

Torob Turbo

AI Shopping Assistant



Amirmohammad Mahdavikia



Appreciation



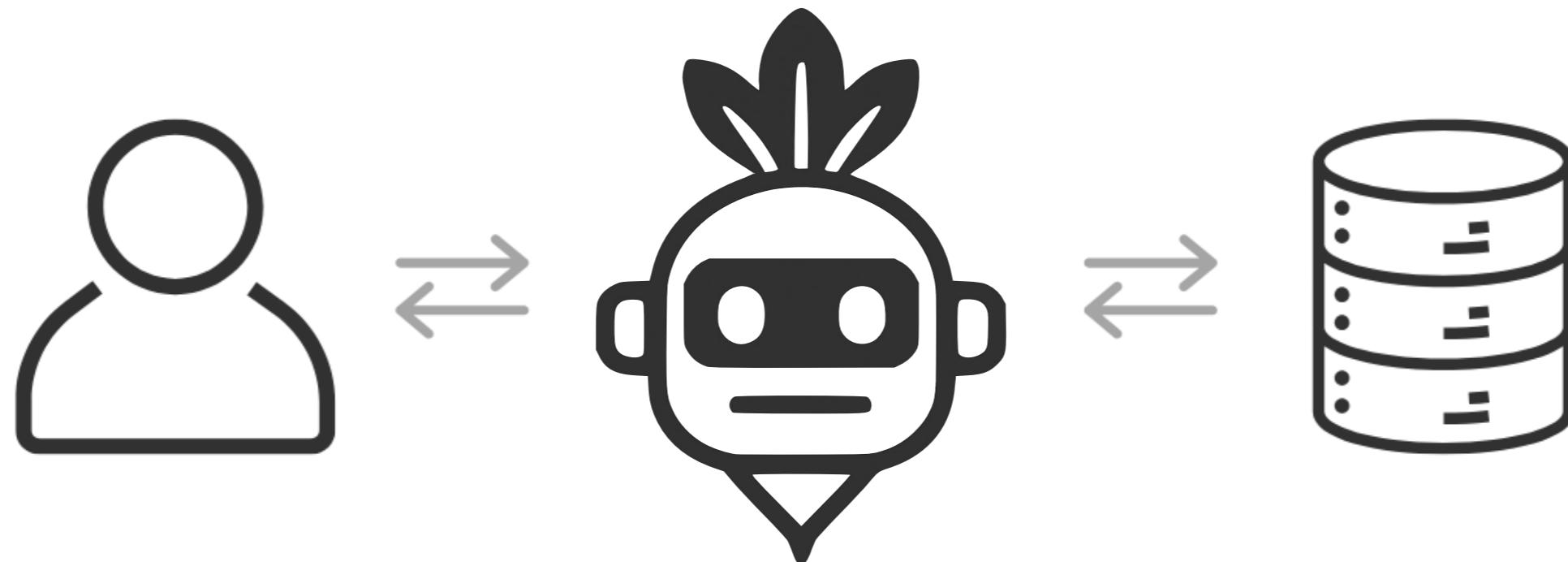
Torob



Quera



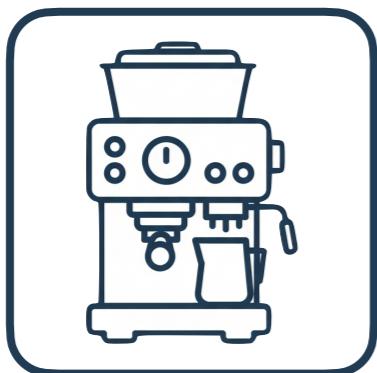
Problem Statement



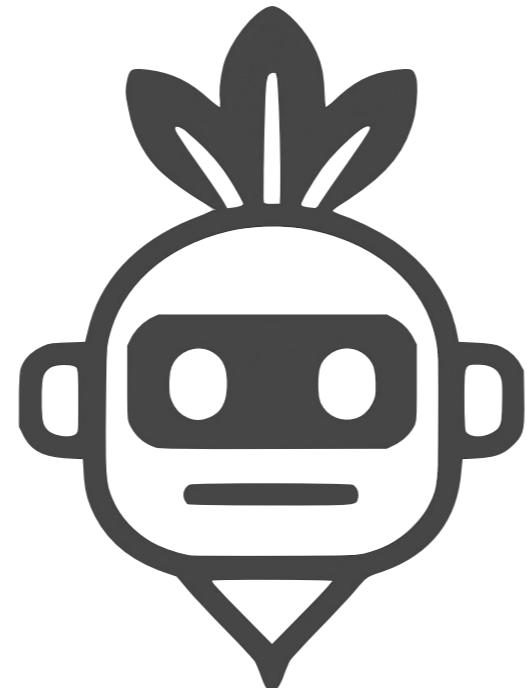


Main Challenge

