LLaMa

Example Generation

Future of Programming
WiSe 2023 / 2024
Software Architecture Group

Christian Raue / Lara Kursawe

Design IT.
Create Knowledge.

www.hpi.de

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Question: Can LLMs generate helpful examples of API interfaces?

You

Show me an example of how to use the Dataframe.empty method from the pandas API

ChatGPT

The `DataFrame.empty` method in the pandas library is used to check if a DataFrame is empty (i.e., it has zero rows and zero columns). It returns a boolean value indicating whether the DataFrame is empty or not.

Here's an example of how to use the `DataFrame.empty` method:

python Copy code

```python
import pandas as pd

# Create an empty DataFrame
empty_df = pd.DataFrame()

# Check if the DataFrame is empty
if empty_df.empty:
    print("The DataFrame is empty.")
else:
    print("The DataFrame is not empty.")
```

In this example, we first create an empty DataFrame using the `pd.DataFrame()` constructor. Then, we use the `empty` method to check if the DataFrame is empty. If it is empty, it will print "The DataFrame is empty," otherwise, it will print "The DataFrame is not empty."

You

Show me an example of how to use the PreTrainedTokenizer.convert_ids_to_tokens method from HuggingFace

ChatGPT

As of my last knowledge update in January 2022, Hugging Face's Transformers library provides the `PreTrainedTokenizer.convert_ids_to_tokens` method for converting token IDs back to tokens. Note that the library may have evolved, and it's always a good idea to check the latest documentation for any changes or updates.

Here's an example of how you might use `convert_ids_to_tokens`:

python Copy code

```python
from transformers import PreTrainedTokenizer, AutoTokenizer

# Replace 'bert-base-uncased' with the desired pre-trained model name o
model_name = 'bert-base-uncased'
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Encode a sentence to obtain token IDs
text = "Hello, how are you doing?"
token_ids = tokenizer.encode(text, add_special_tokens=True)

# Convert token IDs back to tokens
tokens = tokenizer.convert_ids_to_tokens(token_ids)

print("Original text:", text)
print("Token IDs:", token_ids)
print("Tokens:", tokens)
```

Question: Can LLMs generate helpful examples of API interfaces?

Yes!

Thank you for your attention!

convert_ids_to_tokens

<source>

```
( ids: typing.Union[int, typing.List[int]], skip_special_tokens: bool = False ) → str or List[str]
```

Parameters

- **ids** (`int` or `List[int]`) — The token id (or token ids) to convert to tokens.
- **skip_special_tokens** (`bool`, *optional*, defaults to `False`) — Whether or not to remove special tokens in the decoding.

Returns `str` or `List[str]`

The decoded token(s).

Converts a single index or a sequence of indices in a token or a sequence of tokens, using the vocabulary and added tokens.

convert_tokens_to_ids

<source>

```
( tokens: typing.Union[str, typing.List[str]] ) → int or List[int]
```

Parameters

- **tokens** (`str` or `List[str]`) — One or several token(s) to convert to token id(s).

Returns `int` or `List[int]`

The token id or list of token ids.

Converts a token string (or a sequence of tokens) in a single integer id (or a sequence of ids), using the vocabulary.

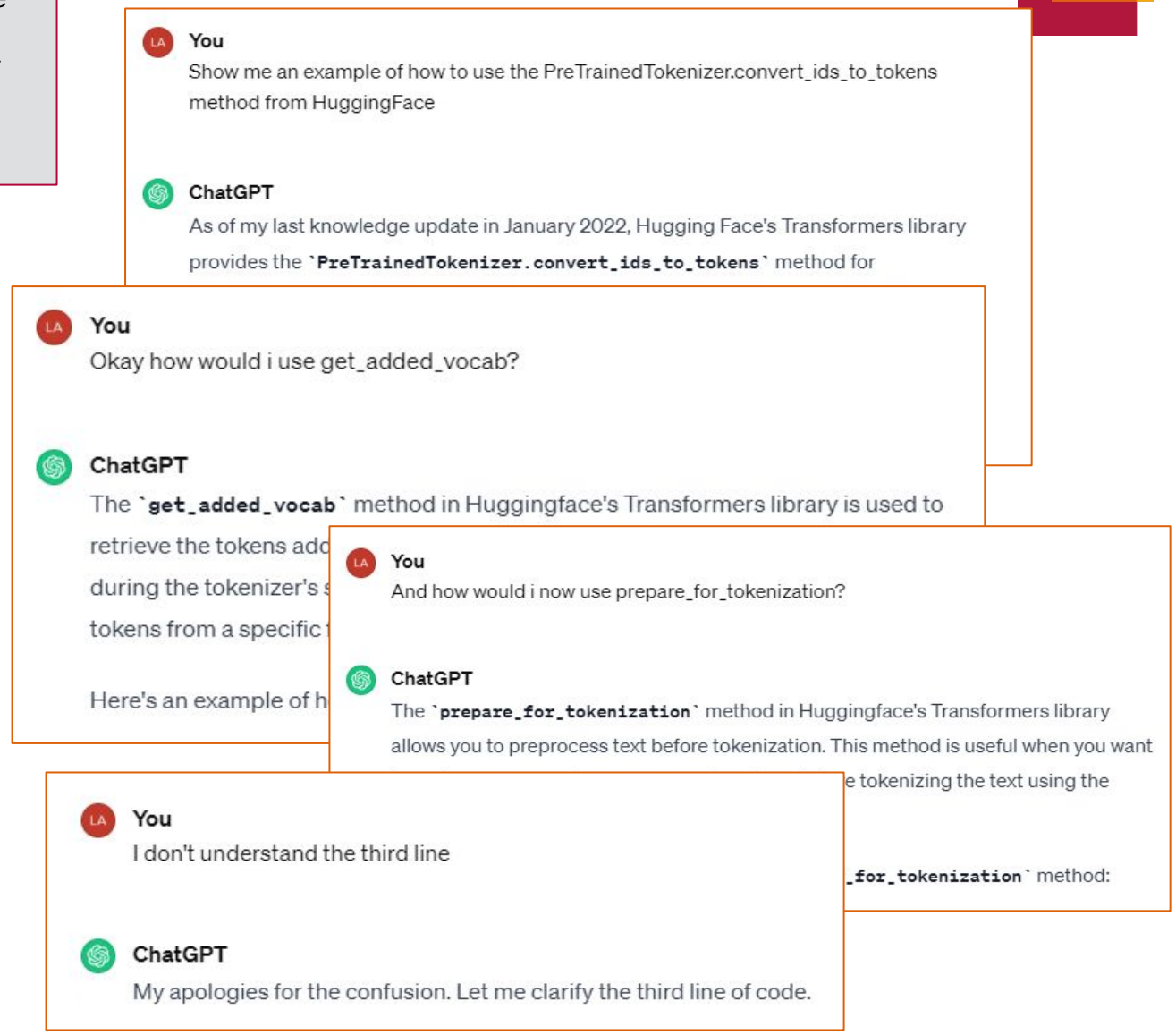
Many APIs don't provide any examples in their documentation, i.e. HuggingFace. This makes it difficult for inexperienced users to get familiar with the API, having to use other resources to help them find examples.

Idea

When using a less known API, you might even have to provide the API documentation for every function you need an example for.

Chatbots

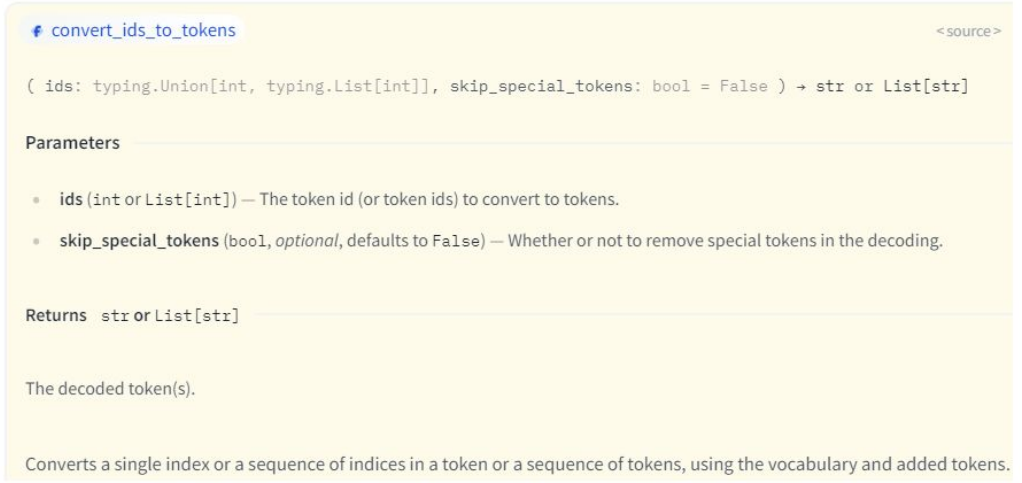
- Helpful when working and learning new tools, but often cumbersome to use
  - Might need long prompts
  - First results may be unhelpful
  - Multiple follow up prompts can be annoying
- Different LLMs give different results (of different quality)



Idea

Chatbots

- Helpful when working and learning new tools, but often cumbersome to use
  - Might need long prompts
  - First results may be unhelpful
  - Multiple follow up prompts can be annoying
- Different LLMs give different results (of different quality)



convert_ids_to_tokens <source>

(ids: typing.Union[int, typing.List[int]], skip_special_tokens: bool = False) → str or List[str]

Parameters

- **ids** (int or List[int]) — The token id (or token ids) to convert to tokens.
- **skip_special_tokens** (bool, *optional*, defaults to False) — Whether or not to remove special tokens in the decoding.

Returns str or List[str]

The decoded token(s).

Converts a single index or a sequence of indices in a token or a sequence of tokens, using the vocabulary and added tokens.

APIs

- Some API references are very detailed and helpful, others are not

xLLaMa lets you generate code examples for APIs, without having to use any prompts, making it much faster and convenient to get examples.

DEMO

Who would use xLLaMa?

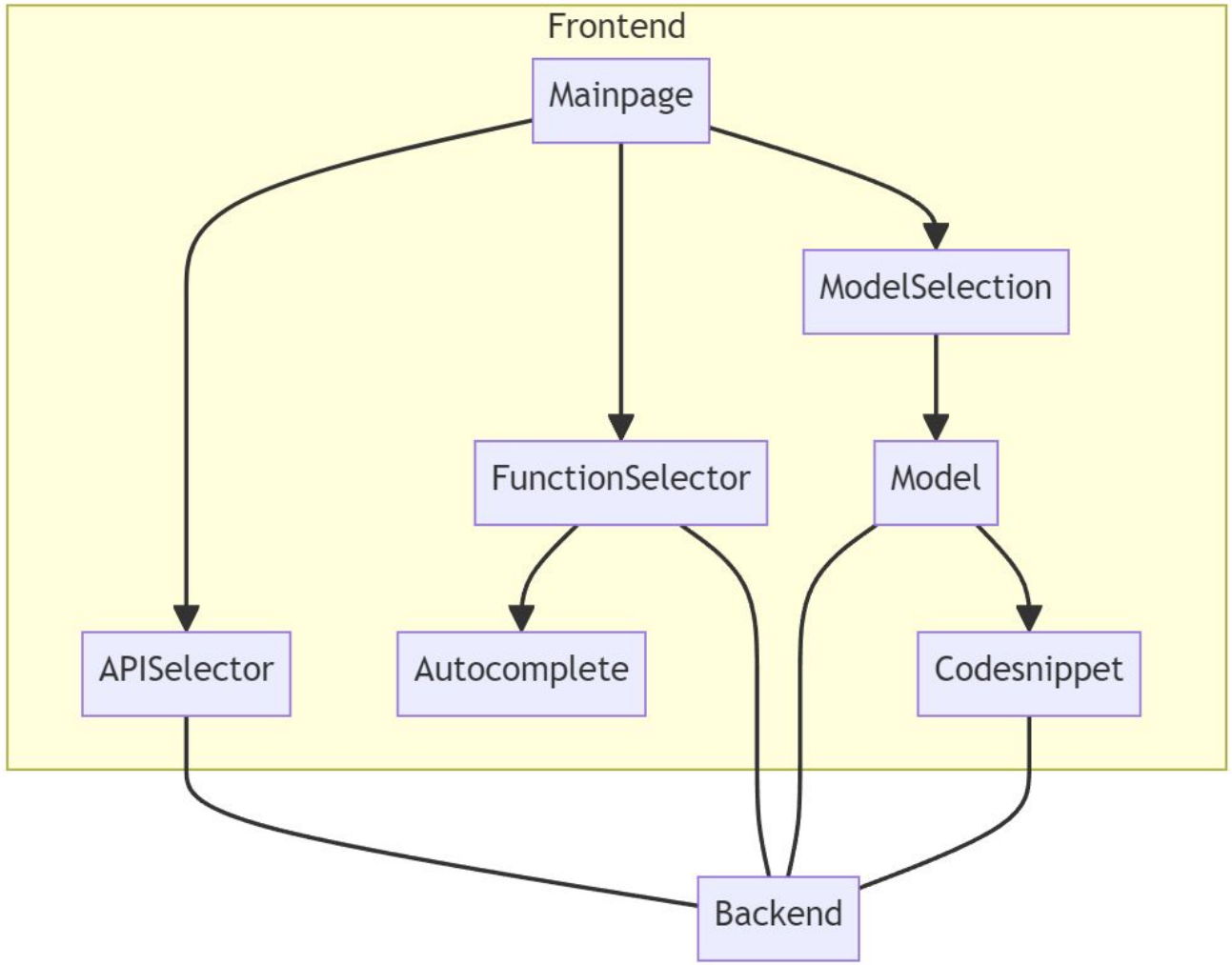
Students and Software Developers who…

- … are new to one of the frameworks / APIs
- … find themselves often wondering “wait, how does this work again…?”
- … want quick results, without having to type a lot to an LLM
- … want multiple different results to choose from
- … use multiple different IDEs and source-code editors, or editors without integrated AI assistants

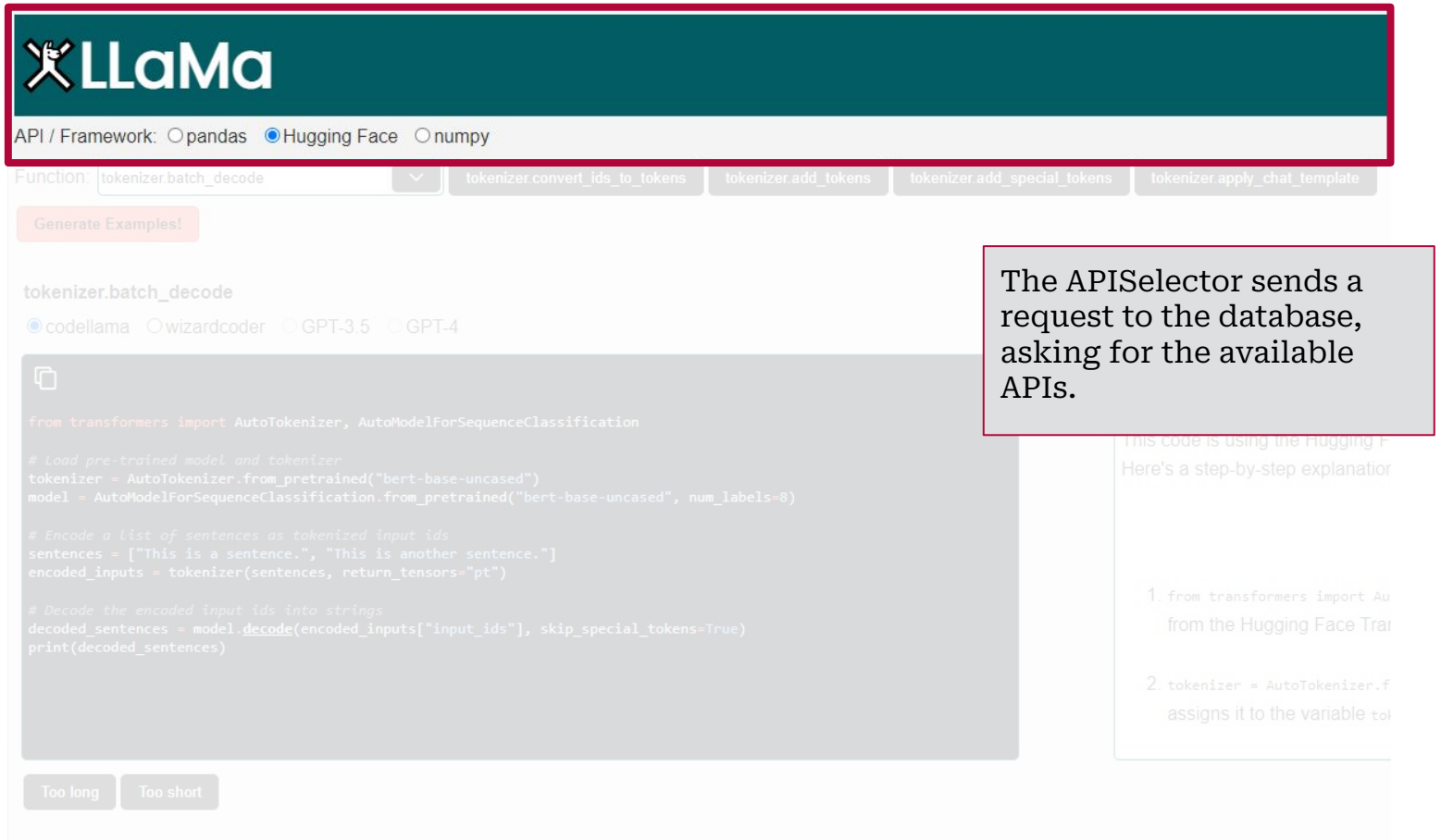
Architecture - Frontend

Vue.js

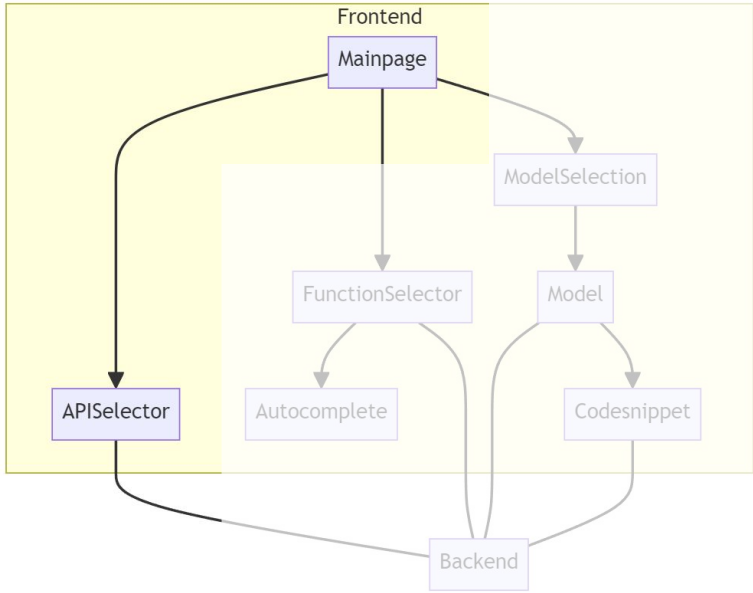
- Builds on top of standard HTML, CSS, and JavaScript
- Component-based programming model



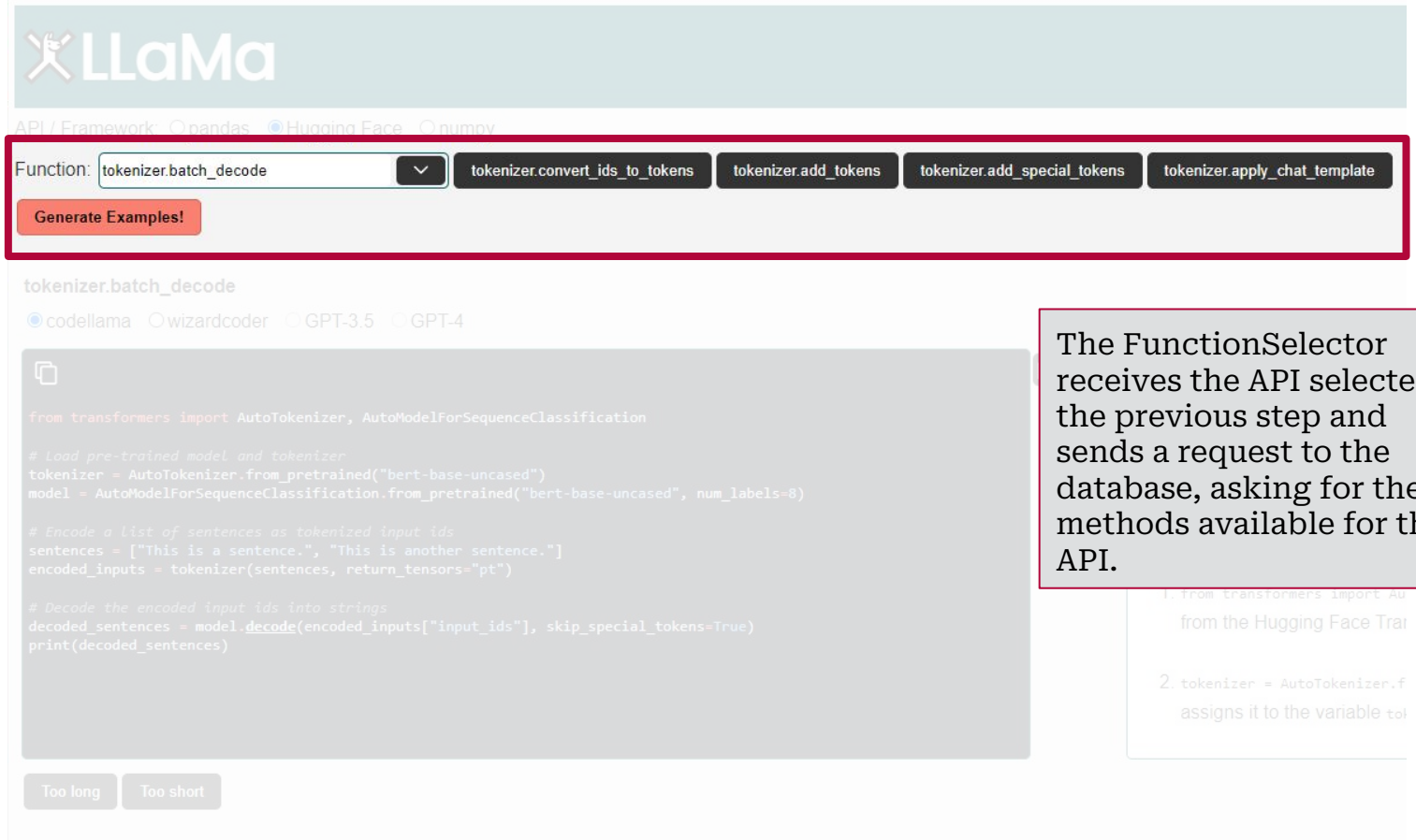
Architecture - Frontend



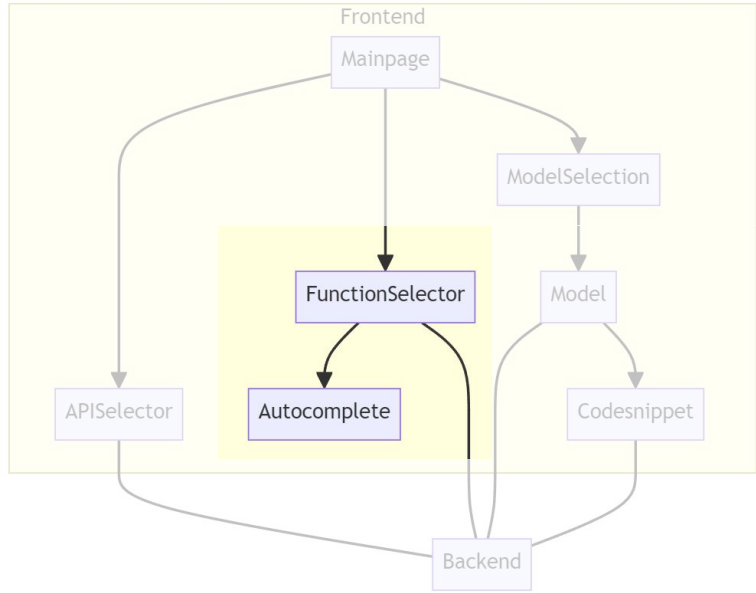
The APISelector sends a request to the database, asking for the available APIs.



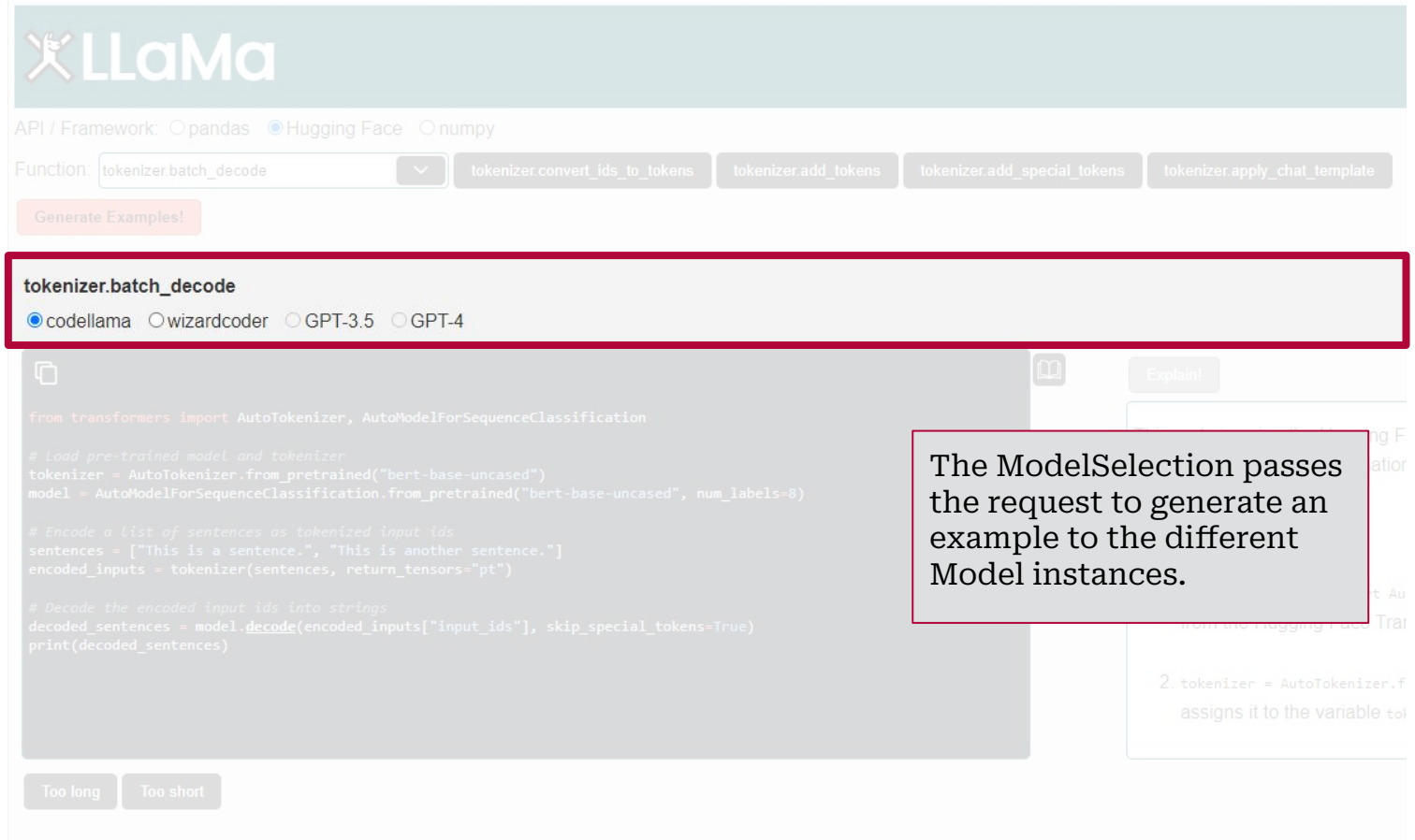
Architecture - Frontend



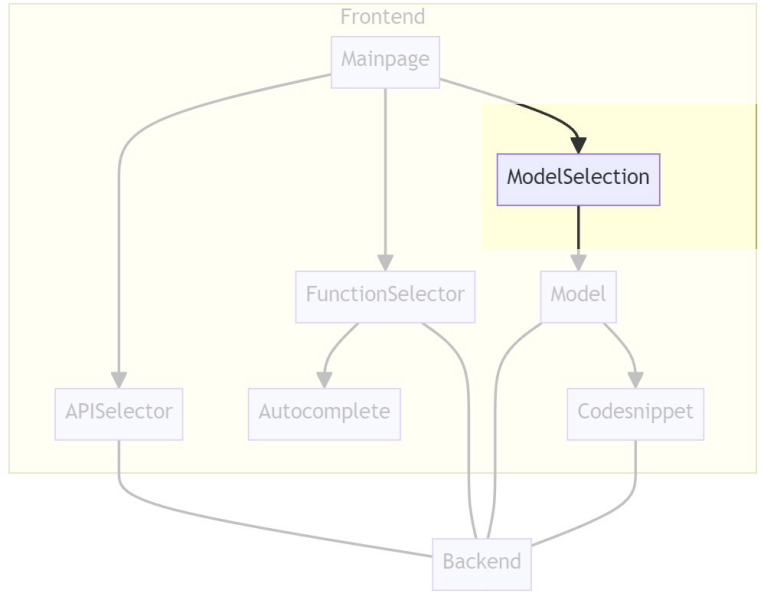
The FunctionSelector receives the API selected in the previous step and sends a request to the database, asking for the methods available for this API.



