

From Idea to Implementation:

A Hands-on Approach
to Developing an AI-Powered
Meeting Minutes Tool



Florian Schepers



07. June 2024



UFO - Unterwelt

1

Motivation

New Challenges:

- Working in IT has a steep learning curve

Mentorship & Inspiration:

- Opportunity to learn from experienced colleagues

Challenge to Innovate:

- Encouraged to build something from scratch, to foster technical and personal growth

What's my personal project?



AI Disruption:

- AI driven tools like Chat GPT change the way we work

Curiosity & Concerns:

- Potential of AI – thrilling and threatening at the same time

Embracing AI:

- Learn and leverage AI to boost productivity



Remote Work Challenges:

- Increased distractions and multitasking demands in digital environments

The Illusion of Multitasking:

- Multitasking makes us less efficient [1]

Meeting Inefficiencies:

- Taking Meeting Minutes – a tedious task distracting you from the actual meeting



[1] http://news.bbc.co.uk/2/hi/uk_news/4471607.stm

2

Idea & Concept

Automated Meeting Minutes:

- Uses AI tools to transcribe and summarize meetings
- Freeing the user from the distraction of note-taking

Open Source and Secure:

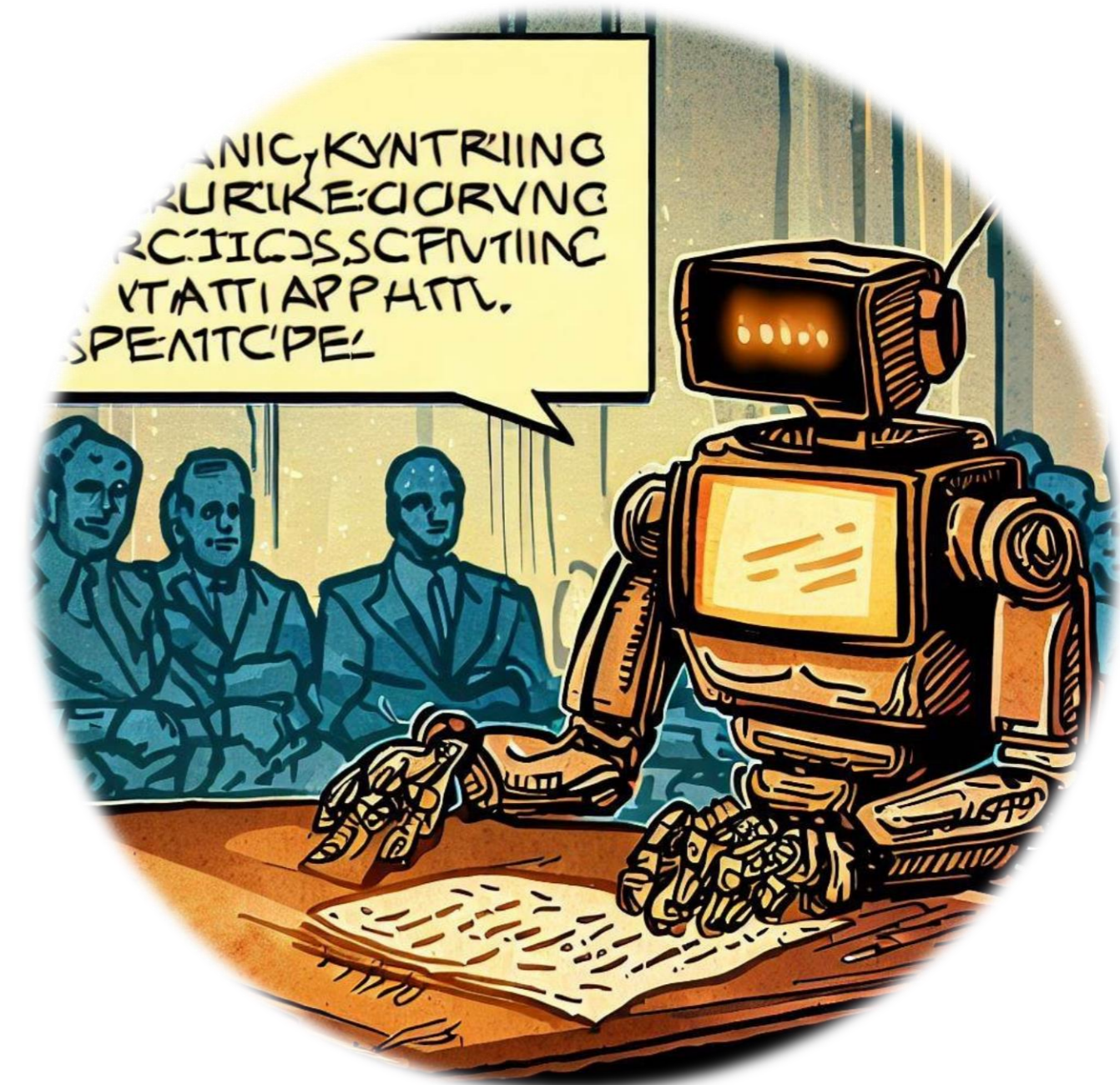
- Ensures transparency and allows users full control over their data

Data Sovereignty:

- Supports local processing or user-selected trusted services, enhancing data privacy and security

Works out of the box:

- No training or fine-tuning needed



<https://github.com/FlorianSchepers/Meminto>



Audio Recording



Meeting Minutes



3

AI-Tools



Speaker Diarization



Audio Recording



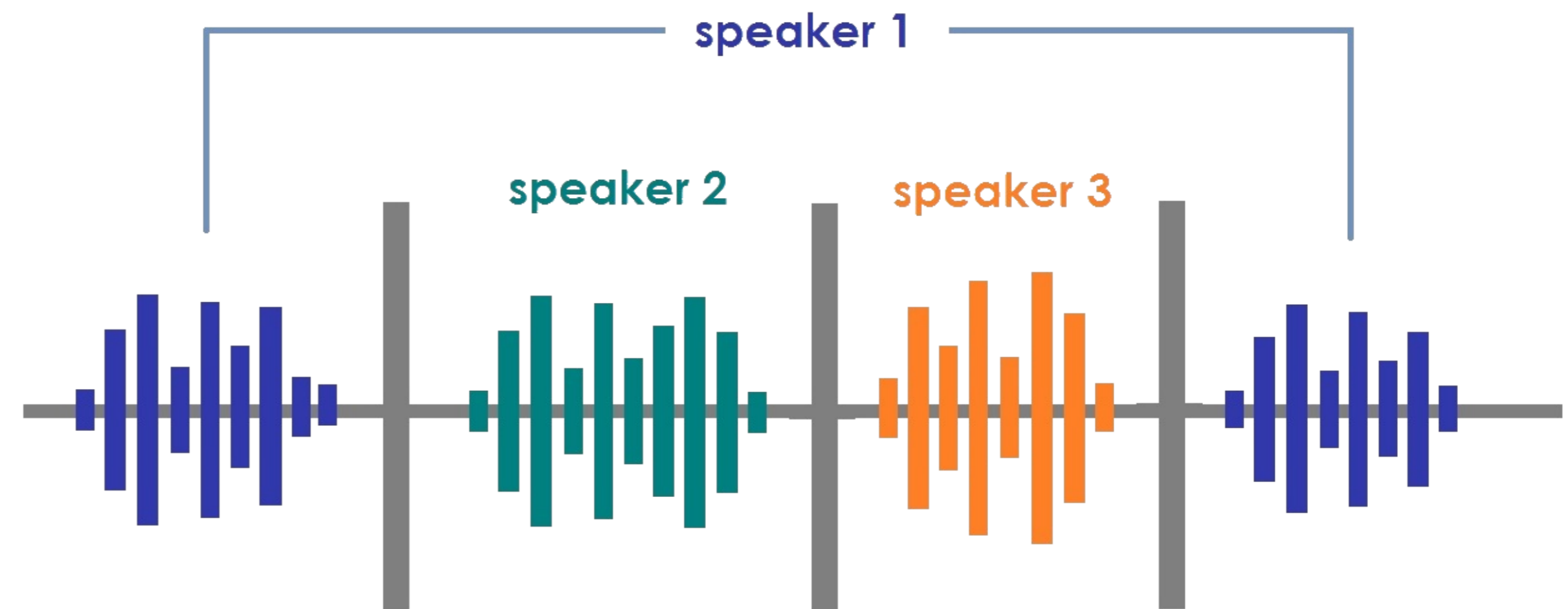
Meeting Minutes





Who speaks when?

- Essential for accurate meeting minutes
- Utilizing `pyannote.audio` for cutting-edge speaker recognition

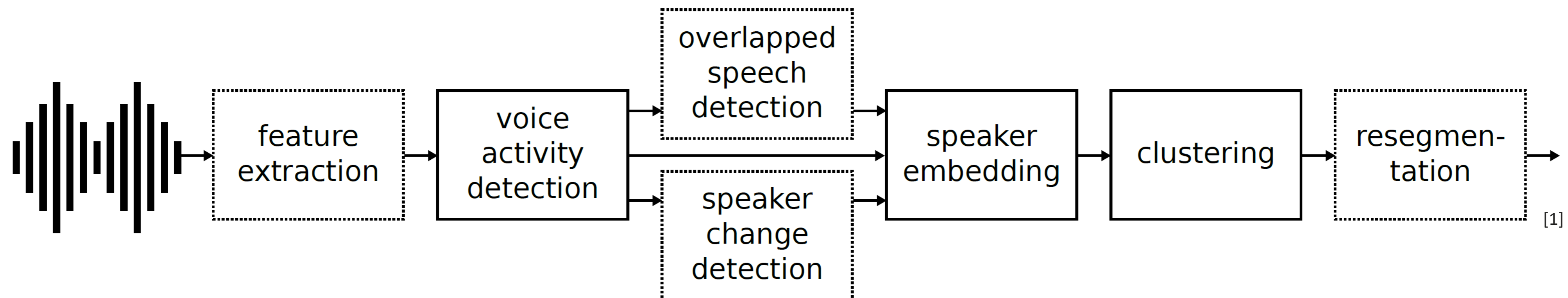


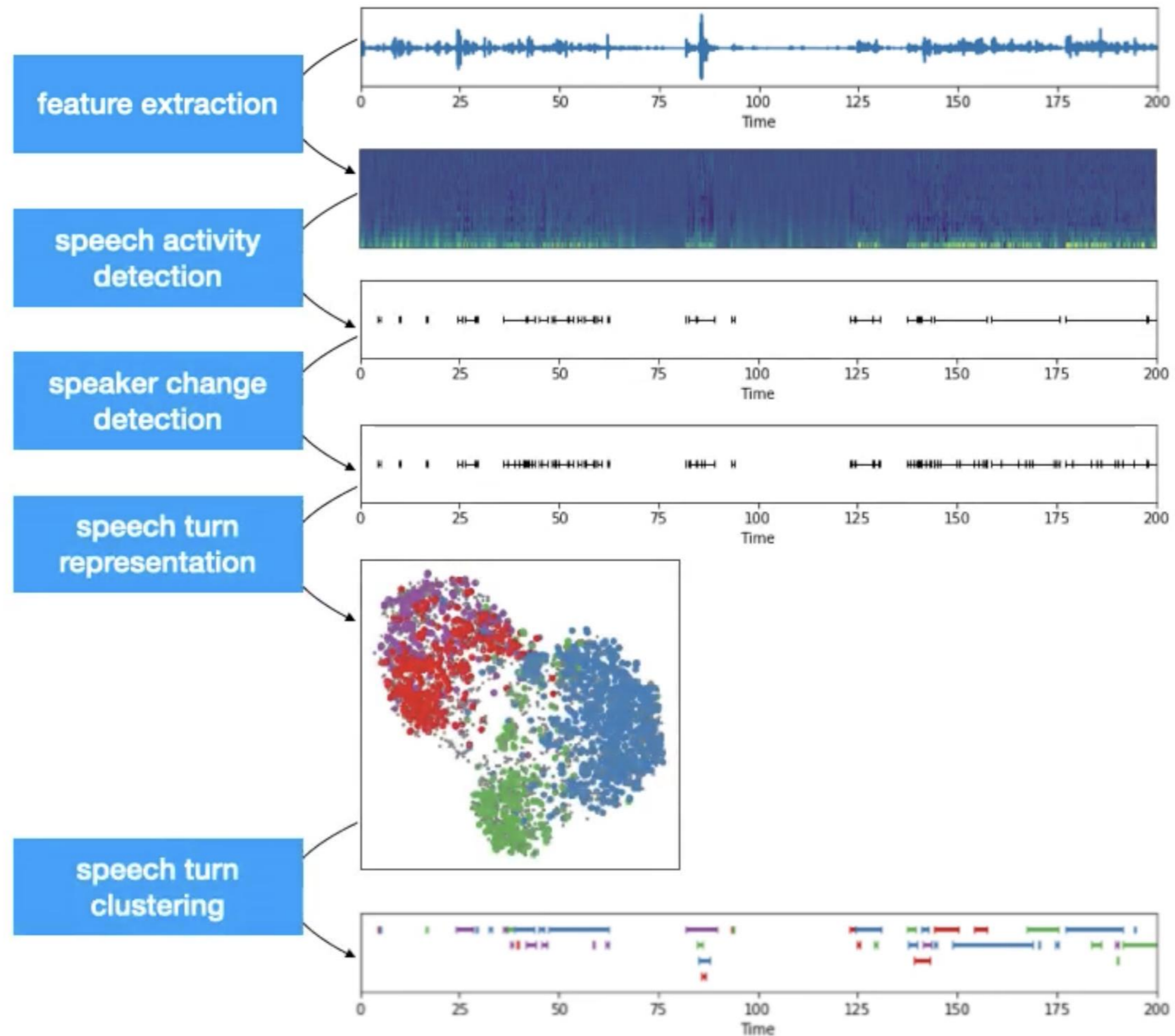


- An open-source toolkit written in Python
- Based on PyTorch
- Provides a set of trainable end-to-end neural building blocks to build speaker diarization pipelines
- Pretrained models available
- Runs locally



