

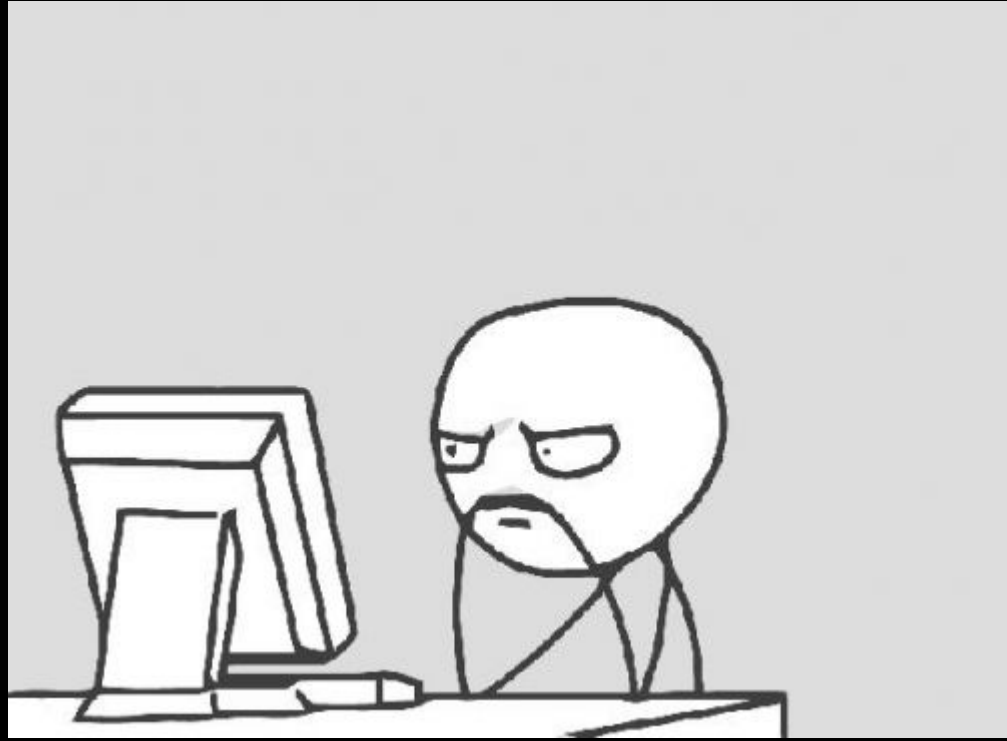
# BUILDING MACHINE LEARNING DEMOS WITH PYTHON

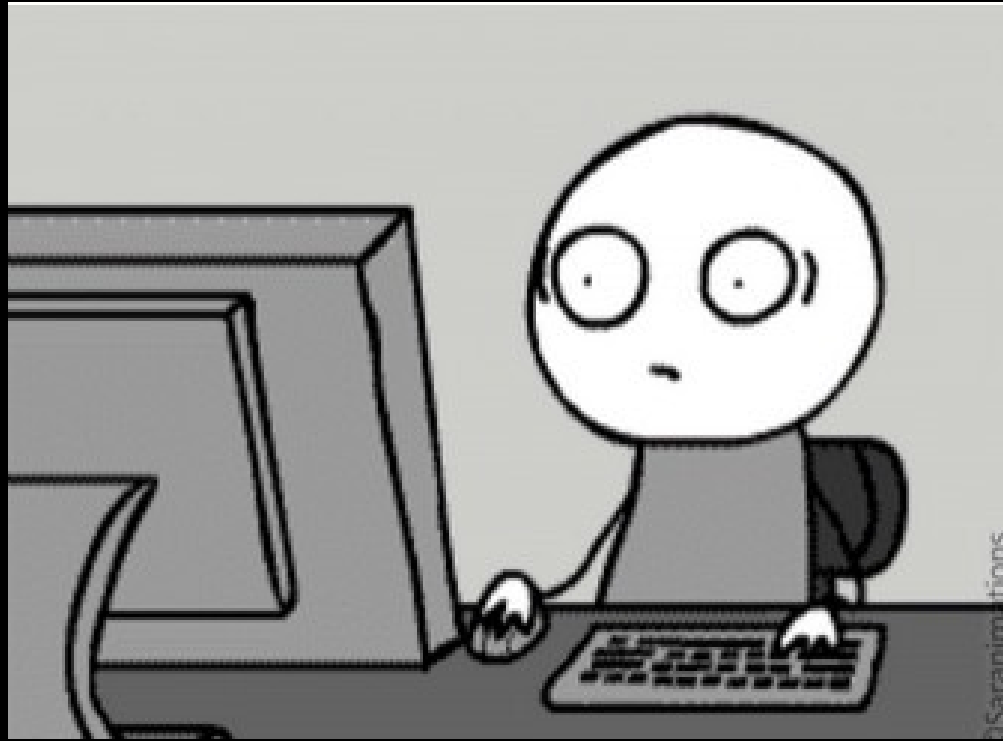
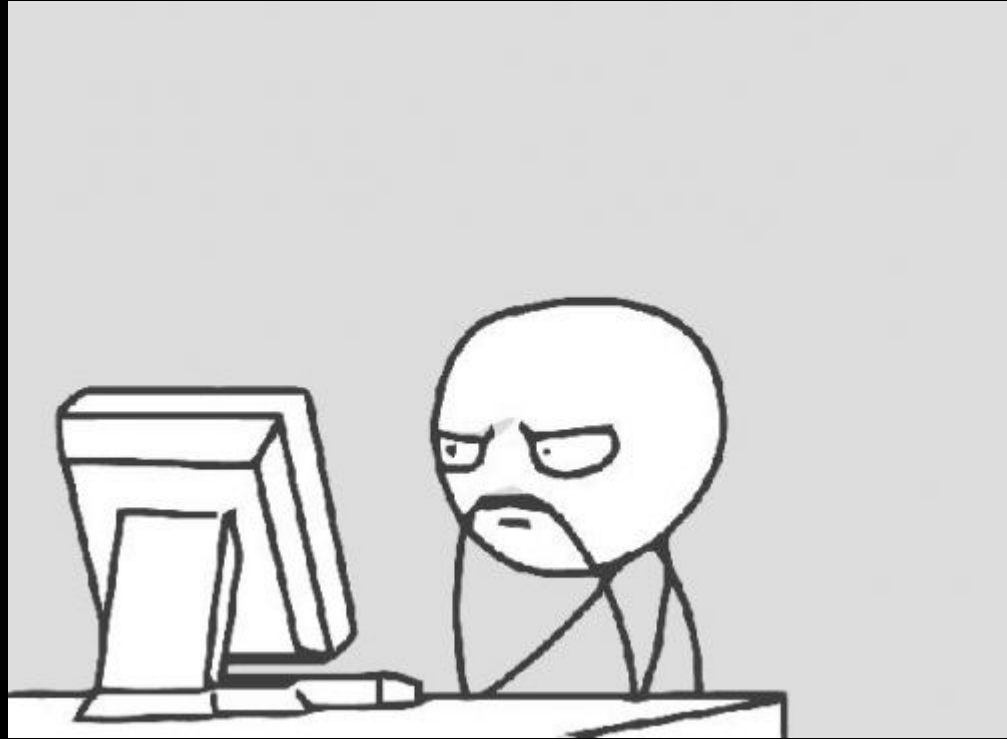