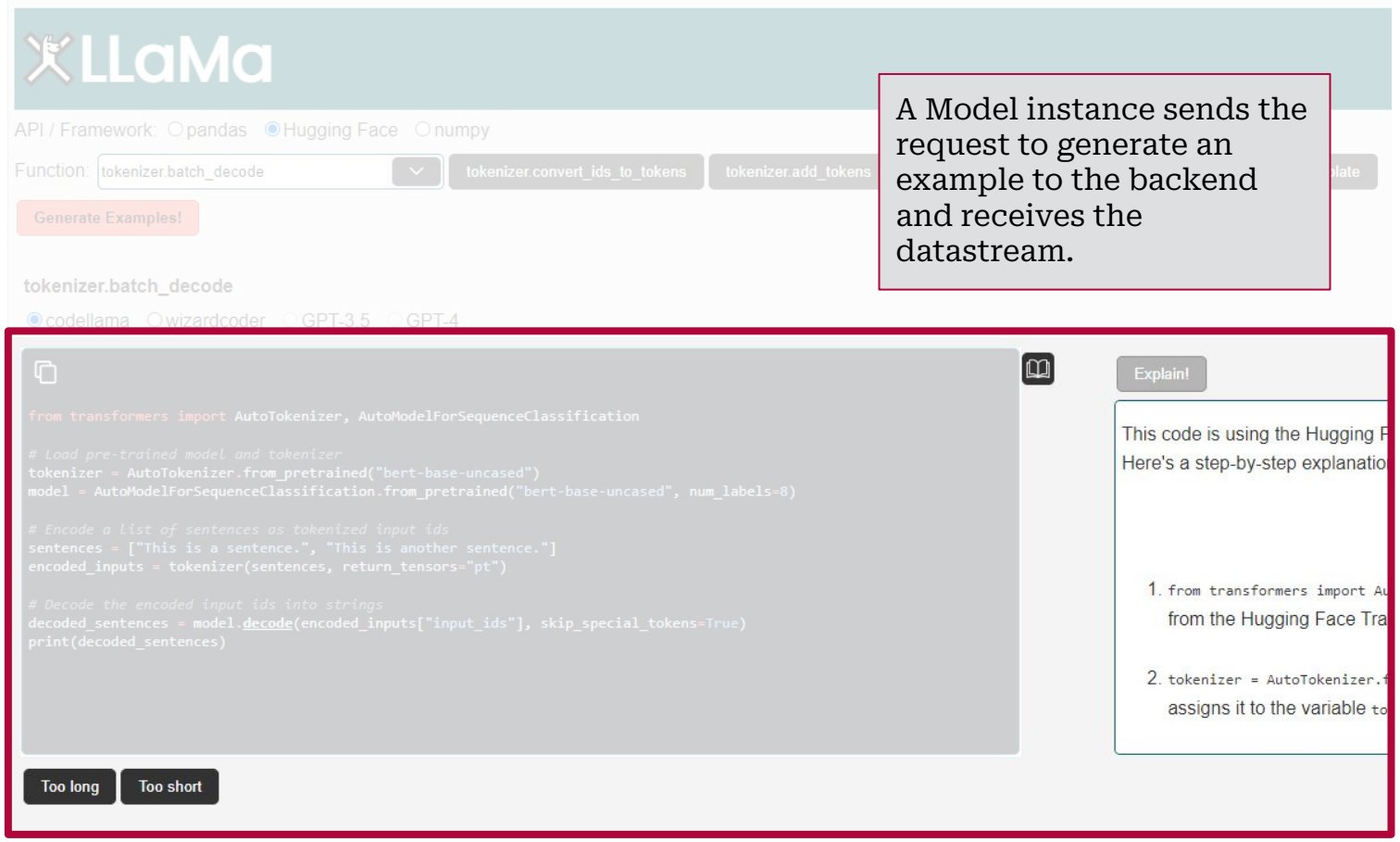
Architecture - Frontend



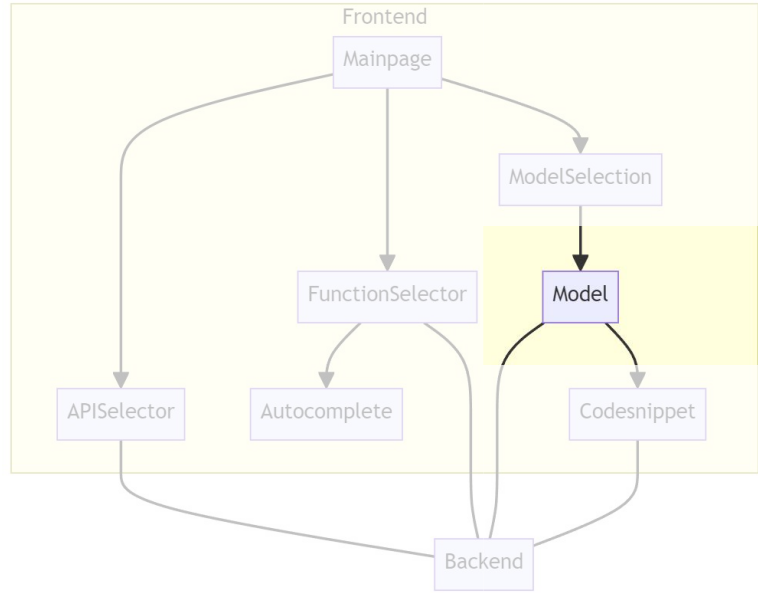
The ModelSelection passes the request to generate an example to the different Model instances.



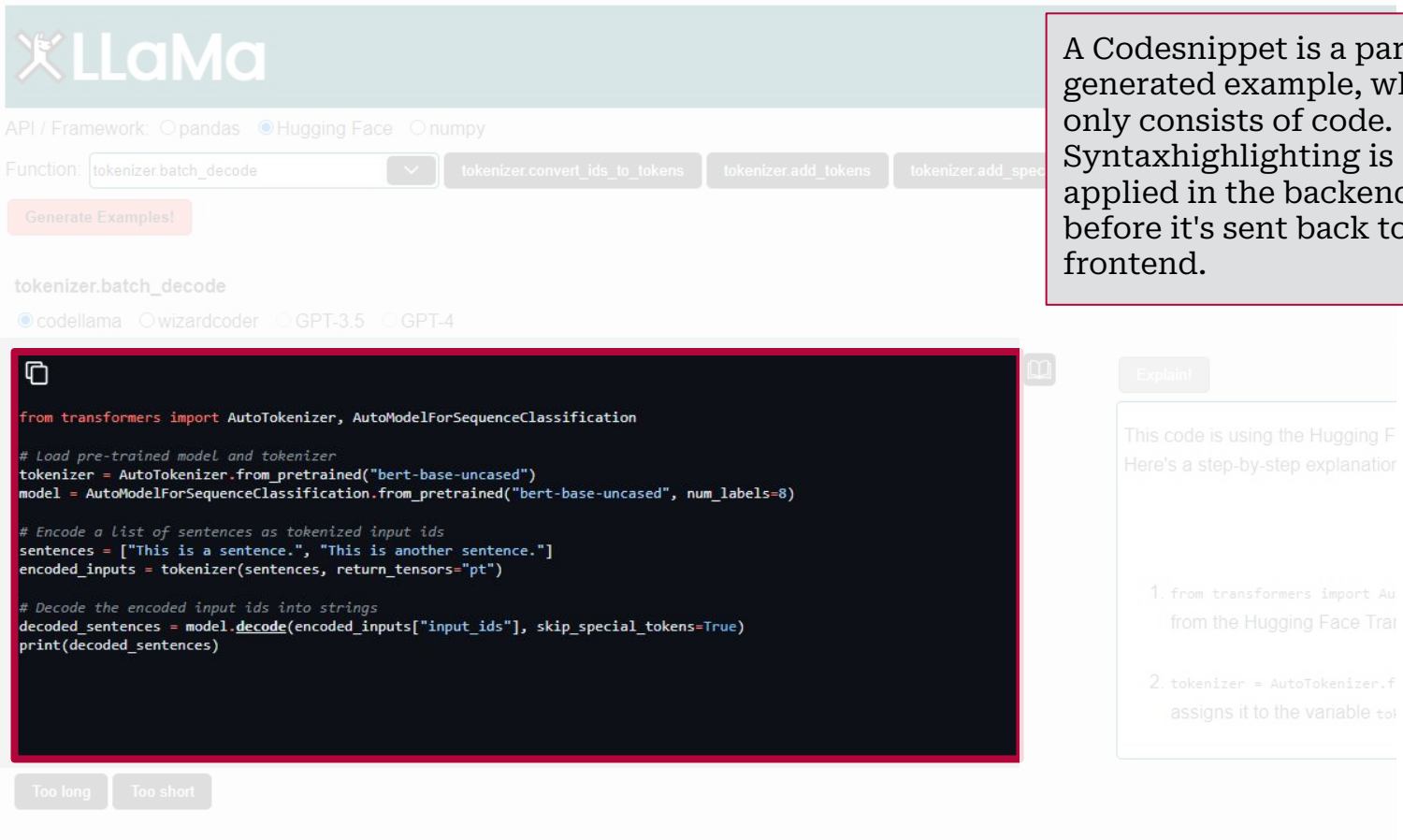
Architecture - Frontend



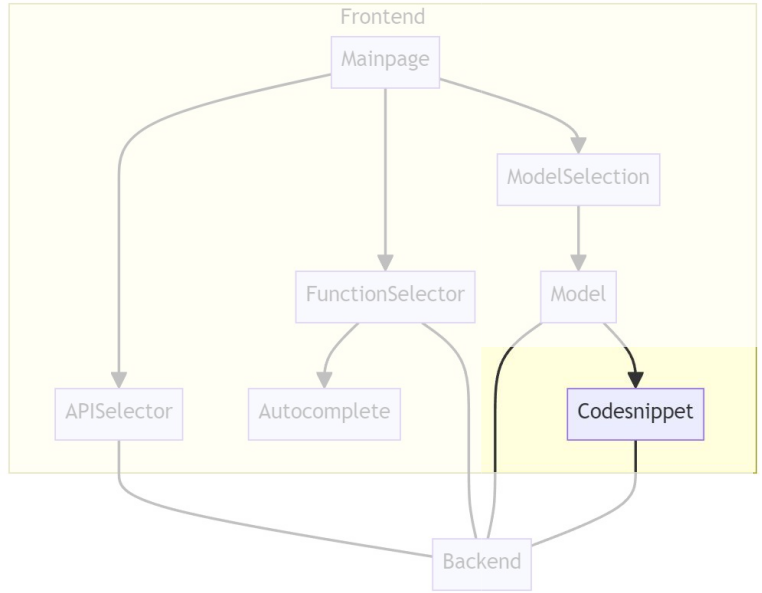
A Model instance sends the request to generate an example to the backend and receives the datastream.



Architecture - Frontend



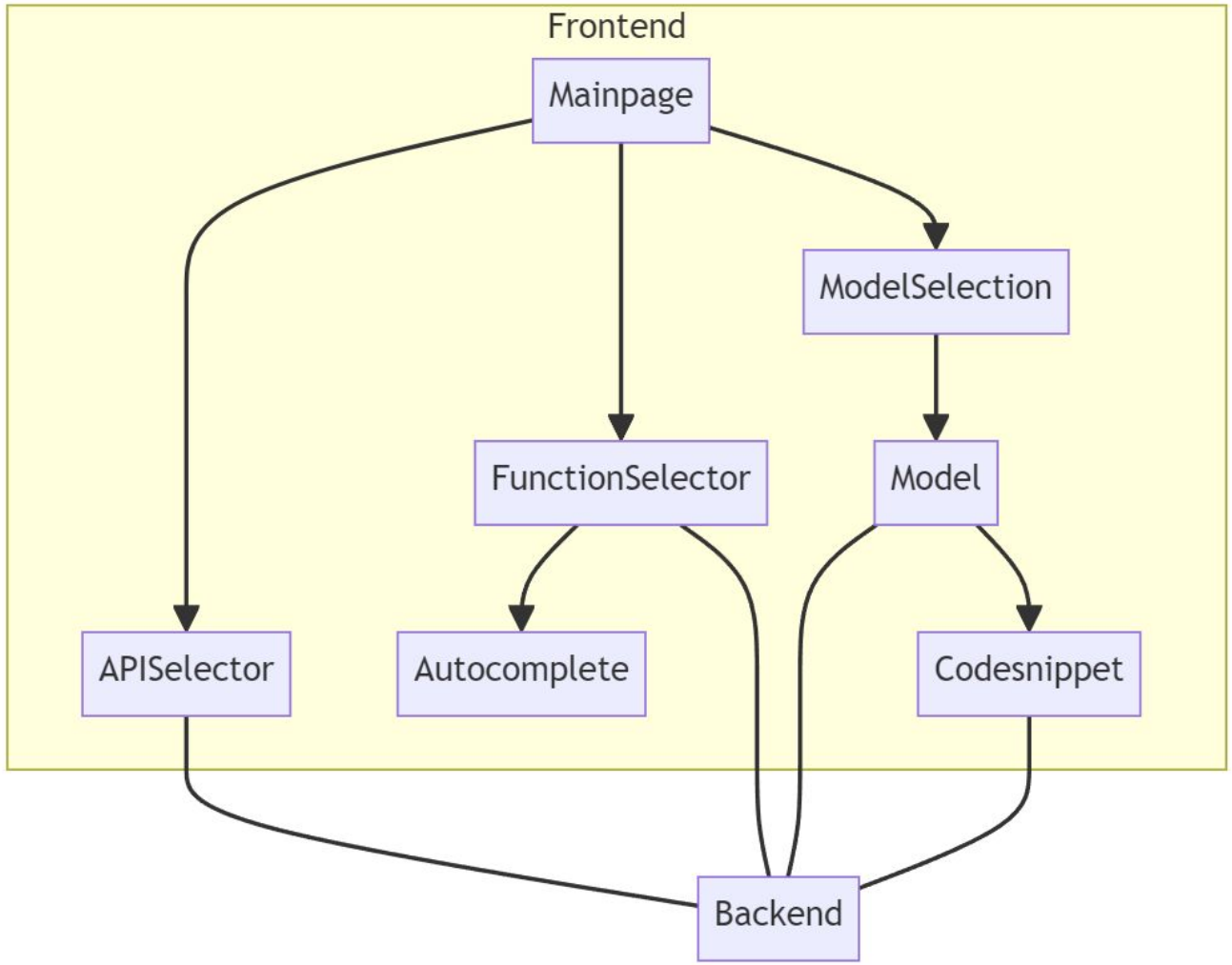
A Codesnippet is a part of a generated example, which only consists of code. Syntaxhighlighting is applied in the backend before it's sent back to the frontend.



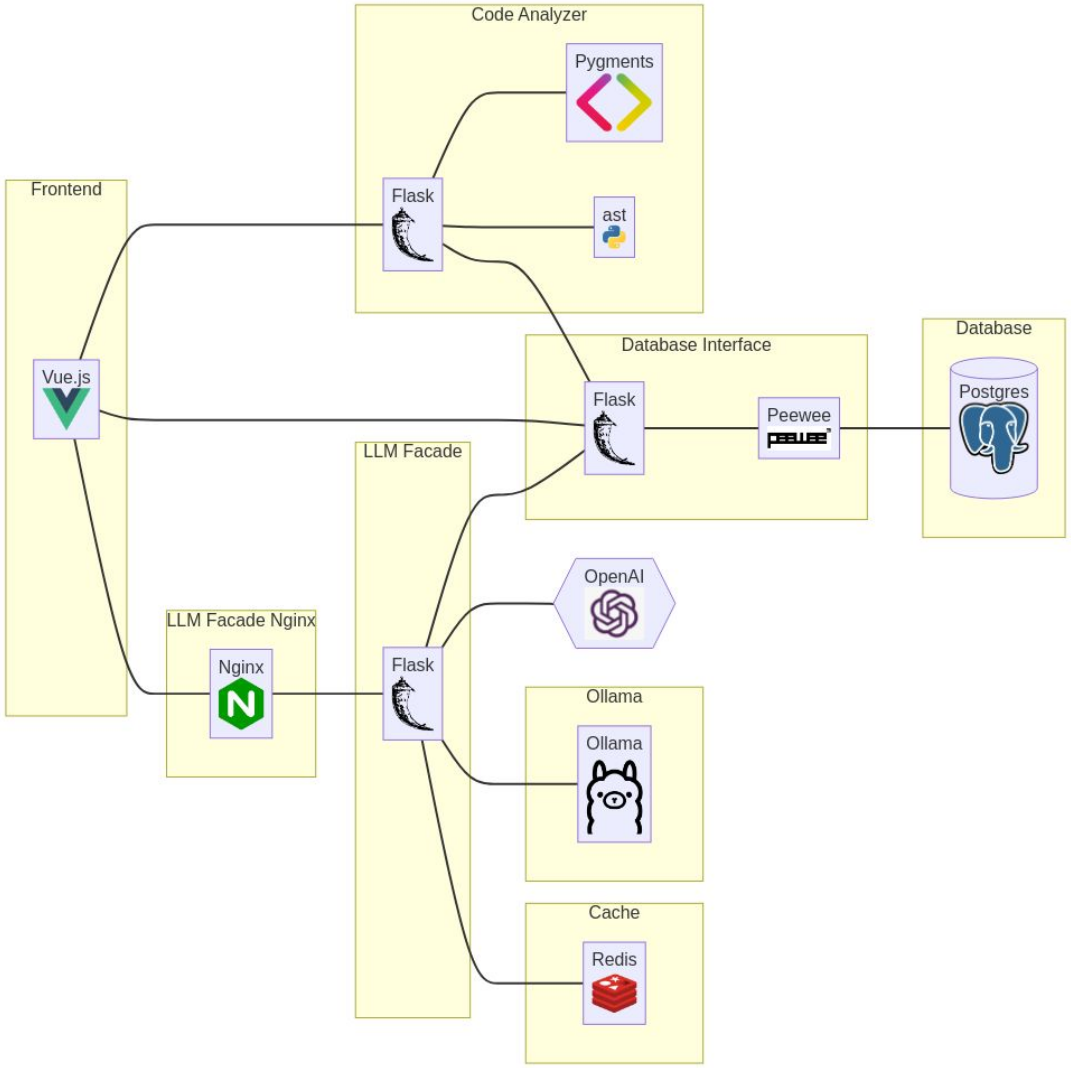
Architecture - Frontend

Vue.js

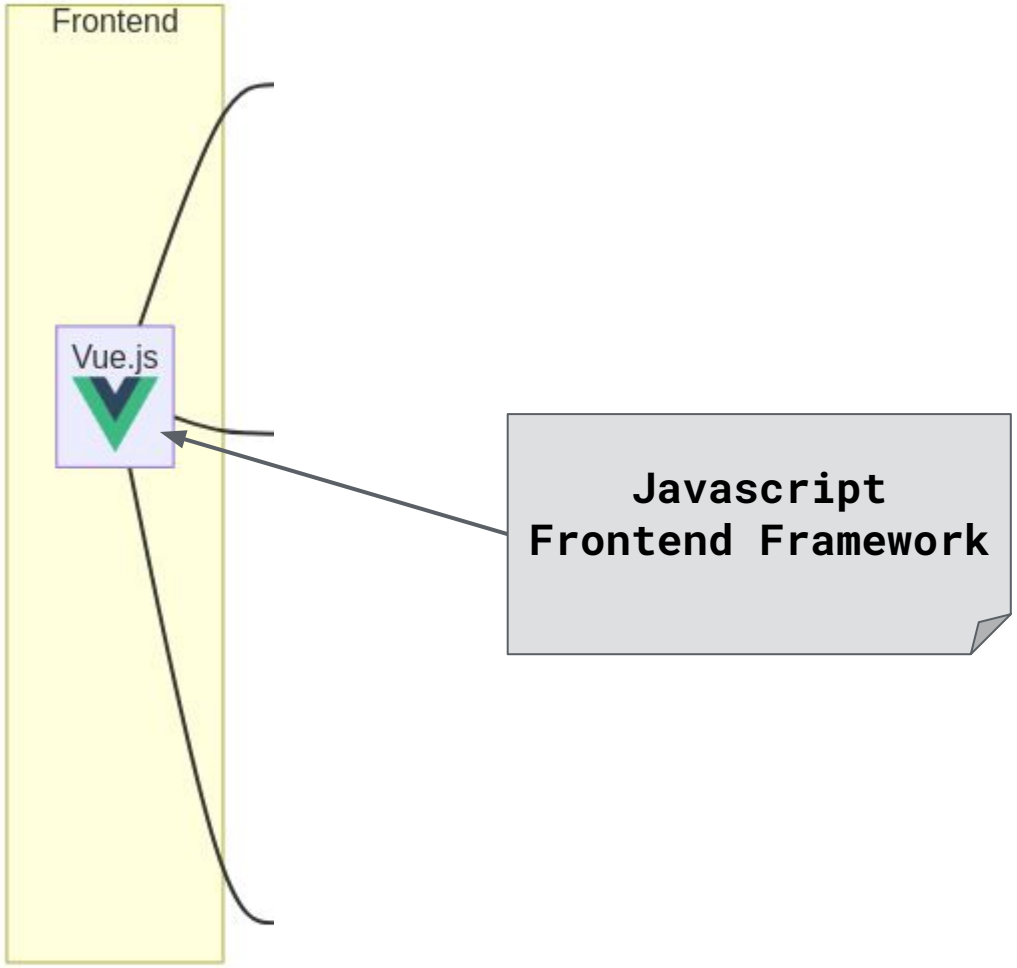
- Builds on top of standard HTML, CSS, and JavaScript
- Component-based programming model

