Firebase

SQLite

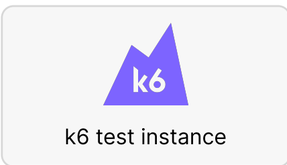
k6

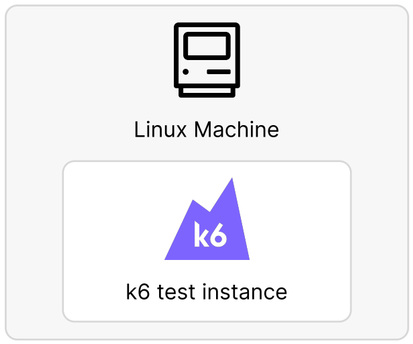
TS

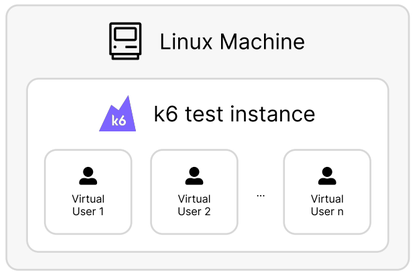
Better Stack

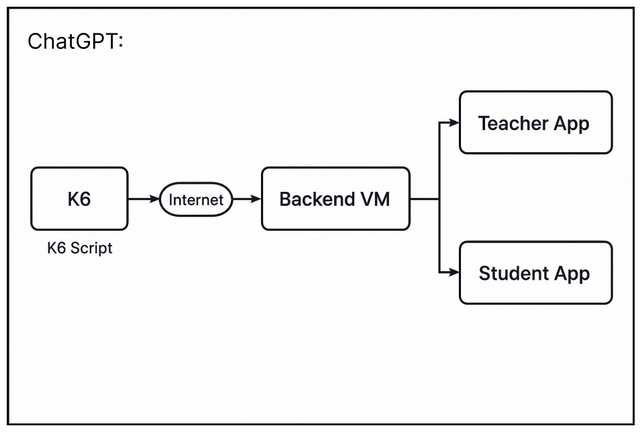
PostHog

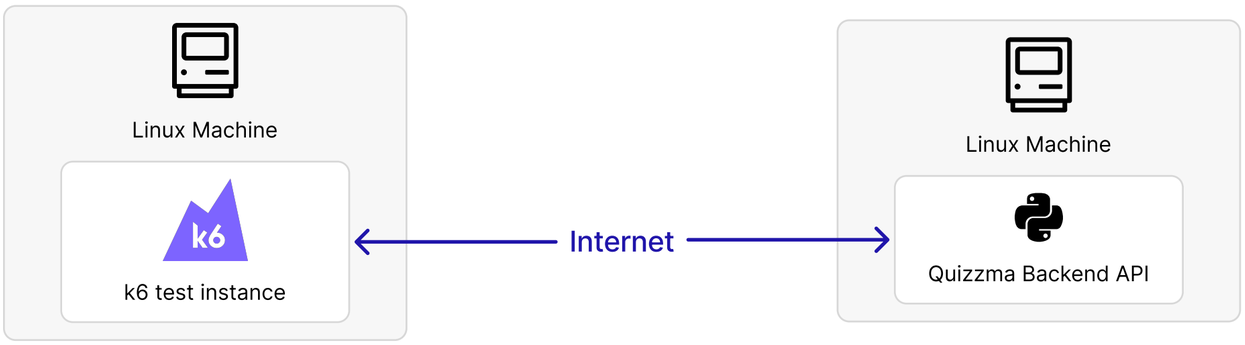
aws

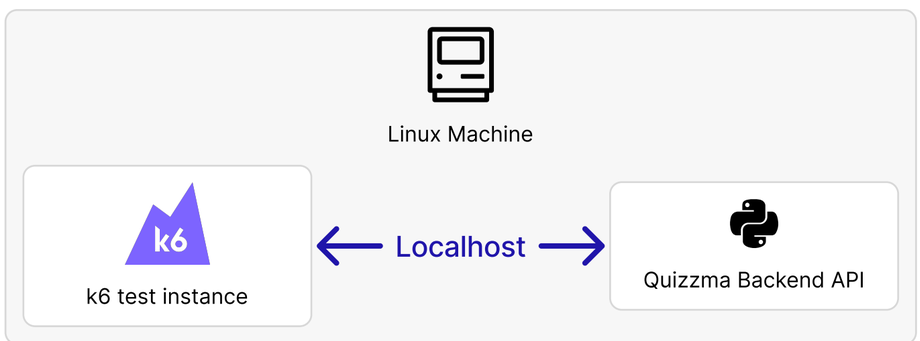

k6
k6 test instance

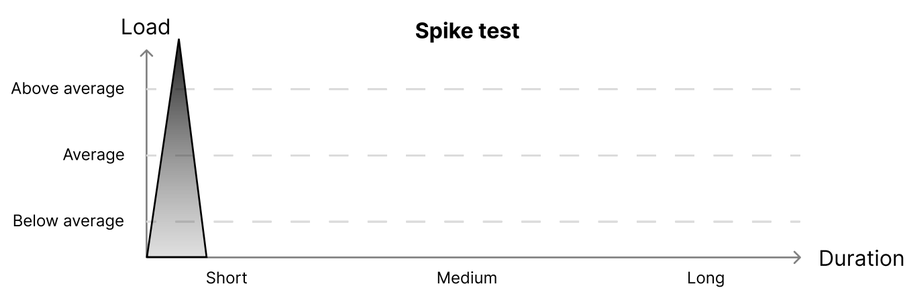

Linux Machine
k6
k6 test instance

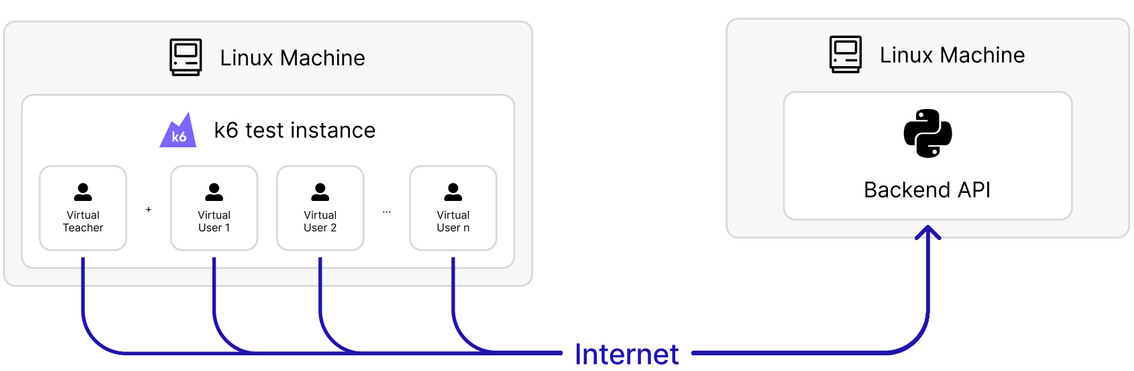