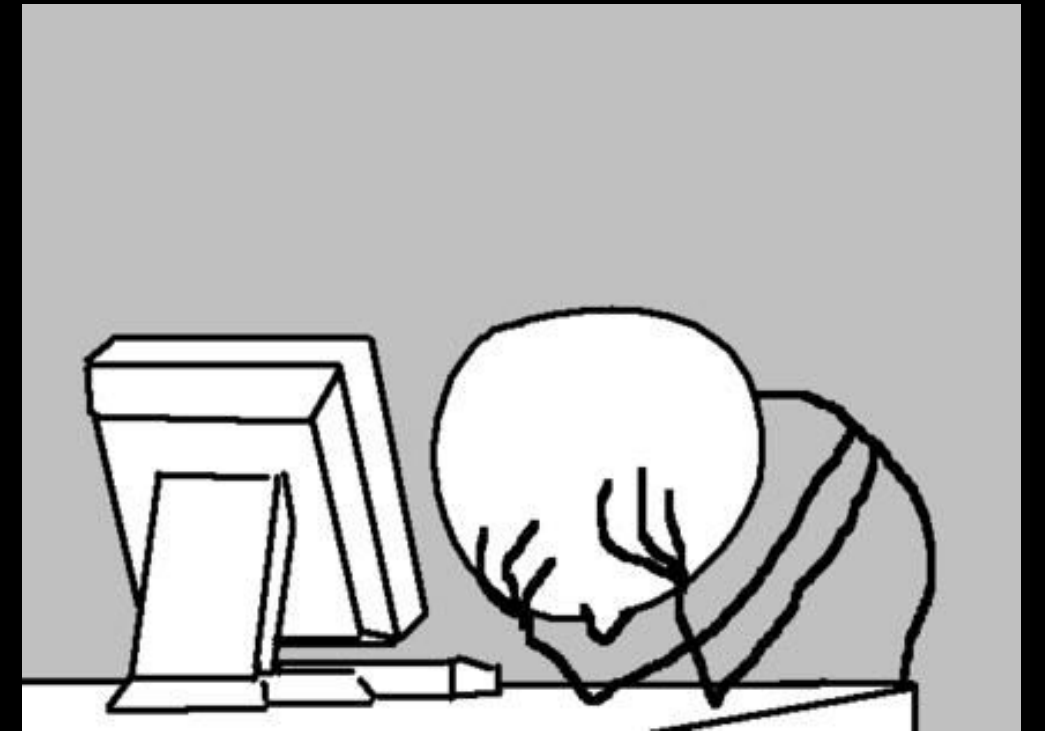
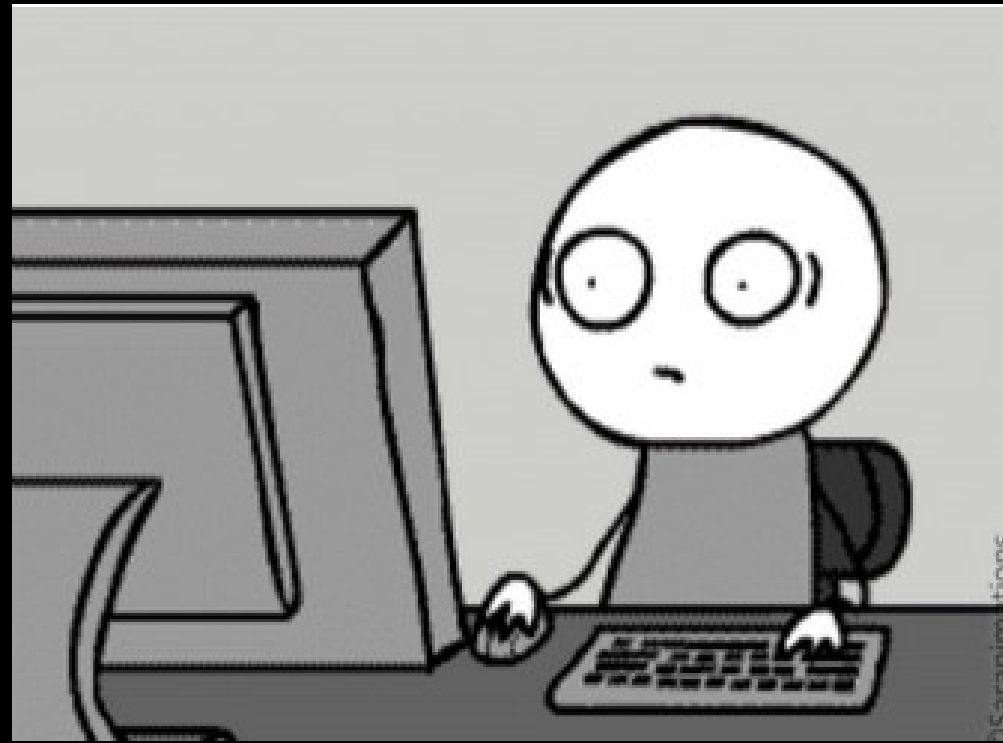
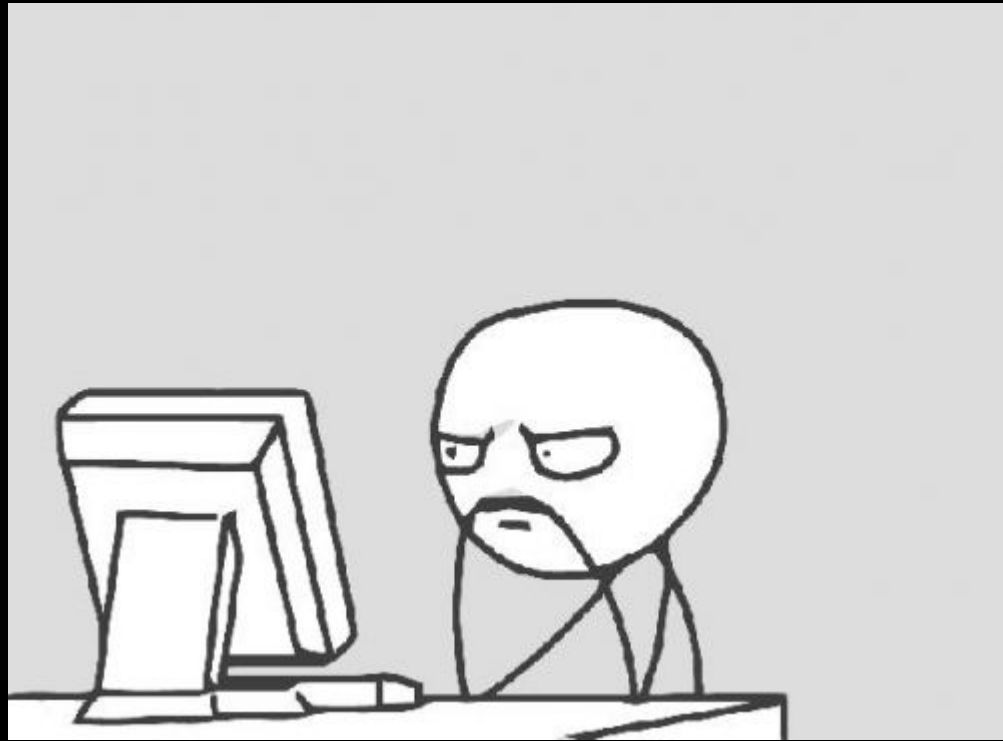
Omar Sanseviero (osanseviero)