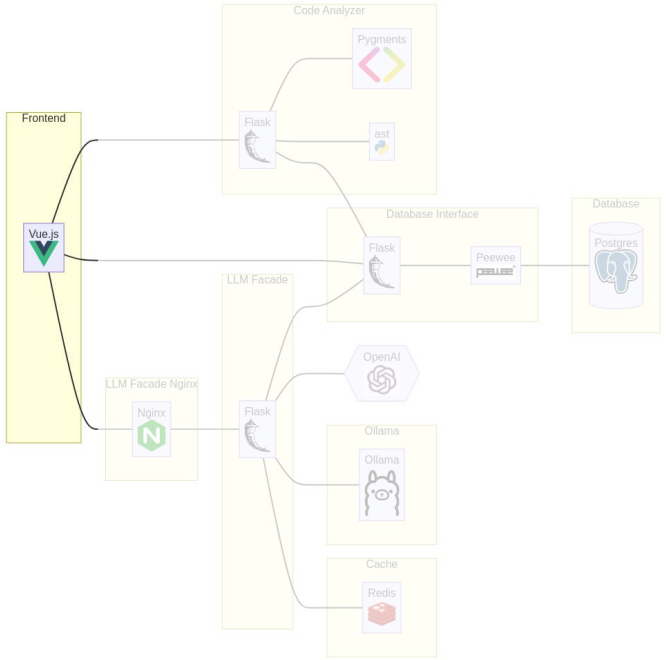


Architecture - Backend

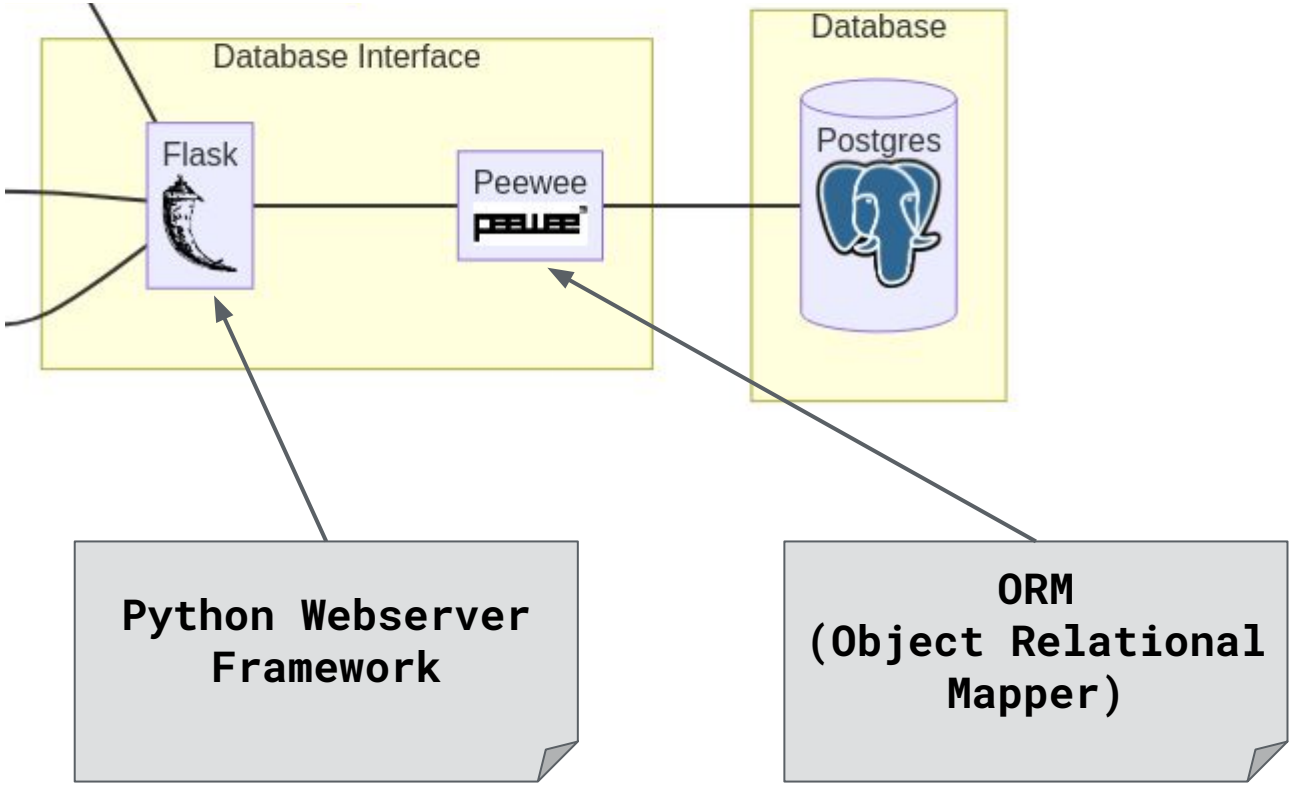


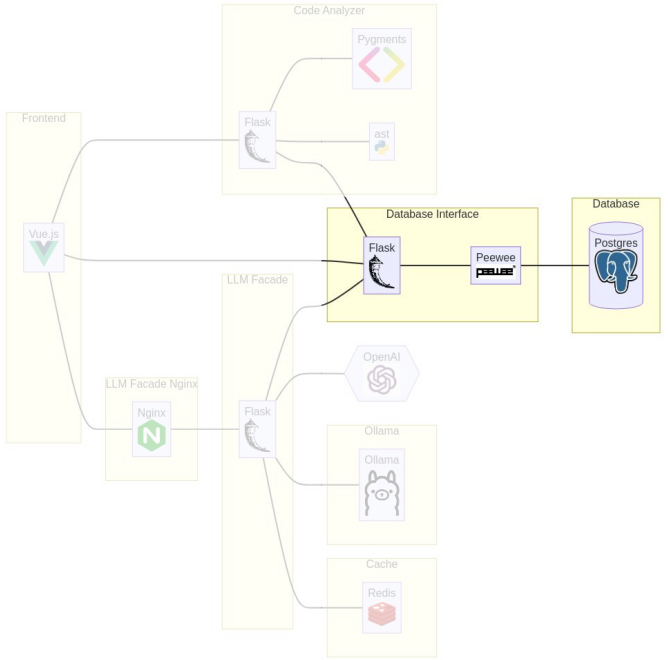
Architecture - Backend



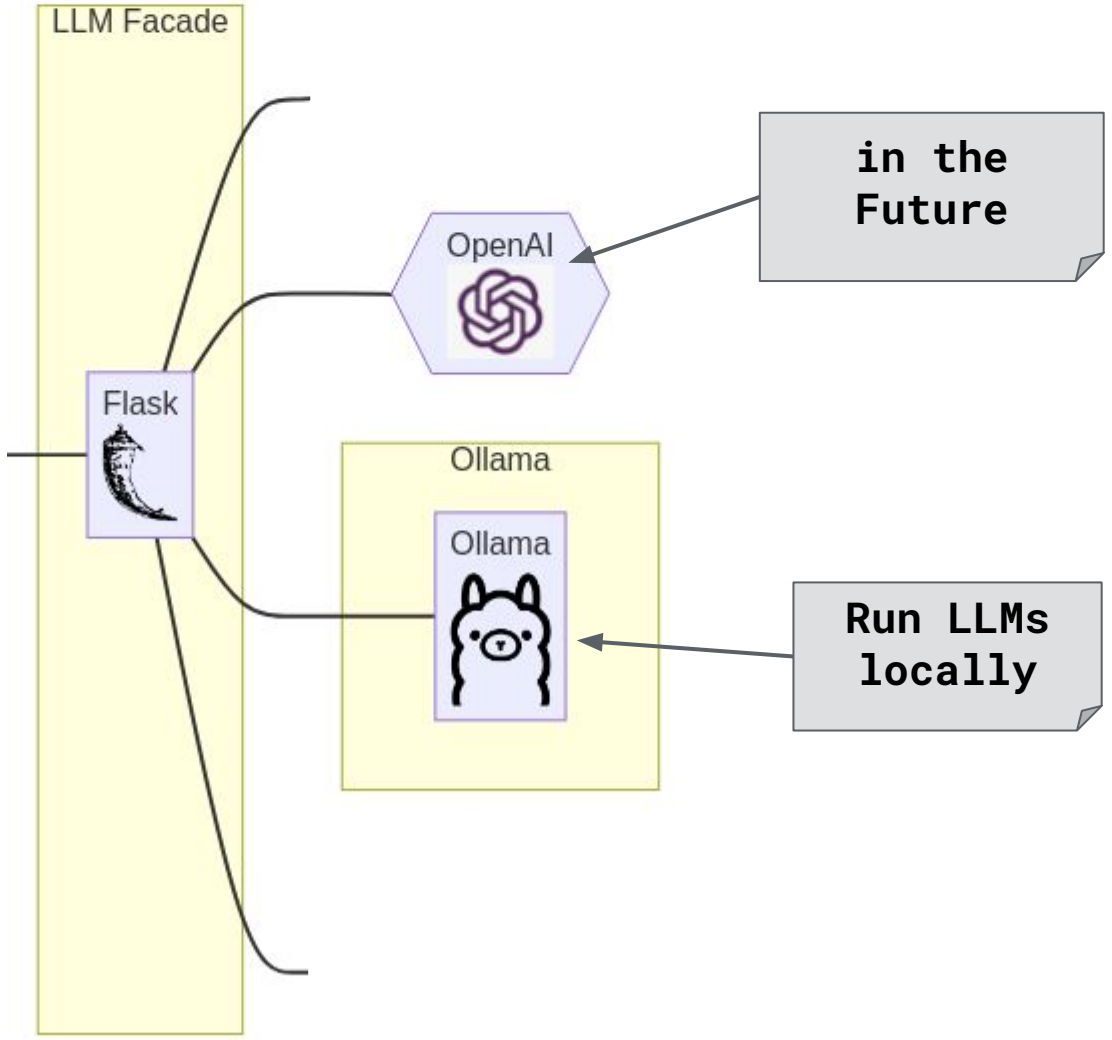


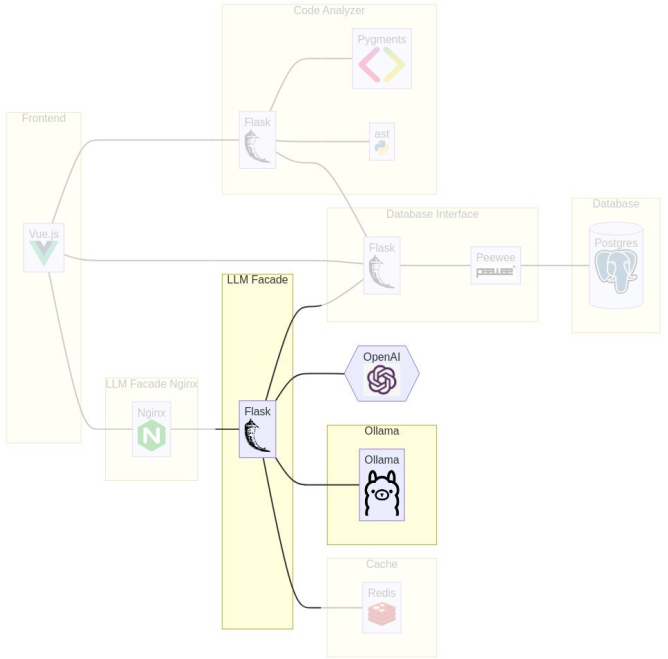
Architecture - Backend



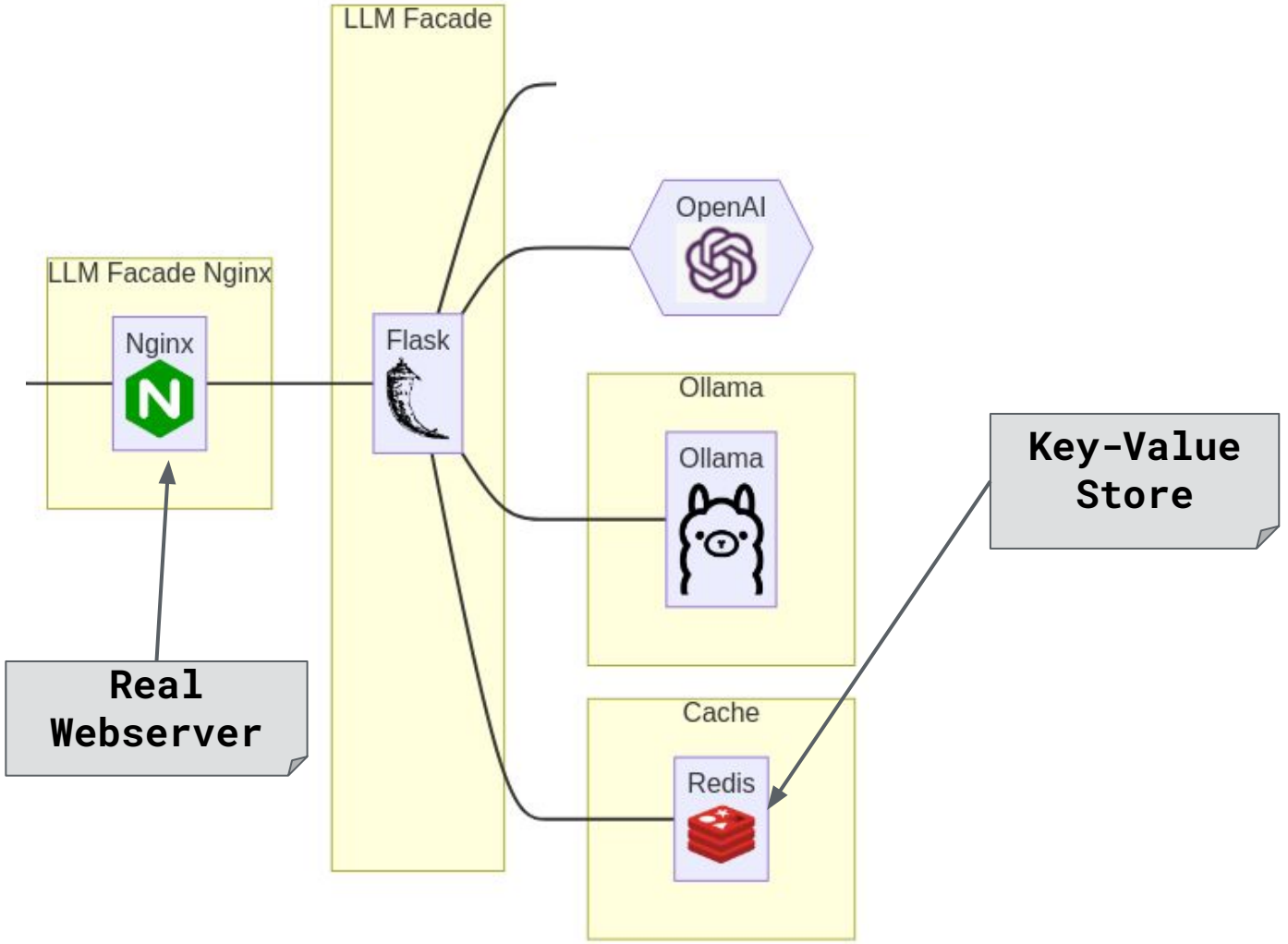


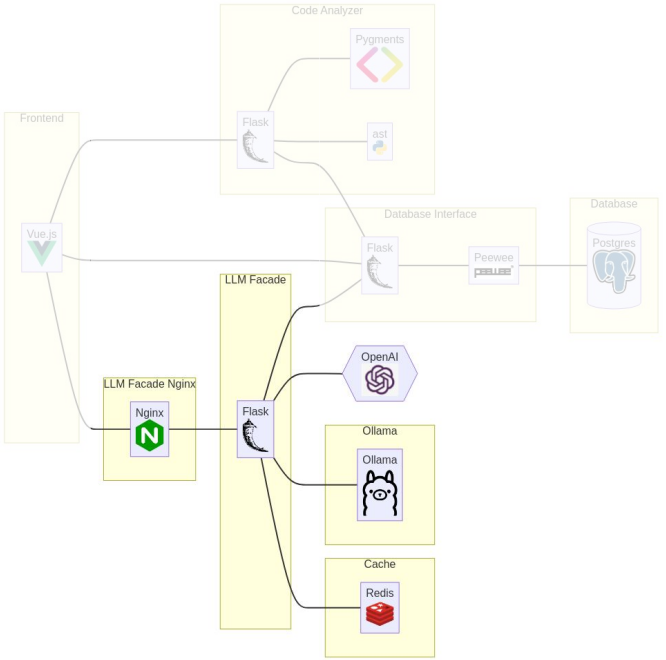
Architecture - Backend



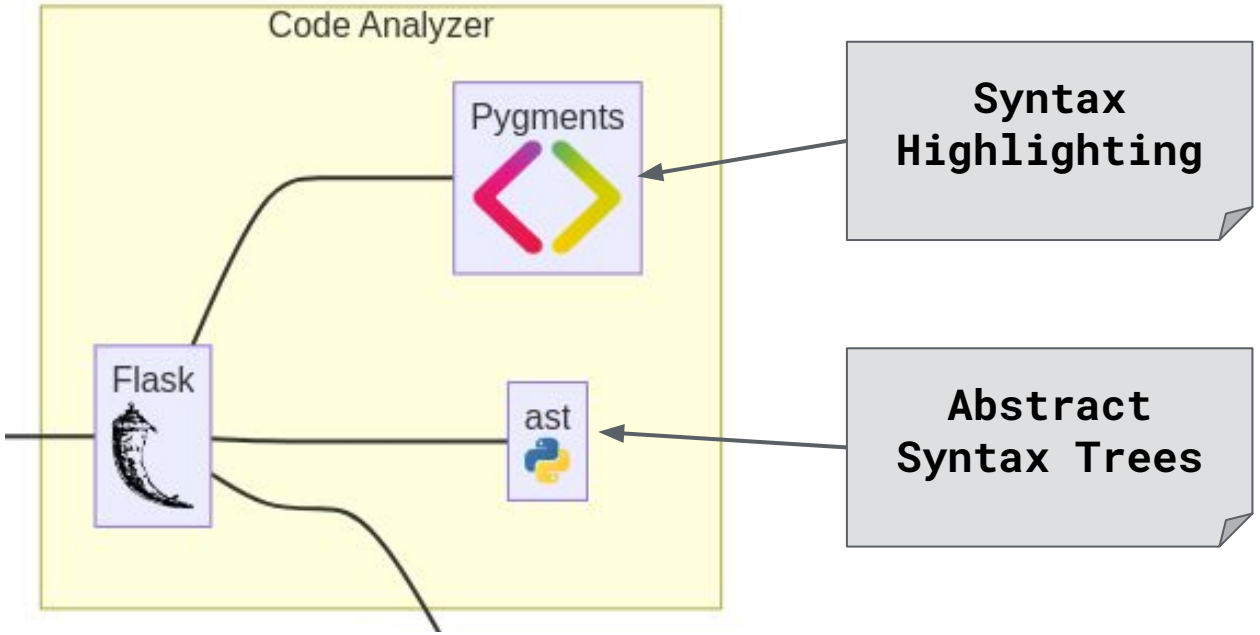


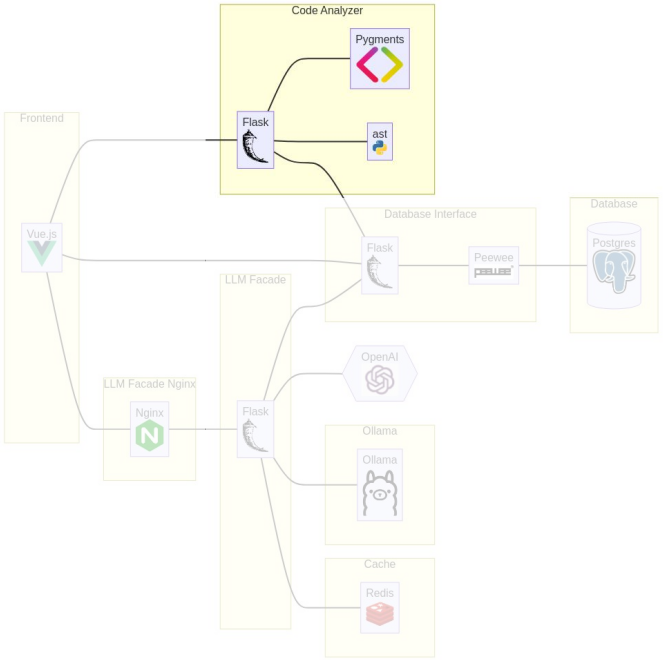
Architecture - Backend



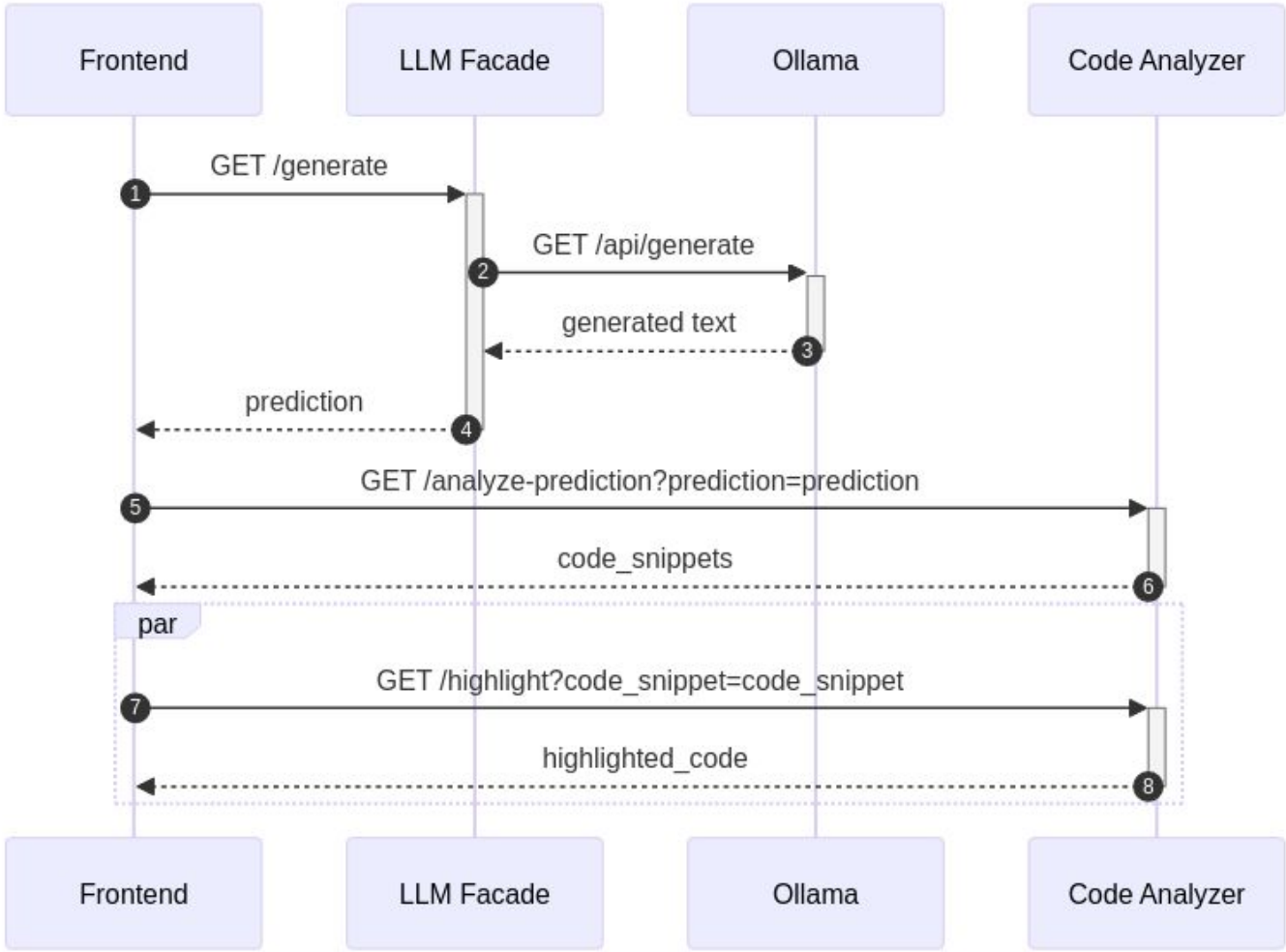


Architecture - Backend

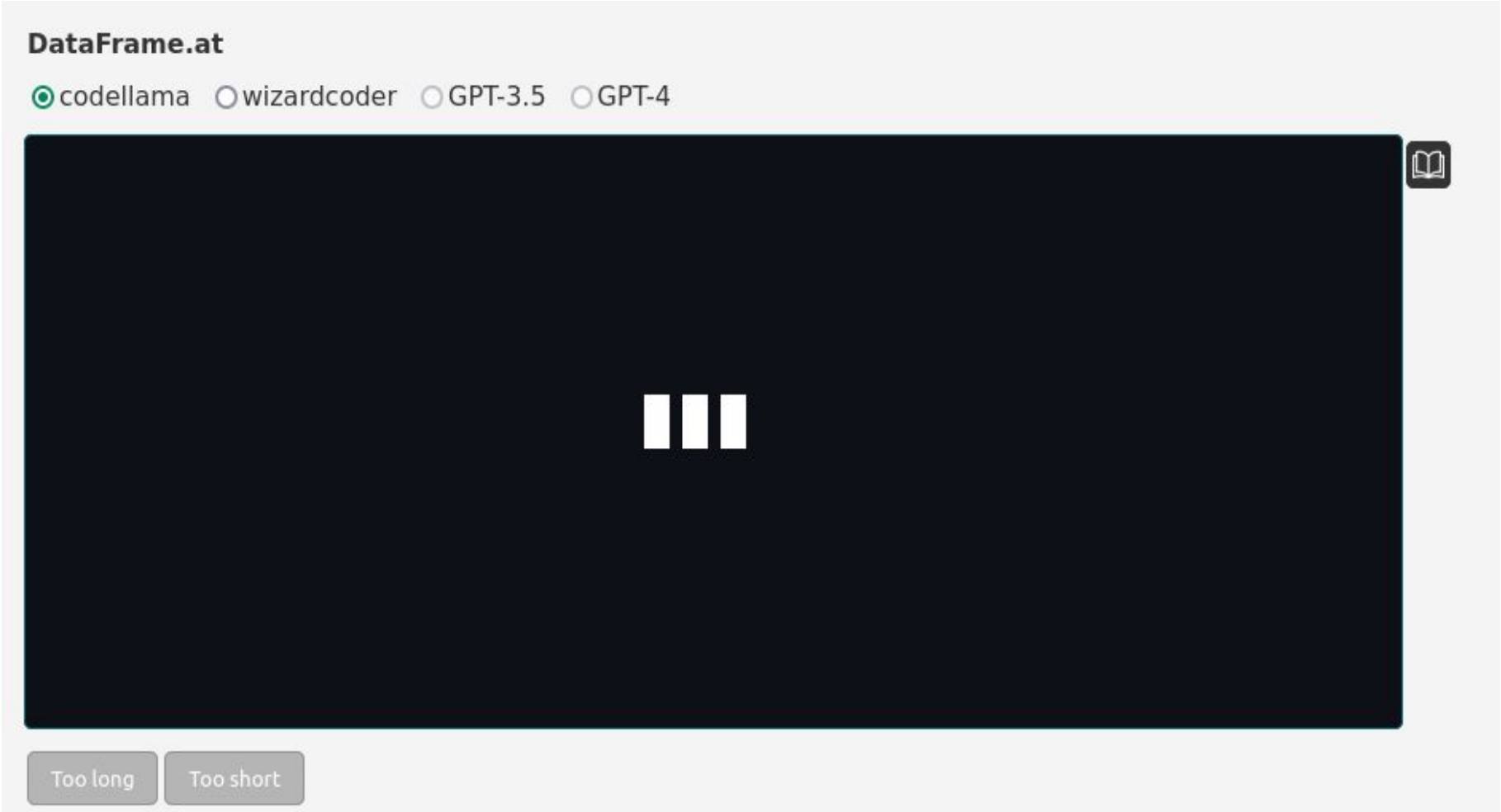




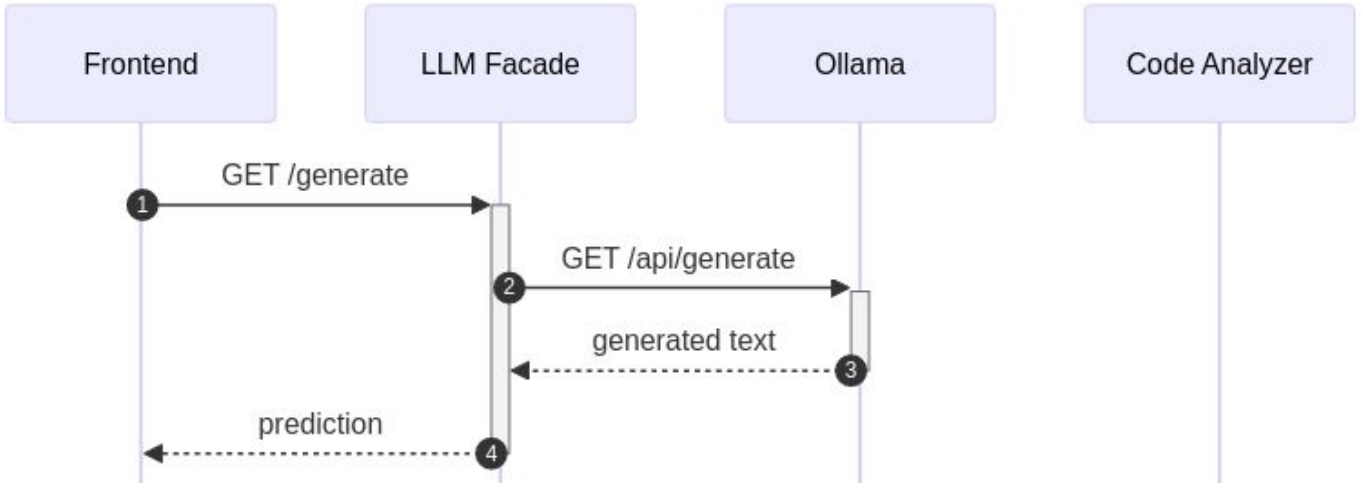
Example Generation

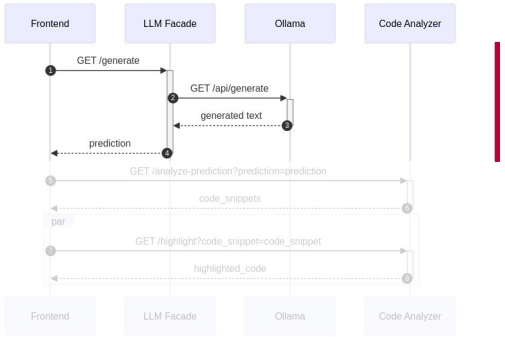


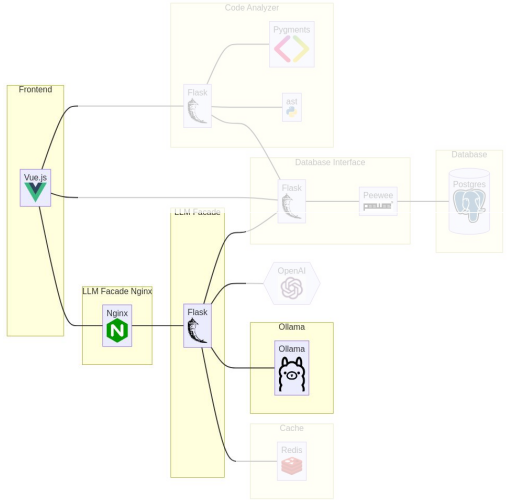
Example Generation



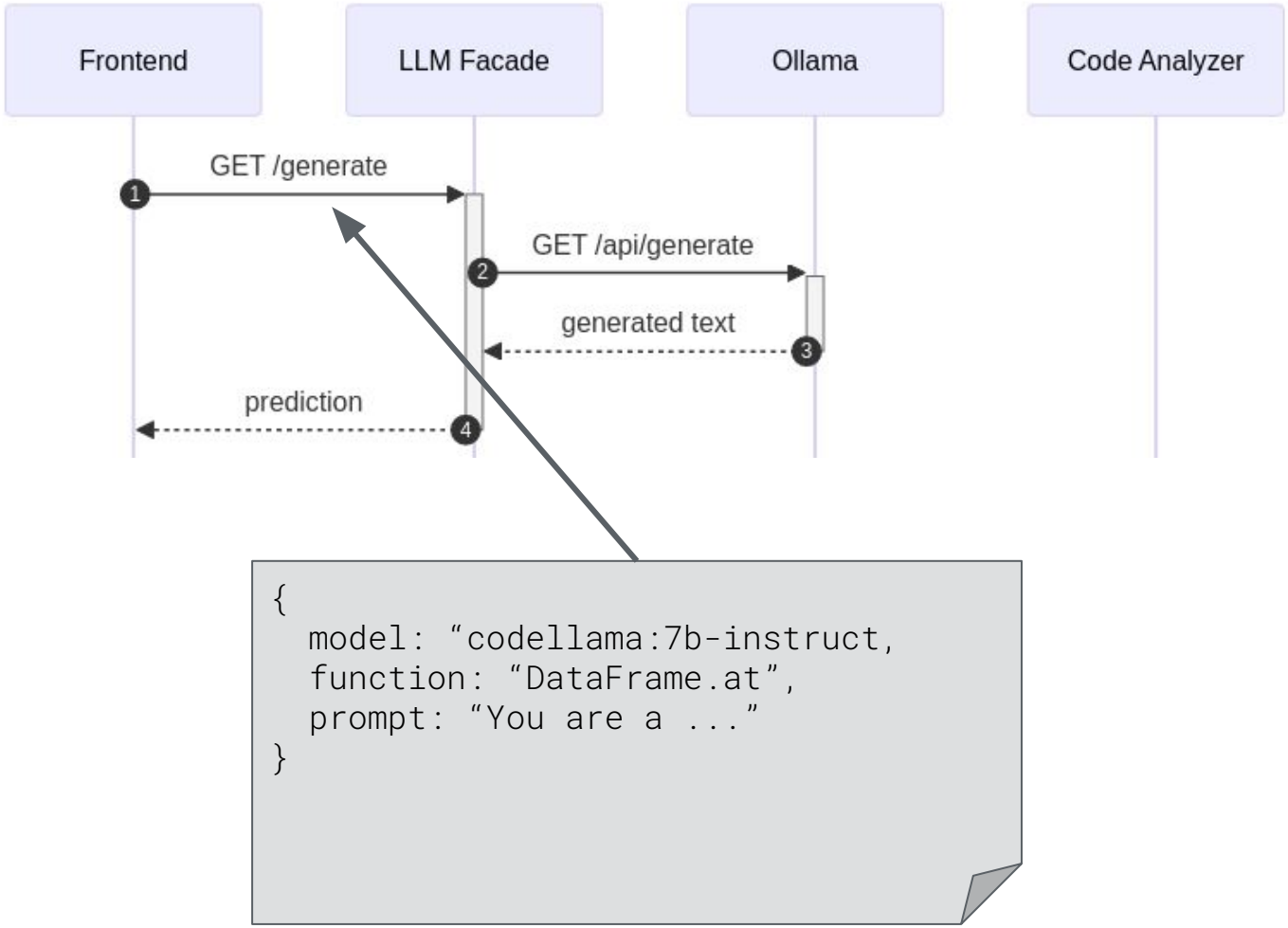
Example Generation

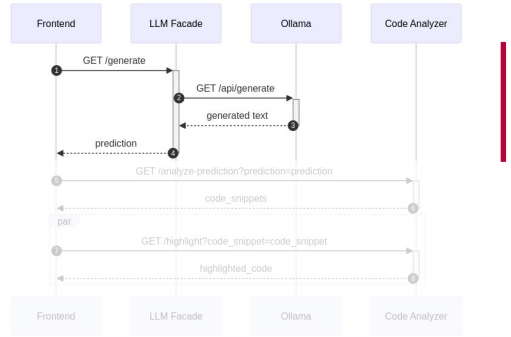


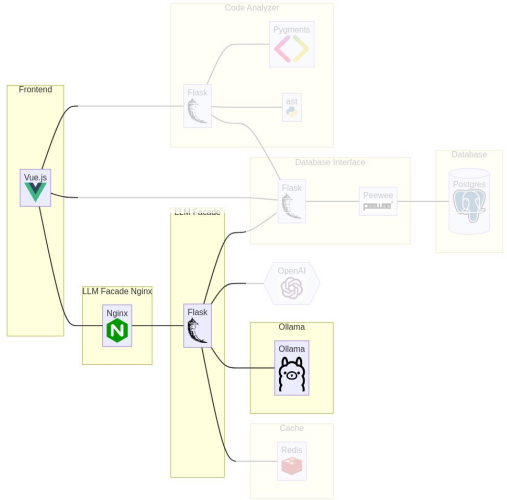




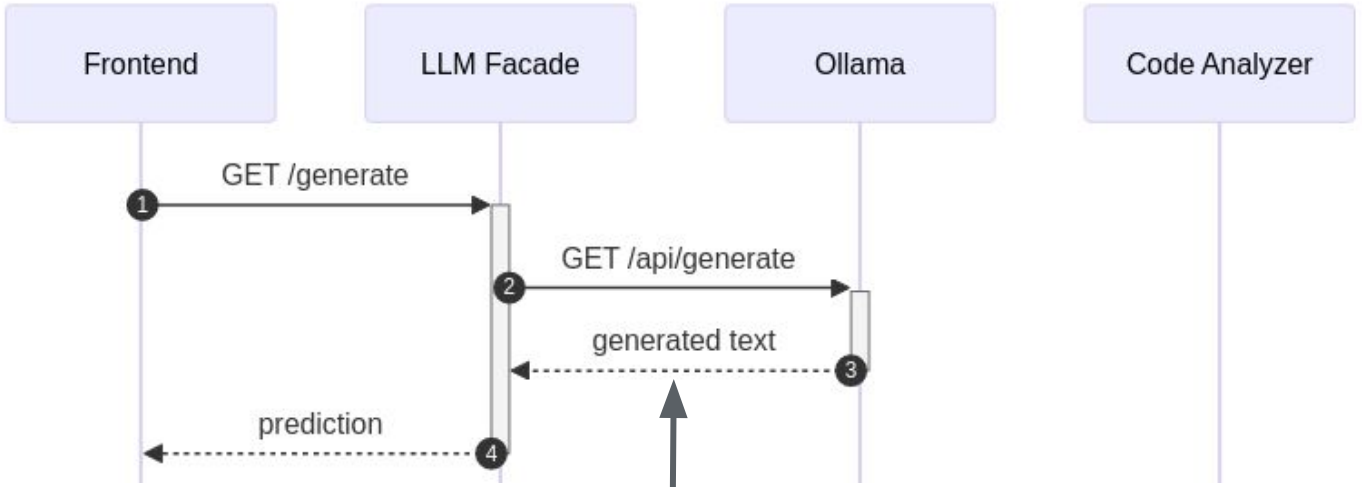
Example Generation



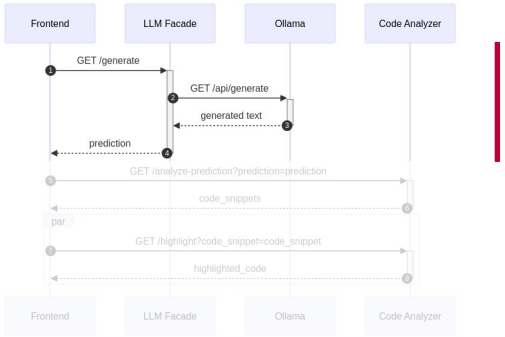


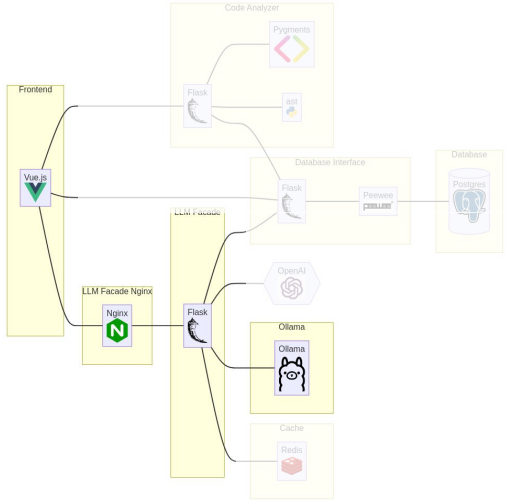


Example Generation

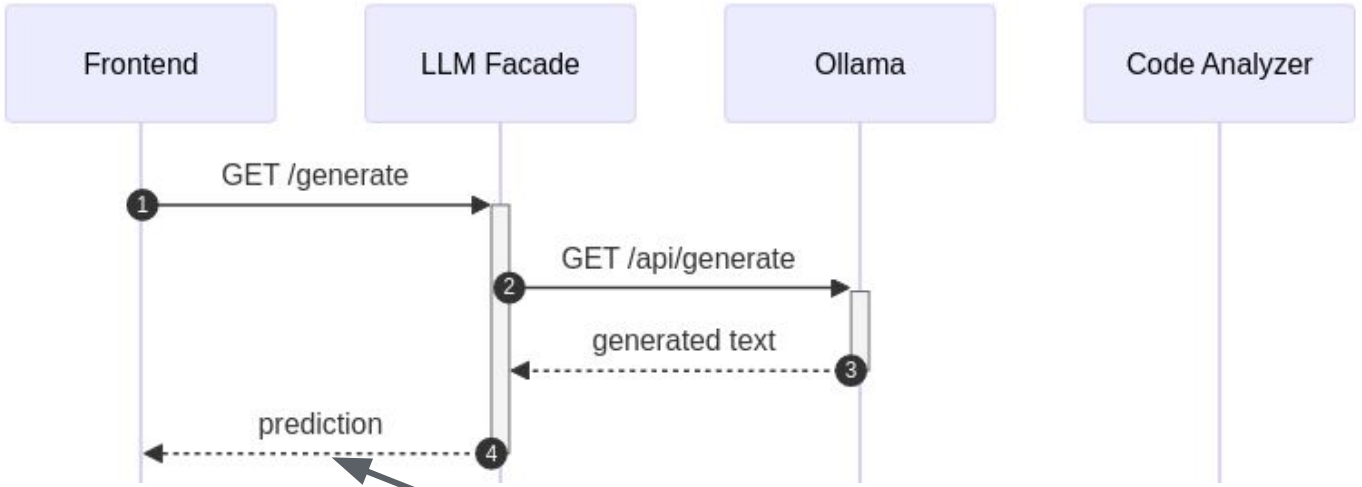



```