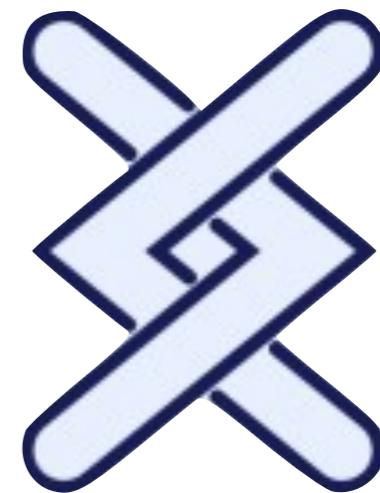
Multi-modality



Espresso
Machine



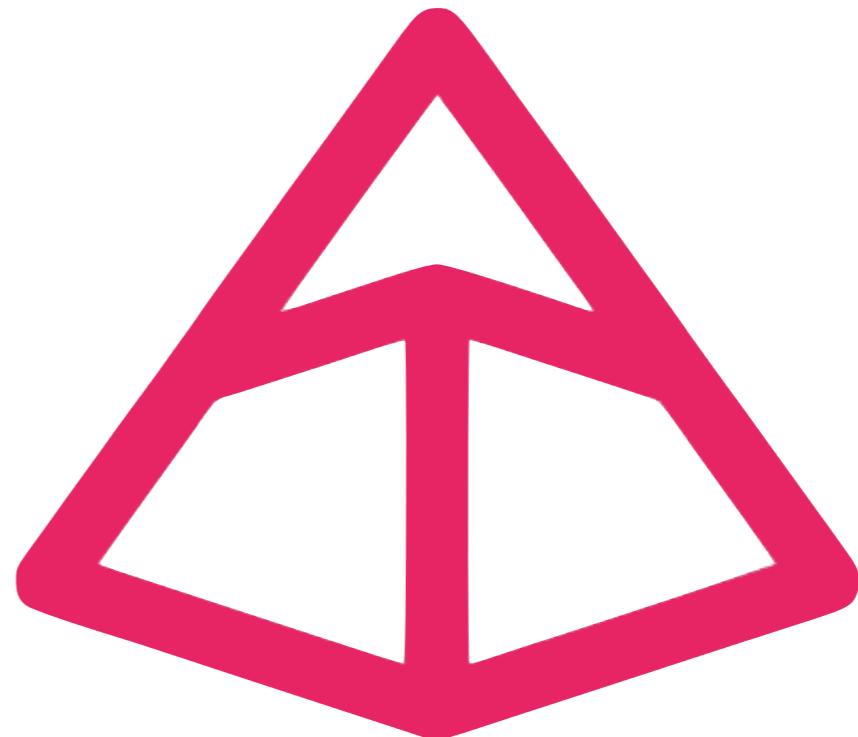
Hosting





Architecture

```
app/
├── .env
├── .gitignore
├── agent.py
├── app.py
├── db.py
├── Dockerfile
├── logging_config.py
├── requirements.txt
└── utils.py
```