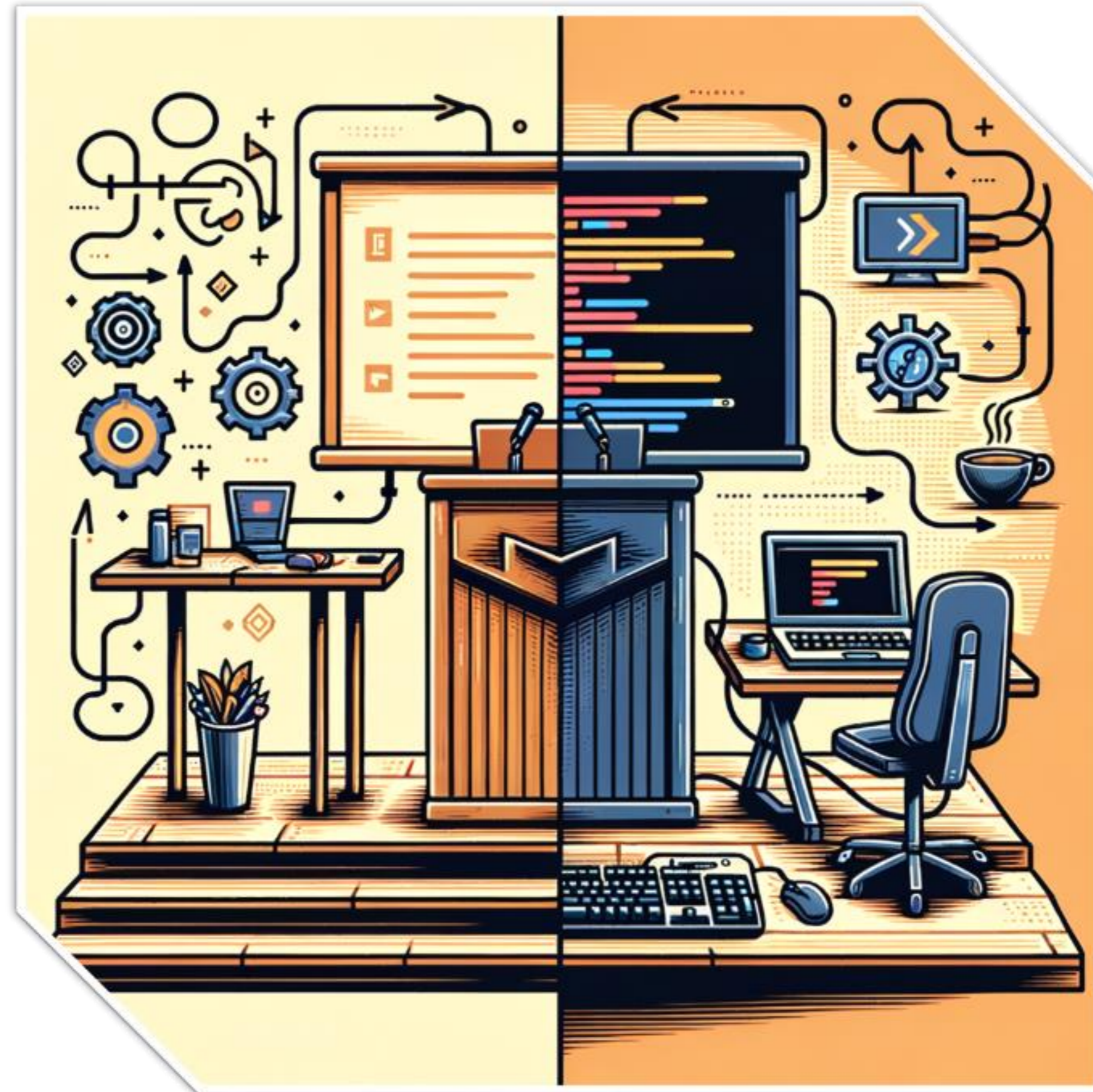
<https://github.com/pyannote/pyannote-audio>







Diarization Implementation





Audio Transcription



Audio Recording

Diarization

Transcription

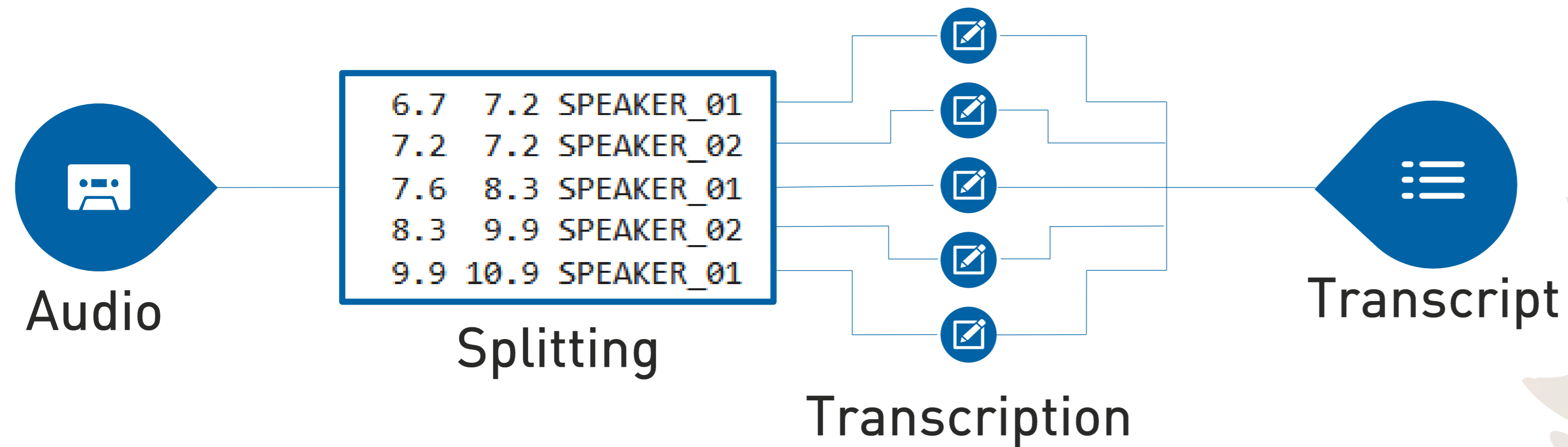
Generation

Meeting Minutes








Audio Transcription



Whisper (OpenAI) [1]



- Open source (MIT license) 
- Runs locally 
- Multiple models of different sizes available 







Multitask training data (680k hours)



English transcription

-  „Ask not what your country can do for ...“
-  Ask not what your country can do for ...



Any-to-English speech translation

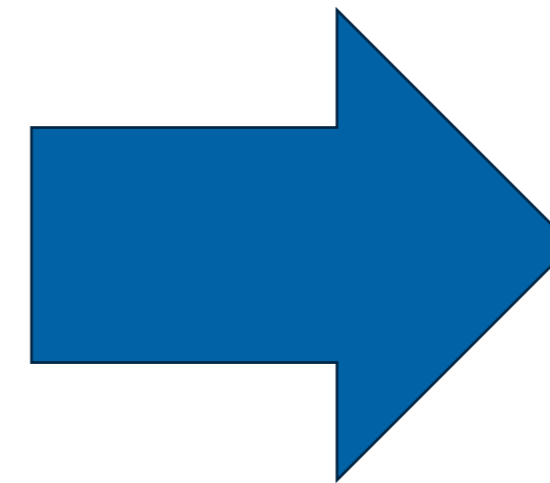
-  „El rápido zorro marrón salta sobre ...“
-  The quick brown fox jumps over ...