# About me

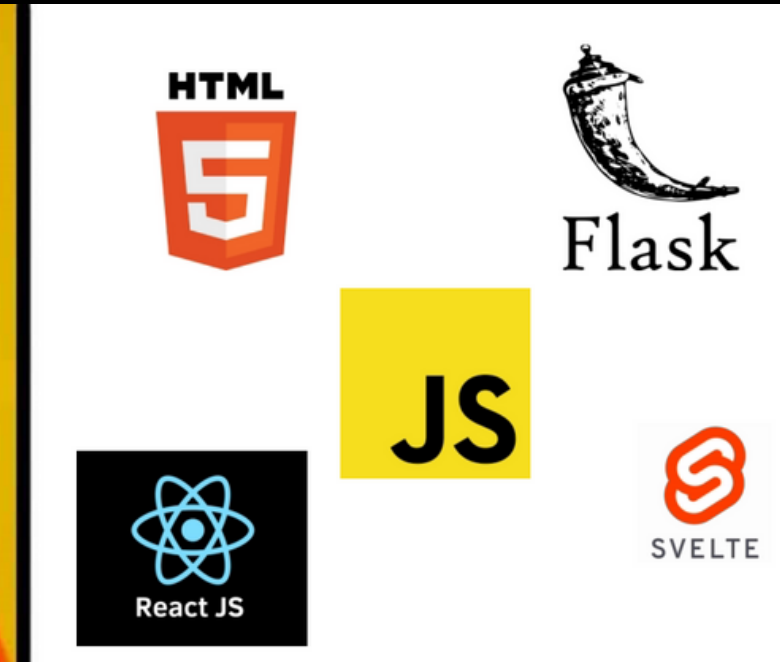
















# Interactive demo: Helsinki-NLP English to Spanish Translation

TEXT

Hello! My name is Omar

Clear

Submit

OUTPUT

0.46s

Hola, mi nombre es Omar.

Screenshot

Flag

## Examples

Hello! My name is Omar

I like this workshop

Interactive c

to Spanish

TEXT

0.46s

Hello! My name is Omar

Flag

Clear

Examples

Hello! My name is Omar





```
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=[["Hello! My name is Omar"], ["I like this workshop"]]
)

iface.launch()
```



```
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=[["Hello! My name is Omar"], ["I like this workshop"]]
)

iface.launch()
```



```
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=[["Hello! My name is Omar"], ["I like this workshop"]]
)

iface.launch()
```



```
import gradio as gr

from transformers import pipeline

pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")

def predict(text):
    return pipe(text)[0]["translation_text"]

title = "Interactive demo: Helsinki-NLP English to Spanish Translation"

iface = gr.Interface(
    fn=predict,
    inputs=[gr.inputs.Textbox(label="text", lines=3)],
    outputs='text',
    title=title,
    examples=[["Hello! My name is Omar"], ["I like this workshop"]]
)

iface.launch()
```

# Interactive demo: Helsinki-NLP English to Spanish Translation

TEXT

Hello! My name is Omar

Clear

Submit

OUTPUT

0.46s

Hola, mi nombre es Omar.

