```
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "gpt-3.5-turbo-0125",
  "system_fingerprint": "fp_44709d6fcb",
  "choices": [{
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "\n\nHello there, how may I assist you today?",},
    "logprobs": null,
    "finish_reason": "stop"
  }],
  "usage": {
    "prompt_tokens": 9,
    "completion_tokens": 12,
    "total_tokens": 21
  }
}
```

When streaming inference transmission is not selected, the most important part is the "messages" content under the "choices" section, which represents the inference results provided by the model.

In contrast, when streaming inference transmission is enabled, the server returns a Response object, which includes the output results of the model at different points in time during the streaming inference process. The data content at each moment is as follows:

```
{
"id": "chatcmpl-123",
  "object": "chat.completion.chunk",
  "created": 1677652288,
  "model": "gpt-3.5-turbo-0125",
  "system_fingerprint": "fp_44709d6fcb",
  "choices": [{
    "index": 0,
    "delta": {
      "role": "assistant",
      "content": "\n\nHello there, how may I assist you today?",},
    "logprobs": null,
    "finish_reason": "stop"
  }]
}
```

After receiving the data from streaming transmission, users need to focus on the "delta" data section within the "choices" part. Additionally, when "finish_reason" is empty (None), it indicates that the model is still in the inference state, and the data has not been fully generated yet. It's only when "finish_reason" returns "stop" that the streaming inference is considered finished.

In the provided deployment example code and API access examples for RKLLM-Server-Flask, you can see identical definitions of the transmission data structure, ensuring the generality of the deployed