Text Retrieval

BM25
OR
Embedding



Both

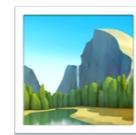
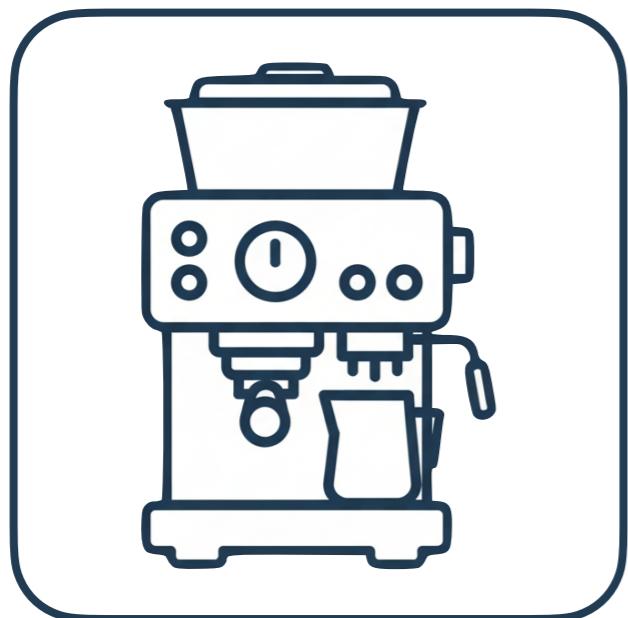
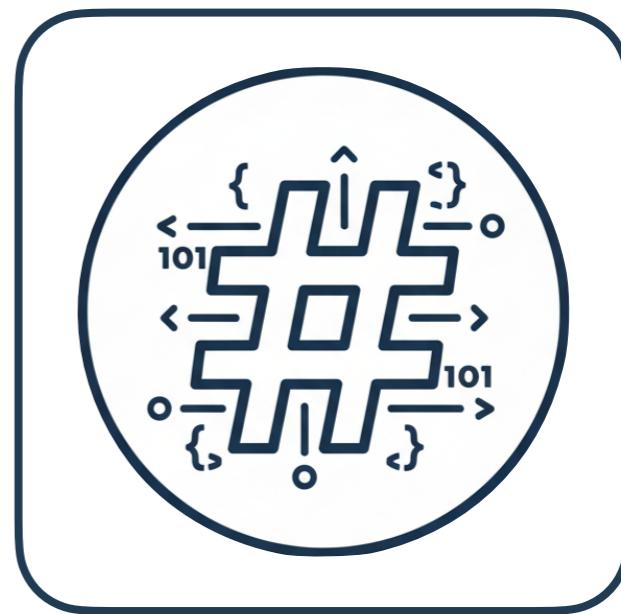