Screenshot

Flag

## Examples

Hello! My name is Omar

I like this workshop



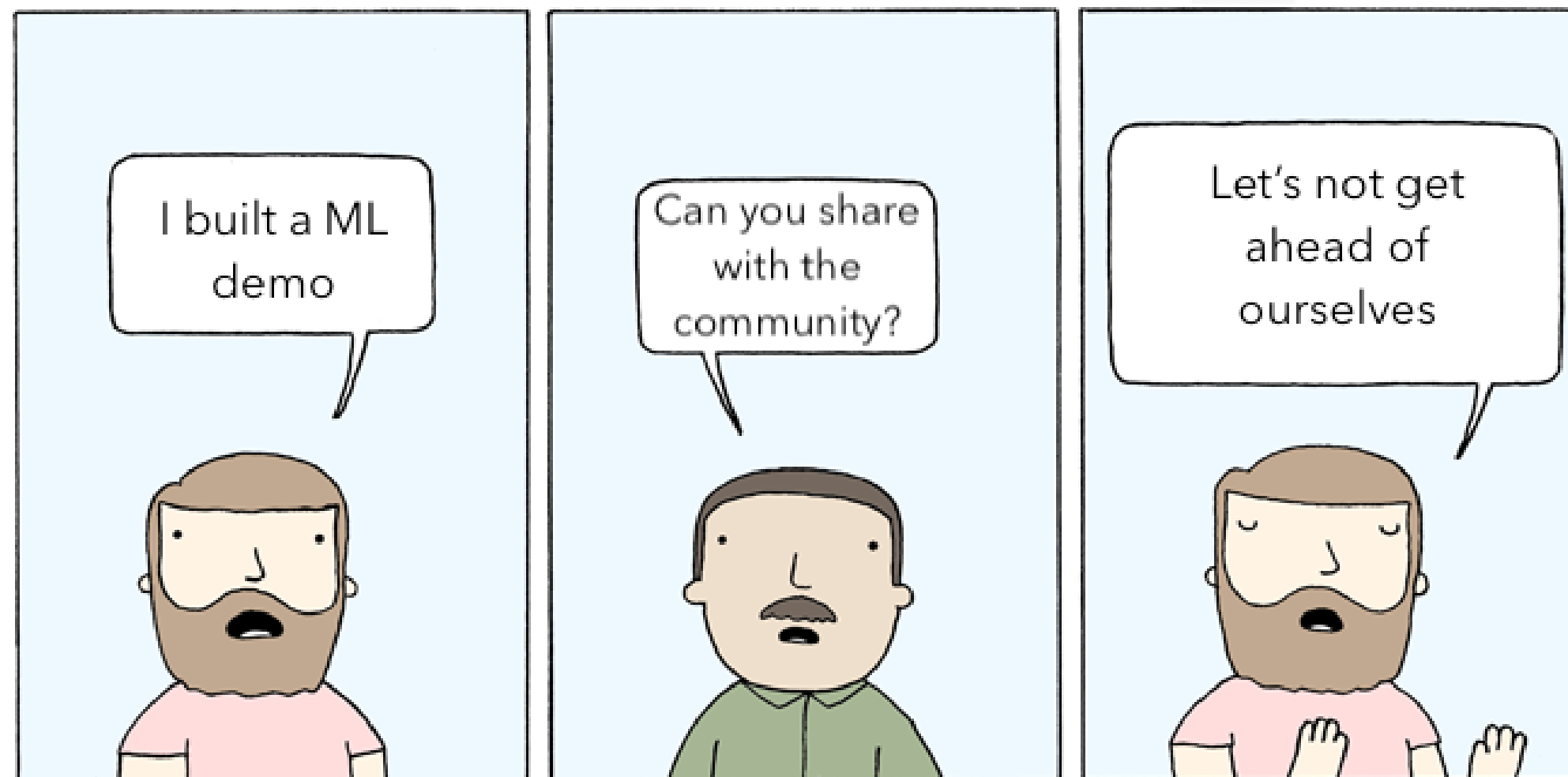


# Gradio vs Streamlit

- Gradio
  - Build extremely easy to use interfaces
  - Only specify input and outputs
- Streamlit
  - For flexible to a fully-fledged web apps
  - Community components

Both are brilliant!





Use [hf.co/spaces](https://hf.co/spaces)

**hf.co/spaces/**  
hf.co/spaces/to-the-moon





## Create a new Space

A repository contains all space files, including the revision history.

Owner

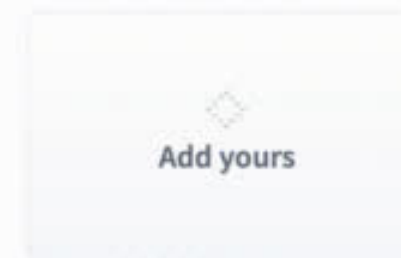
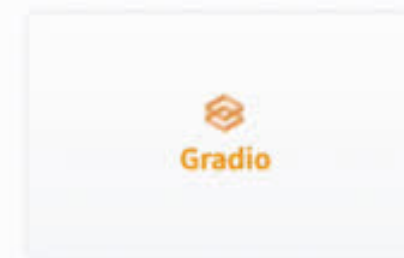
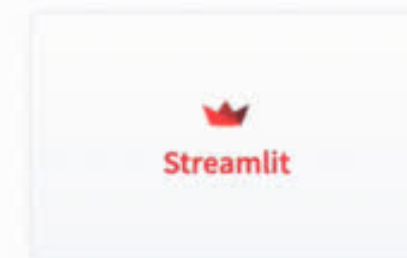
julien-c

Space name

M

### Select the Space SDK

You can chose between Streamlit and Gradio for your Space. Contact us if you need a custom solution.



#### Public

Anyone on the internet can see this space. Only you (personal space) or members of your organization (organization space) can commit.