import pandas as pd

# create a sample DataFrame
df = pd.DataFrame({'A': [0, 2, 3],
'B': [4, 1, 6], 'C': [3, 2, 1]},
index=[4, 5, 6])

...
```

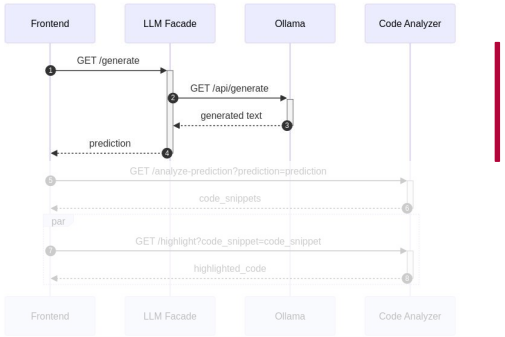


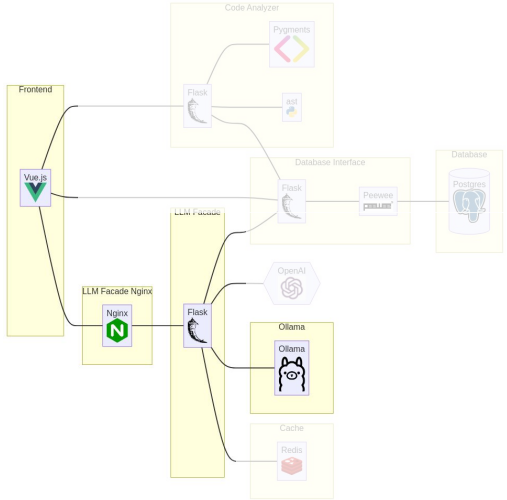


Example Generation



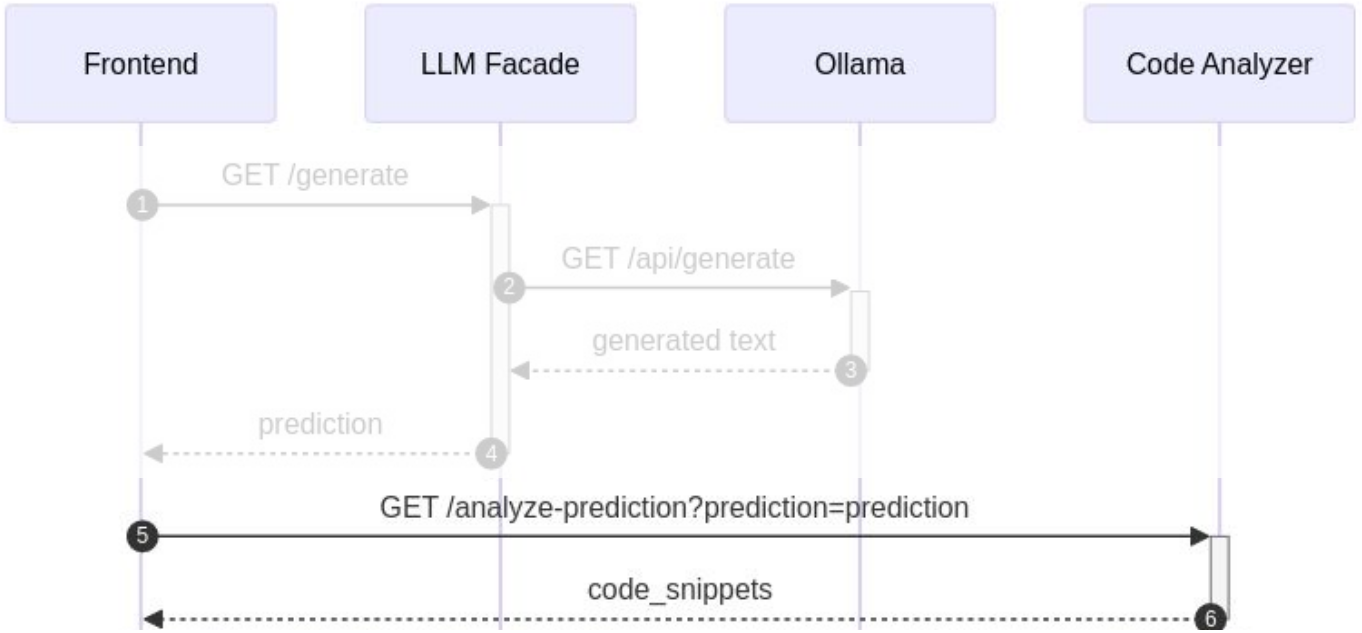

```
{
  text: “import pandas as pd ...”,
  model: “codellama:7b-instruct”,
  token_amount: 456
}
```

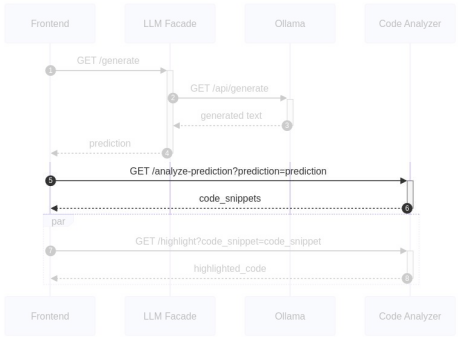


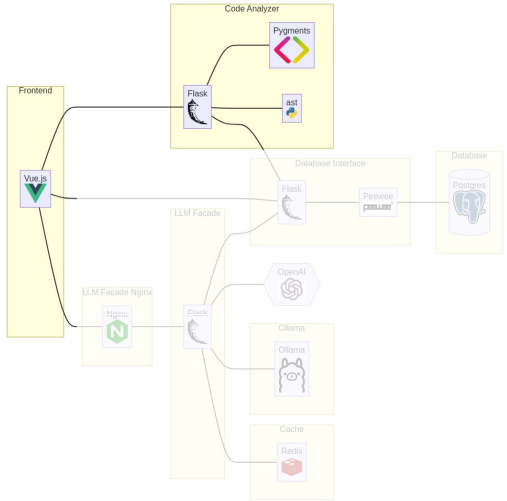


Example Generation

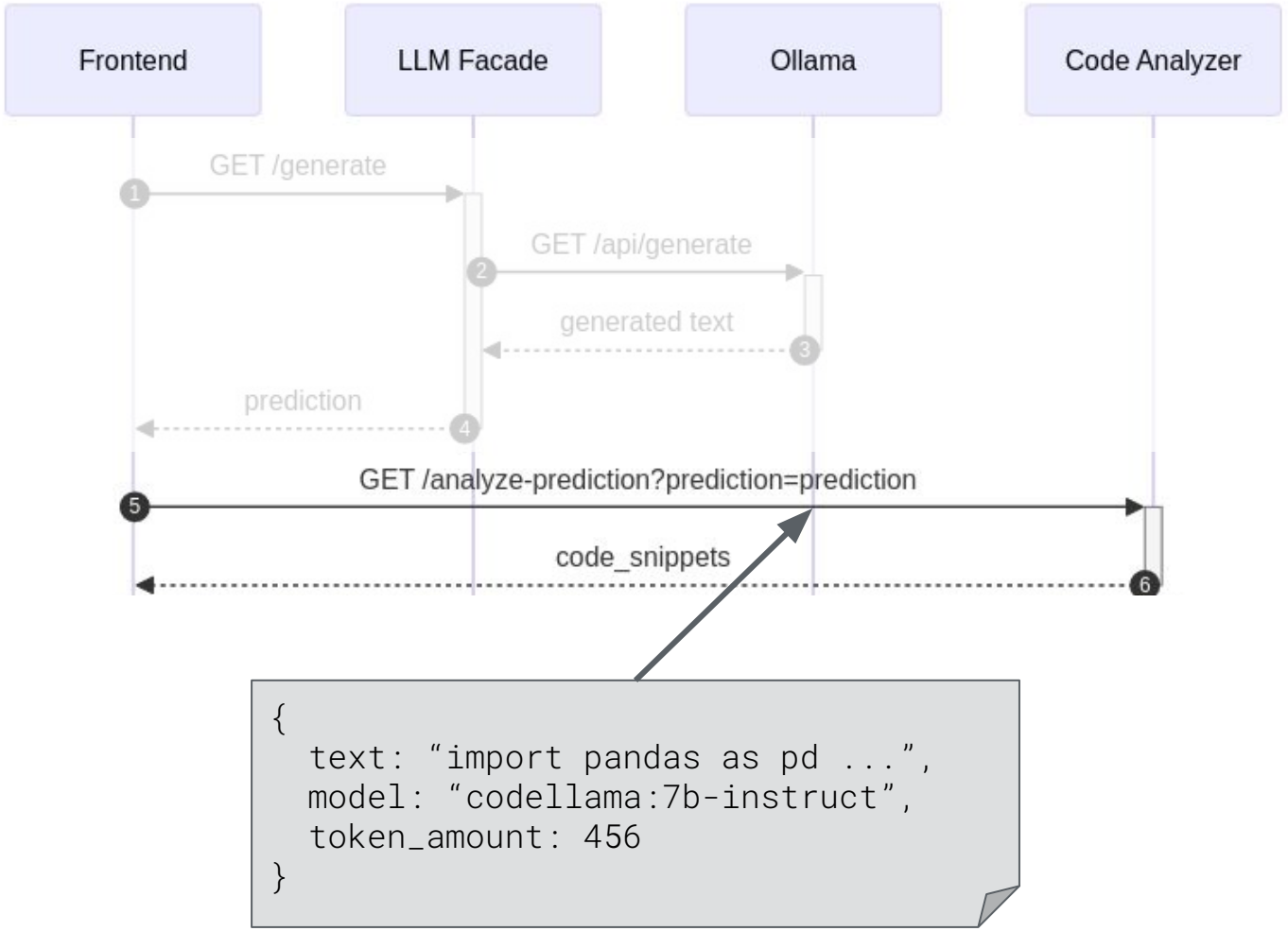
Example Generation

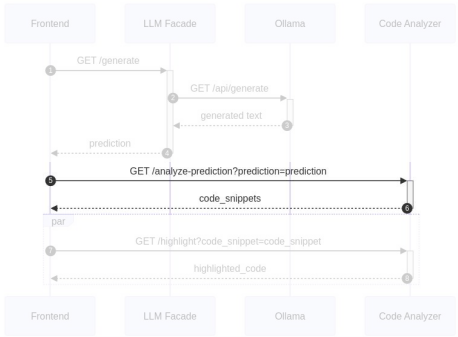


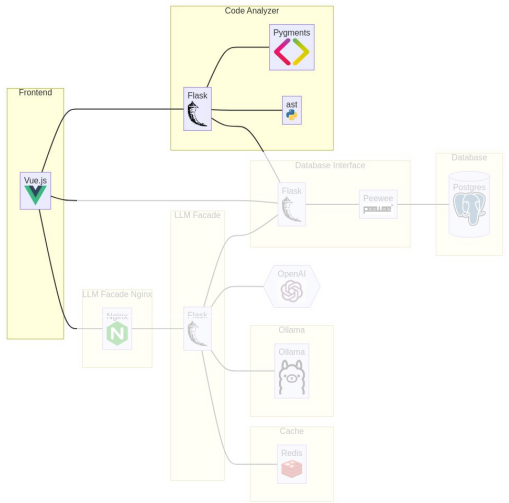




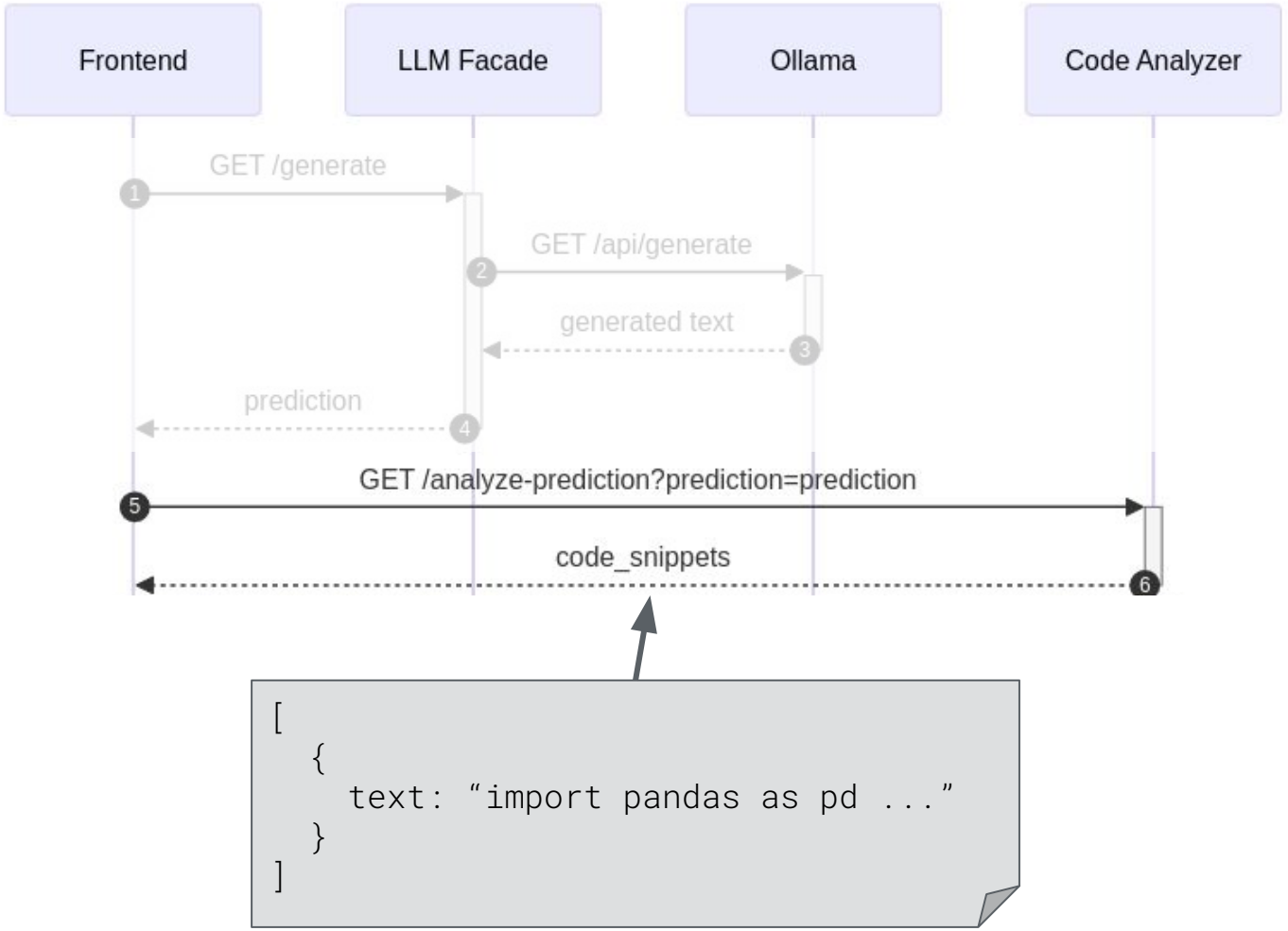
Example Generation

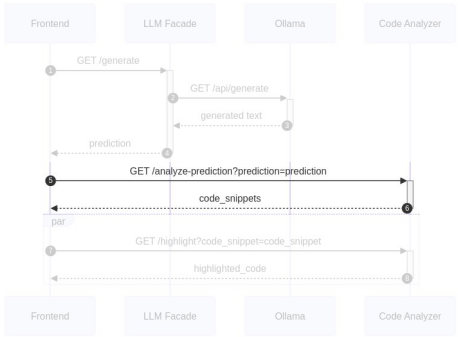


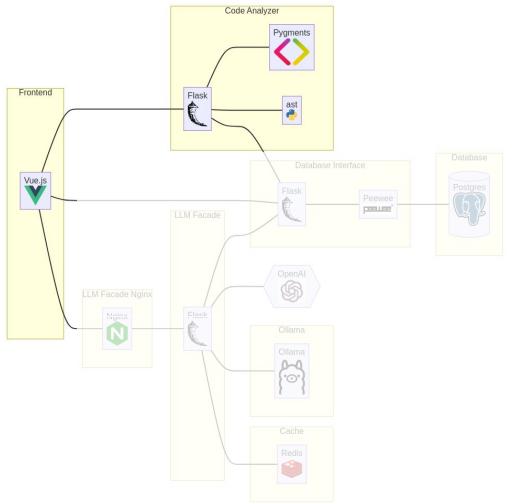




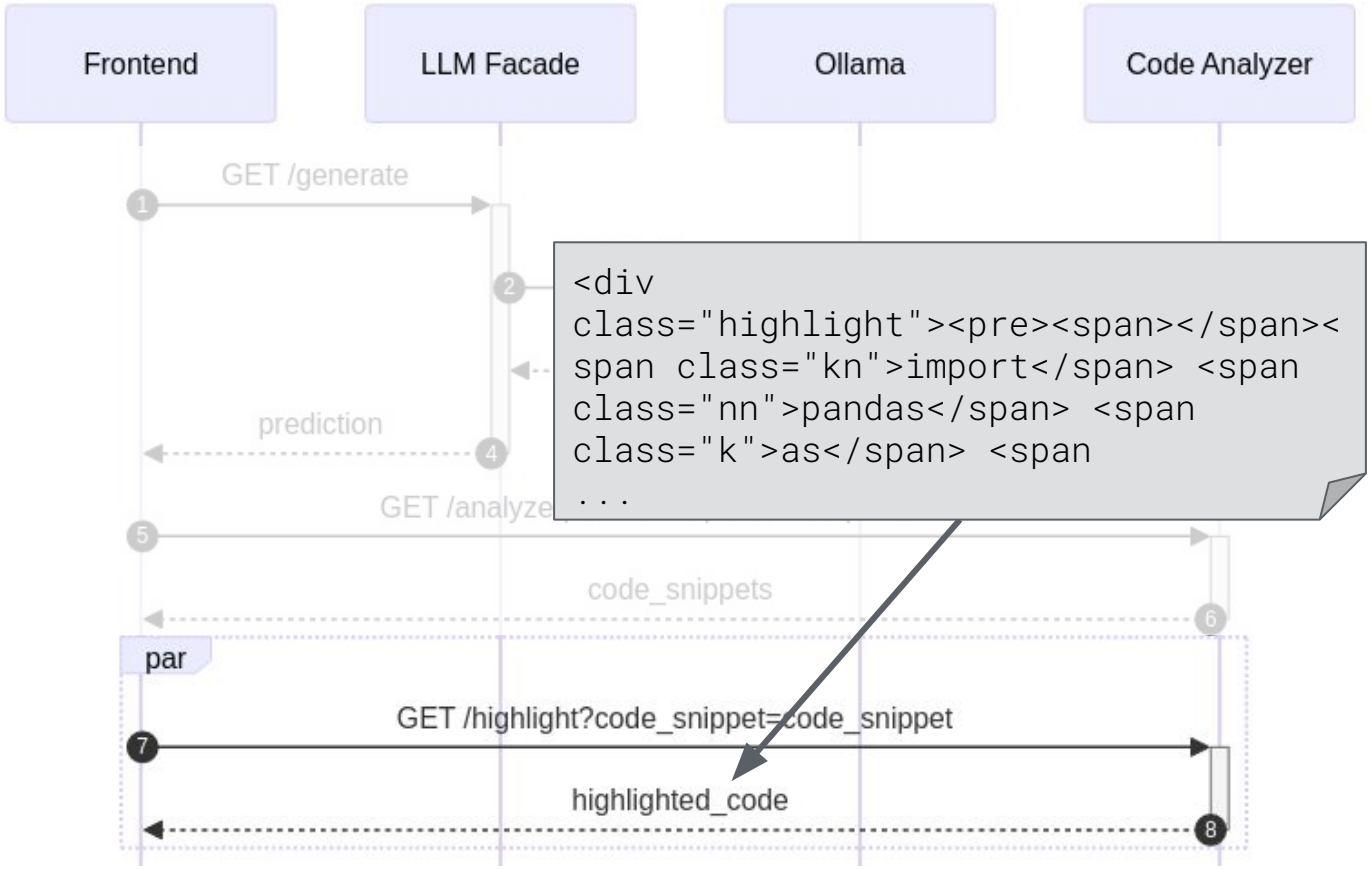
Example Generation

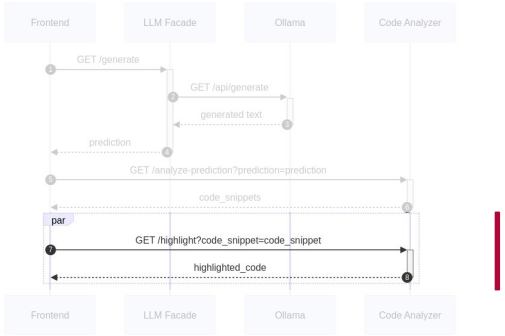


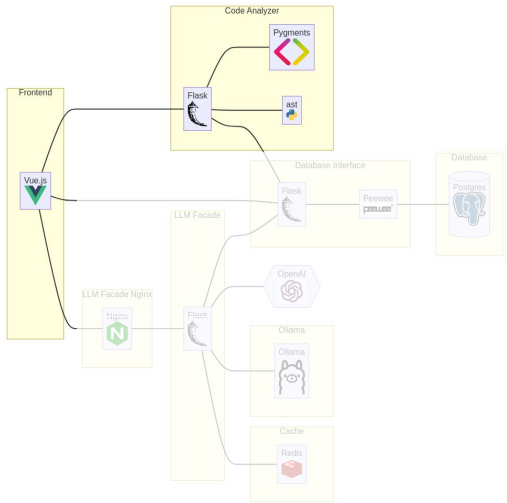




Example Generation







Example Generation

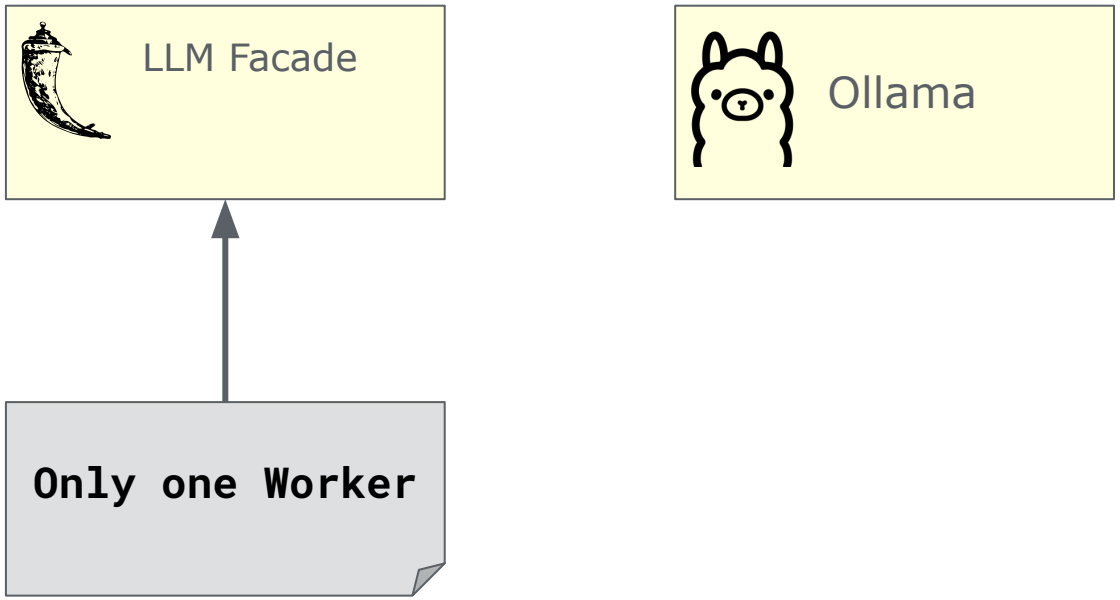
Simultaneous Example Generation

LLM Facade

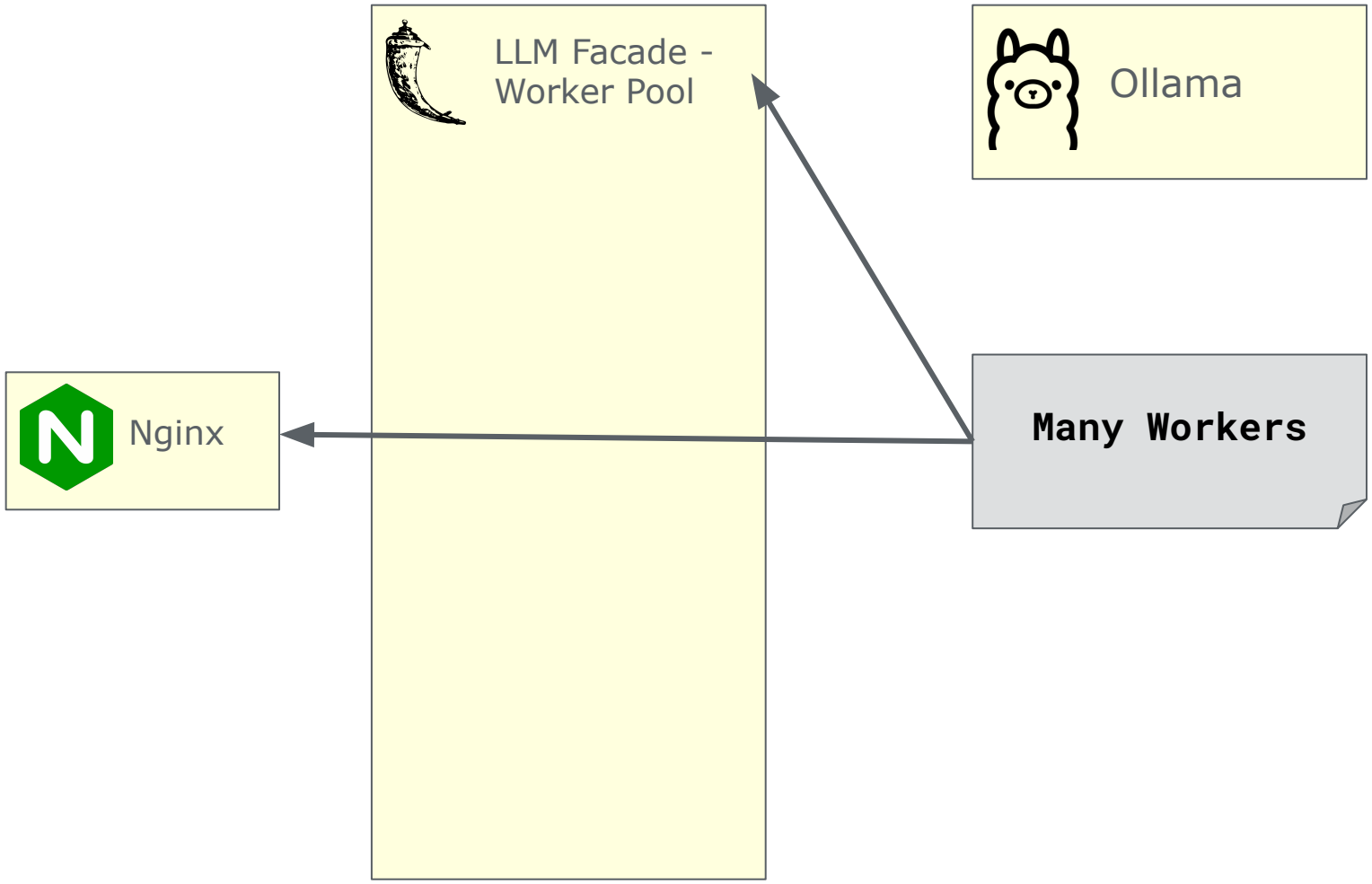
Ollama

Can it handle multiple requests?

