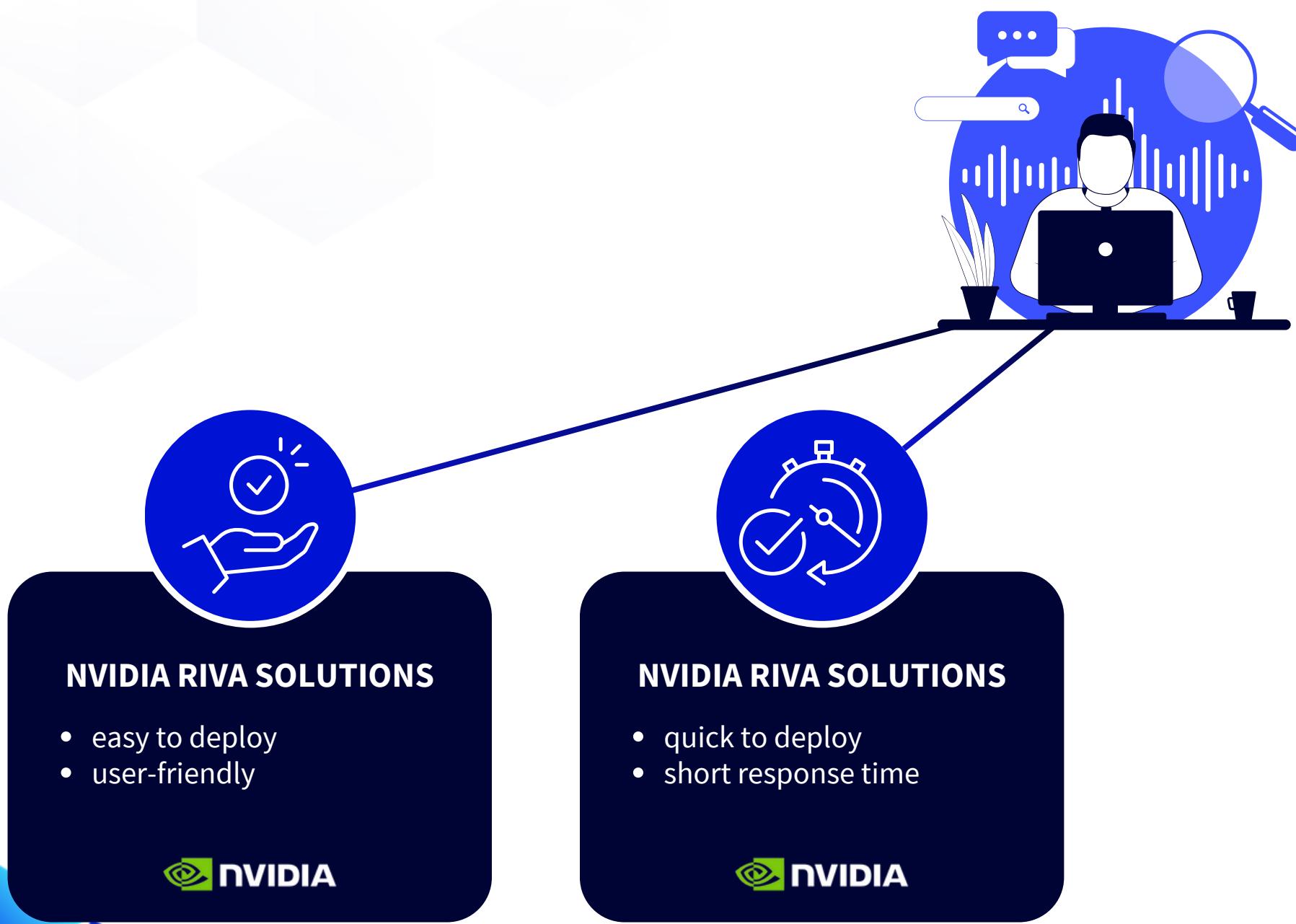
## NVIDIA RIVA SOLUTIONS

- easy to deploy
- user-friendly



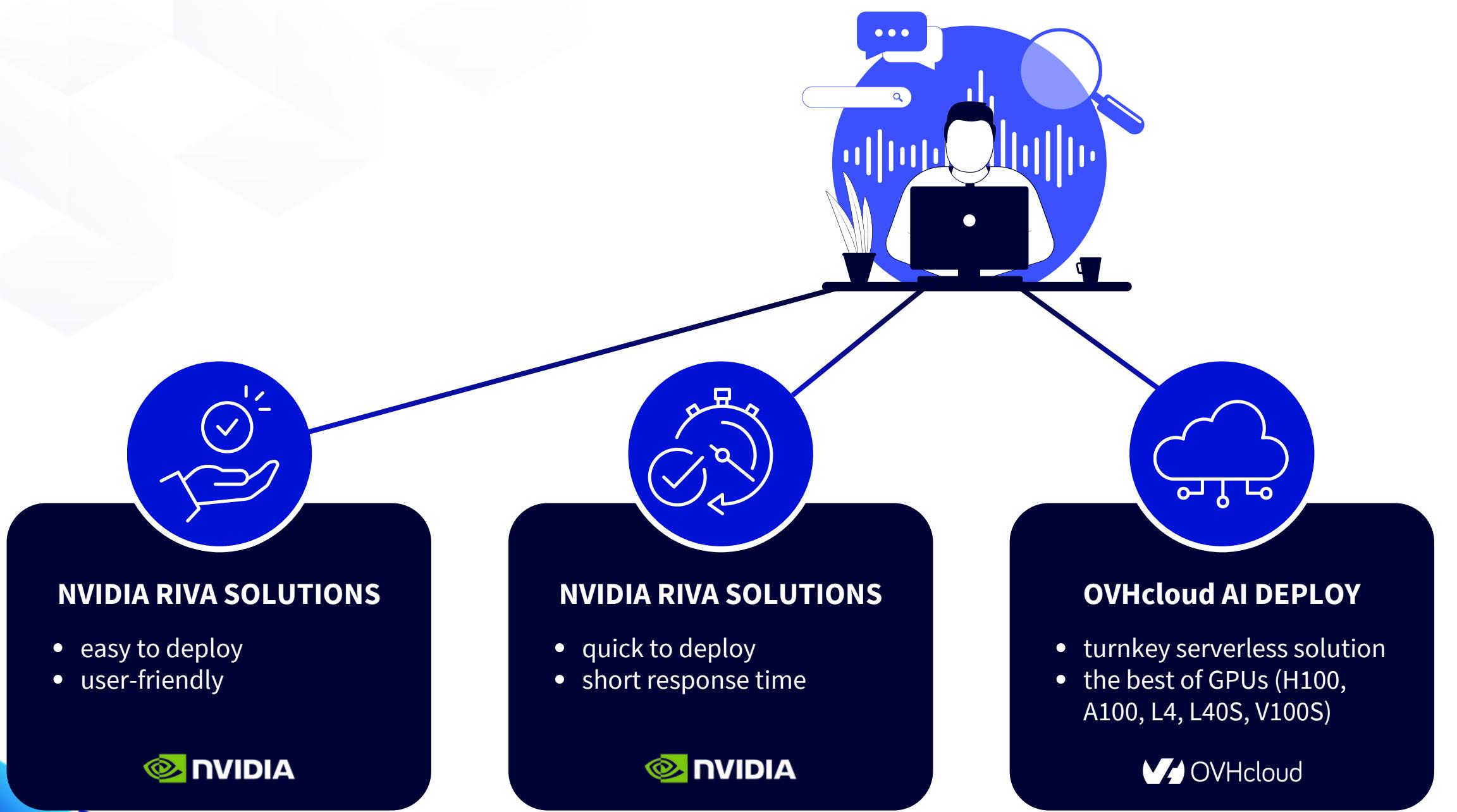
# CONCLUSION

*The **multimode multimedia translator** in a nutshell!*



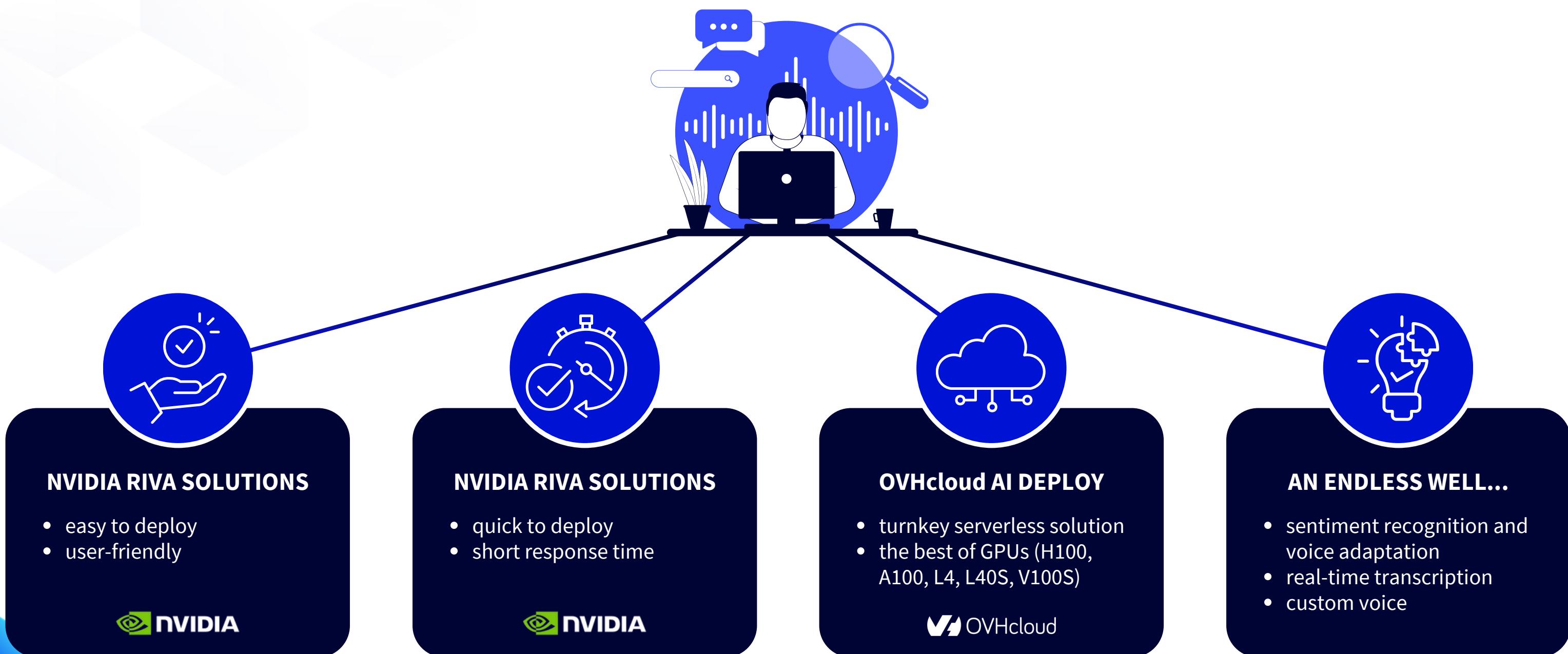
# CONCLUSION

*The **multimode multimedia translator** in a nutshell!*



# CONCLUSION

*The **multimode multimedia translator** in a nutshell!*



# GET CONNECTED WITH US

## Contact Information



<https://www.ovhcloud.com/>



[elea.petton@ovhcloud.com](mailto:elea.petton@ovhcloud.com)



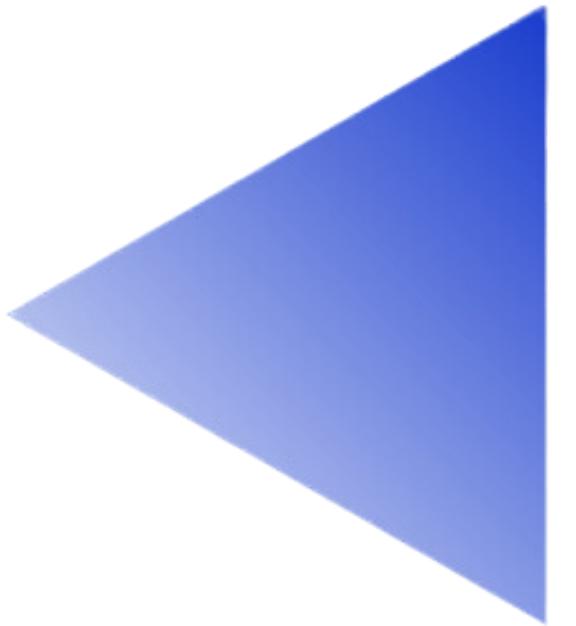
Eléa Petton



@EleaPetton



# IT'S UP TO YOU!



# SPEECH & TRANSLATION AI AT GTC 2024

Join to learn the latest speech & translation AI achievements & how to use them with GenAI-based conversational applications