Non-English transcription

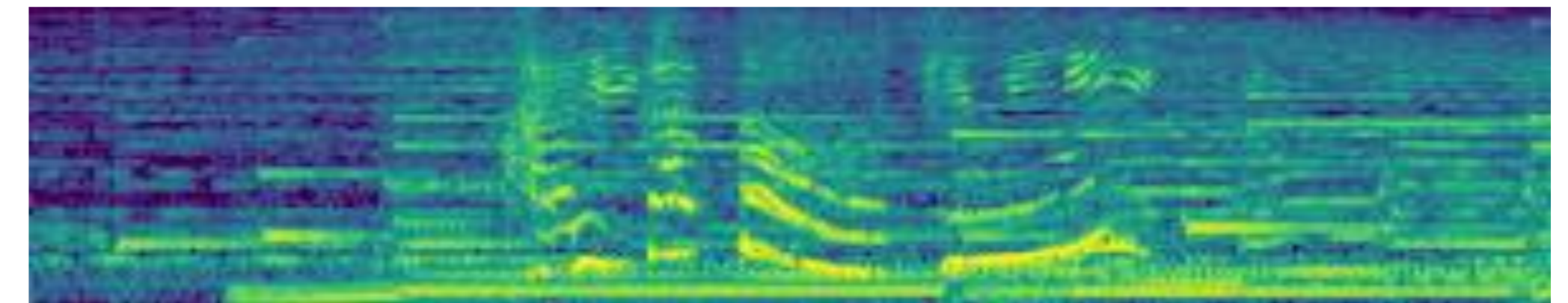
-  „Mein Text“
-  Mein Text

No speech

-  (background music playing)
-  nothing



Frequency

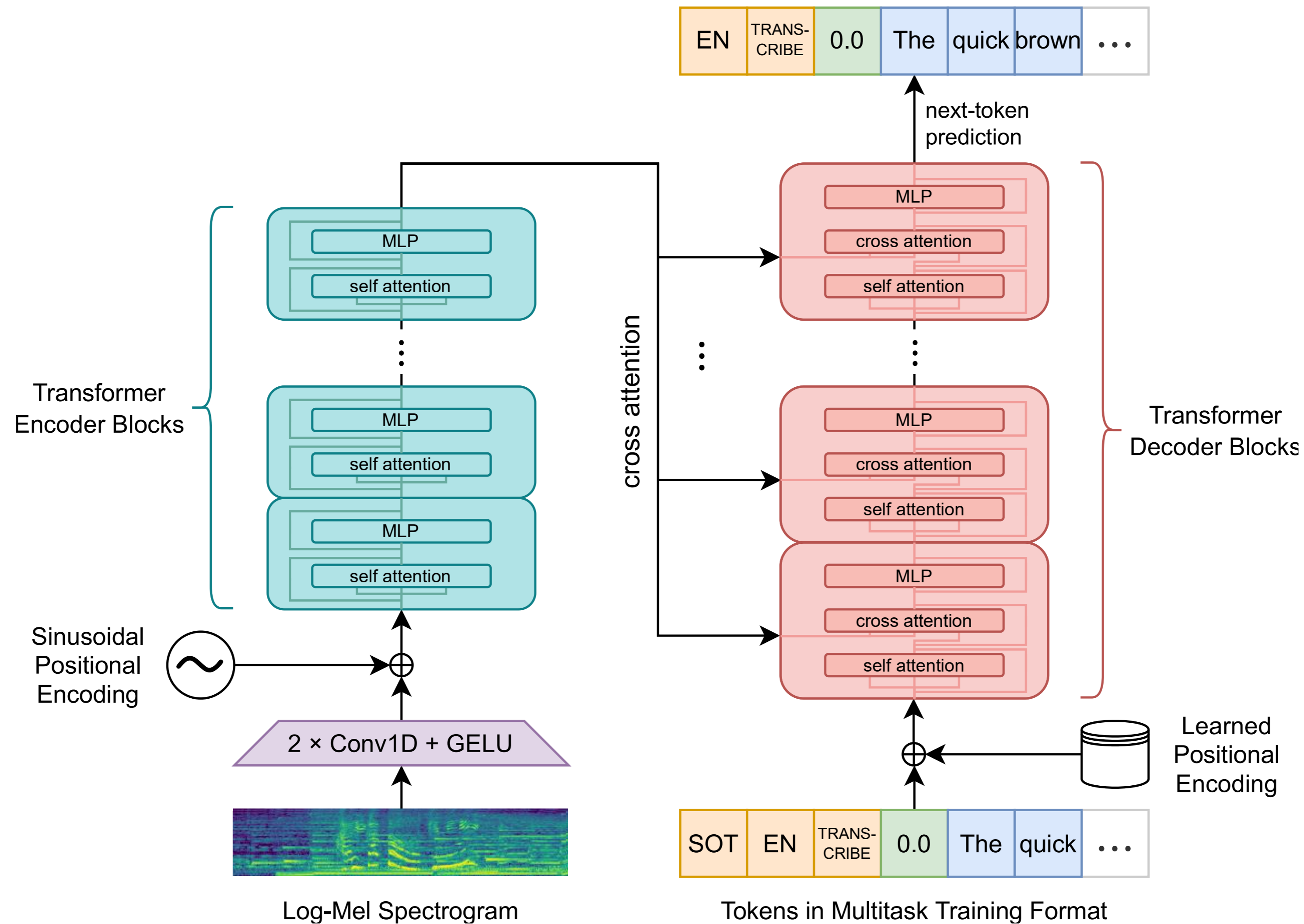


Log-Mel-Spectrogram

Time

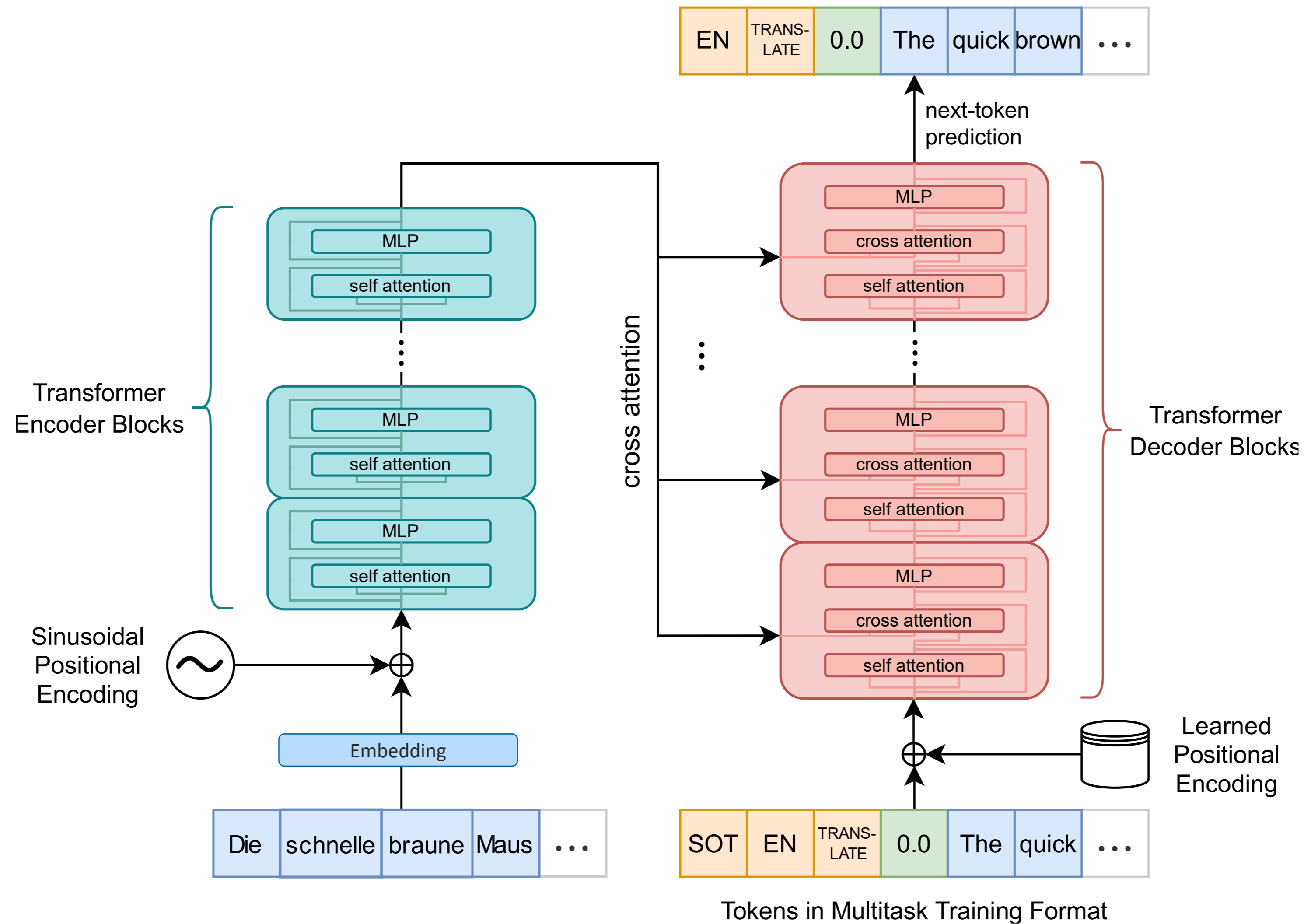


Whisper Transformer Architecture



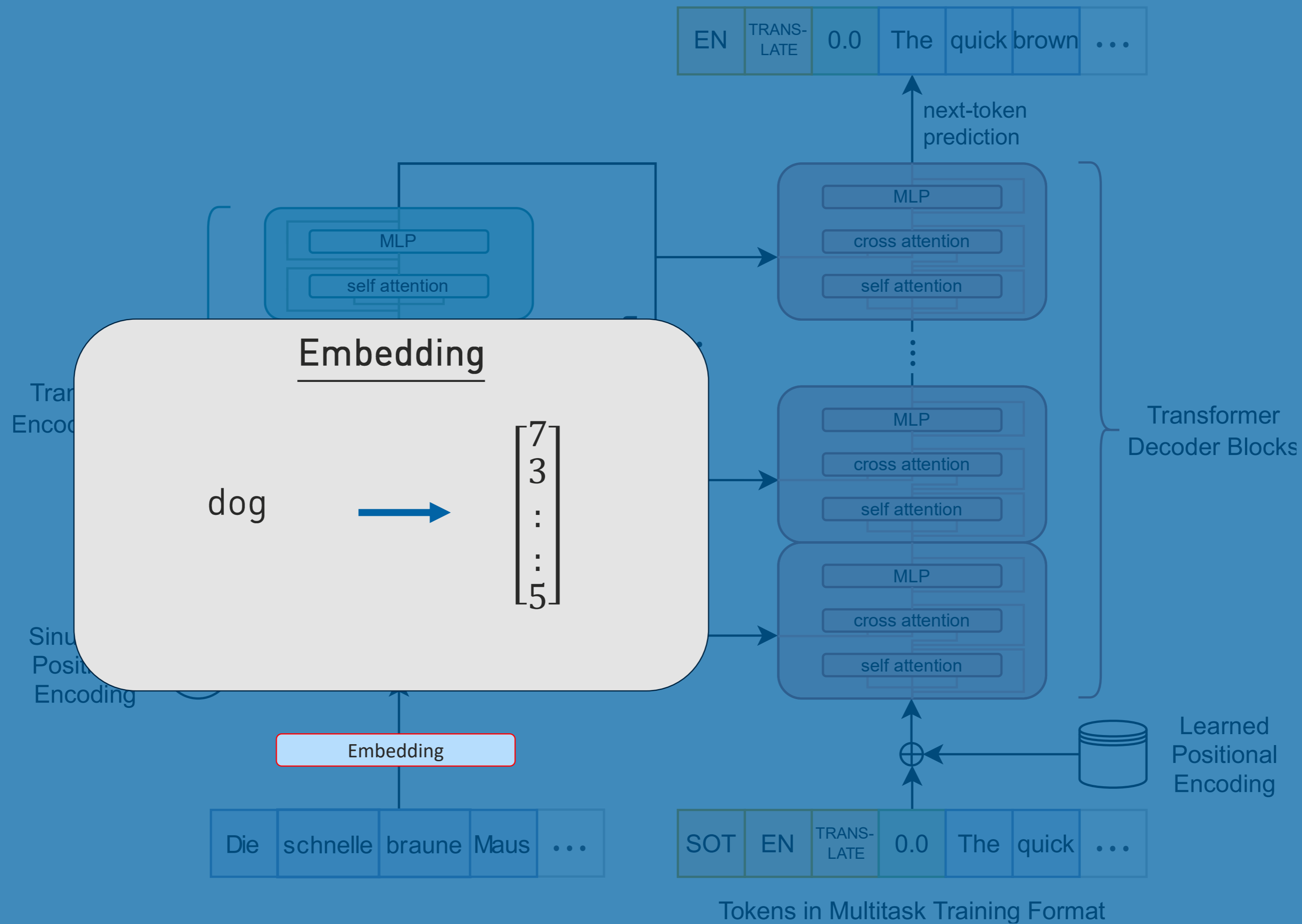


Whisper Transformer Architecture



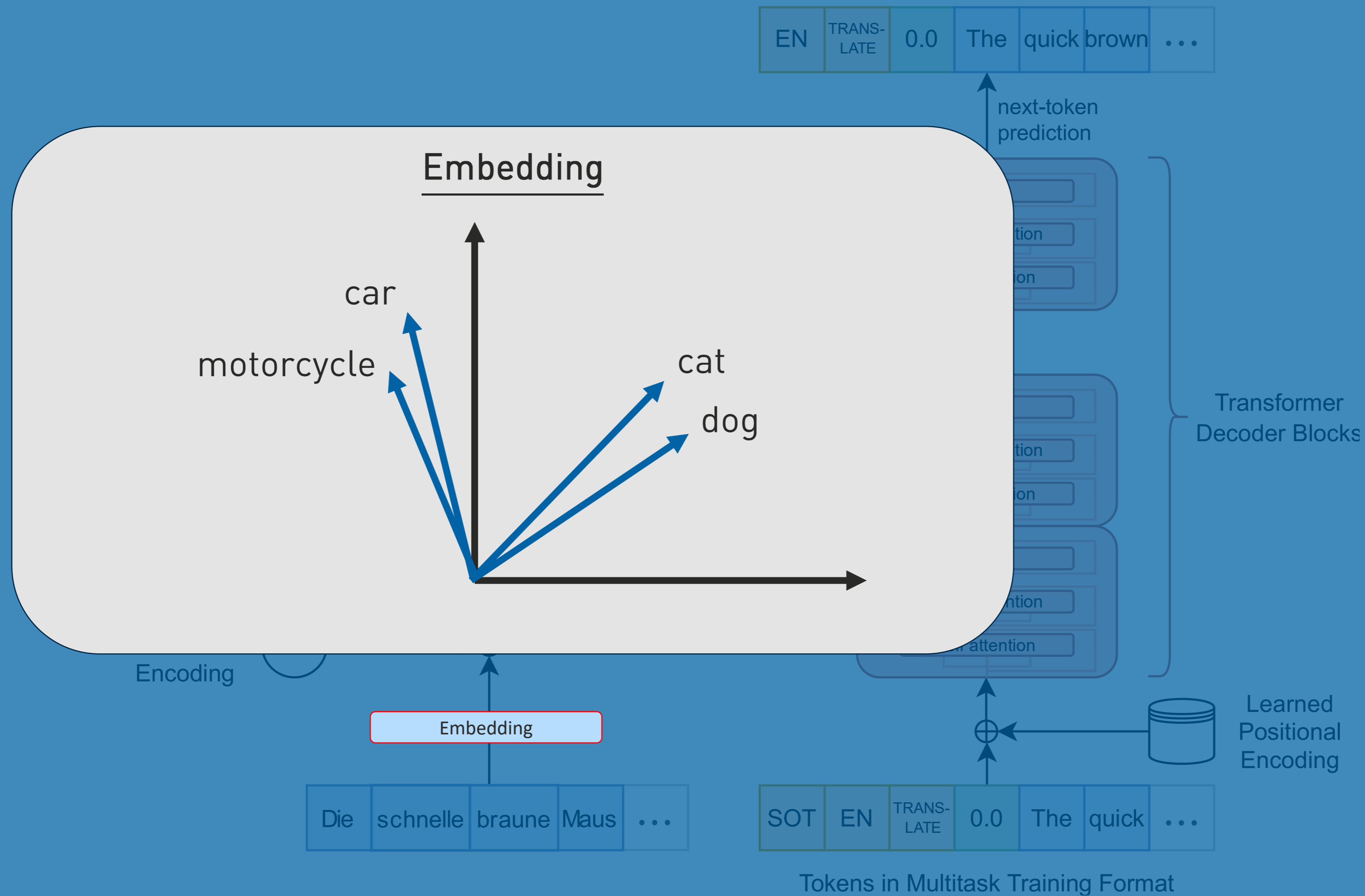


Whisper Transformer Architecture



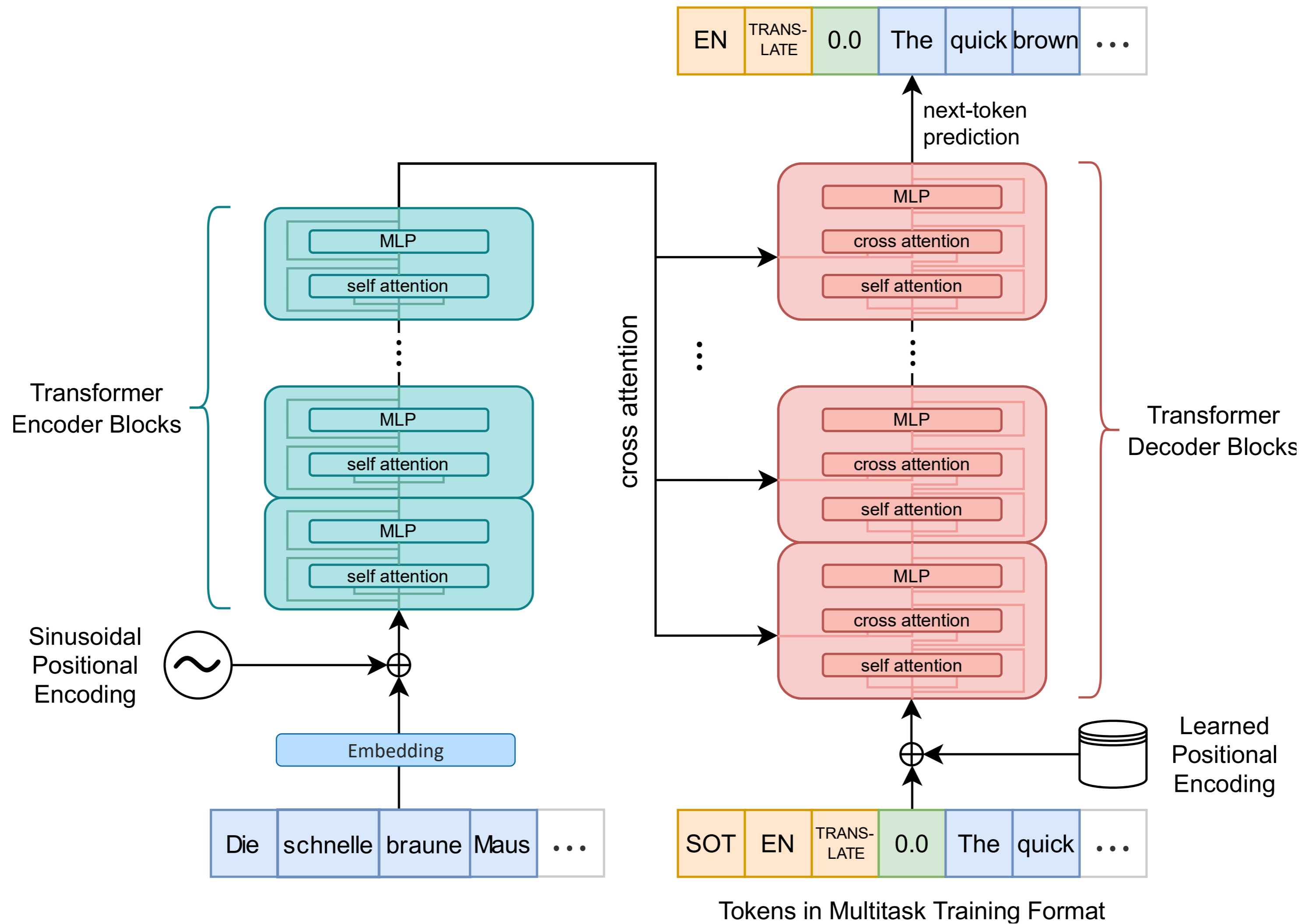


Whisper Transformer Architecture



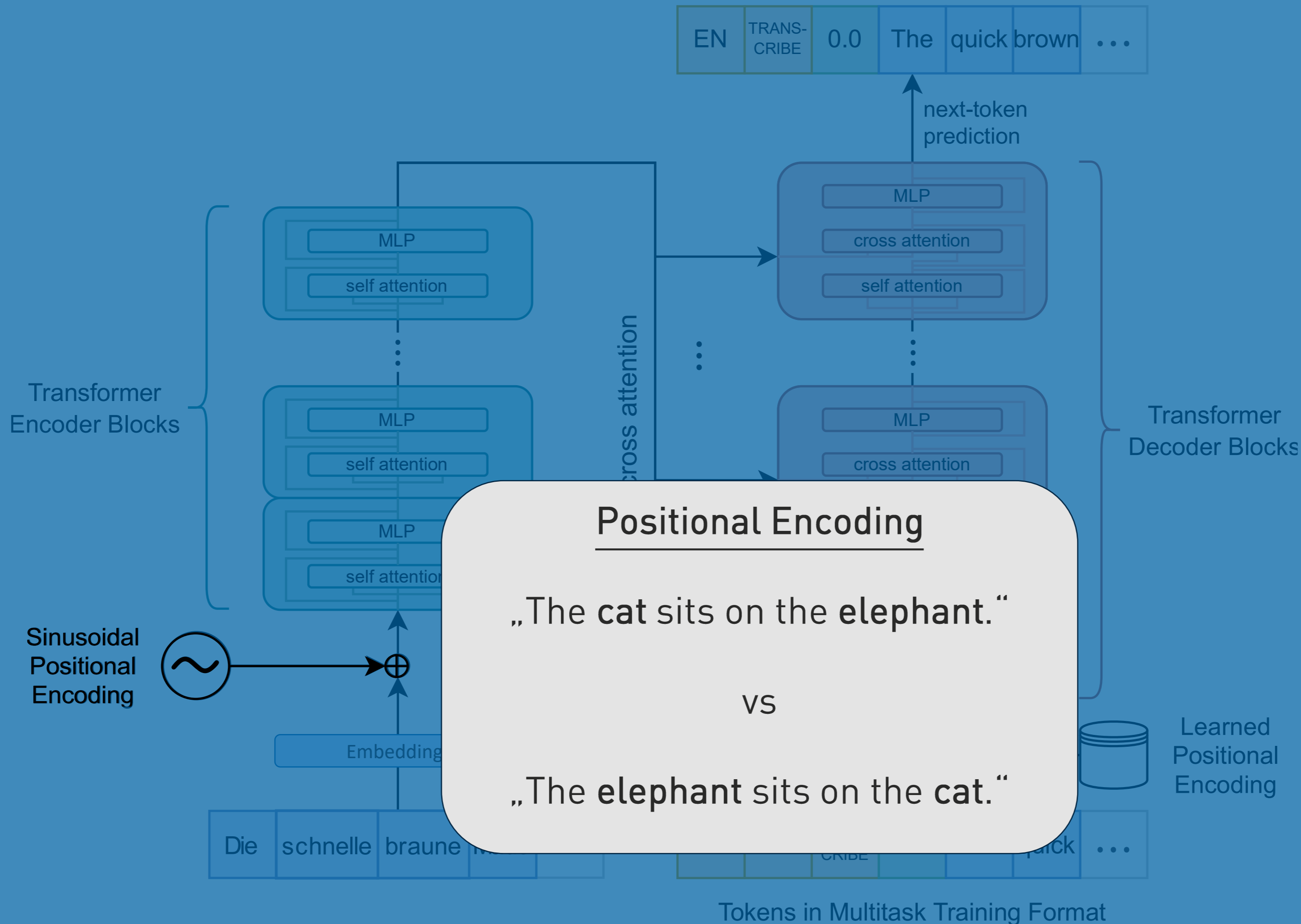