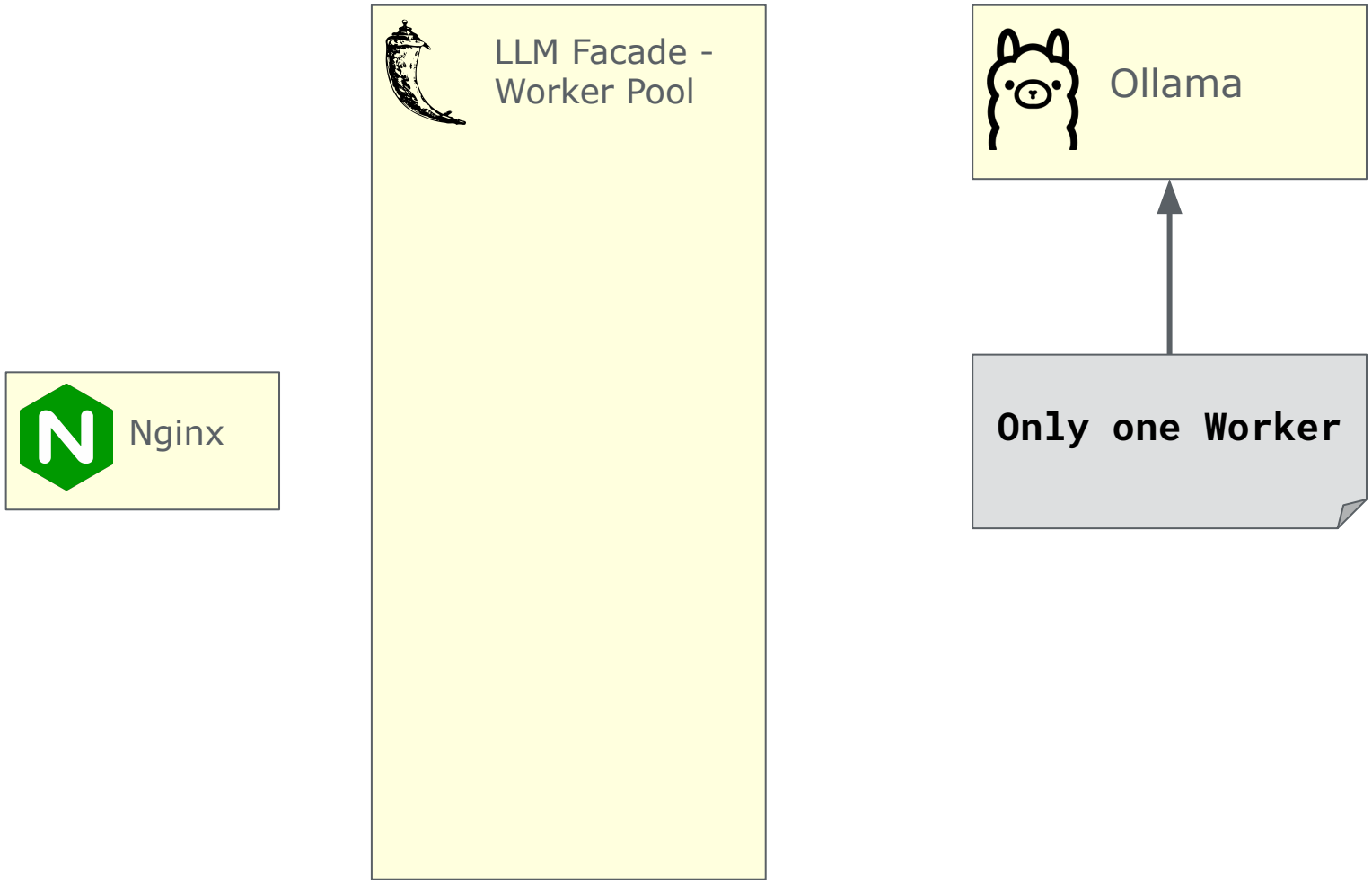
Simultaneous Example Generation



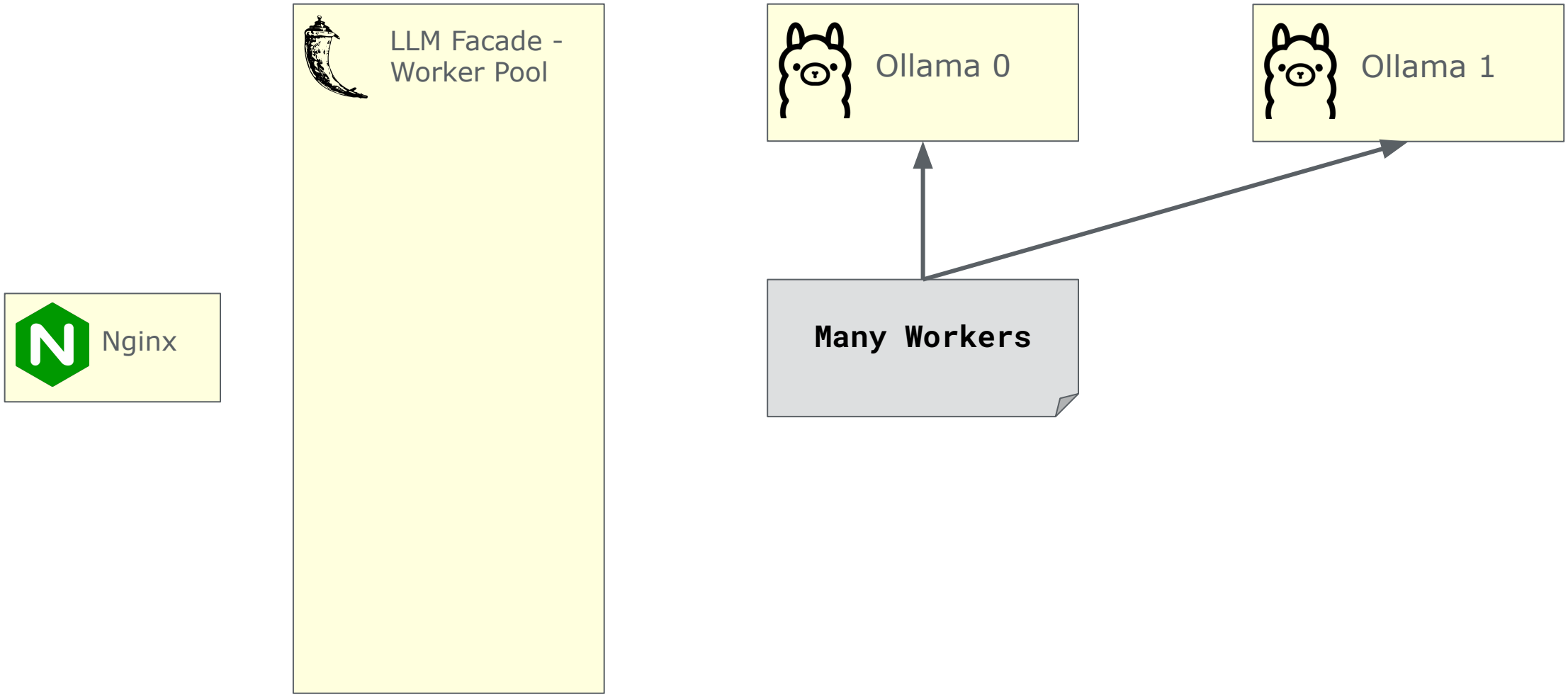
Simultaneous Example Generation



Simultaneous Example Generation



Simultaneous Example Generation



Simultaneous Example Generation

LLM Facade - Worker Pool

Ollama 0

Ollama 1

Nginx

Scheduling?

Simultaneous Example Generation

LLM Facade - Worker Pool

Ollama 0

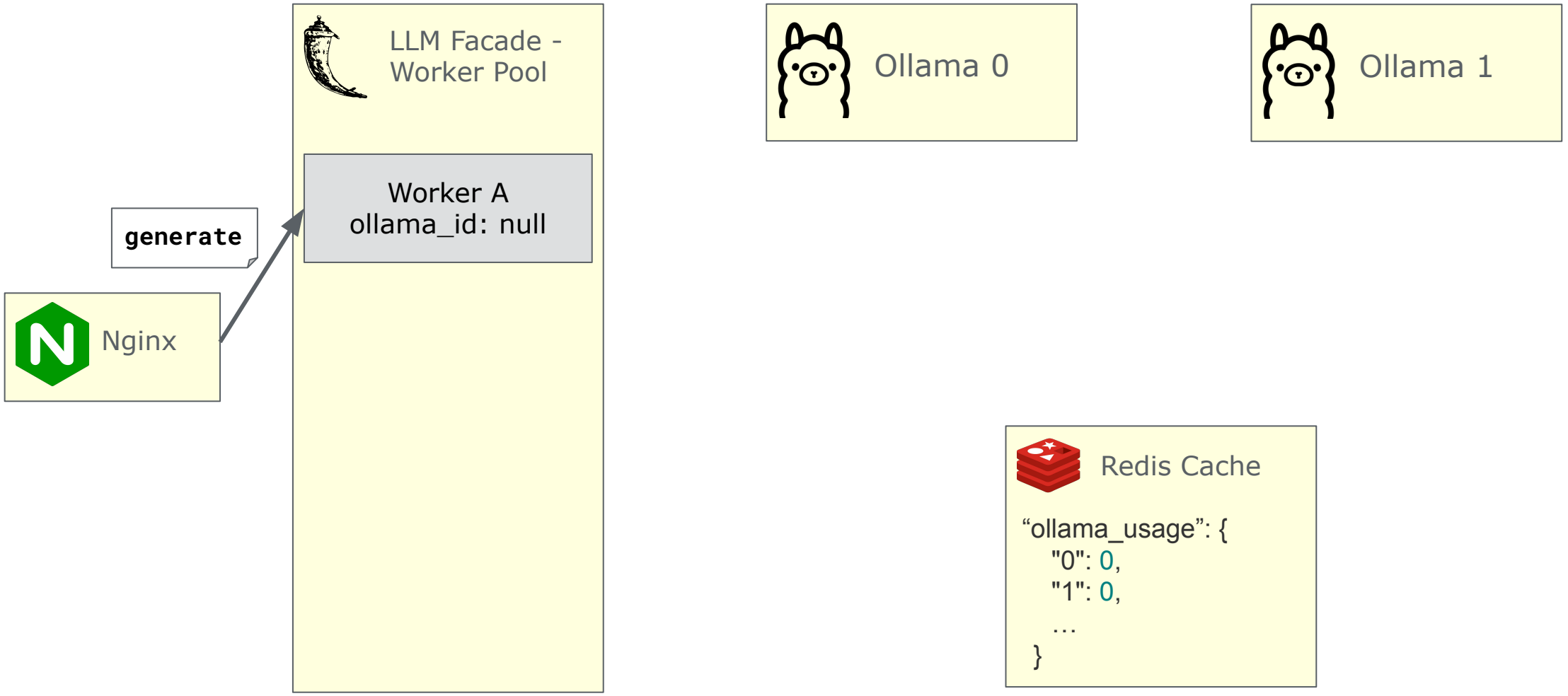
Ollama 1

Nginx

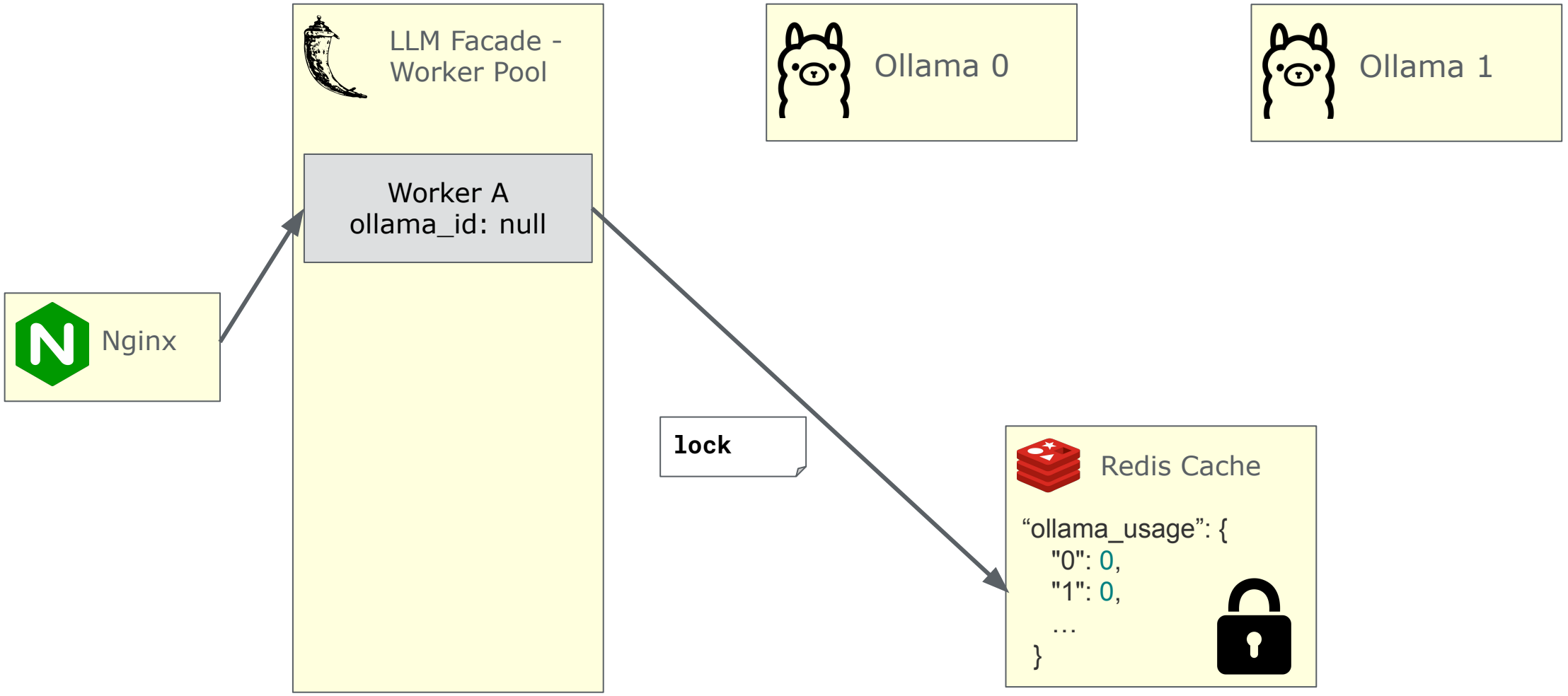
Redis Cache

```
“ollama_usage”: {
   "0": 0,
   "1": 0,
   …
 }
```