Image Retrieval



exact hash
→
phash



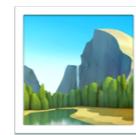


Image Retrieval

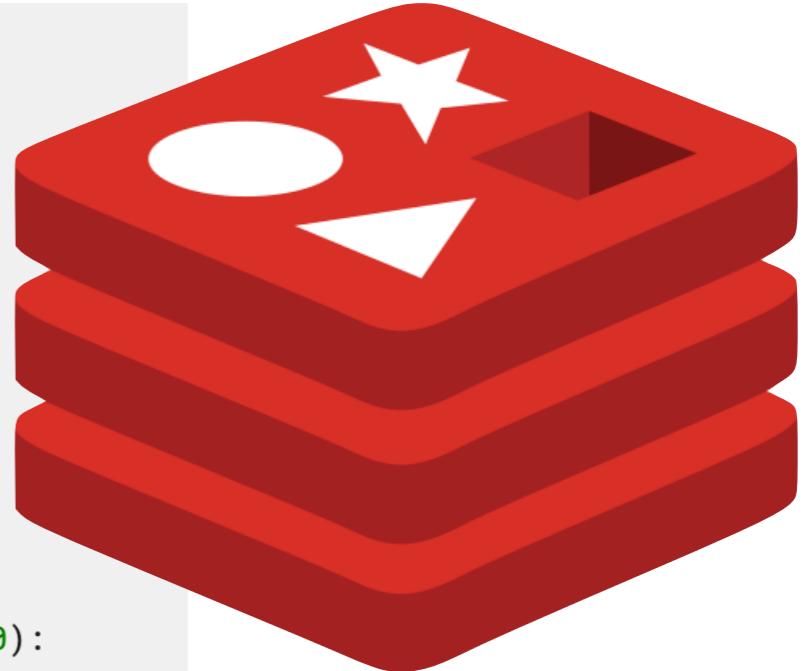


openai/clip-vit-base-patch32



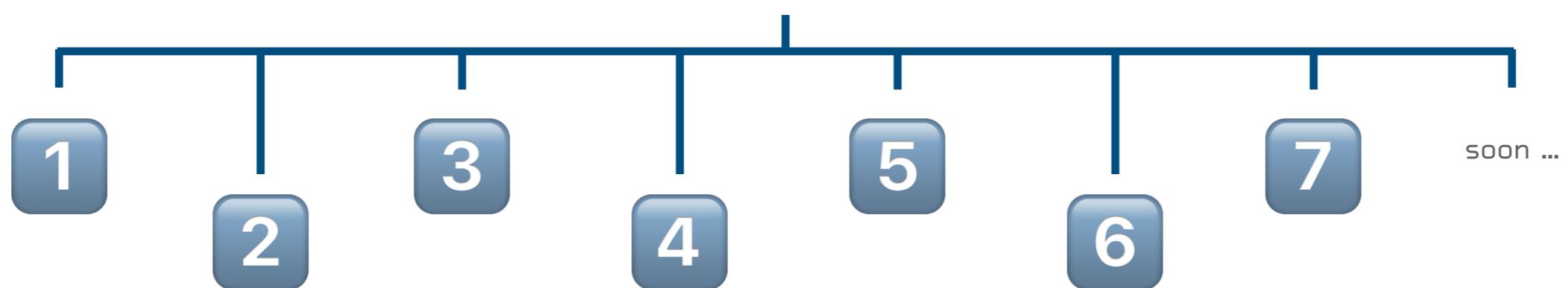
Caching

```
def get_chat(chat_id: str) -> Dict[str, Any]:  
    """Retrieve chat data from Redis and deserialize Python objects."""  
    raw = r.get(chat_id)  
    if not raw:  
        return {}  
  
    try:  
        data = pickle.loads(raw)  
        return data  
    except (pickle.UnpicklingError, EOFError) as e:  
        logger.error(f"Redis GET error for {chat_id}: {e}")  
        return {}  
  
def set_chat(chat_id: str, data: Dict[str, Any], expire_seconds: int = 3600):  
    """Store chat data in Redis with proper object serialization."""  
    try:  
        serialized_data = pickle.dumps(data)  
        r.set(chat_id, serialized_data, ex=expire_seconds)  
    except (pickle.PicklingError, redis.RedisError) as e:  
        logger.error(f"Redis GET error for {chat_id}: {e}")
```



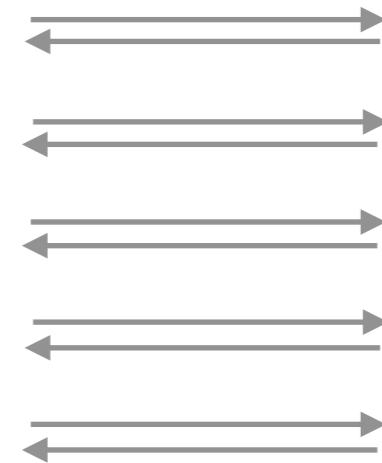
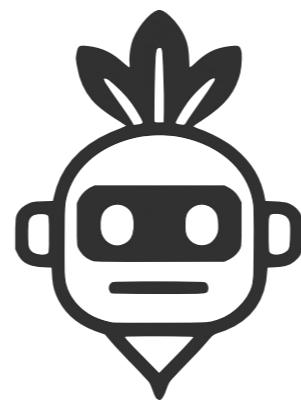
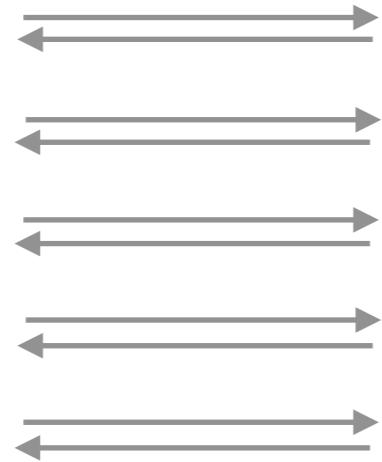


Routing



4

Tricky Scenario



Summary

ammomahdavikia.github.io

mahdavikiaamirmohammad@gmail.com