Whisper Transformer Architecture





Whisper Transformer Architecture

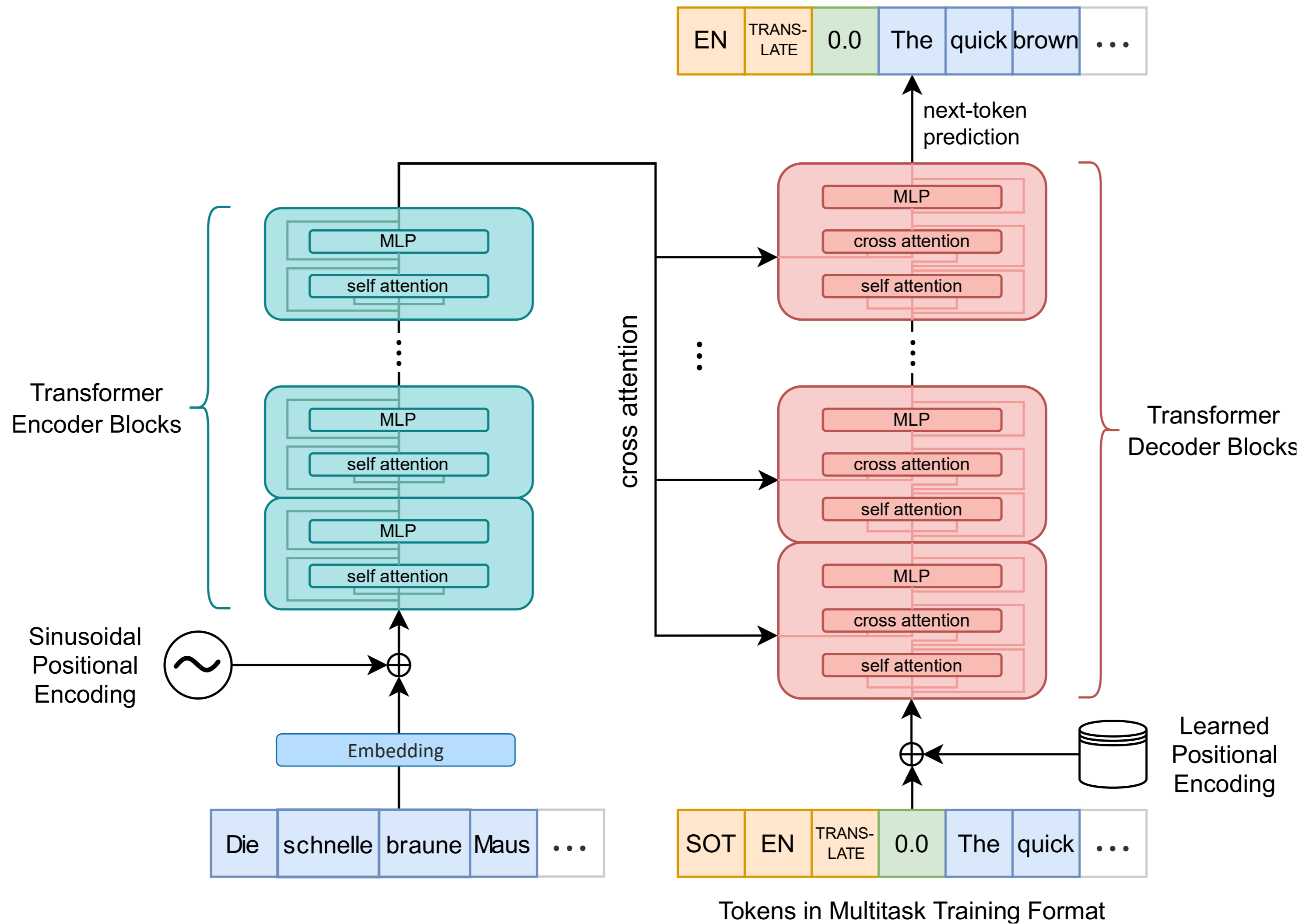


Positional Encoding
 „The cat sits on the elephant.“
 VS
 „The elephant sits on the cat.“

[1] Adapted from Alec Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision", arXiv:2212.04356, [2022]

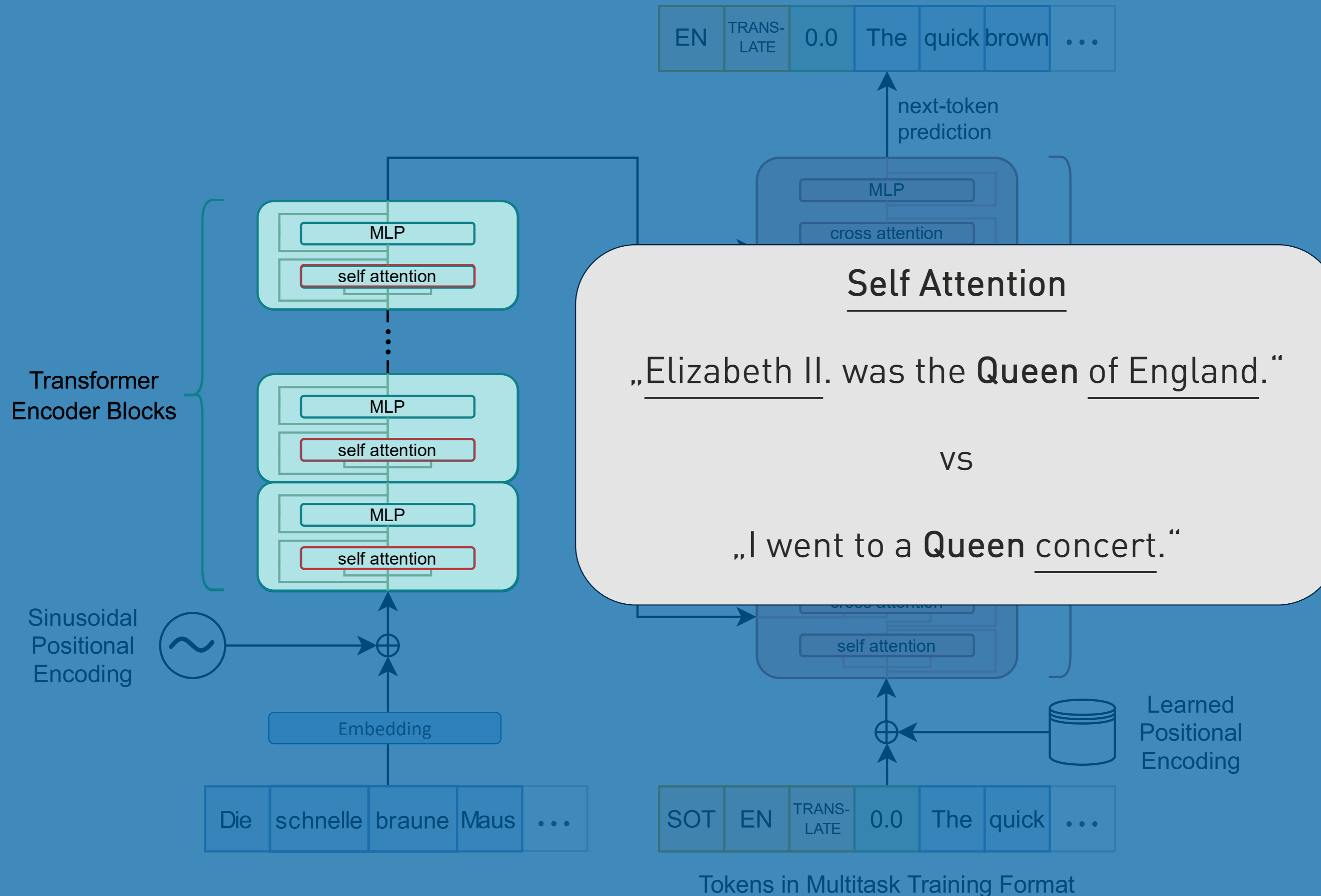


Whisper Transformer Architecture





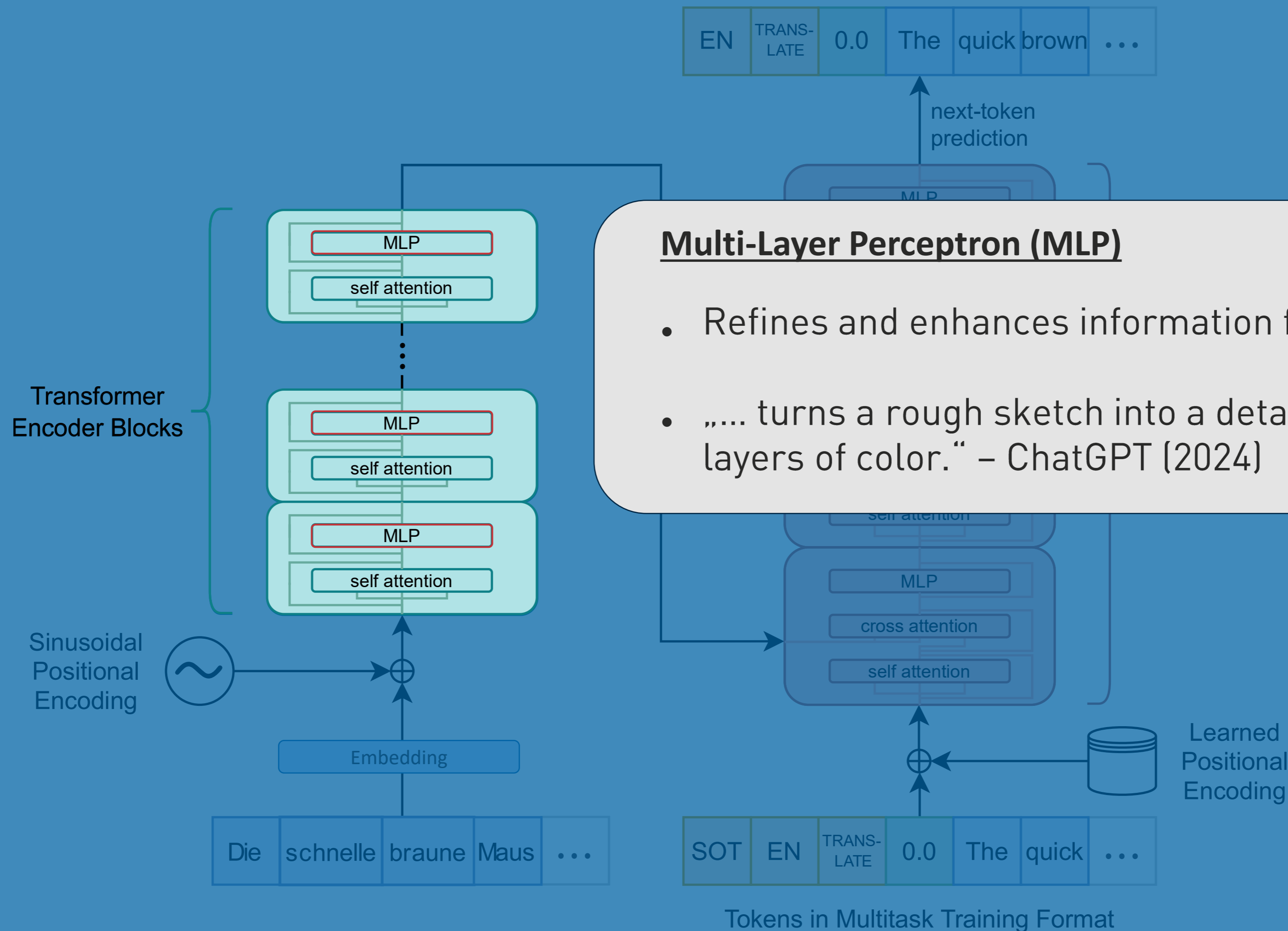
Whisper Transformer Architecture



[1] Adapted from Alec Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision", arXiv:2212.04356, [2022]