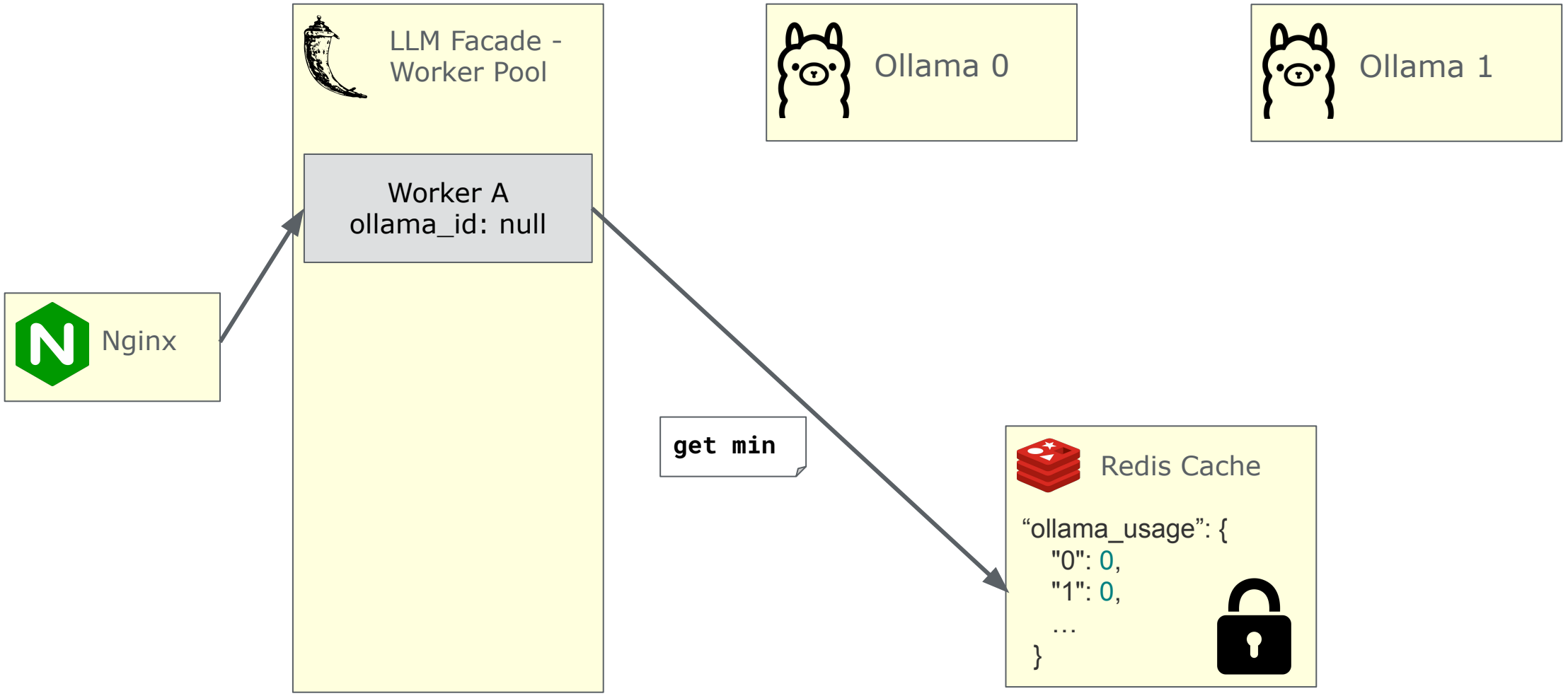
Simultaneous Example Generation



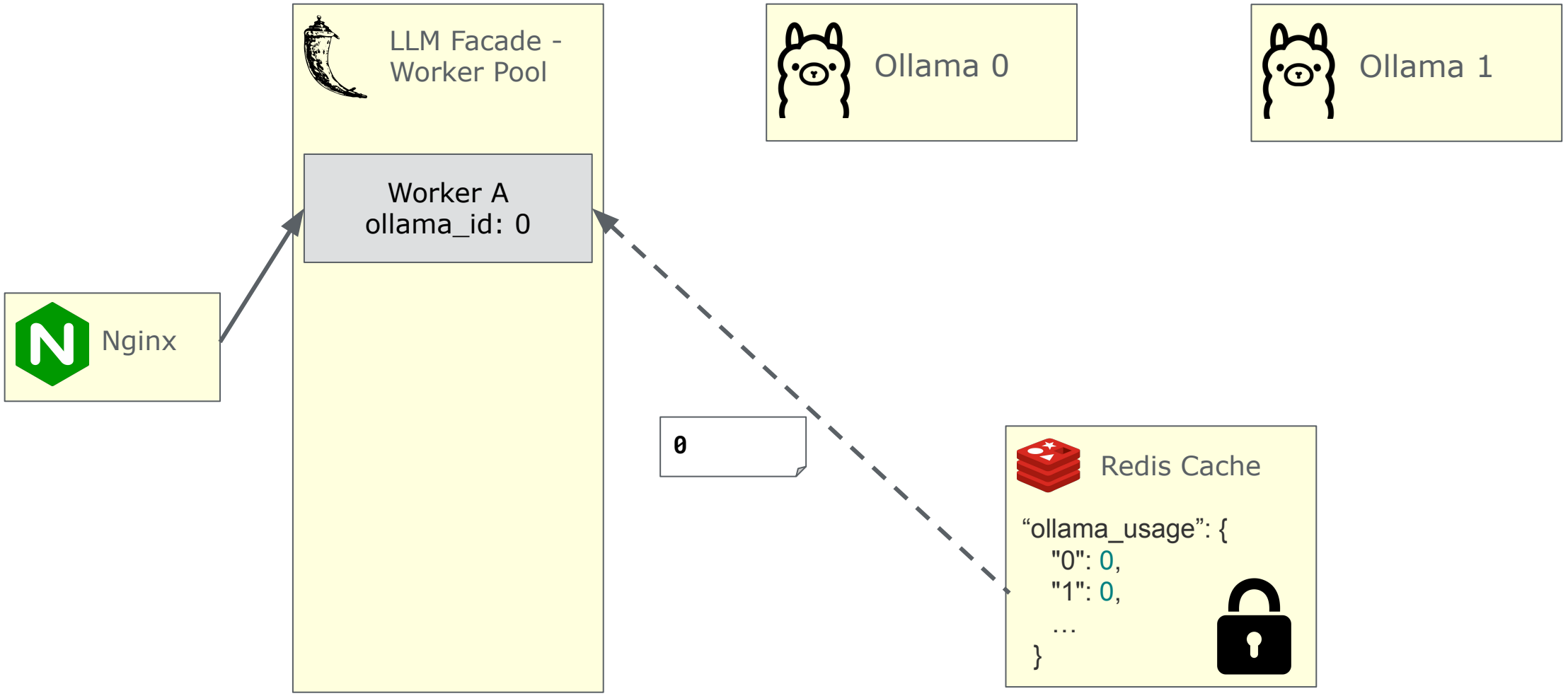
Simultaneous Example Generation



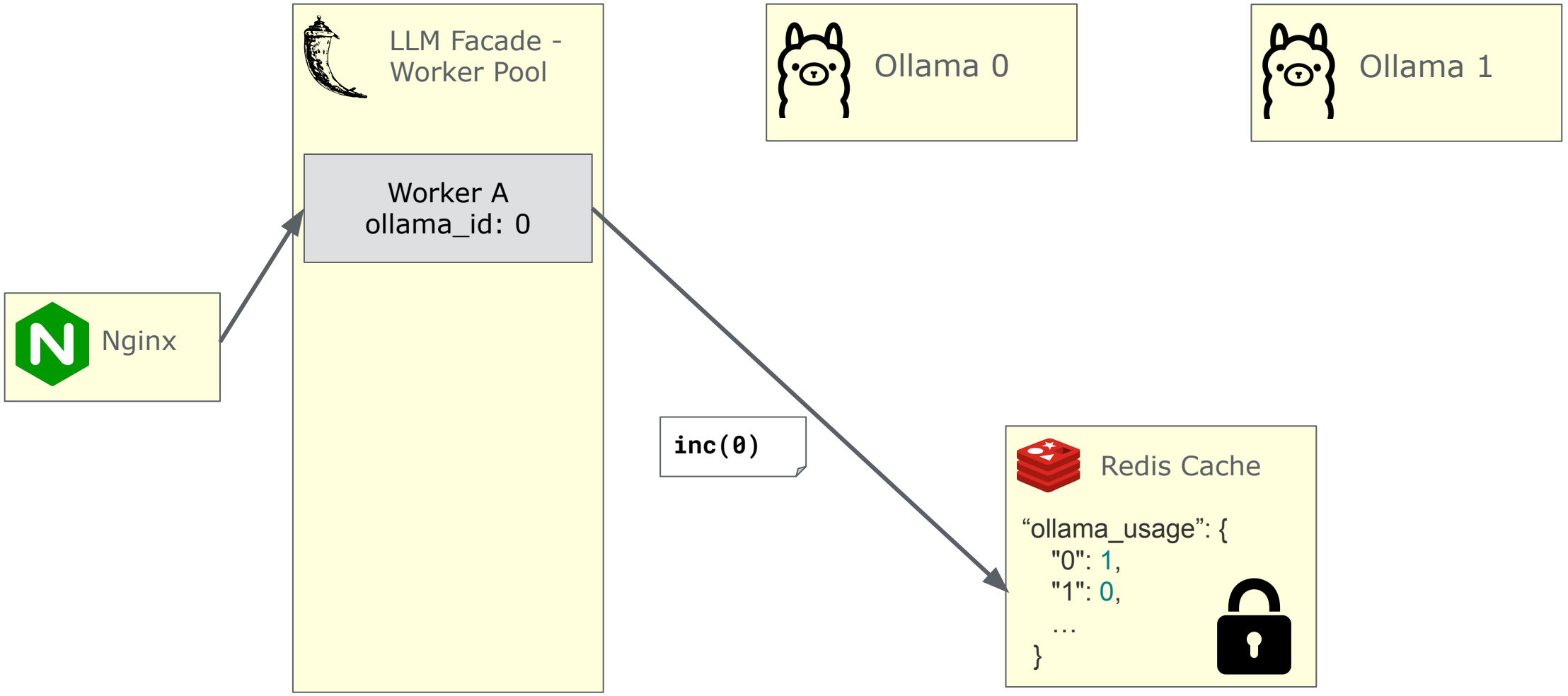
Simultaneous Example Generation



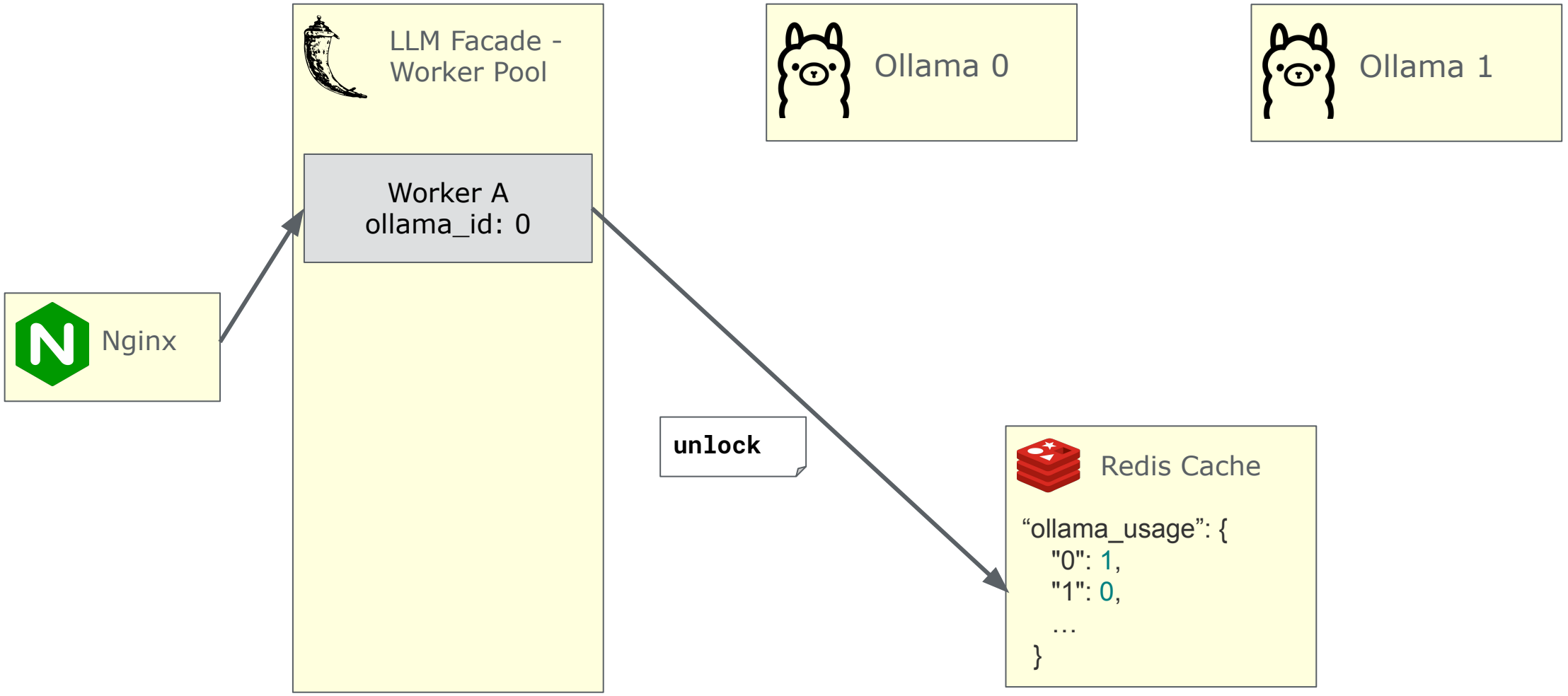
Simultaneous Example Generation



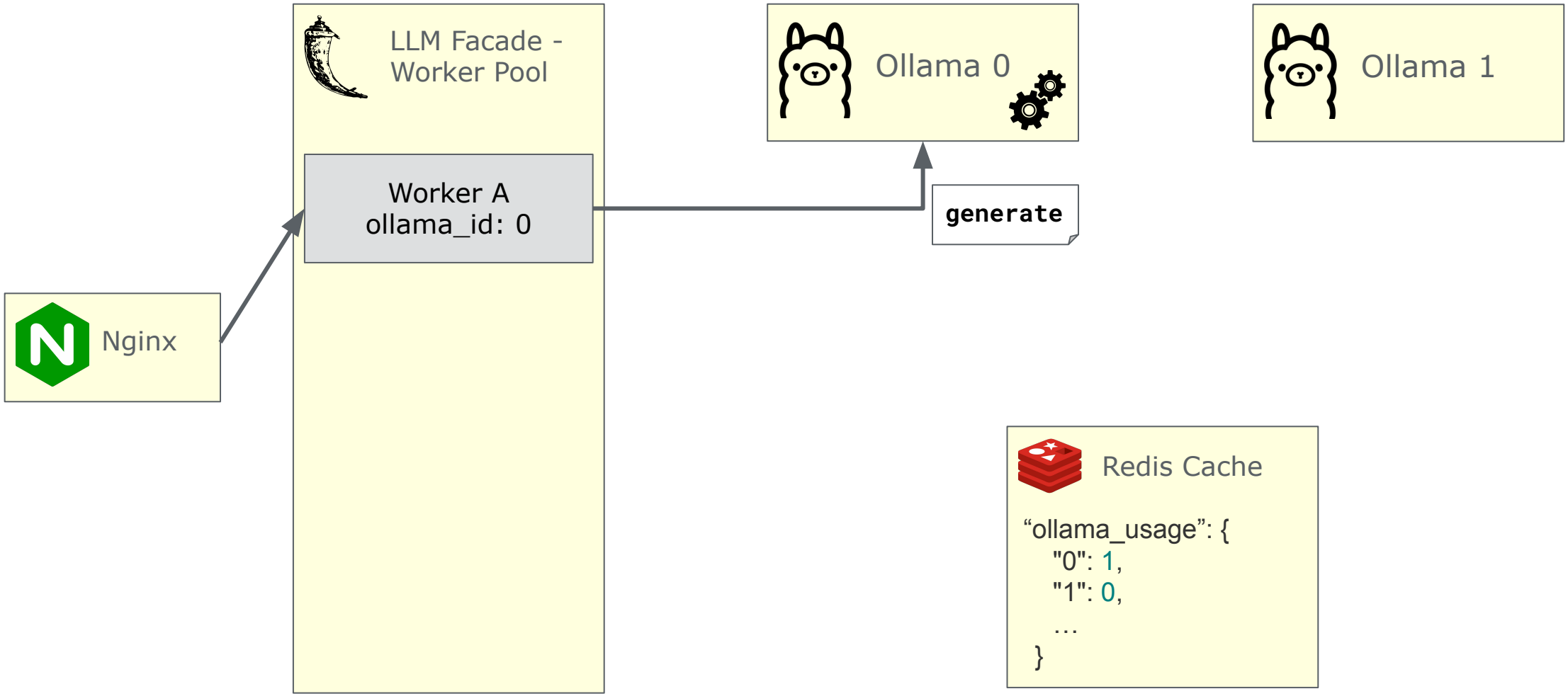
Simultaneous Example Generation



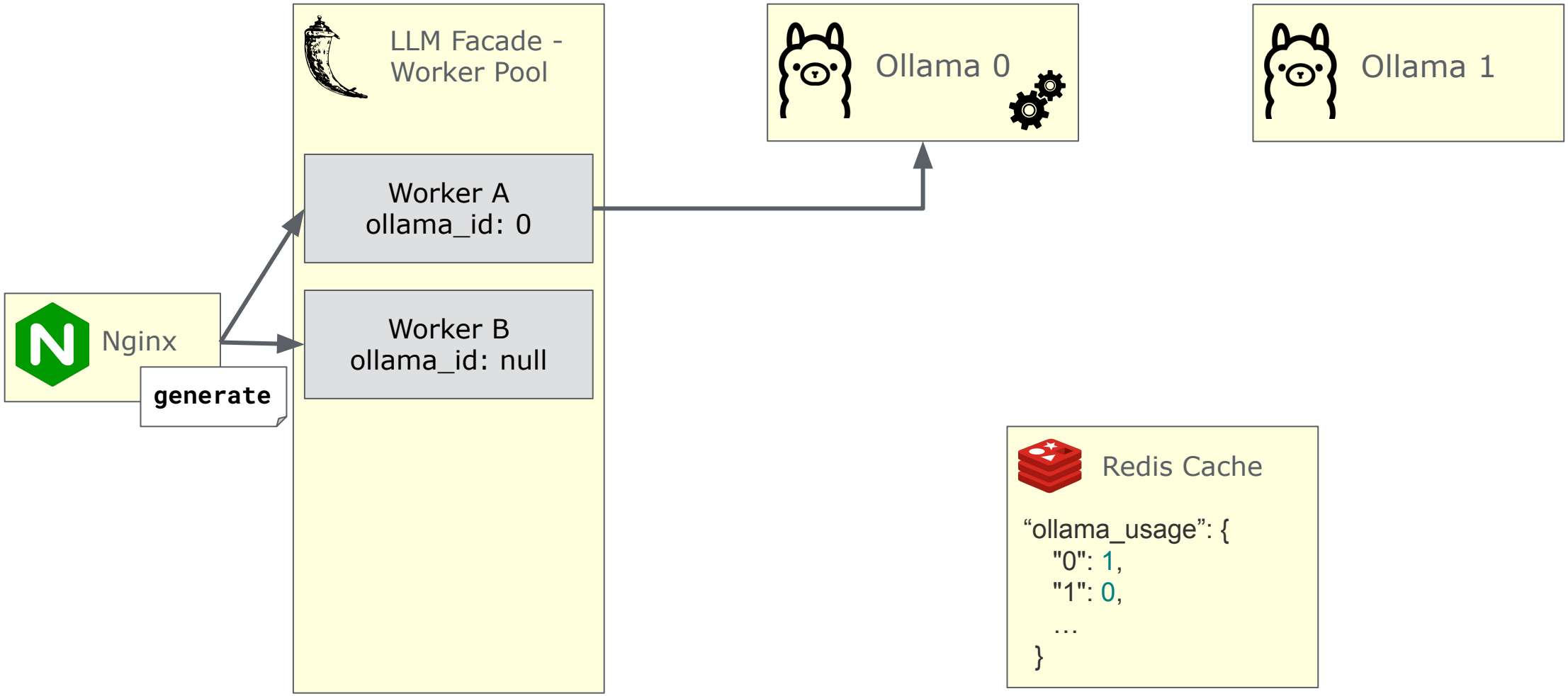
Simultaneous Example Generation



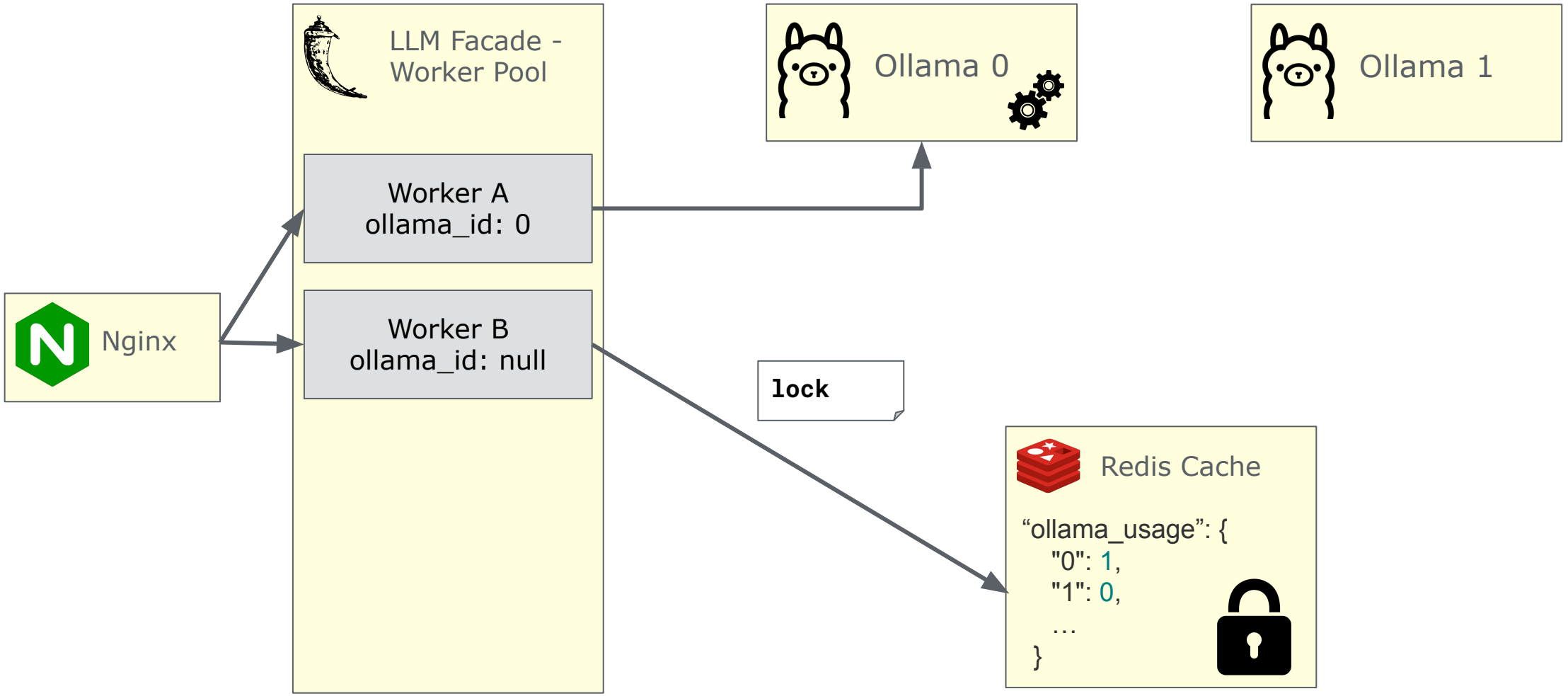
Simultaneous Example Generation



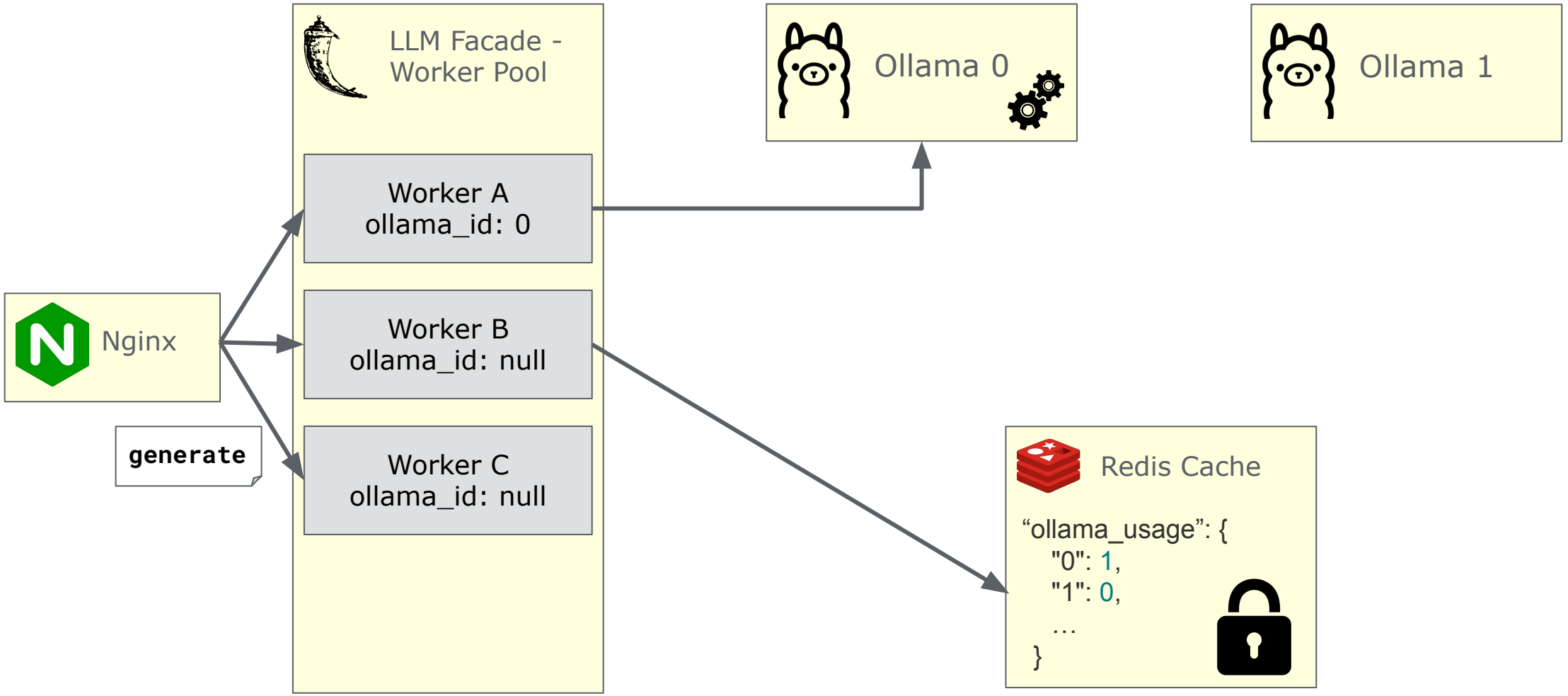
Simultaneous Example Generation



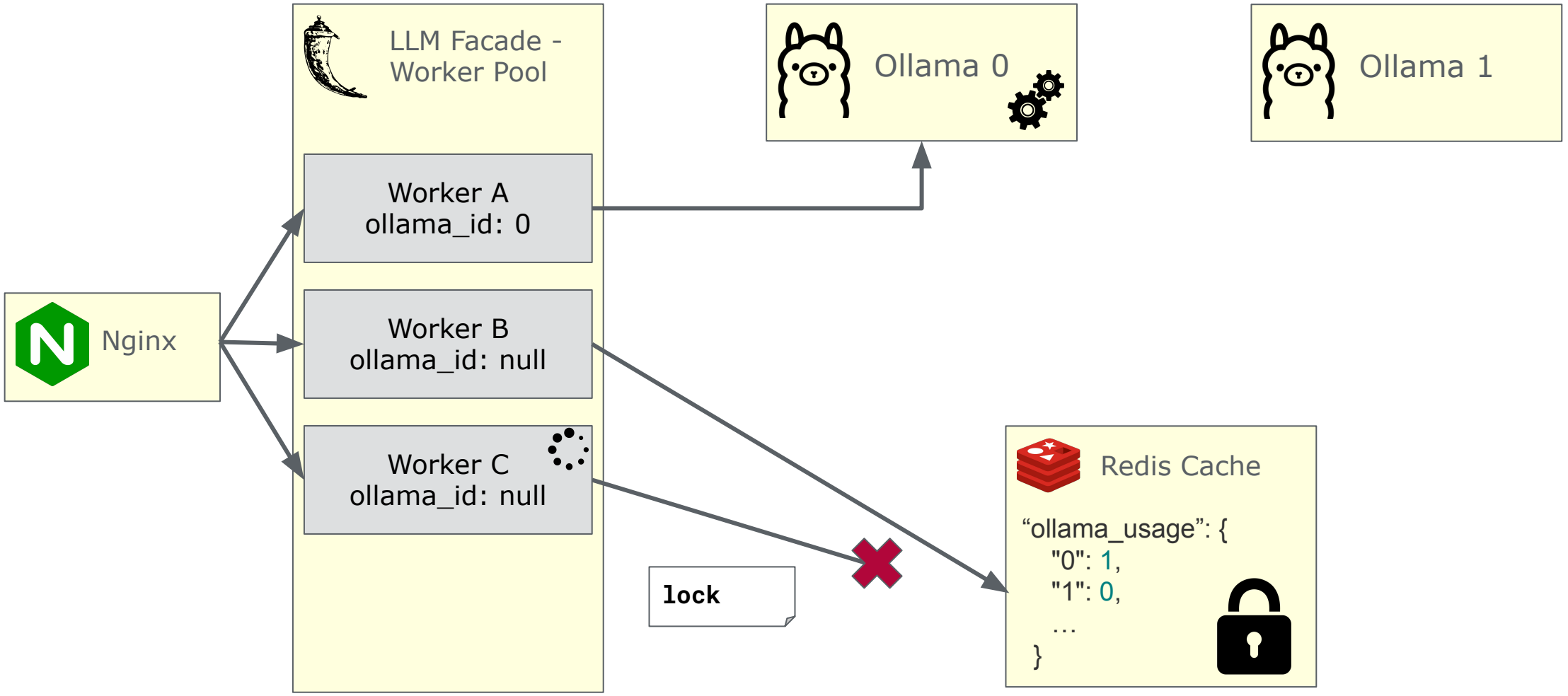
Simultaneous Example Generation



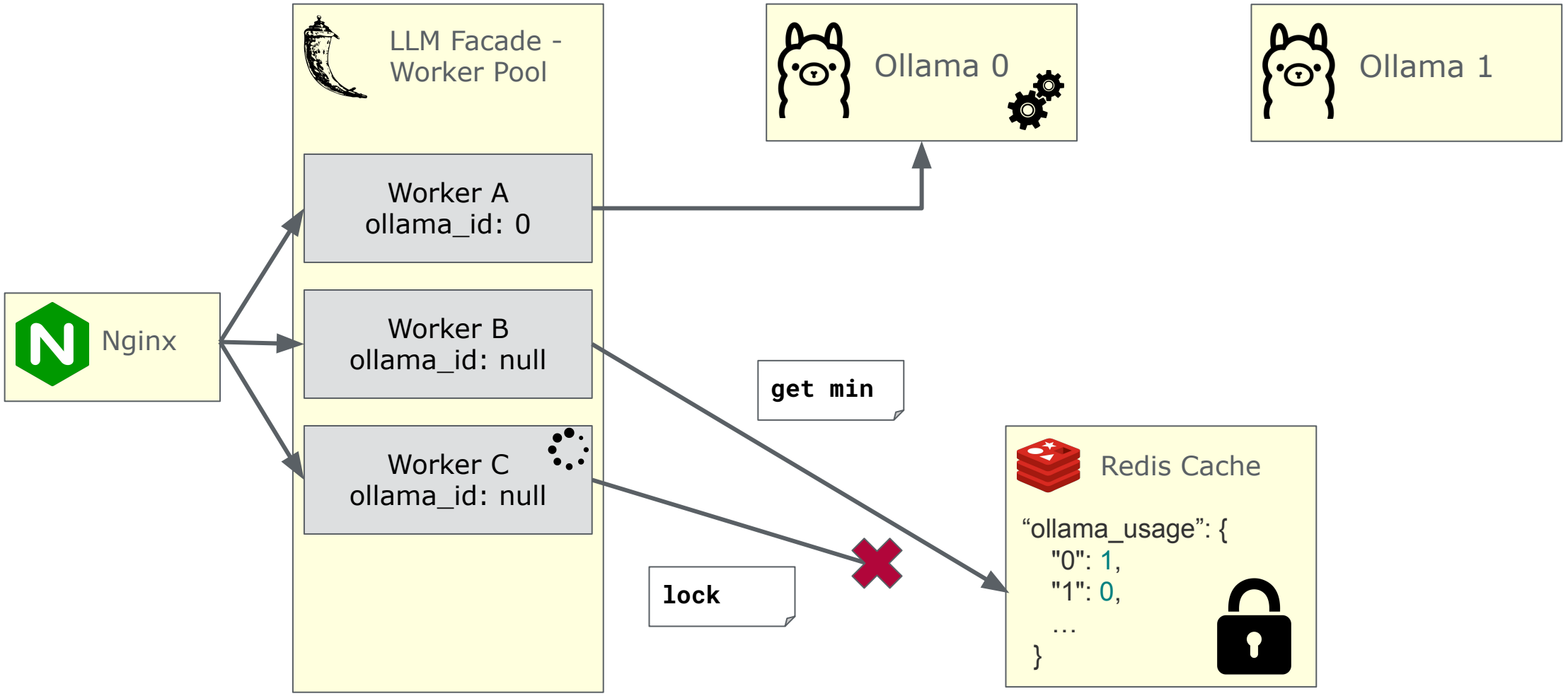
Simultaneous Example Generation



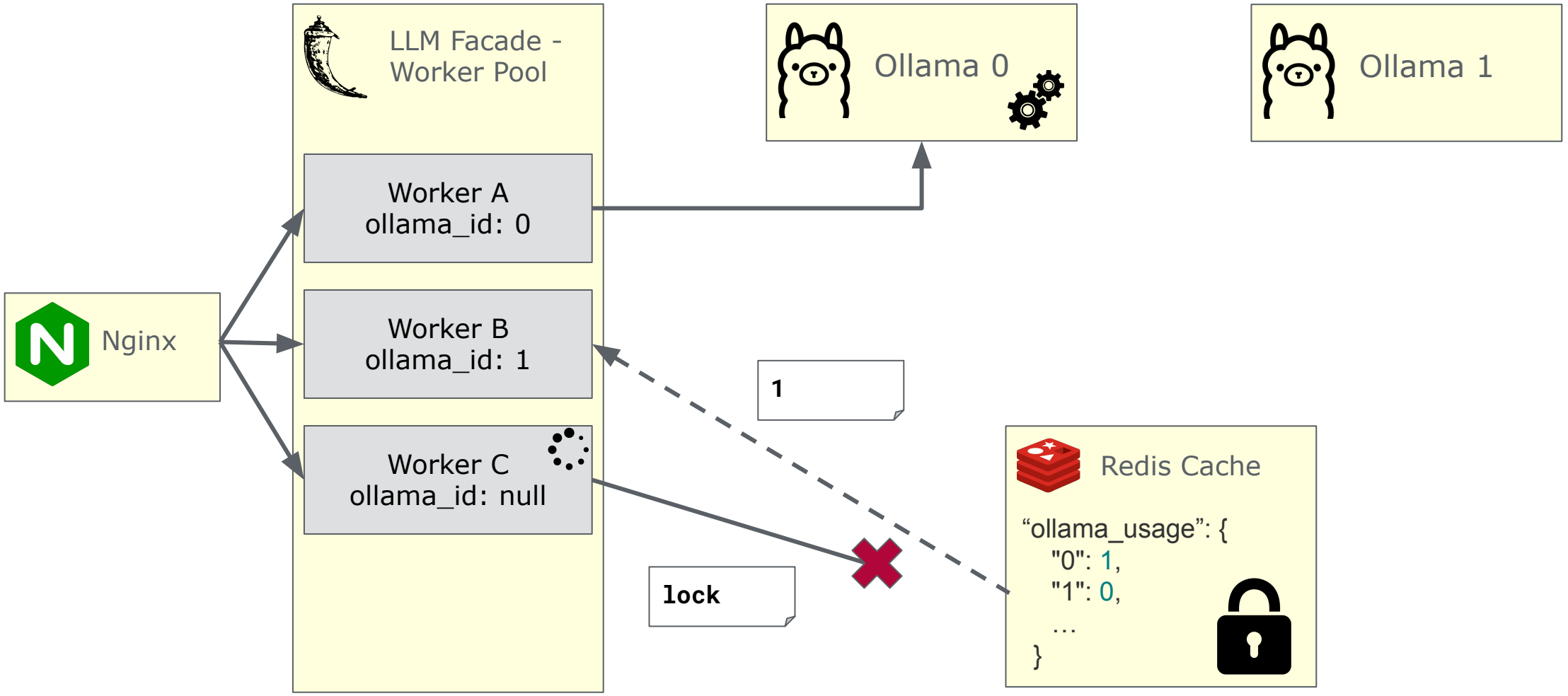
Simultaneous Example Generation



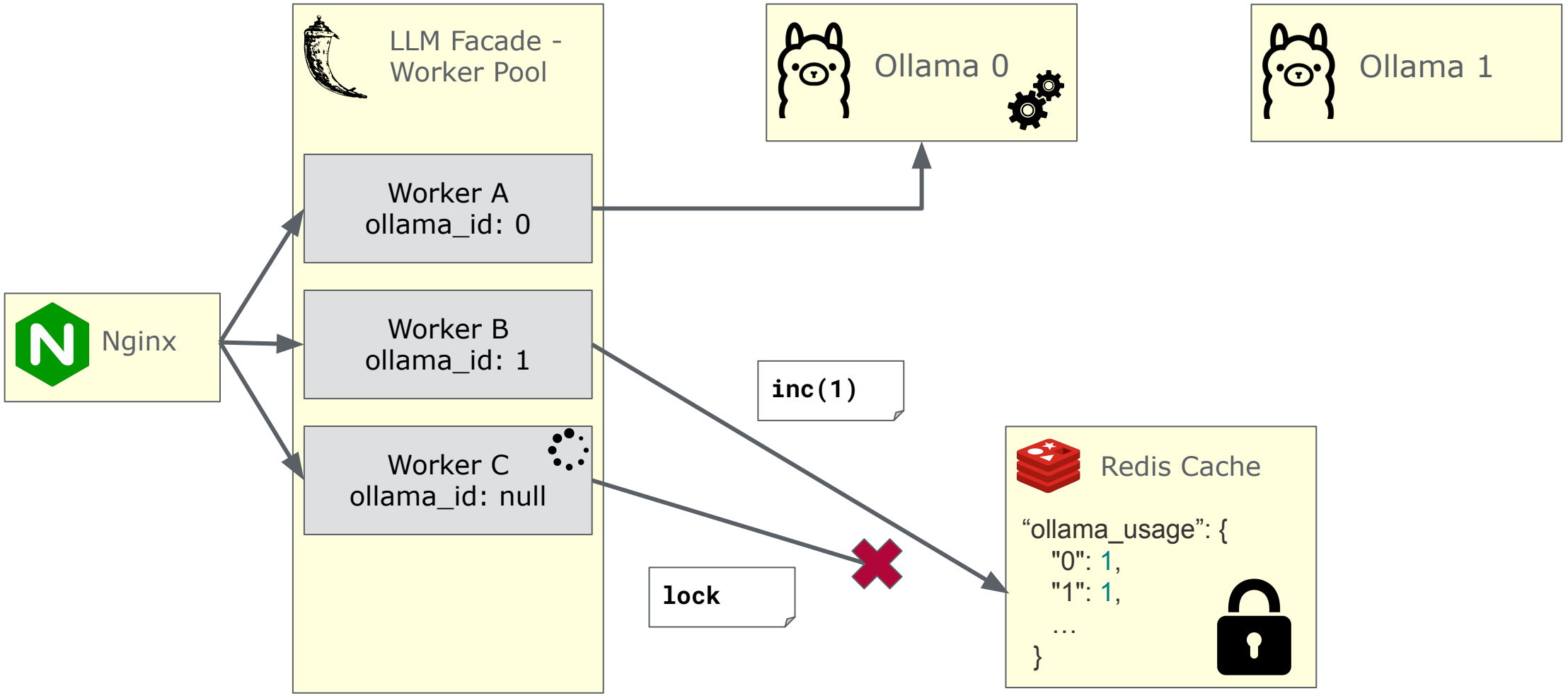
Simultaneous Example Generation



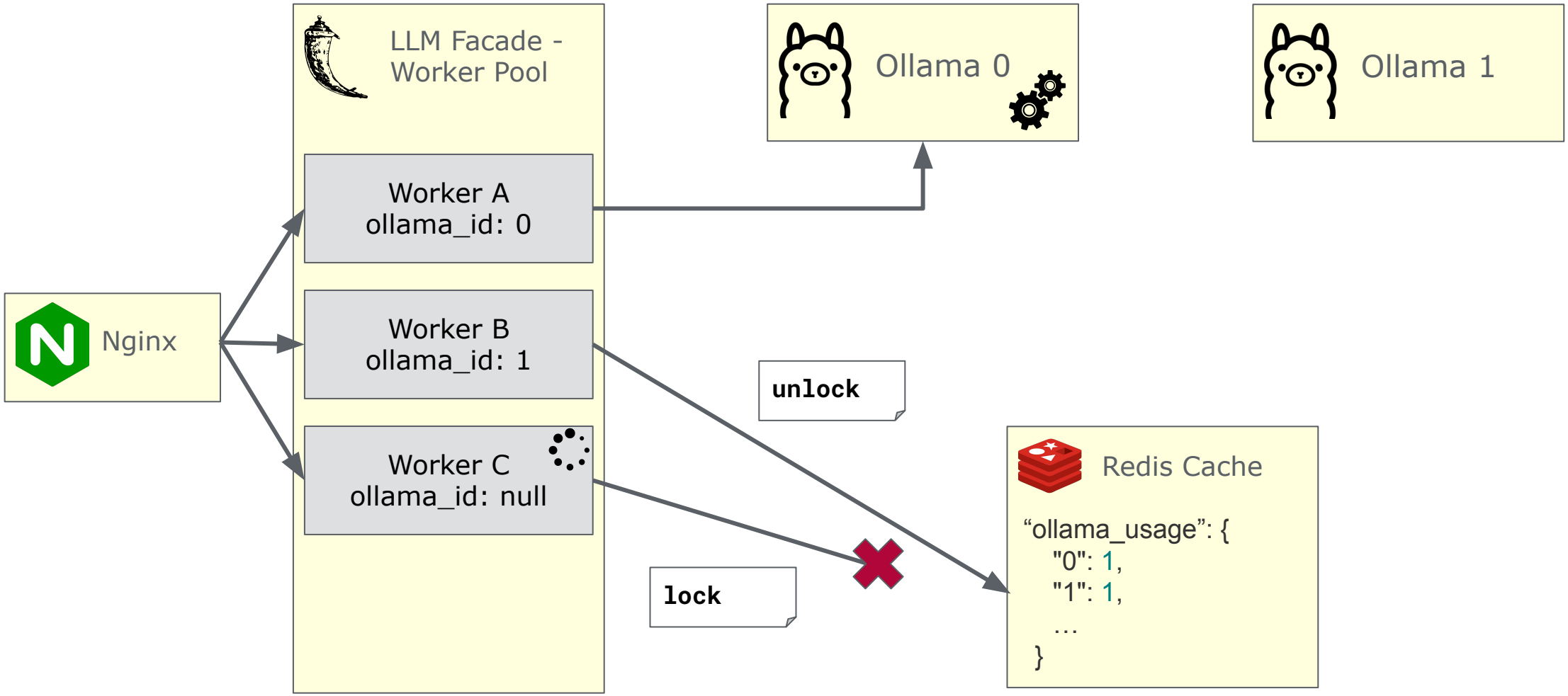
Simultaneous Example Generation



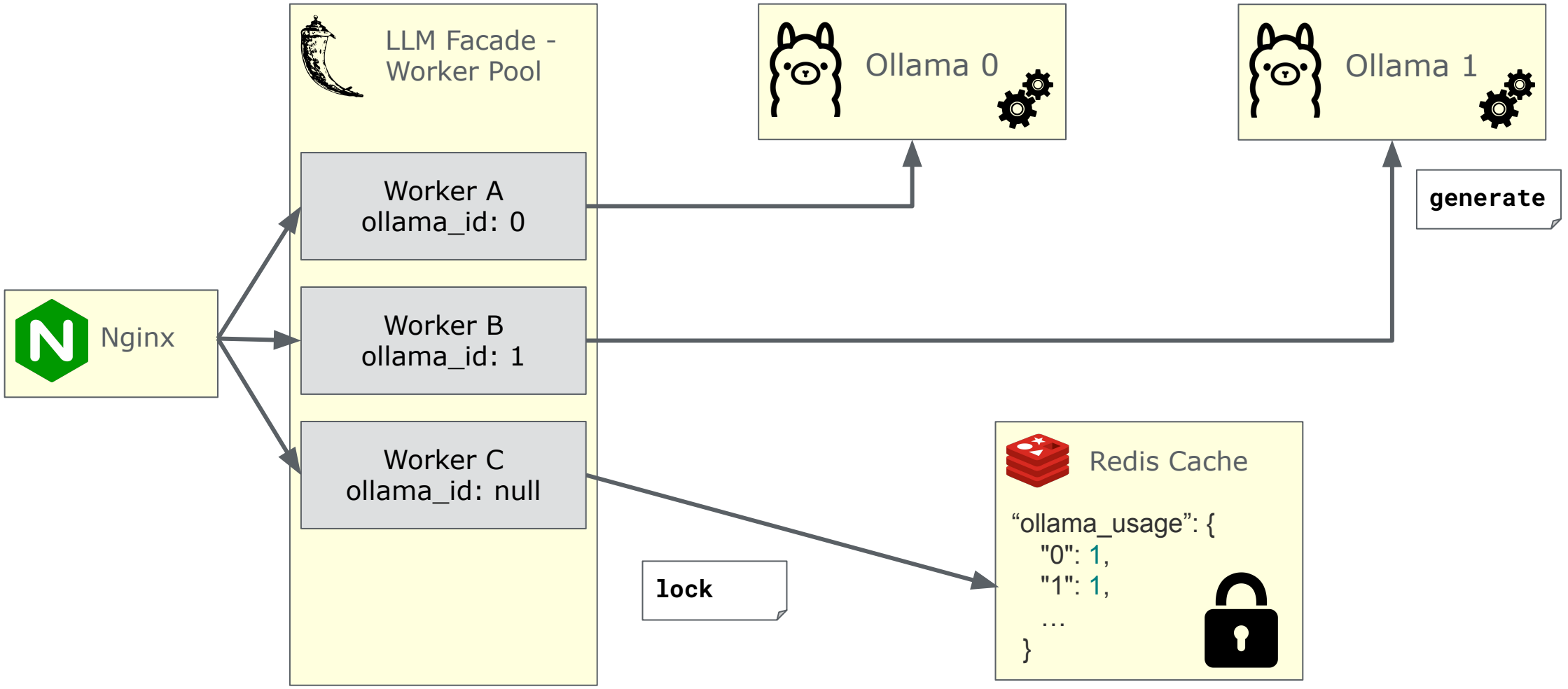
Simultaneous Example Generation



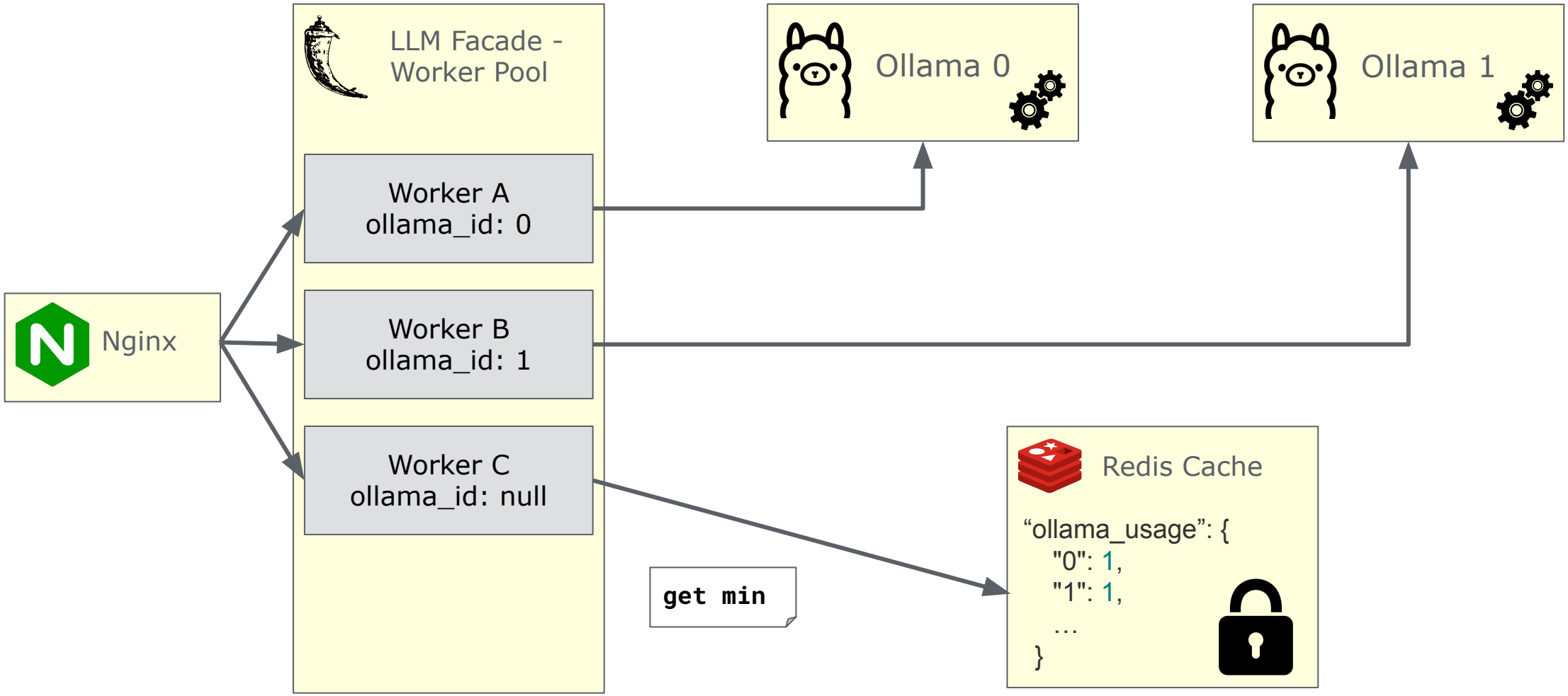
Simultaneous Example Generation



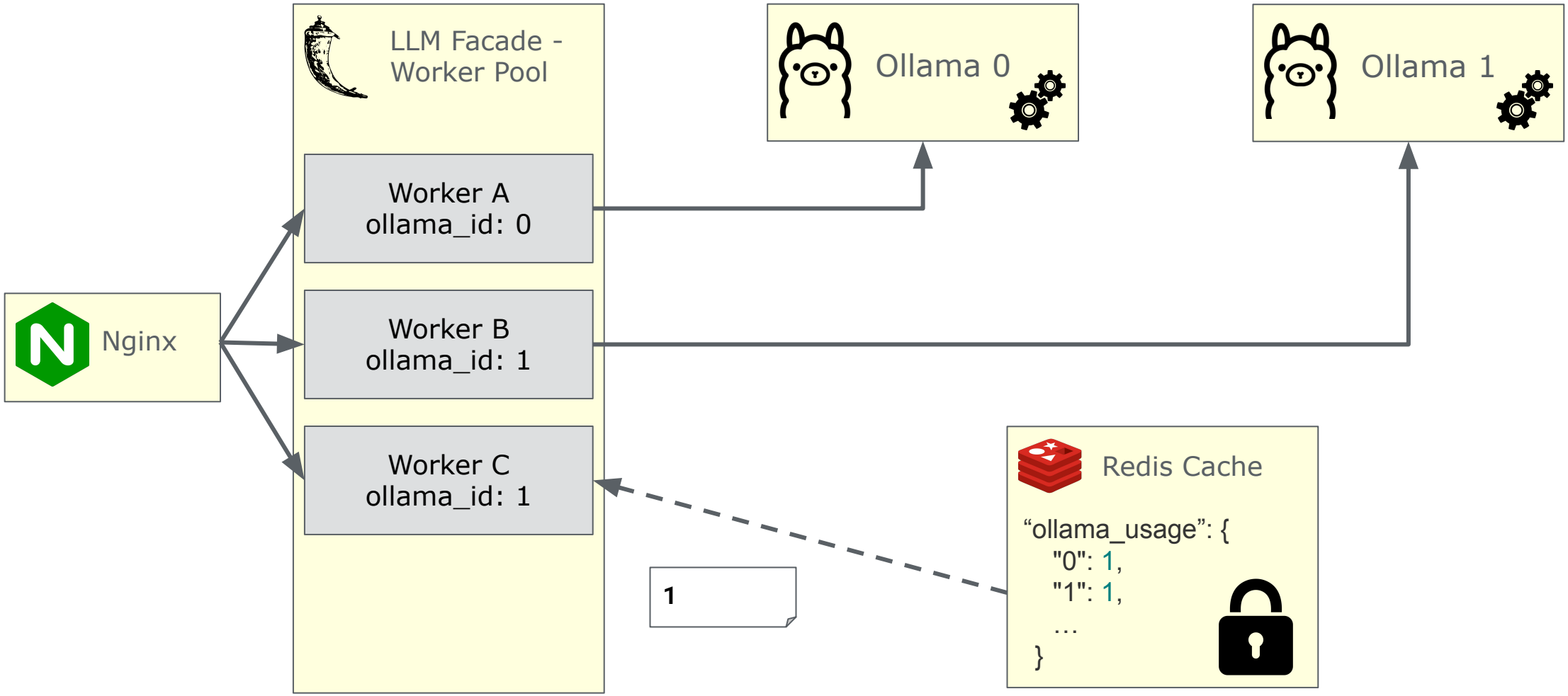
Simultaneous Example Generation



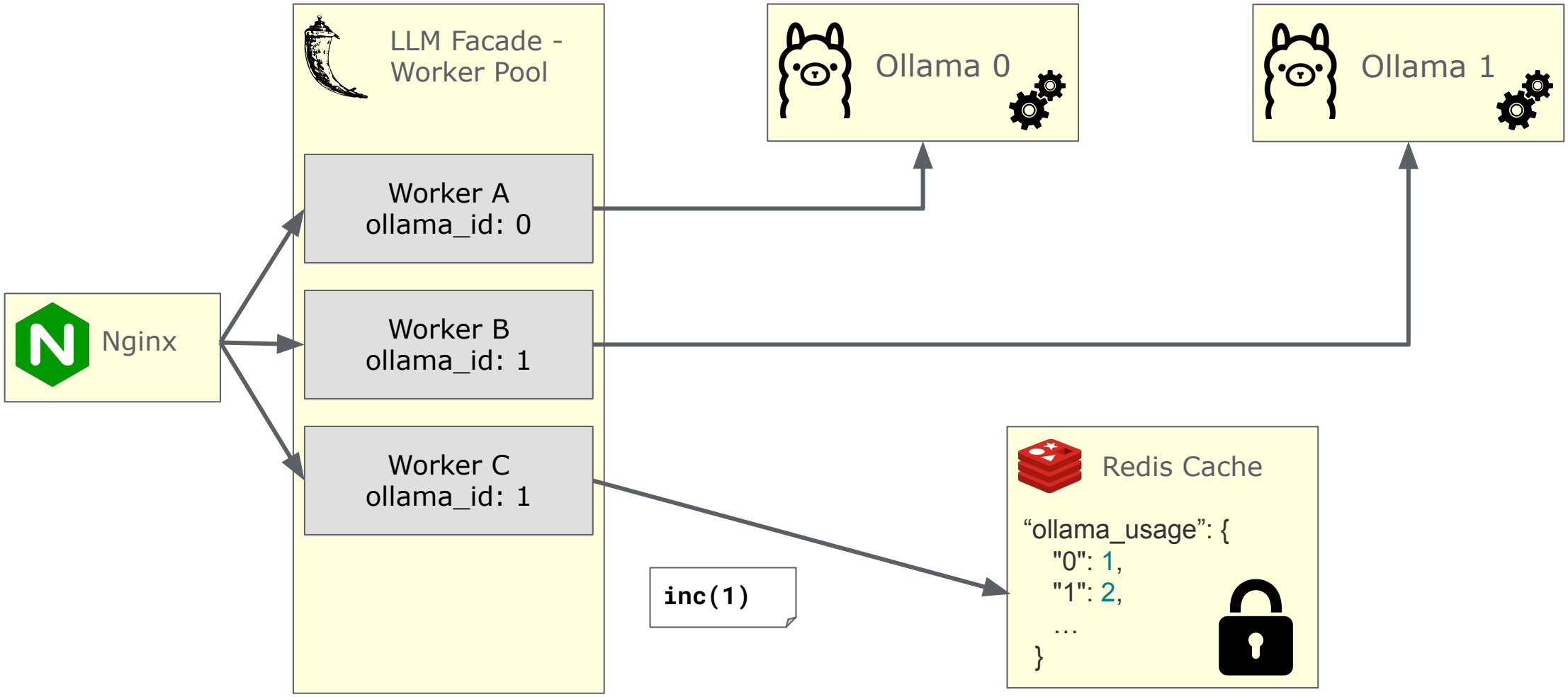
Simultaneous Example Generation



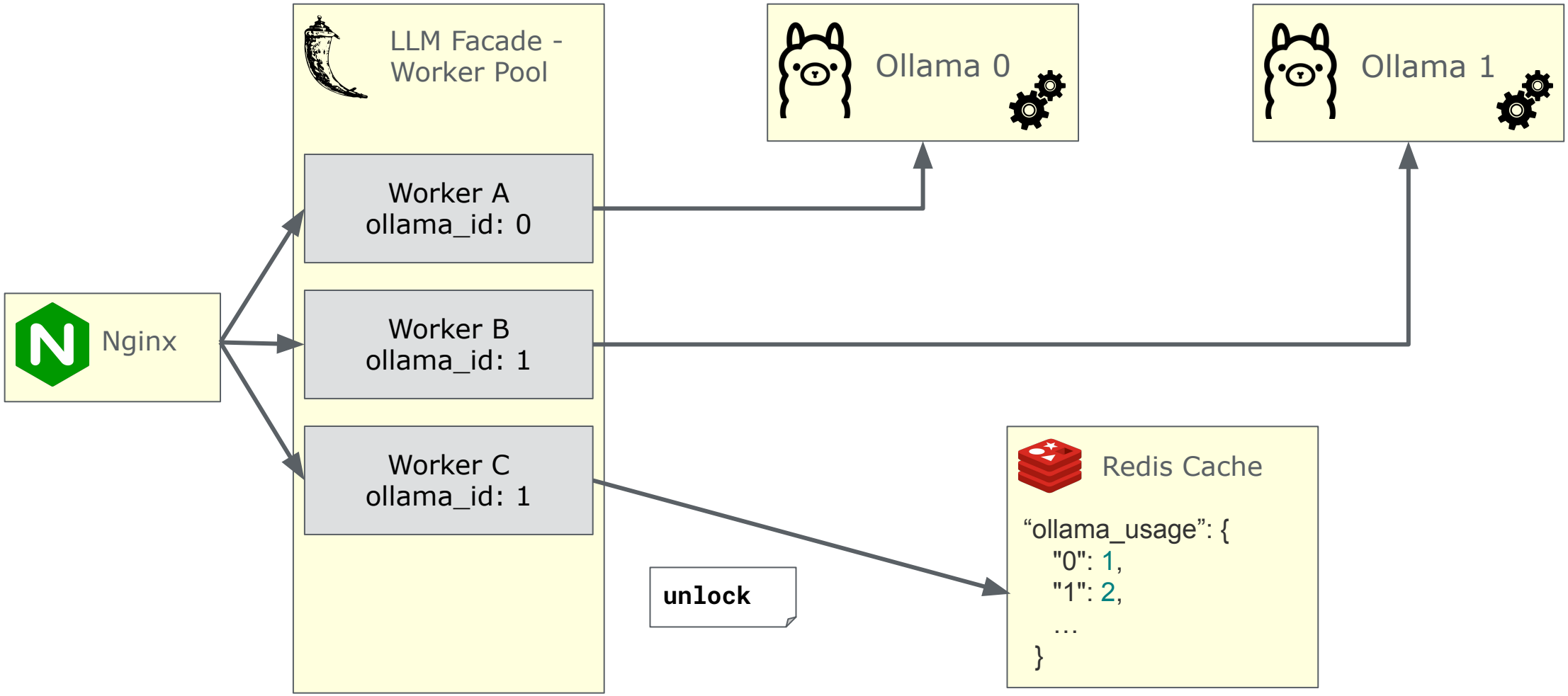
Simultaneous Example Generation



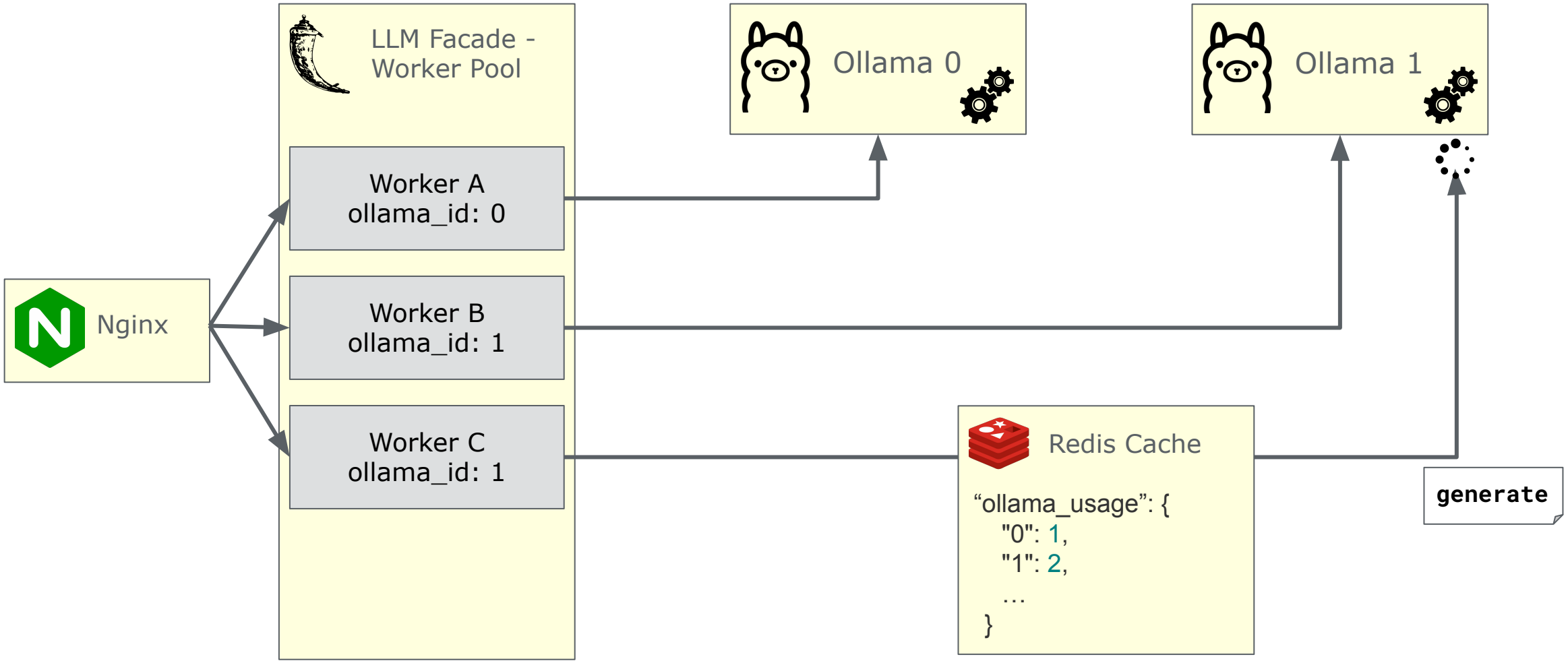
Simultaneous Example Generation



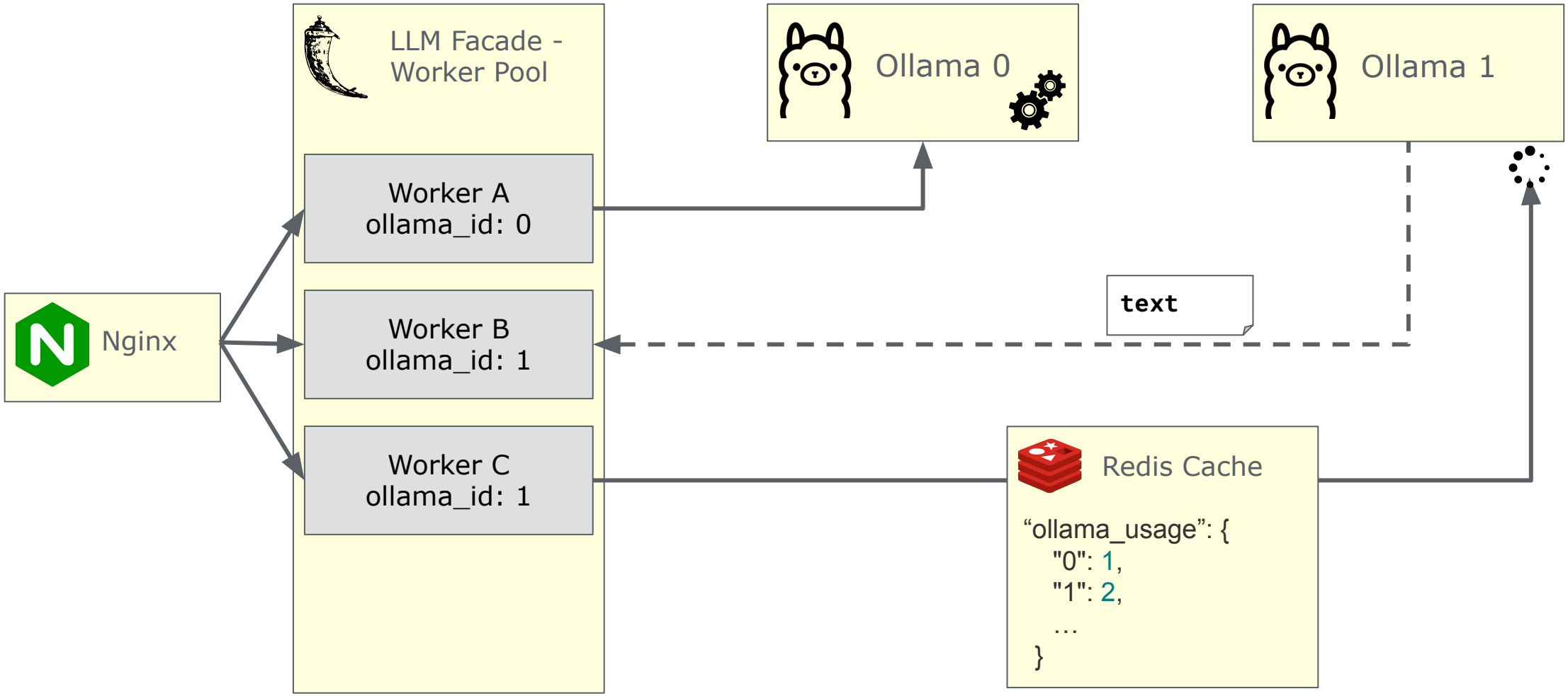
Simultaneous Example Generation



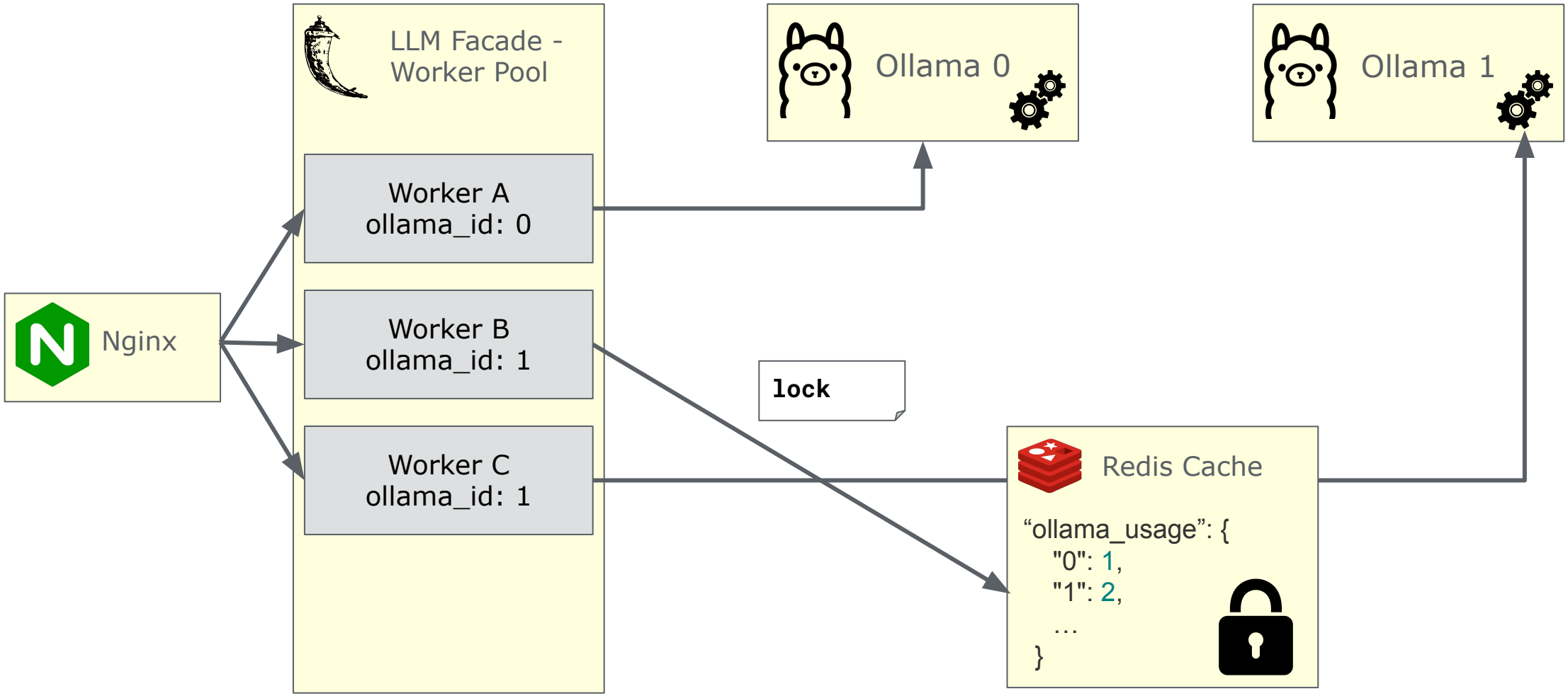
Simultaneous Example Generation



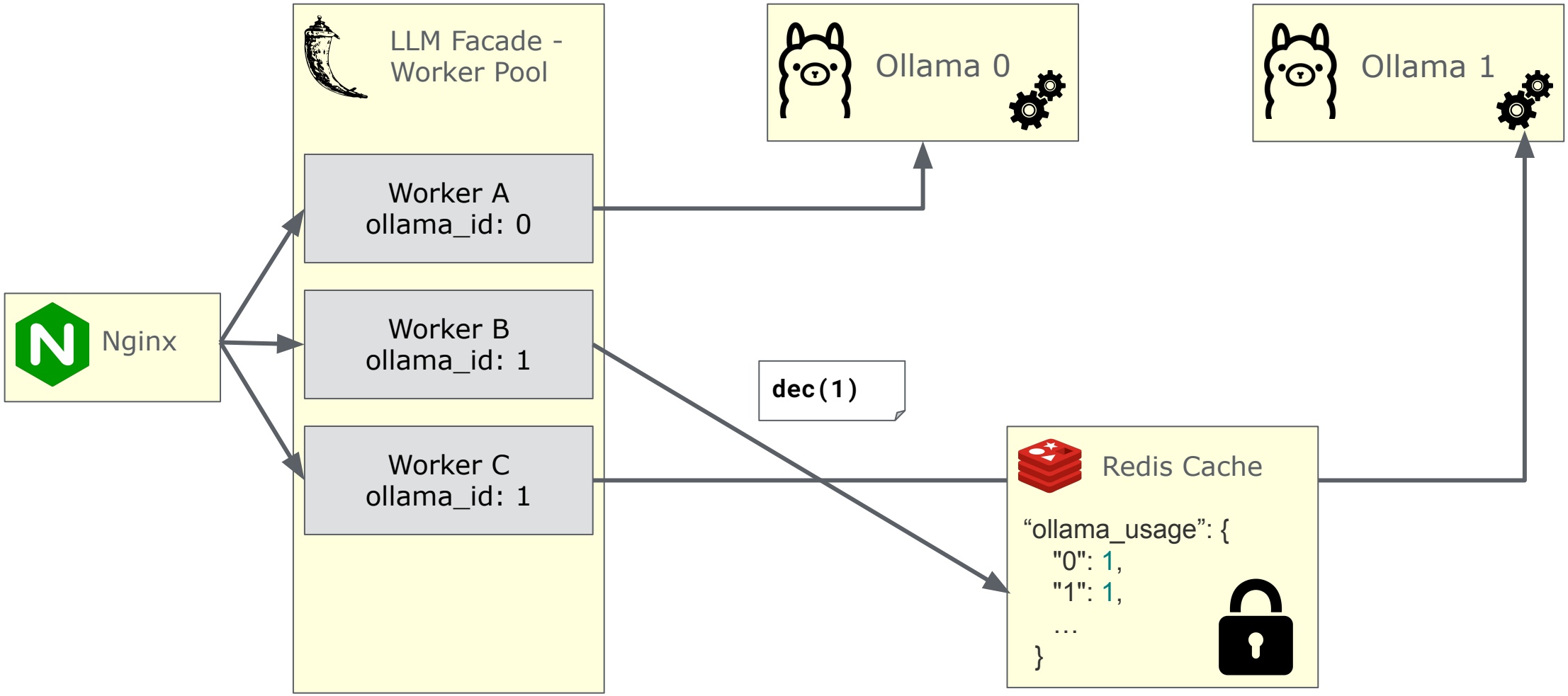
Simultaneous Example Generation



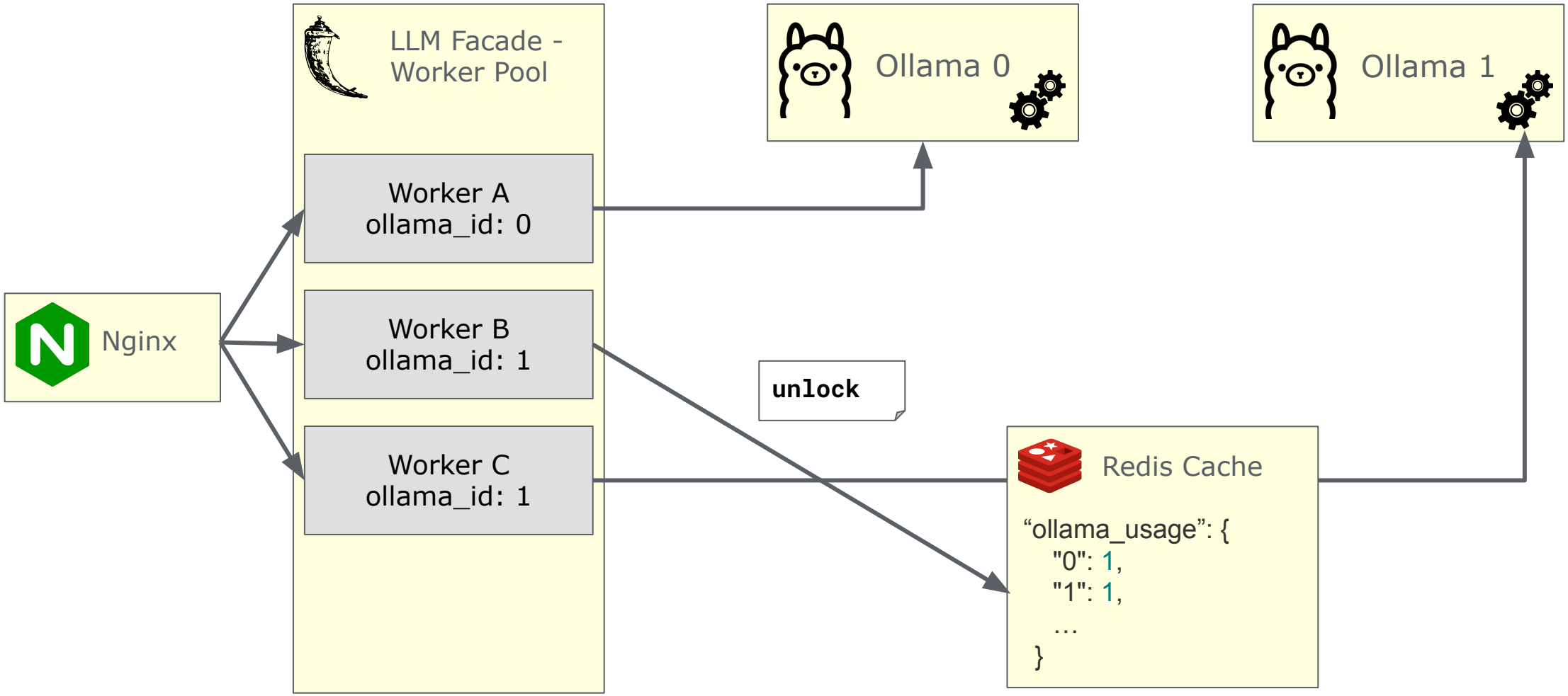
Simultaneous Example Generation



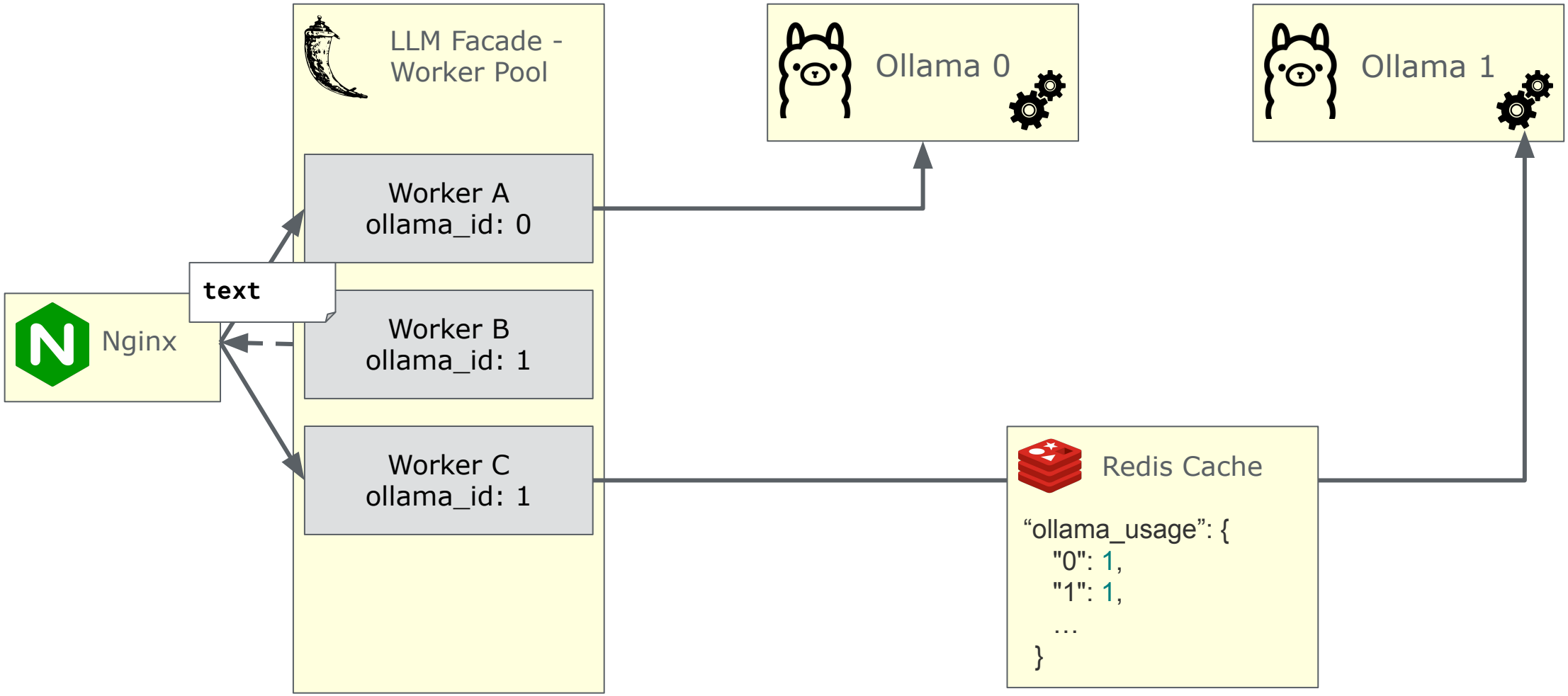
Simultaneous Example Generation



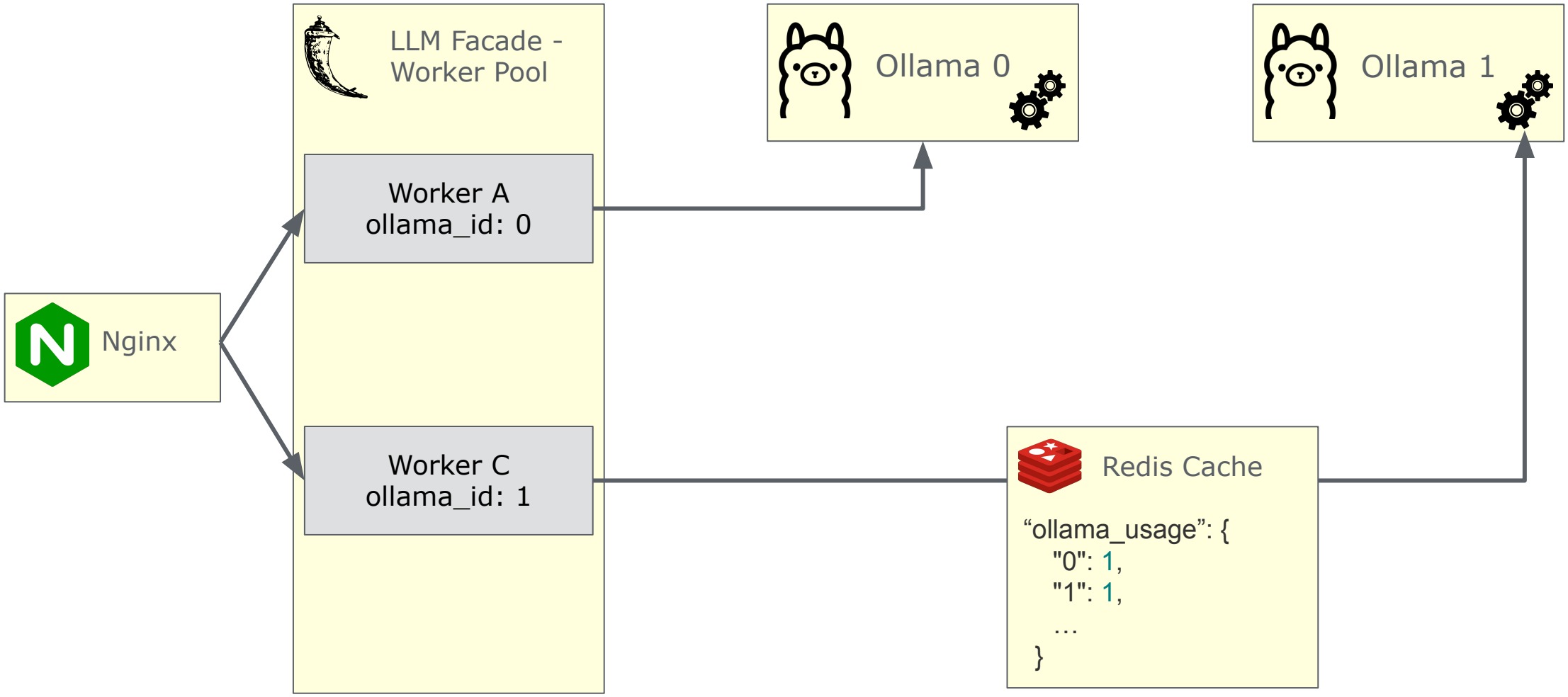
Simultaneous Example Generation



Simultaneous Example Generation



Simultaneous Example Generation



Challenges

Code Related

- Simultaneous Generation
- Syntax Highlighting
- Prompt Engineering to avoid explanation in code
- Hard to find suitable metrics
  - Code Length
  - Code Complexity
  - Maintainability

Project Related

- User Feedback: Server restricted to HPI network
- Linux and Windows incompatibilities

Challenges

- Code Related
  - Maintainability