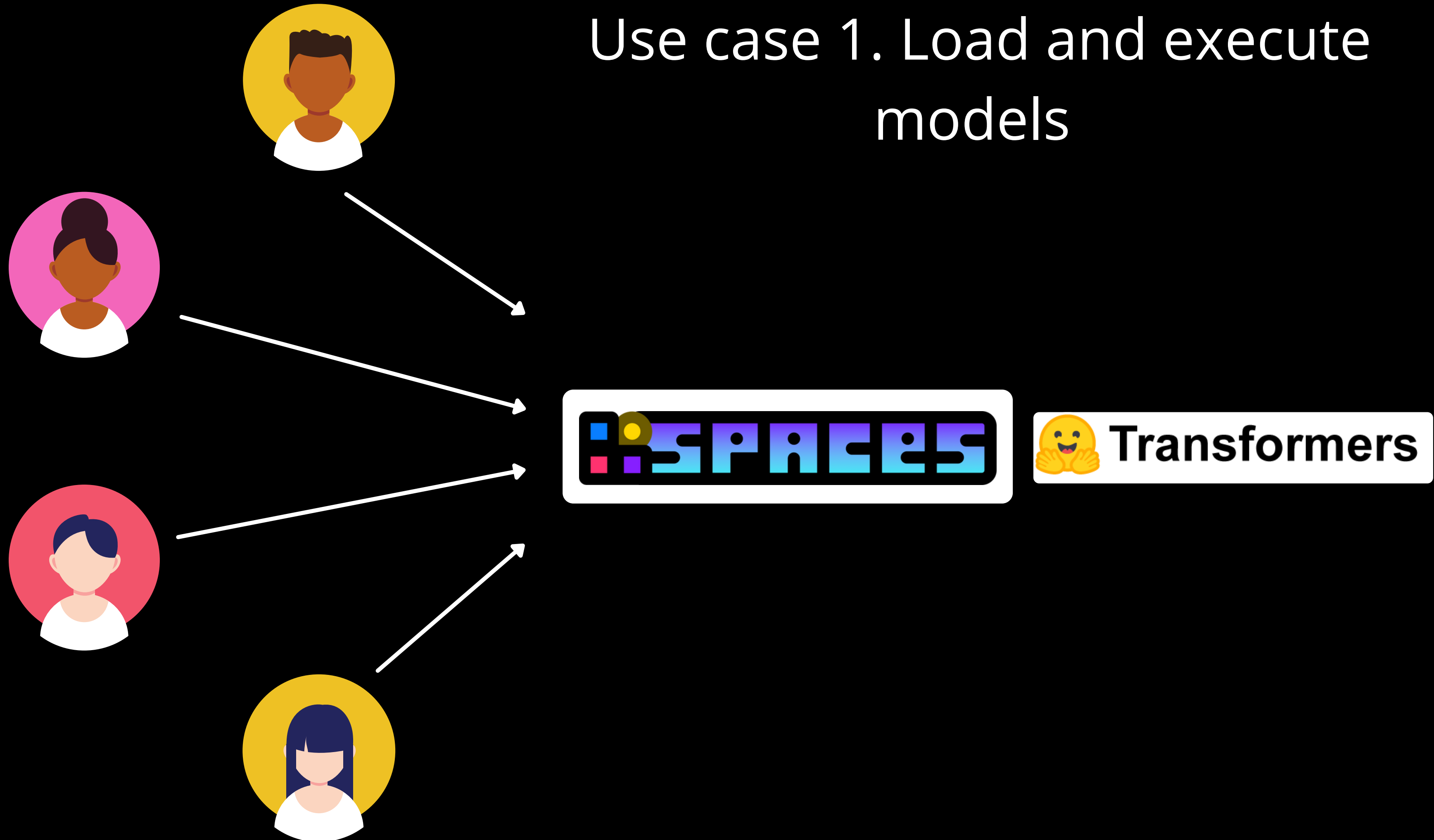


#### Private

Only you (personal space) or members of your organization (organization space) can see and commit to this space.