## **Speaking in Every Language: A Quick Start Guide to TTS Models for Accented, Multilingual Communication [S62517]**

NVIDIA

Mon, 3/18, 8:00 AM - 8:50AM PDT, In-Person, SJCC 230B (L2)

## **Adapting Conformer-Based ASR Models for Conversations Over the Phone [S62441]**

PolyAI

Tue, 3/19, 1:00 AM - 1:25 AM, EMEA Simulive Room 2

## **Secure AI-Driven Translation in Video Conferencing [S61718]**

Pexip

Tue, 3/19, 3:00 AM - 3:50 AM, EMEA Simulive Room 2

## **Multi-Speaker ASR with NVIDIA NeMo Toolkit —Training & Inference [CWE62255]**

NVIDIA

Wed, 3/20, 9:00 AM - 9:50 AM, In-Person, CWE Pod D (LL)

## **Mastering Speech for Multilingual Multimedia Transformation [S62549] & [S62549a]**

OVHcloud, NVIDIA

Tue, 3/19, 9:00 AM - 9:50 AM PDT, In-Person, SJCC 210G (L2) & Fri, 3/22, 5:00 AM - 5:50 AM PDT, EMEA Simulive Room 6

## **Talk to Your Data in Your Native Language [DLI61469]**

NVIDIA

Wed, 3/20, 3:00 PM - 4:40 PM PDT, In-Person, SJCC LL21D (LL)

## **Behind the Scenes of Running a Conversational Character in a 3D Scene [S62570]**

Convai Technologies

Wed, 3/20, 4:30PM – 4:55PM PDT, In-Person, Marriott Ballroom 4 (L2)

## **Build Speech AI for Multilingual Multimedia Transformation [SE62869]**

HPE, Data Monsters, Kore.ai, Quantiphi, NVIDIA

Thur, 3/21, 8:00 AM-12:00 PM PDT, In-Person, Signia Hotel Club Regent (Level 1)

## **Speech AI Demystified [S61523]**

NVIDIA

Thur, 3/21, 10:00 AM - 10:50 AM PDT, In-Person, SJCC 210F (L2)