"You are as experienced as a senior software developer. You will get a piece of code documentation and the source code for this documentation.
Your task is to provide a helpful code example for the documentation. You are encouraged to use also other functions from the same framework to demonstrate interoperability between them if it is appropriate.

You ONLY output python code. If you output anything else than python code 100 cute little rabbits will DIE. You DON'T under any circumstances output any explanation of the code."

Threatening the model helped reduce the output of any text other than code, but could not eliminate it entirely.

Challenges

Code Related

- Simultaneous Generation
- Syntax Highlighting
- Prompt Engineering to avoid explanation in code
- Hard to find suitable metrics
  - Code Length
  - Code Complexity
  - Maintainability

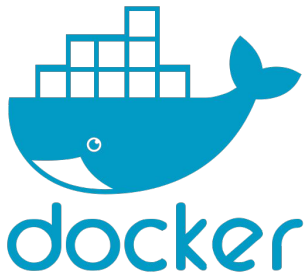
The only metric, which is actually meaningful, is a user's opinion of the example and whether it's helpful to them or not.

Project Related

- User Feedback: Server restricted to HPI network
- Linux and Windows incompatibilities

Learnings / Reflection

New Technologies



ollama

Project Related

- Improve Collective Code ownership
- Two development phases
- Focus on user experience / story

User Feedback

Methodology

- Introduction to subject
- ~15 Minutes
- Communicated incomplete features

We asked several users for feedback in the last 1-2 weeks of development. After giving them a brief introduction to the subject, i.e. telling them that we built an application to generate code examples for different APIs, we let them explore the tool on their own.
Two constraints made the collection of user feedback slightly challenging: The website was only available from the HPI network and we were not able to find users which were currently in a learning process for any of the available APIs. Therefore, we were limited to brief, in-person sessions of about 15 minutes, gathering mostly “theoretical” feedback. The users were asked to generate a few examples, experiment with the workflow and use as many features as possible. When they were using any incomplete features, we made them aware of this.

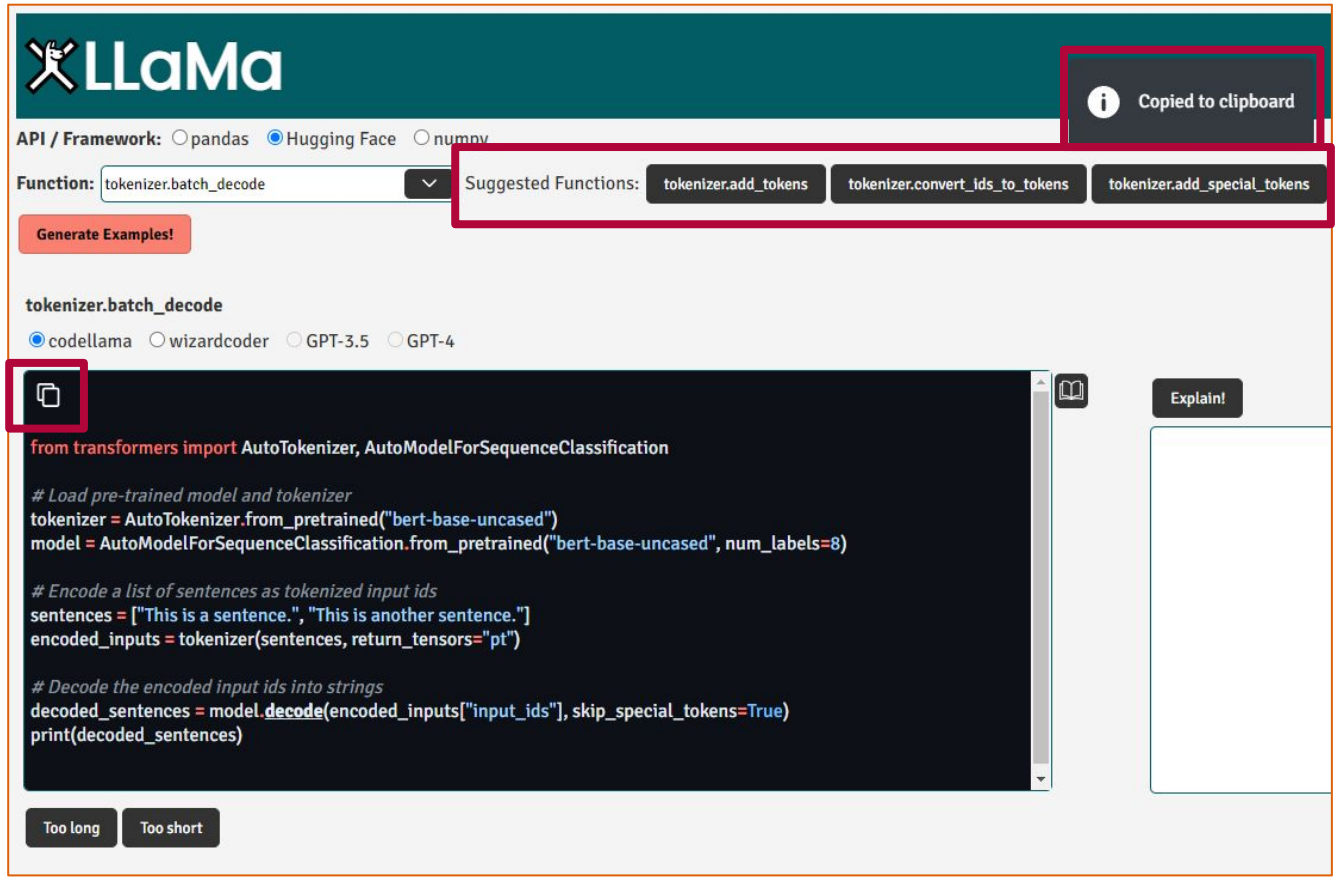
User Feedback

Methodology

- Introduction to subject
- ~15 Minutes
- Communicated incomplete features

Implemented

- Confusing function buttons