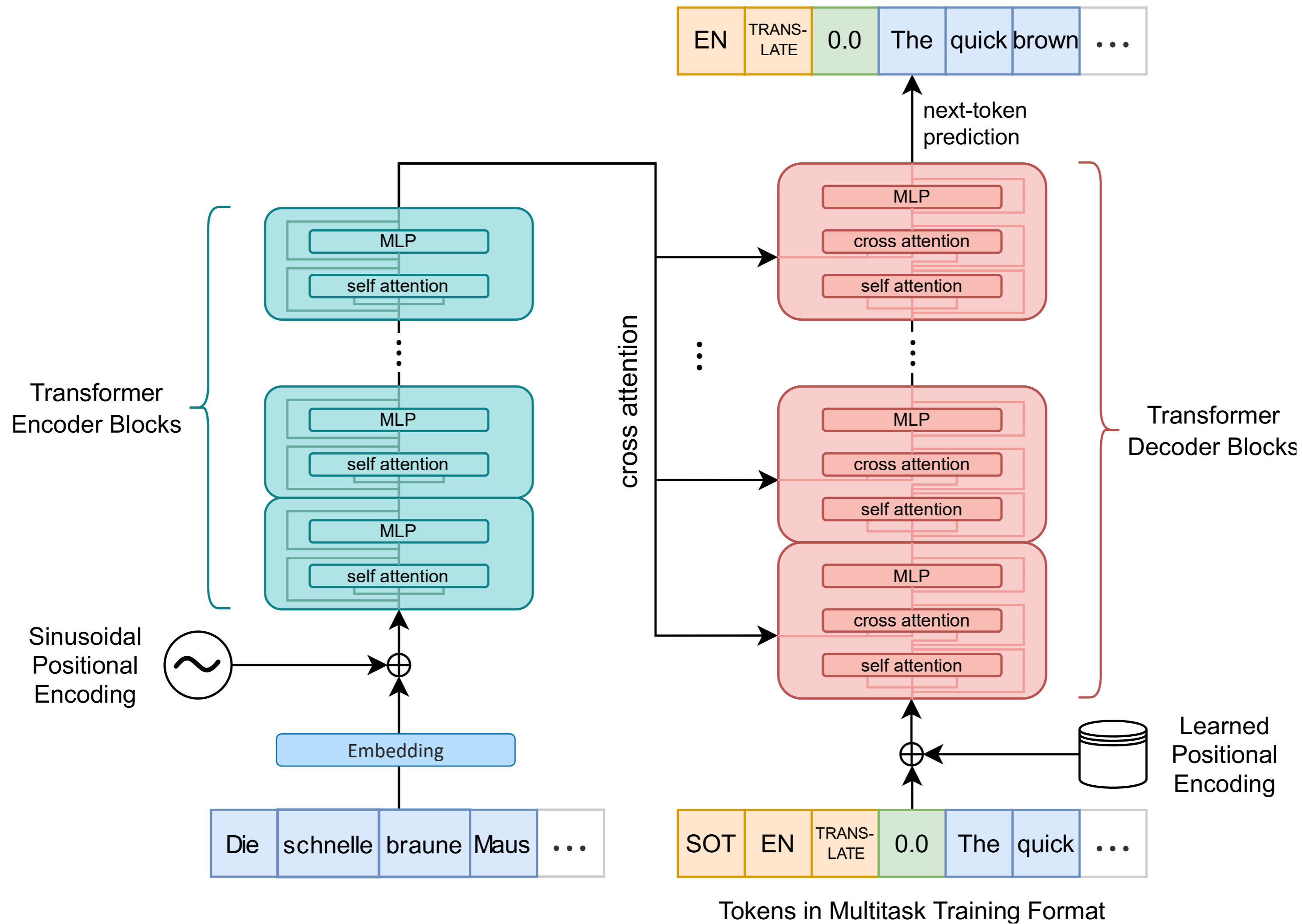


Whisper Transformer Architecture



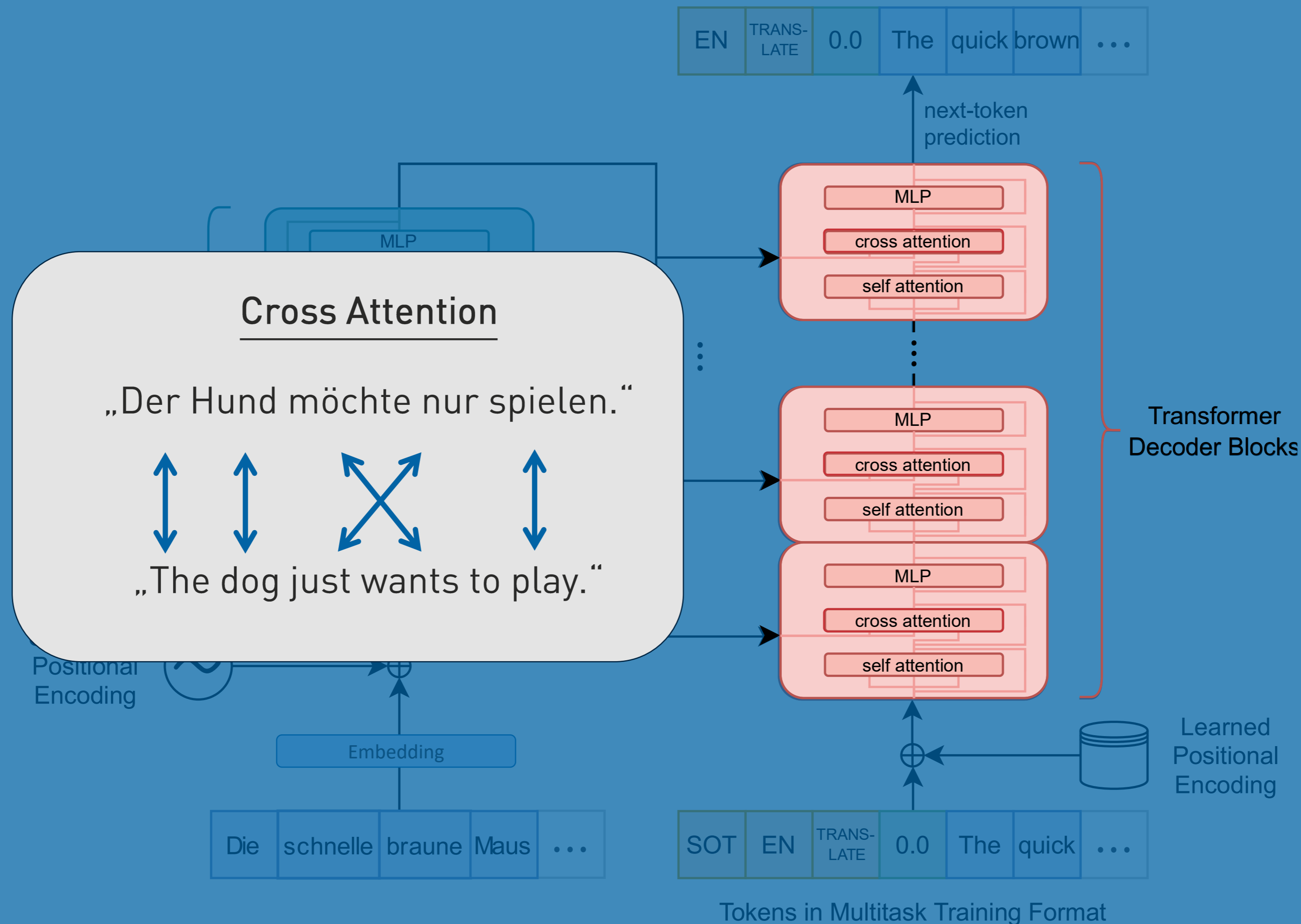


Whisper Transformer Architecture



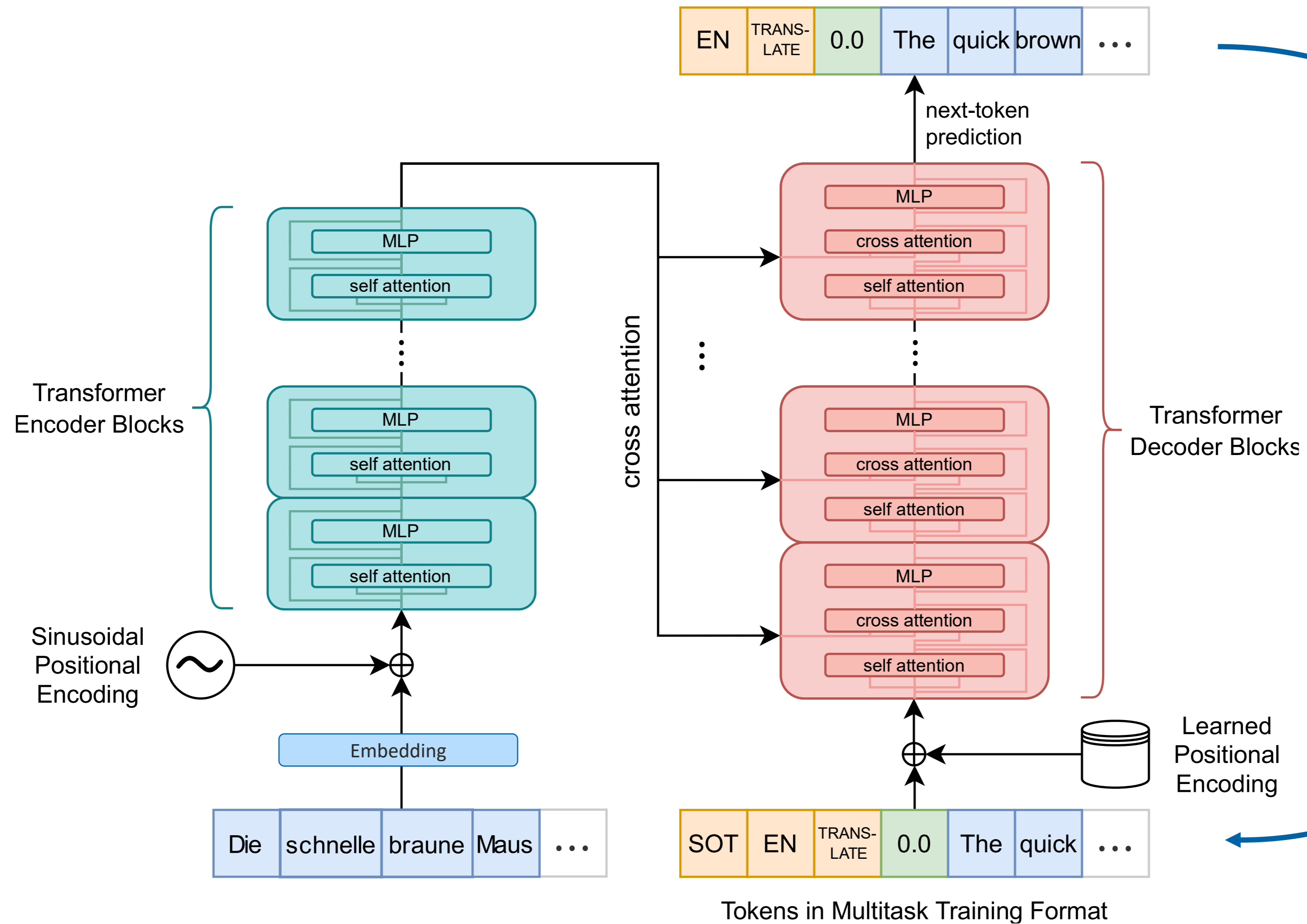


Whisper Transformer Architecture



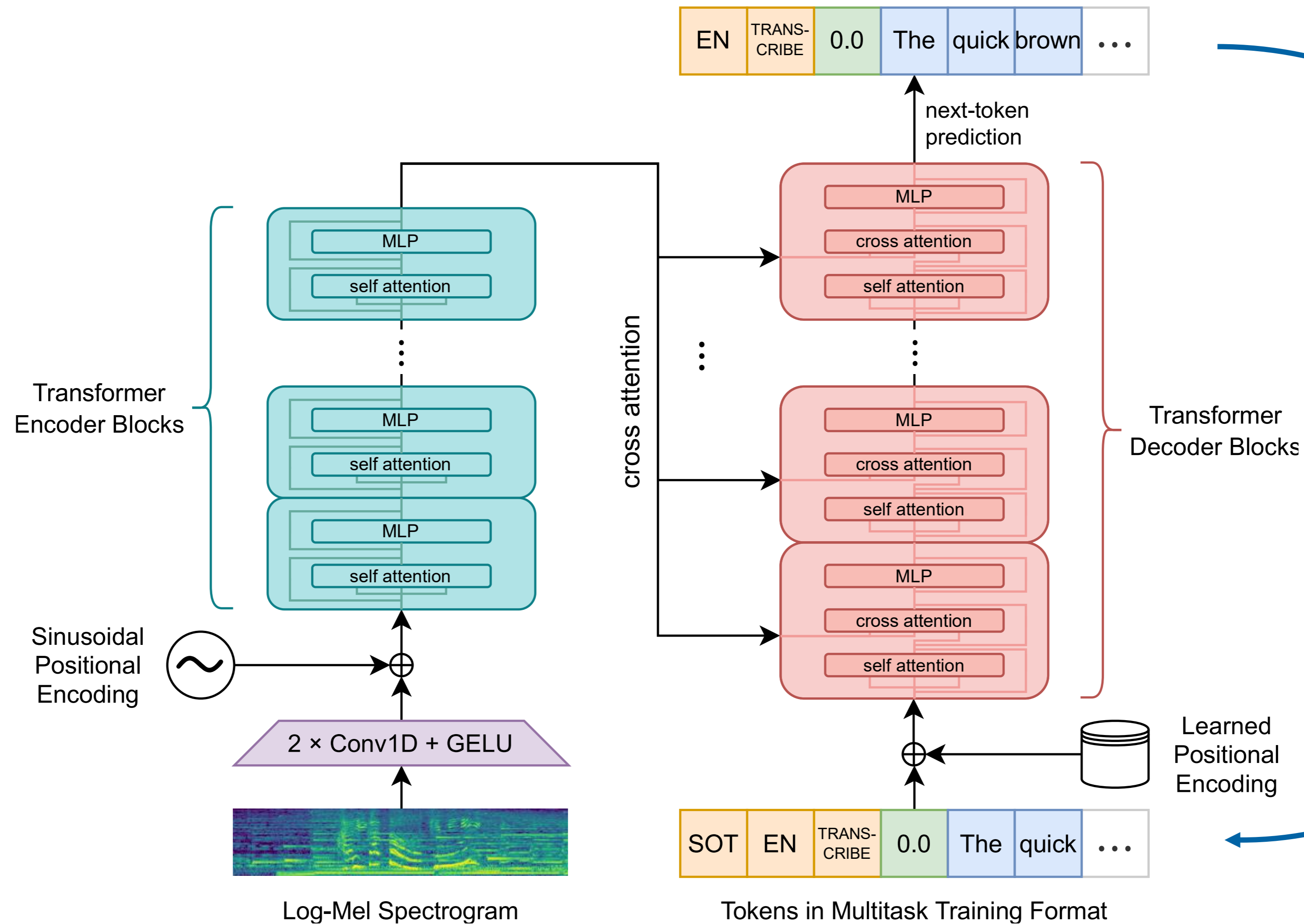


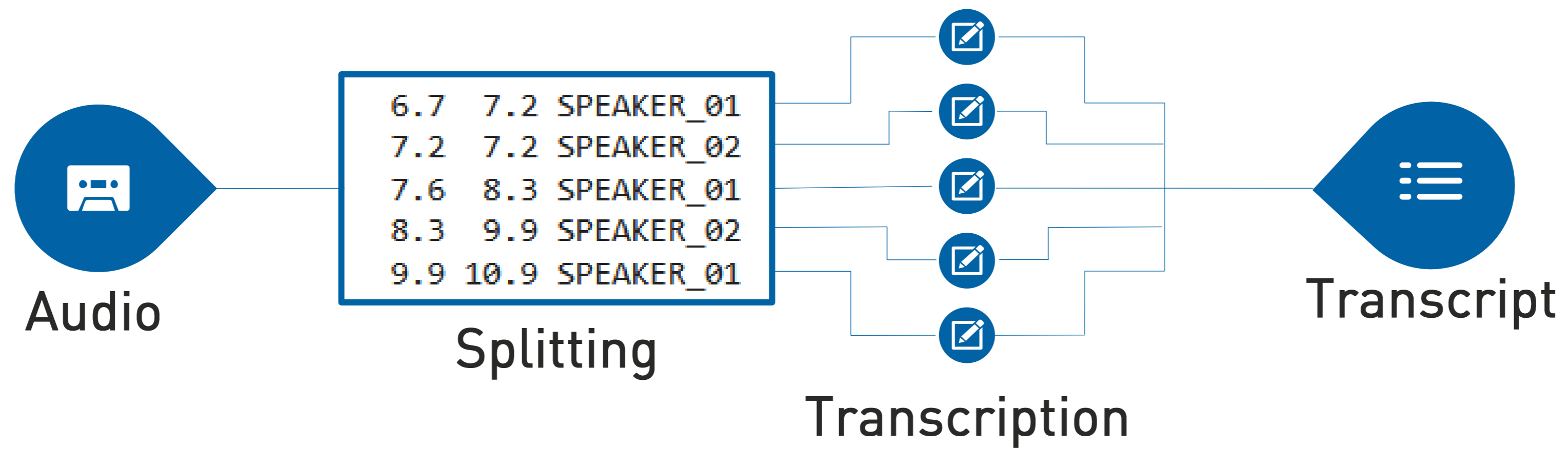
Whisper Transformer Architecture





Whisper Transformer Architecture







Transcription Implementation

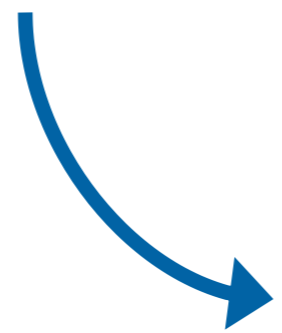




Meeting Minutes Generation



Audio Recording



Diarization



Transcription



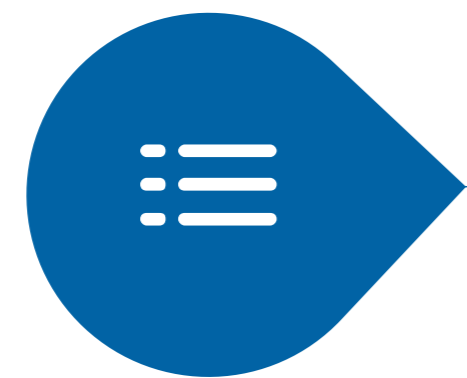
Generation

Meeting Minutes

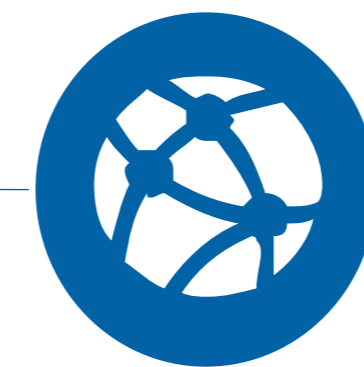




Instructions



Transcript

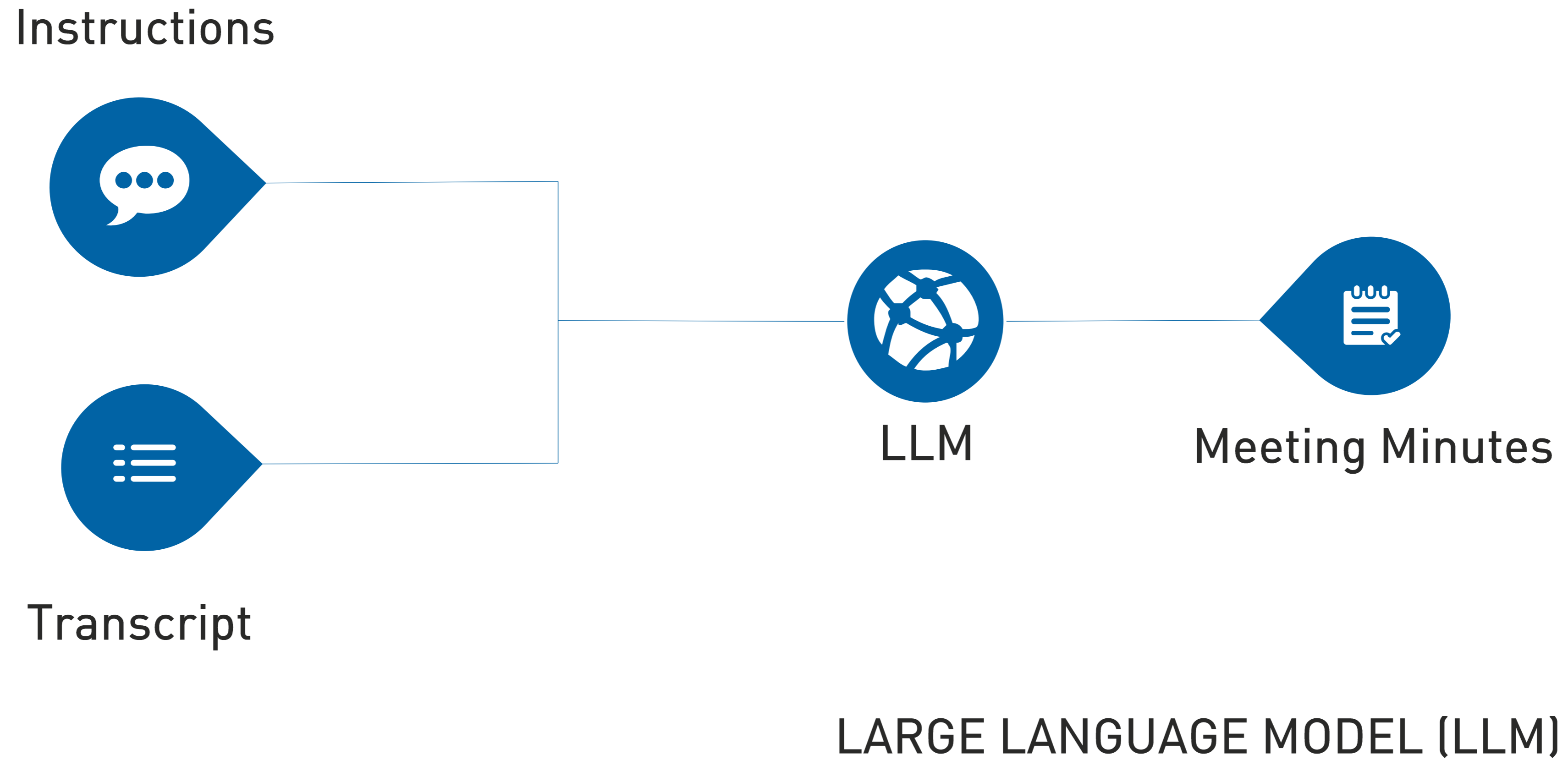


LLM

LARGE LANGUAGE MODEL (LLM)