Create space

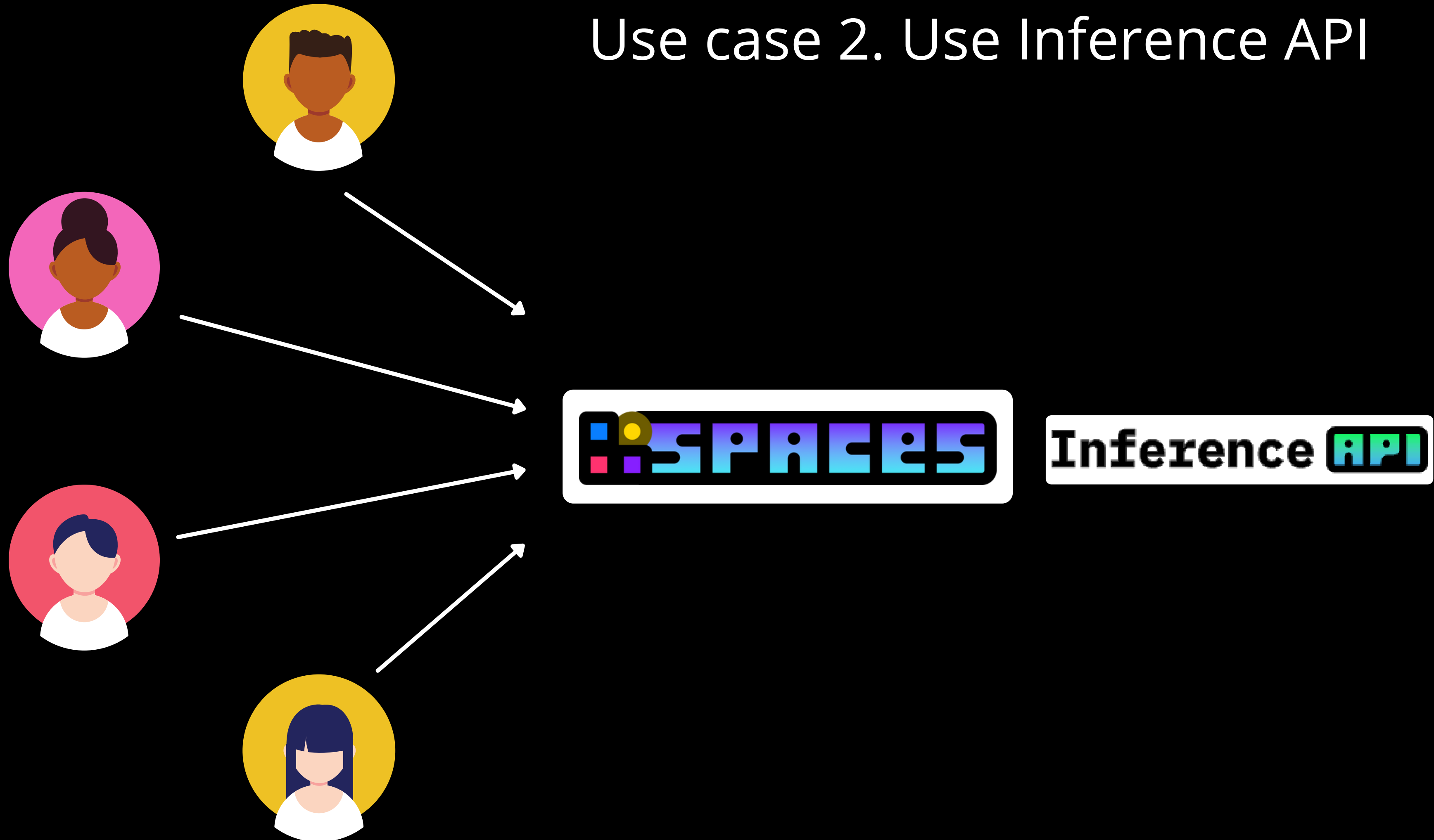
# Use case 1. Load and execute models



# Use case 1. Load and execute models



## Use case 2. Use Inference API



# Use case 2. Use Inference API



```
import gradio as gr

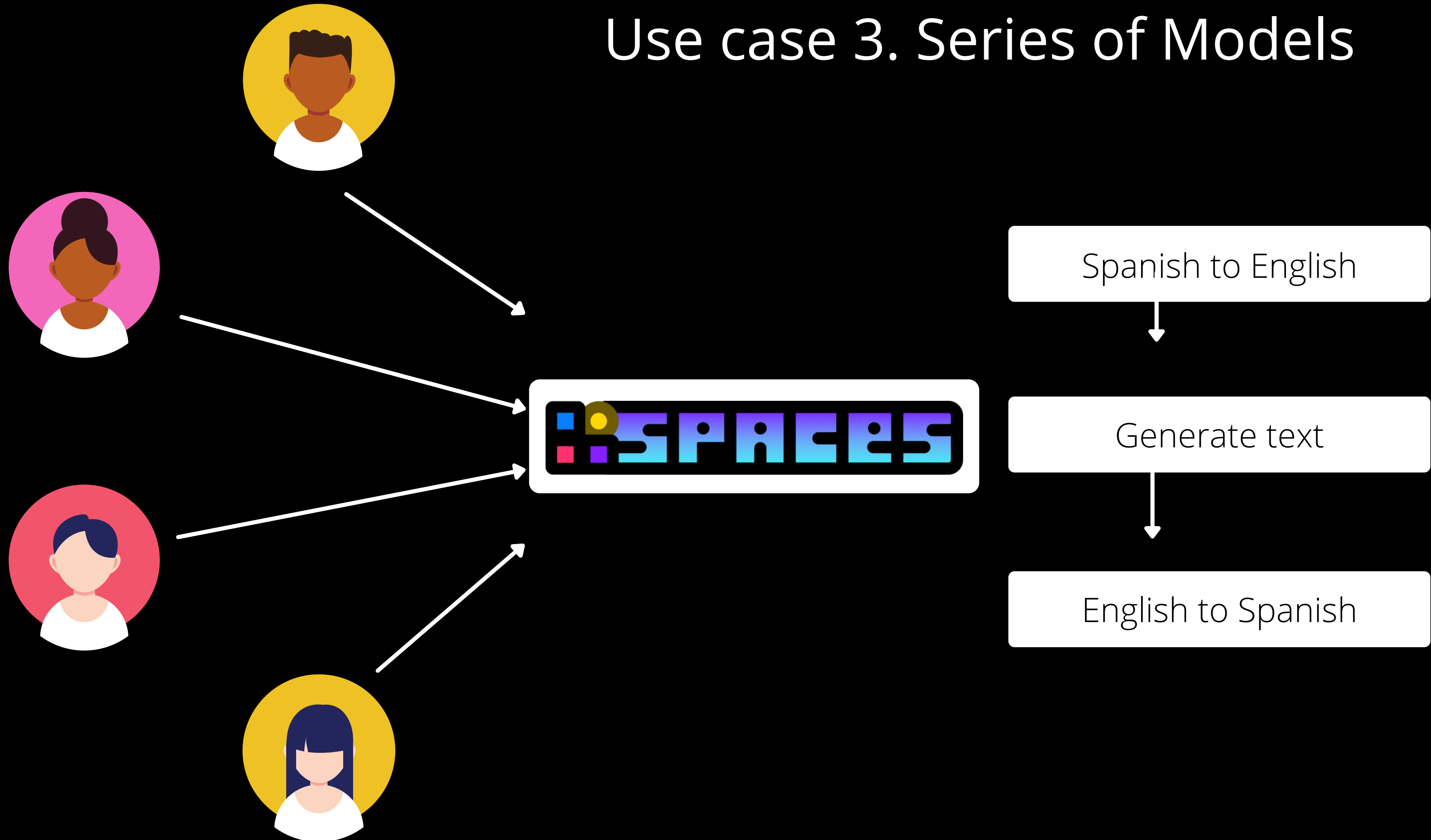
description = "BigGAN text-to-image demo."
title = "BigGAN ImageNet"

interface = gr.Interface.load(
    "huggingface/osanseviero/BigGAN-deep-128",
    description=description,
    title = title,
    examples=[["american robin"], ["chest"], ["soap bubble"]]
)

interface.launch()
```