- Copy button should give visual feedback



User Feedback

User 1 uses pandas for data analysis.

User 1 (Molecular Biology PhD Student)

- Documentation button very helpful
- Did not notice clickable button in example
- More advanced options for generated examples and layout
- History of generated examples
- Would even just use it to read documentation, nice collection of different frameworks

User 2 (Machine Learning Engineer)

- Currently learning new framework with bad documentation, so tool would be useful if all functions were included

User 3 (Software Developer)

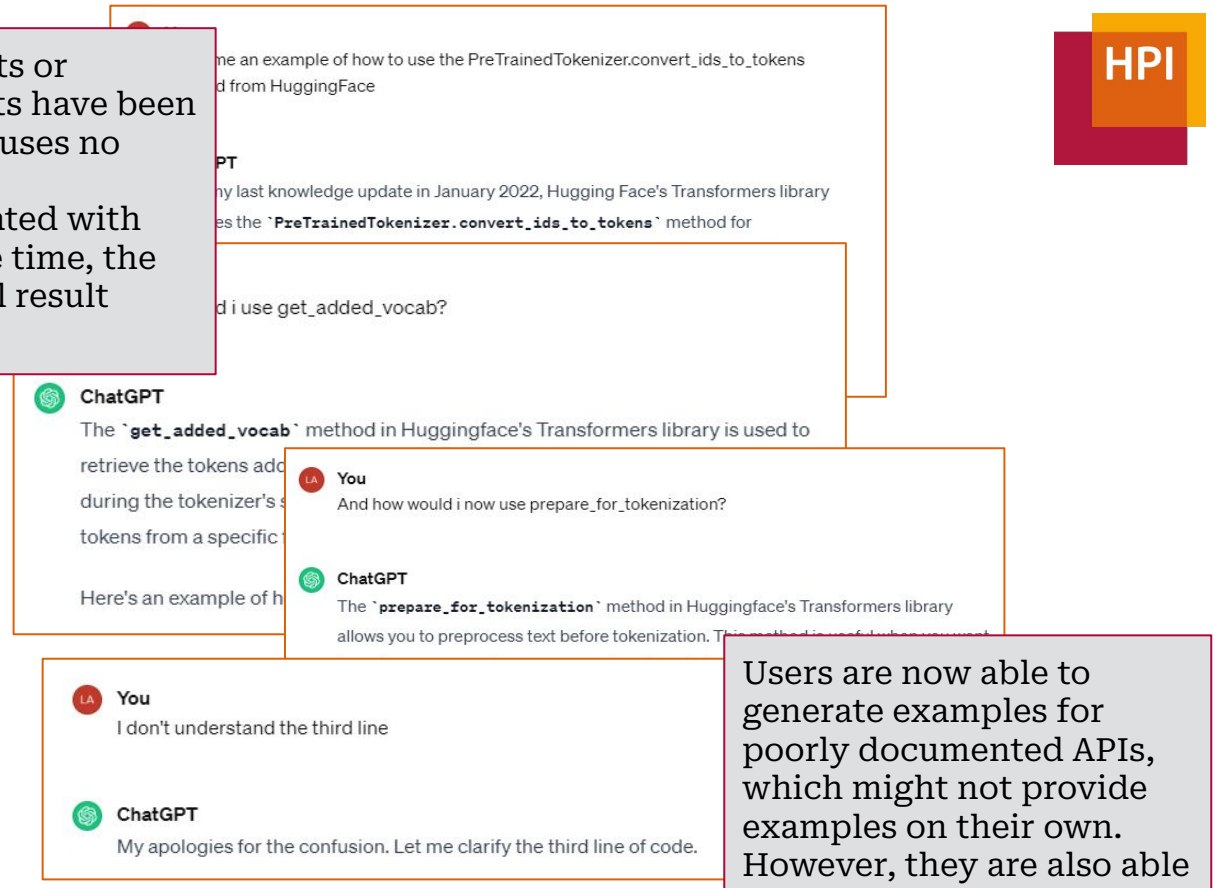
- Would use when learning new API
- Would be nice to have for programming languages, e.g. Java
- Nice to have everything in one place

Idea

Chatbots

- Helpful when working and learning new tools, but often cumbersome to use
  - Might need long prompts ✅
  - First results may be unhelpful (✅)
  - Multiple follow up prompts can be annoying ✅
- Different LLMs give different results (of different quality) ✅

Problems like long prompts or multiple follow up prompts have been eliminated, since xLLaMa uses no prompts whatsoever.
Since examples are generated with multiple LLMs at the same time, the change of having a helpful result immediately are higher.



Users are now able to generate examples for poorly documented APIs, which might not provide examples on their own. However, they are also able to find additional examples for well documented APIs, in a fast and convenient way.

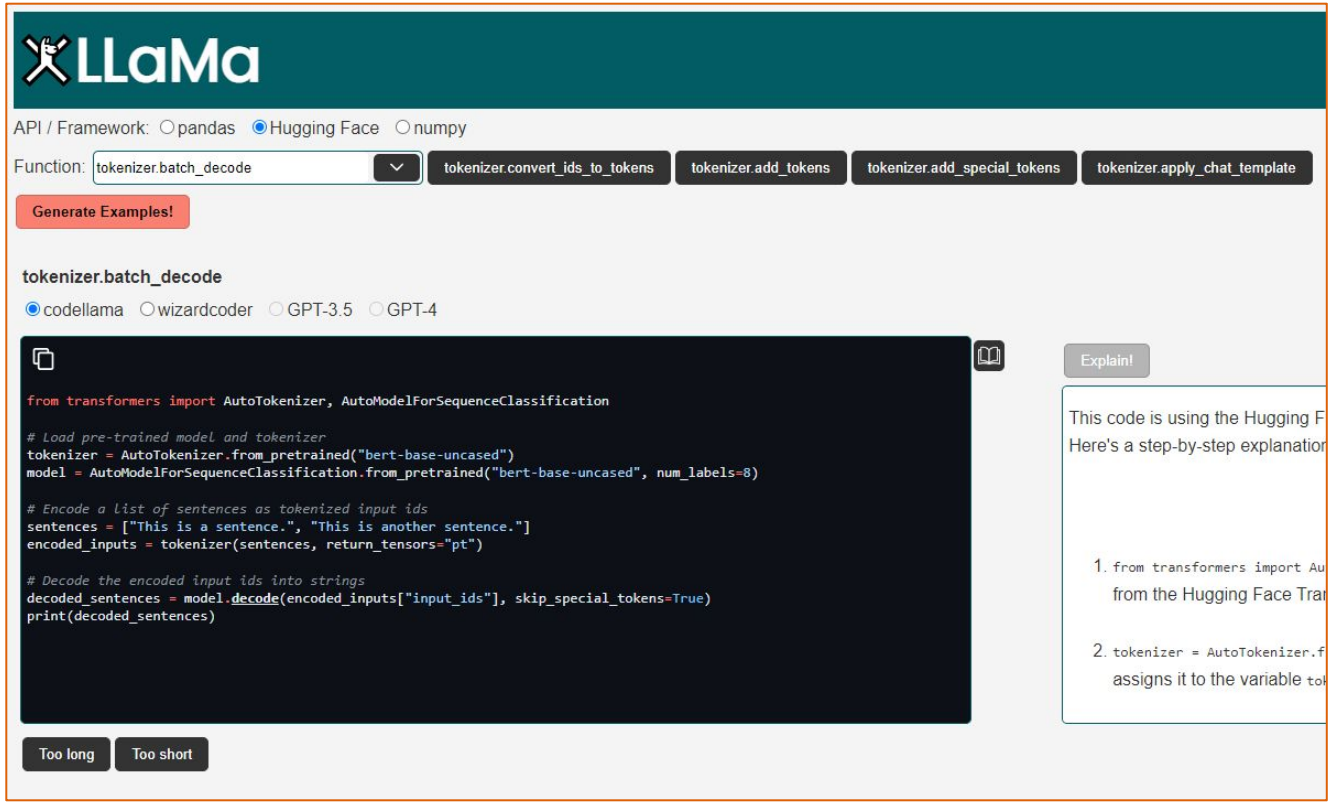
APIs

- Some API references are very detailed and helpful, others are not ✅

Future Work

- Add entire APIs to database
- Add Context integration
- Add GPT or any other LLM support
- Collect and use user feedback

It would be nice to collect and store user feedback, specifically about the helpfulness of an example, store it and use this data to fine-tune a model and generate even more helpful examples in the future.



xLLaMa

Thank you for your attention!

Links:

http://delos.eaalab.hpi.uni-potsdam.de:8080/

https://github.com/CR1337/xLLaMa

Quellen

- https://pandas.pydata.org/docs/
- https://numpy.org/doc/
- https://huggingface.co/docs
- https://ollama.ai/
- https://huggingface.co/codellama/CodeLlama-70b-hf
- https://huggingface.co/WizardLM/WizardCoder-33B-V1.1
- https://chat.openai.com/