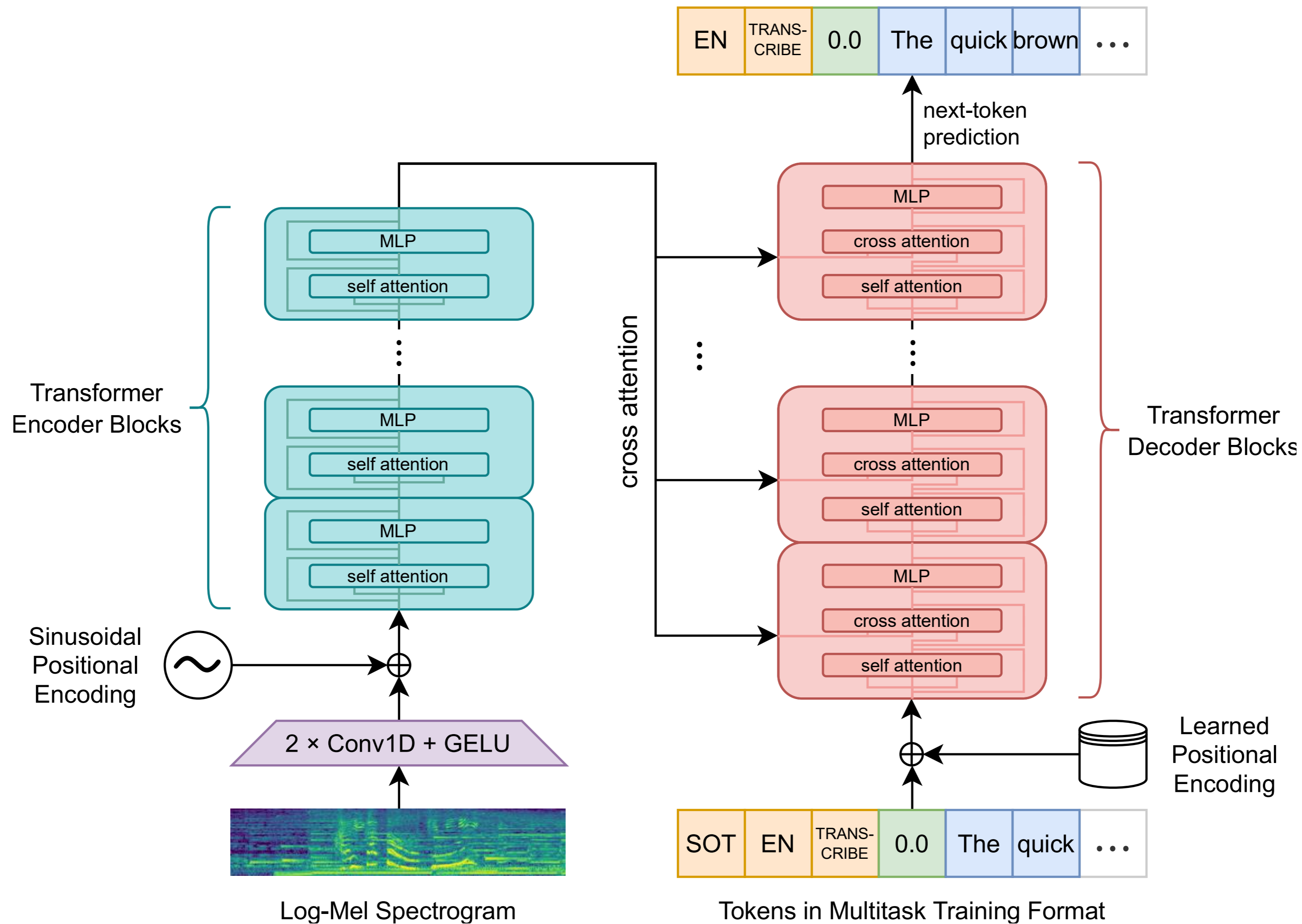


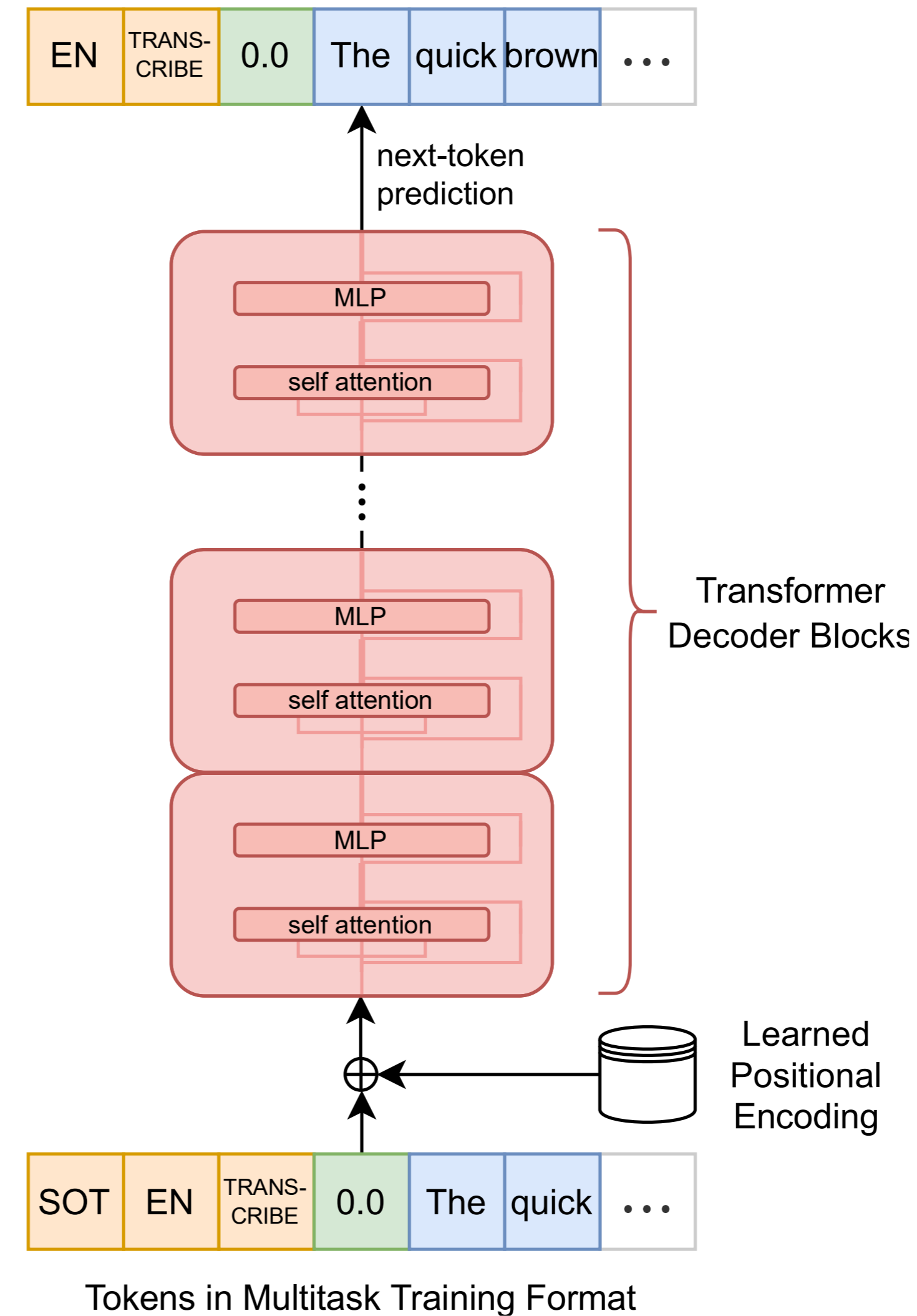
Meeting Minutes





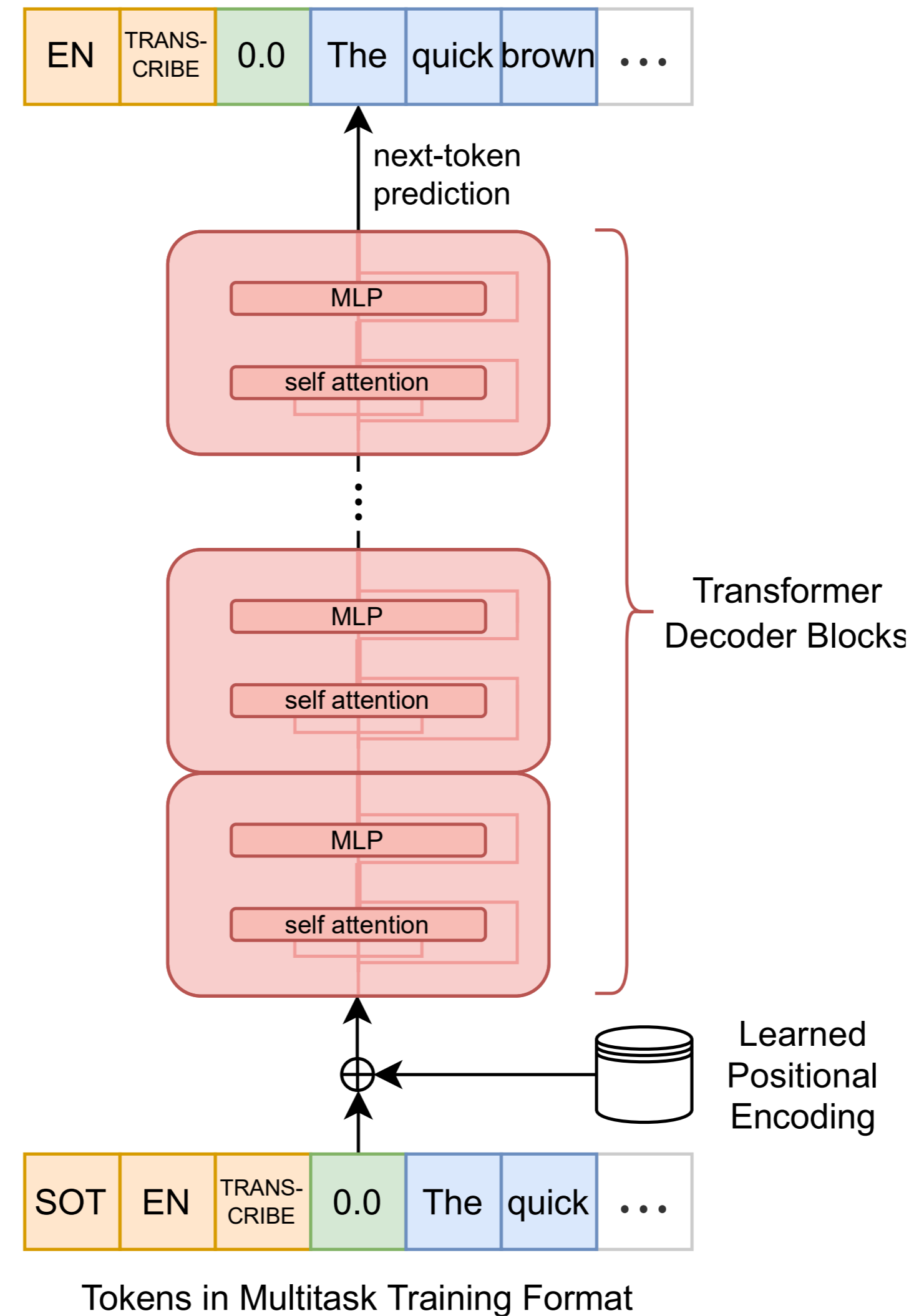
GPT Transformer Architecture

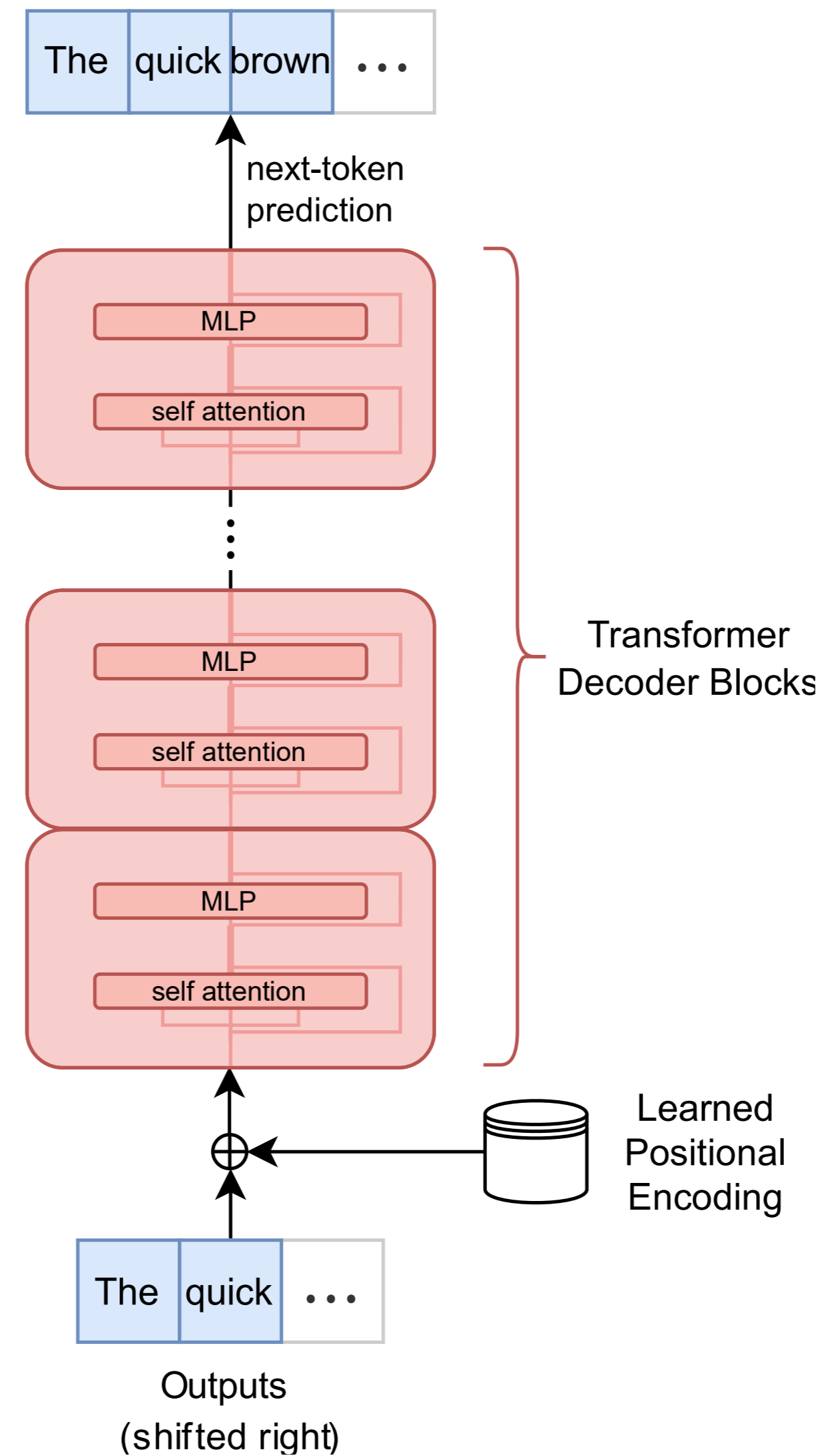


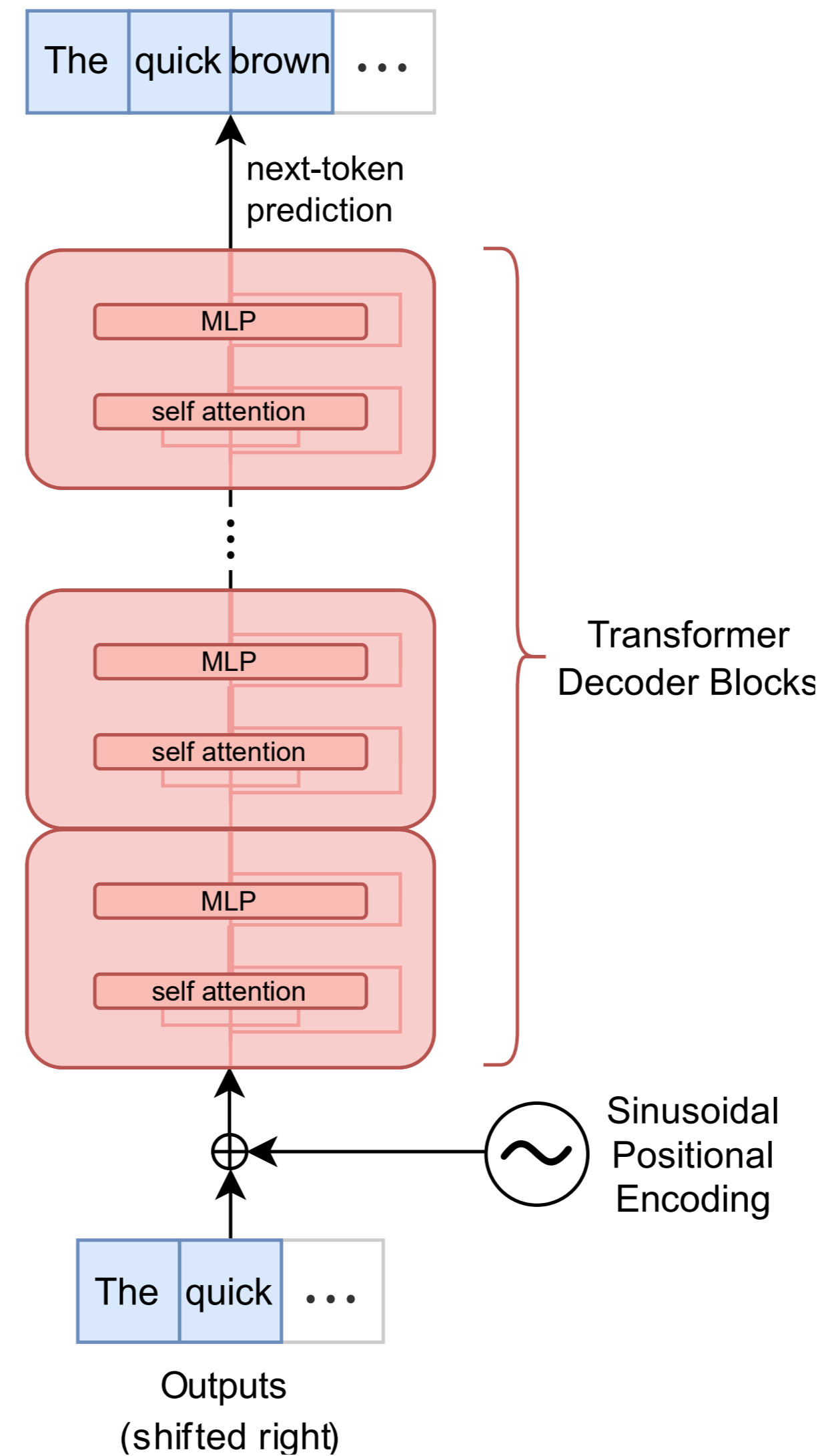


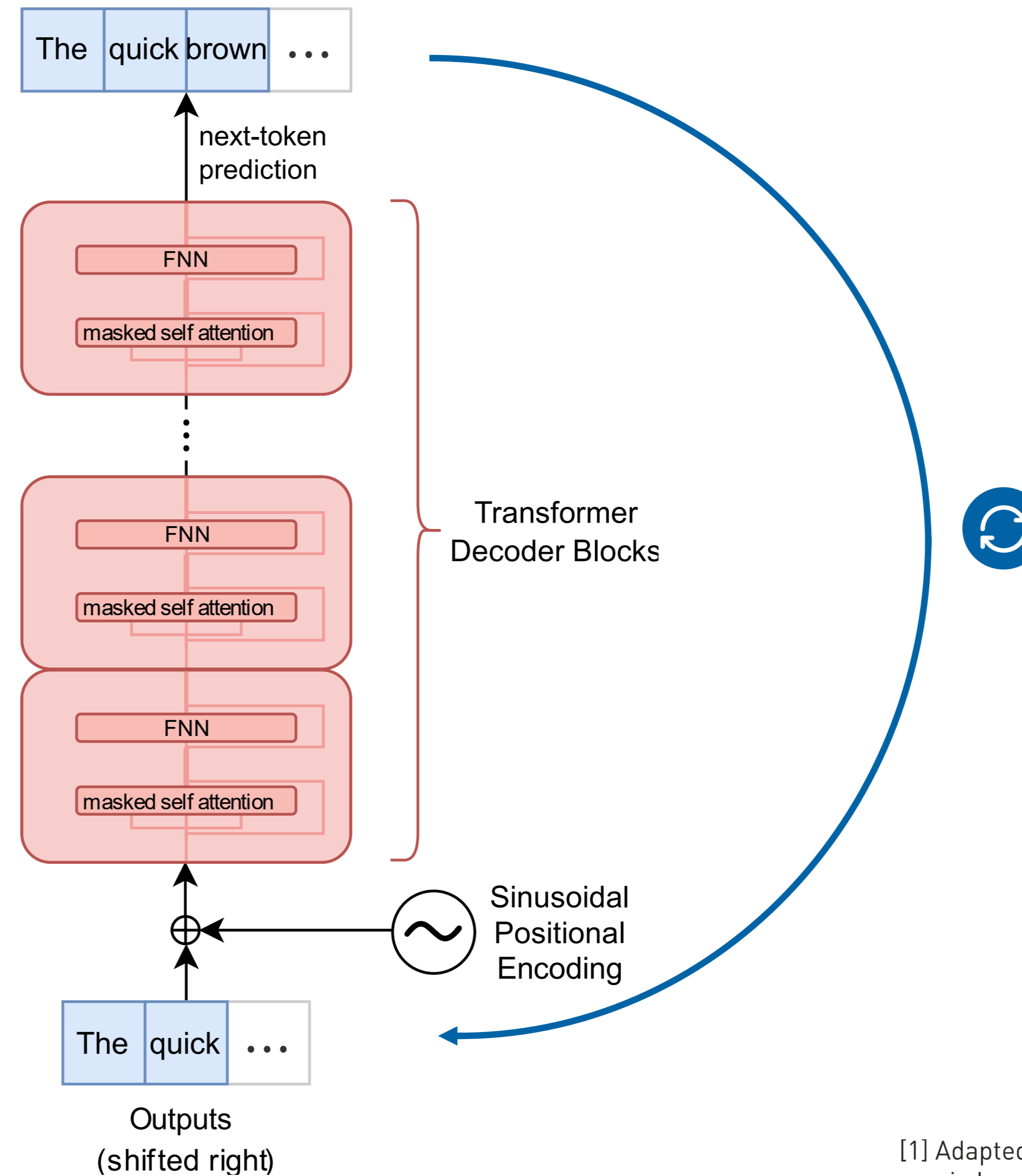


GPT Transformer Architecture



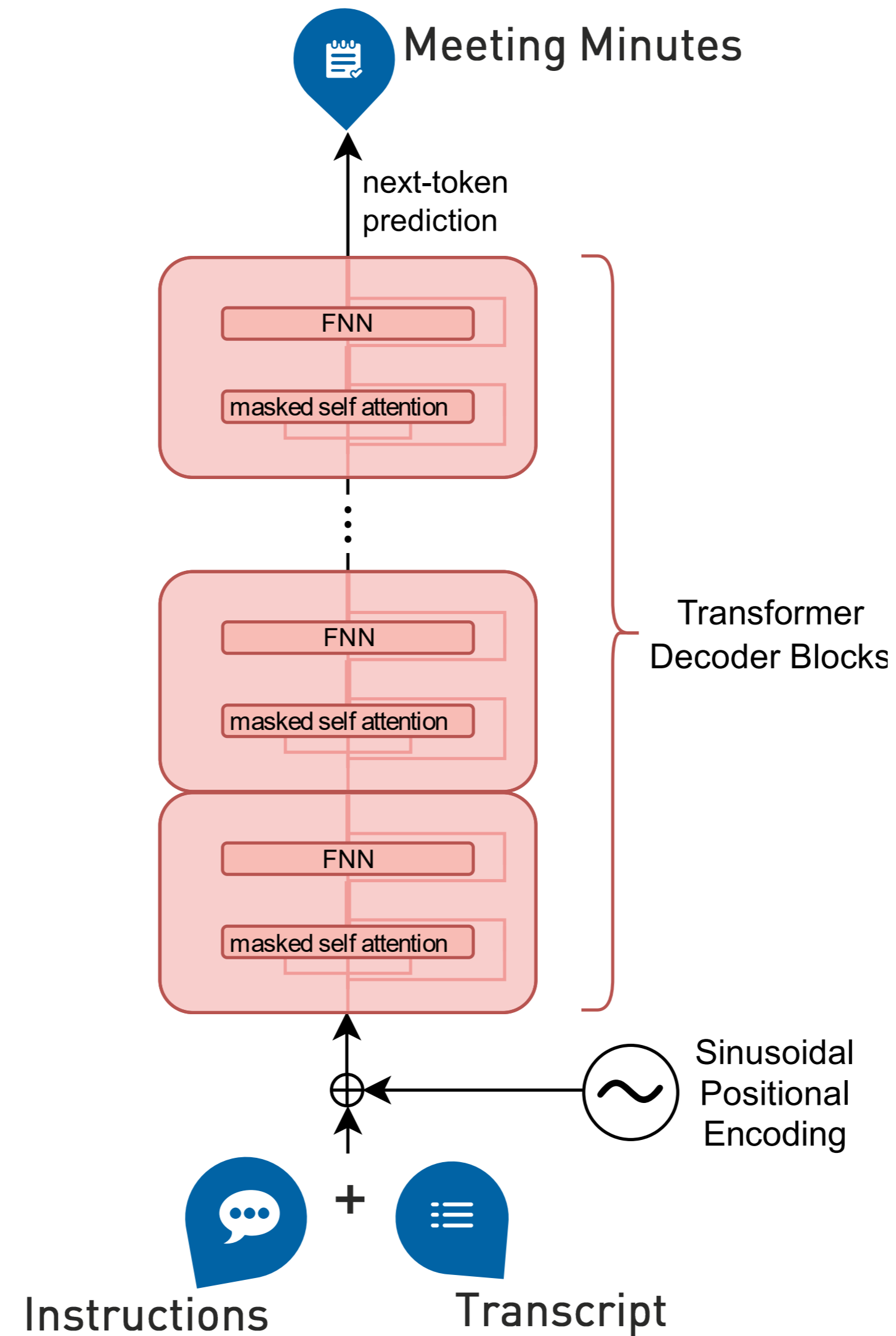








GPT Transformer Architecture





Ollama

<https://github.com/ollama/ollama>



LLaMA C++

Llama.cpp

<https://github.com/ggerganov/llama.cpp>



Both run locally



Open source (MIT license)



Various open-source models available
(e.g. Llama3, Mistral)



User-defined LLM endpoint:

- Meminto calls a user-provided LLM endpoint

Free choice:

- Select model and provider of your choice

Scalability:

- Seamless switching in a fast paced environment



Meminto



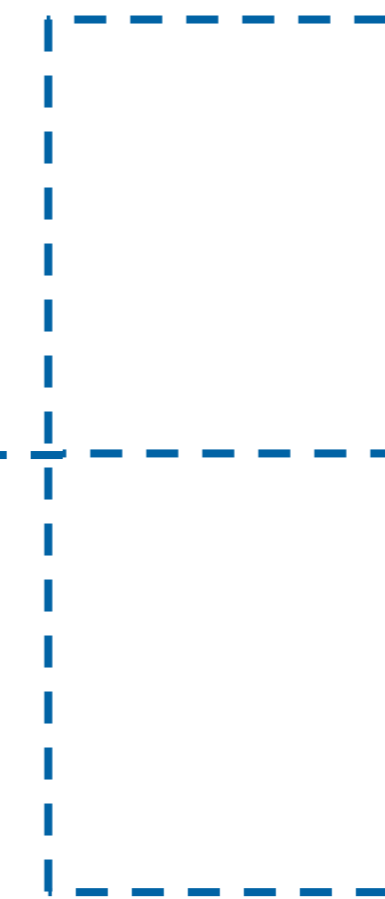
Local LLM



On-Premise LLM

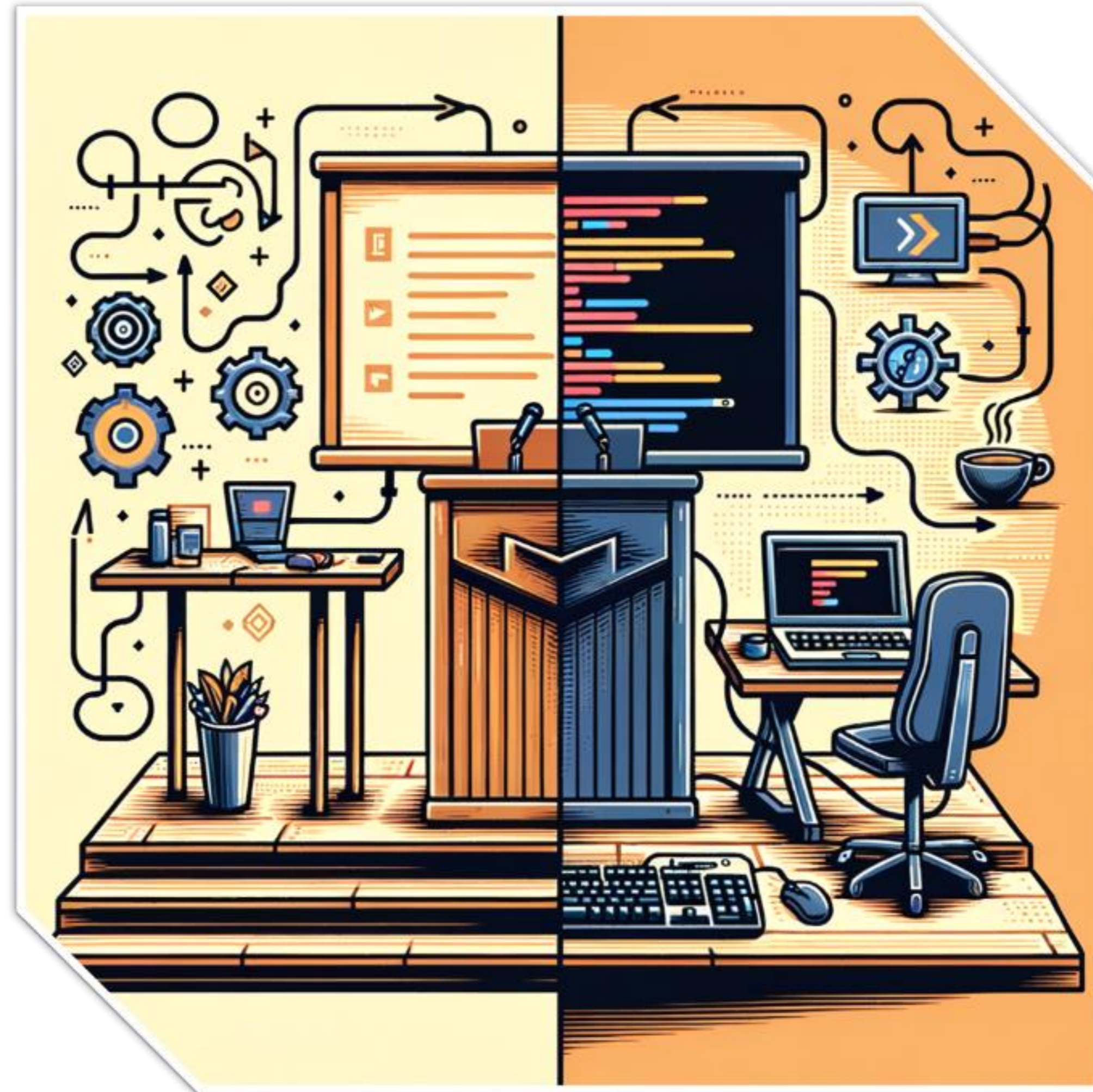


Third-Party LLM





Generation Implementation





Limited Context Size

Context size:

- LLMs can only process a limited number of tokens at once
- The context size is determined during training / fine-tuning

Typical context sizes:

- GPT-4 Turbo: 128k tokens
- Llama 3: 8k tokens

Example:

- Context size: 5

Lunch	is	at	the	Mace
-------	----	----	-----	------

- Context size: 4

Lunch	is	at	the	<?>
-------	----	----	-----	-----

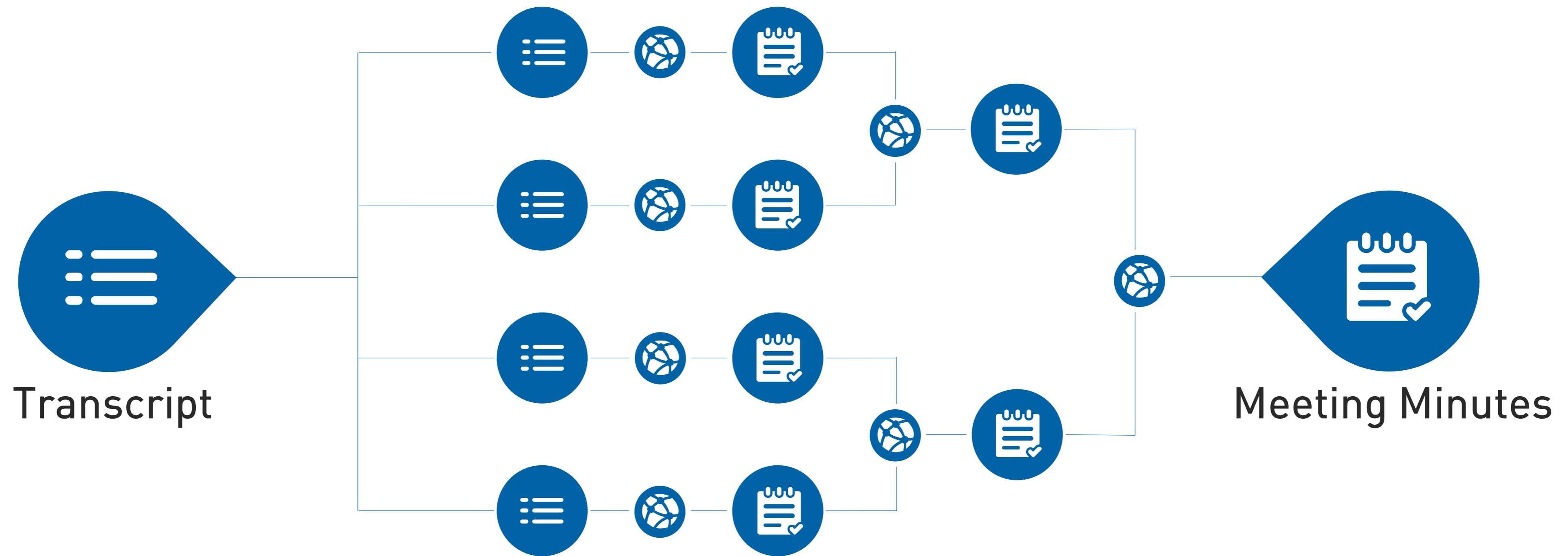




Chunking

Parallel Generation

 LLM





Chunking Implementation





Output

****Goals:****

- Plan the new high scoreboard for the game Pegasus
- Store and show the achieved score and player name for each game played
- Discuss and plan the implementation of a game leaderboard with a limit of 10,000 stored games
- Implement a sortable table with search function

****Decisions:****

Scoreboard:

- Ask players for their names at the end of the game
- Provide an option for players to opt out of having their name shown on the board
- Use an SQLite database to store player scores for the beginning
- Limit the score to a certain number of entries (exact number to be determined)

Leaderboard:

- Store the top 10,000 games to ensure good performance
- Implement a sortable table by score and name
- Add a search function to find games of a specific player faster

****Assigned Tasks:****

- SPEAKER_01: Implement the frontend part of the table and talk to legal
- SPEAKER_00: Take care of the backend part of the table implementation

****Additional Notes:****

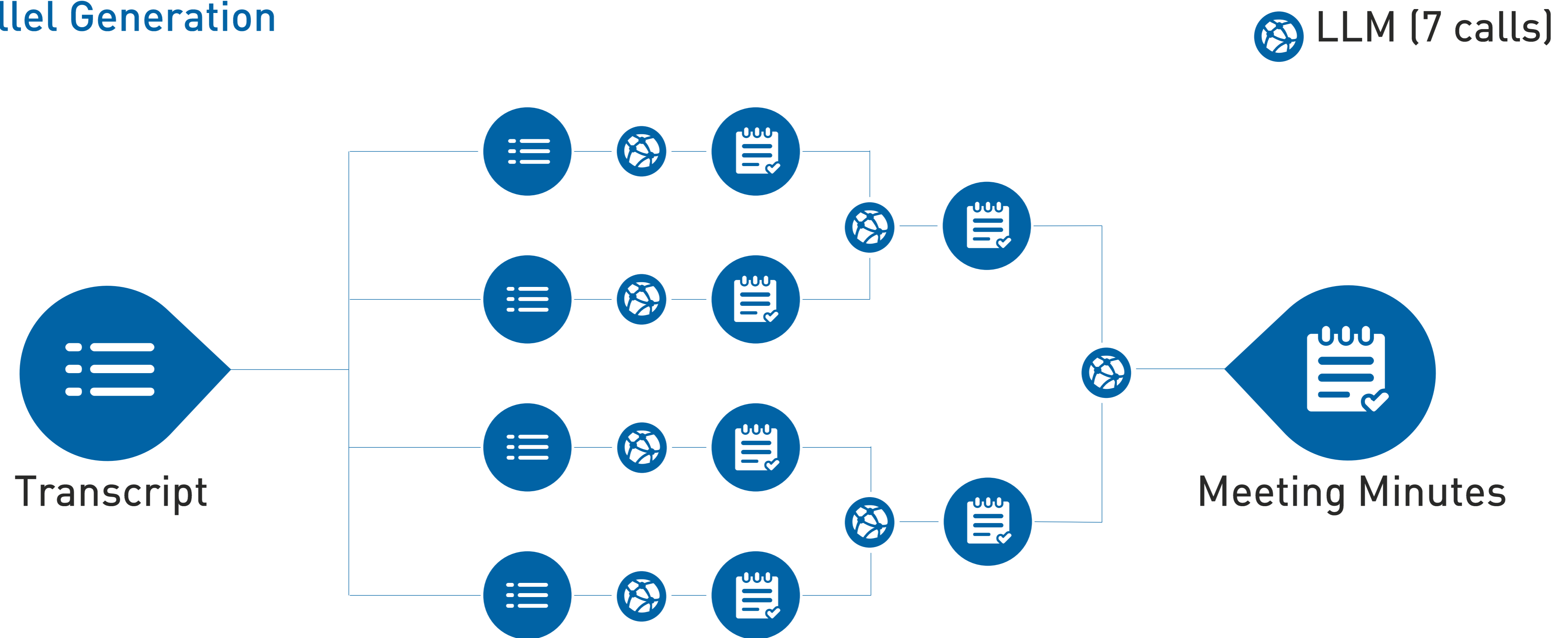
- Consider compliance-related issues and consult with GleeGlyph
- Potential issue with long loading times if too many scores are stored in the database
- Discussion around the ideal number of entries to limit the scoreboard to (10 vs 50k)
- The meeting was considered productive and goals were achieved
- The discussion about grabbing a beer at the Hercules bar at five was not related to the main meeting topic

4

Outlook

Alternative Chunking Approach

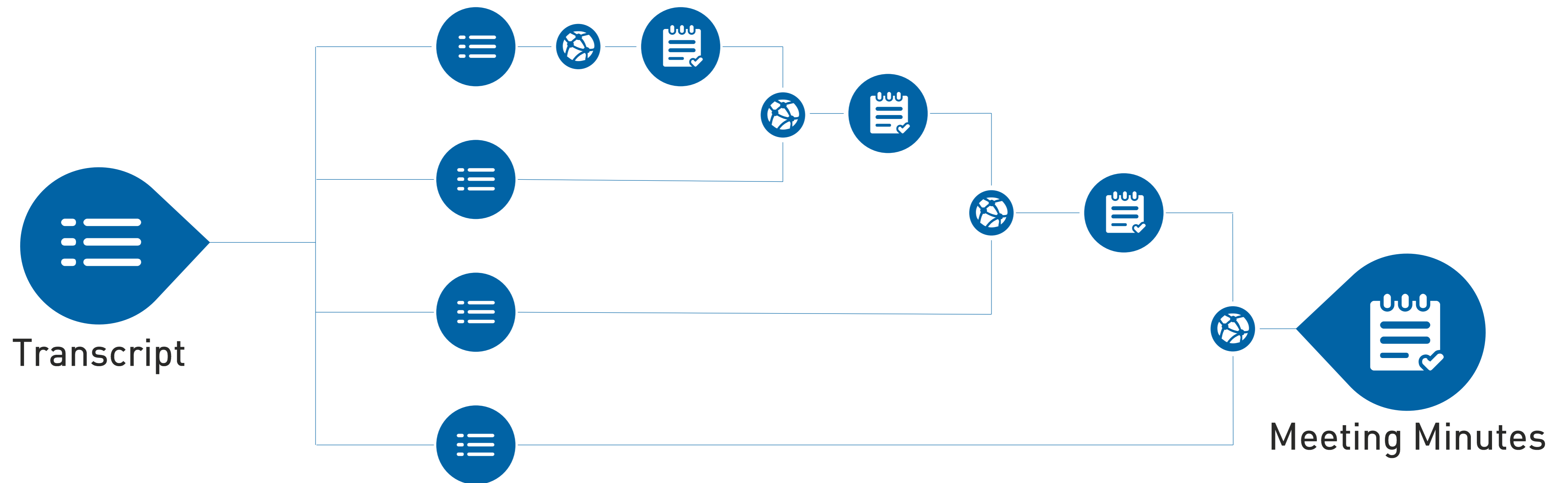
Parallel Generation



Alternative Chunking Approach

Consecutive Generation

 LLM (4 calls)

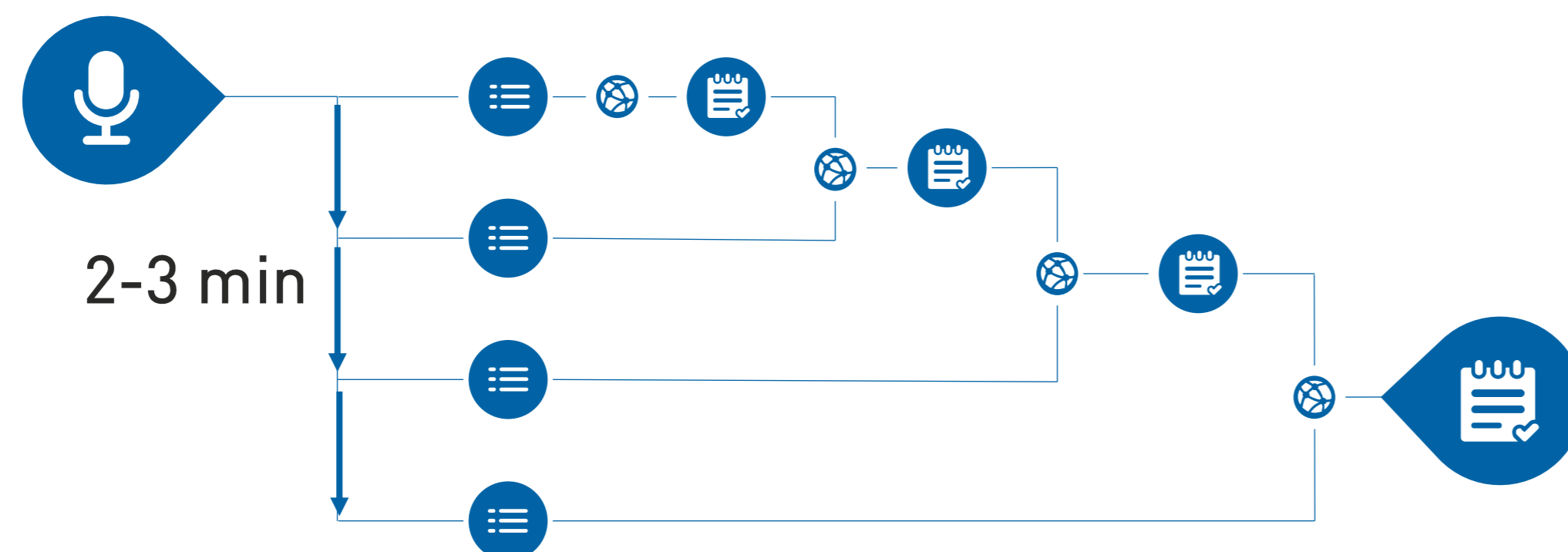


On-the-fly Generation:

- Generate meeting minutes during the meeting.
- Record audio over a given time frame (2-3 minutes)
- Update meeting minutes based on the most recent recording

On-the-fly Modification:

- Participants can modify or correct meeting minutes during the meeting
- Changes will be considered during the next iteration



Expanded technical horizon:

- Practical experience with cutting-edge technologies

Getting started was easier than expected:

- Many well-maintained and easy to use open-source libraries

Building connections:

- Expanding my professional network and fostering a sense of community

What's your next project?



Thank you for your attention!



Florian Schepers
Software Consultant
florian.schepers@tngtech.com