# Use case 3. Series of Models



# Use case 3. Series of Models

```
import gradio as gr
from gradio.mix import Series

description = "Generate your own D&D story!"
title = "Spanish Story Generator using Opus MT and GPT-2"

translator_es = gr.Interface.load("huggingface/Helsinki-NLP/opus-mt-es-en")
story_gen = gr.Interface.load("huggingface/pranavpsv/gpt2-genre-story-generator")
translator_en = gr.Interface.load("huggingface/Helsinki-NLP/opus-mt-en-es")

examples = [["La aventura comienza en"]]

interface = Series(translator_es, story_gen, translator_en, description = description,
                  title = title,
                  examples=examples,
                  inputs = gr.inputs.Textbox(lines = 10)
                  )

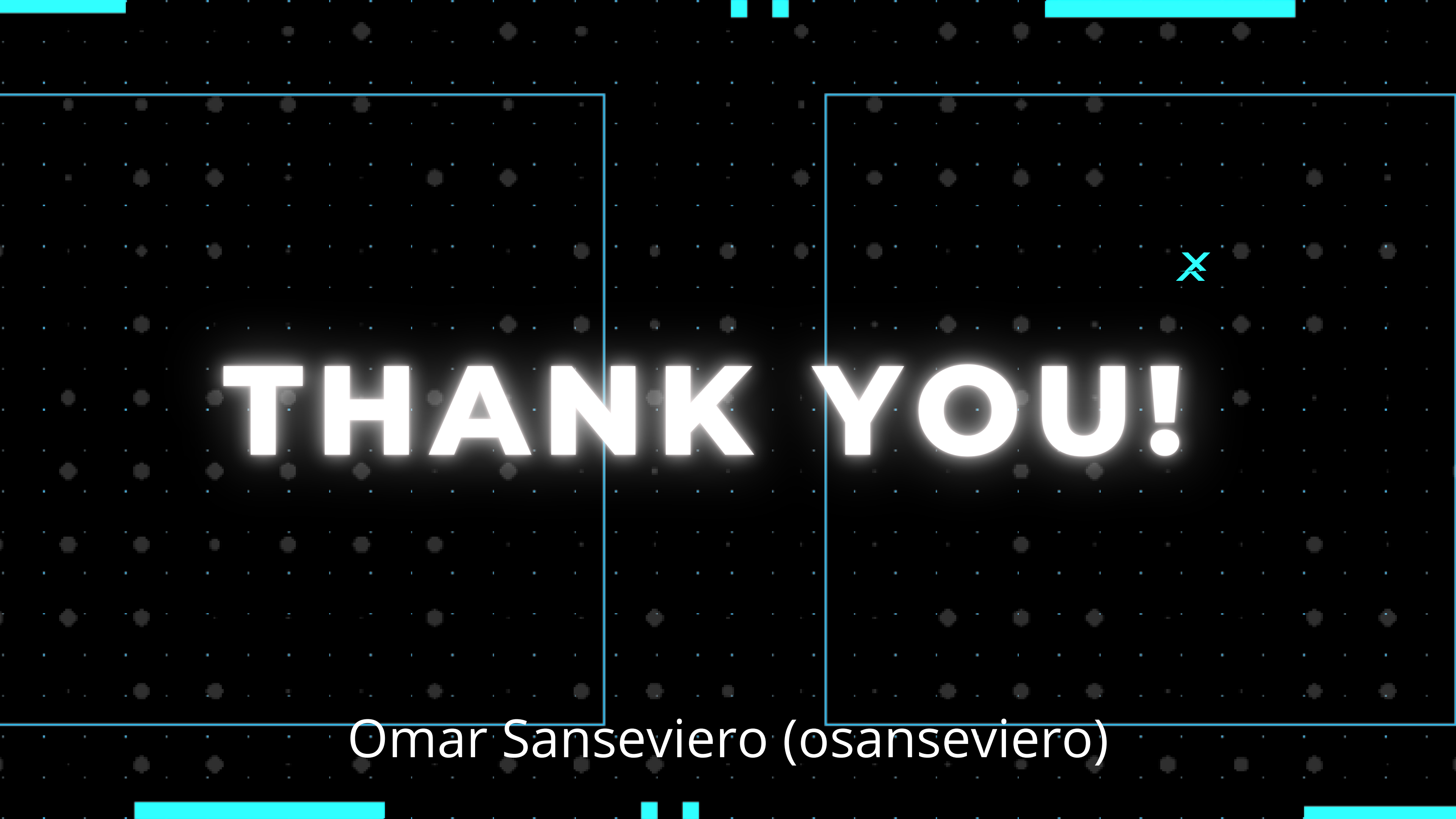
interface.launch()
```



# Selected Spaces

- DALLE Mini
- BERTIN
- DOCTR
- Spacy\_Visualization
- HUBERT





**THANK YOU!**

Omar Sanseviero (osanseviero)

# Bonus Content



Streamlit