Linux Machine
k6
k6 test instance
Virtual User 1
Virtual User 2
...
Virtual User n

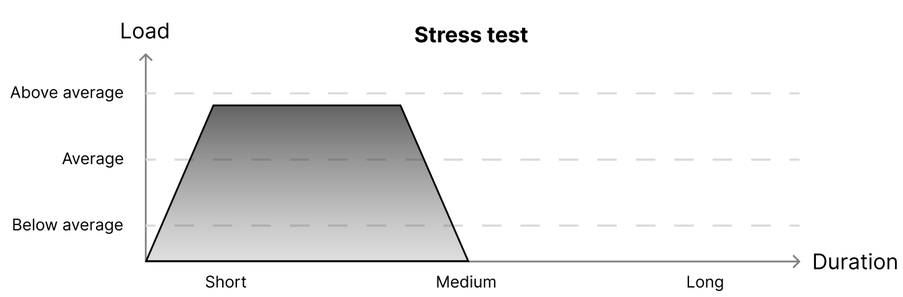

ChatGPT:
K6
K6 Script
Internet
Backend VM
Teacher App
Student App

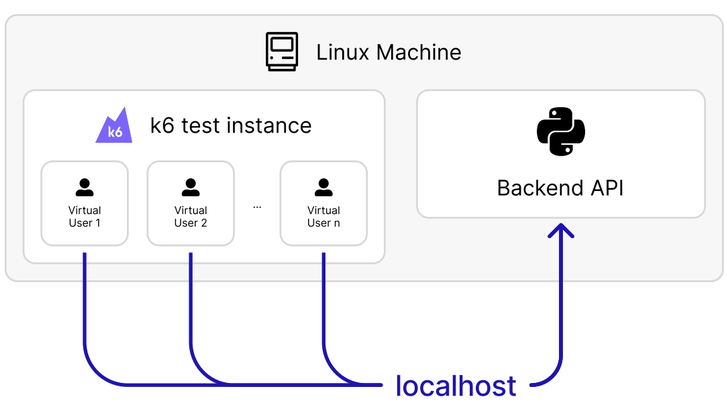

Linux Machine
k6
k6 test instance
Internet
Linux Machine
Quizzma Backend API

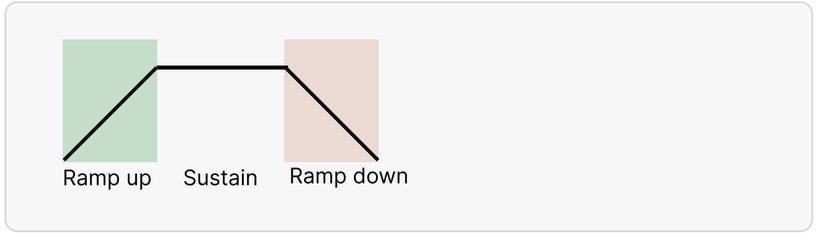

Linux Machine
k6
k6 test instance
Localhost
Quizzma Backend API

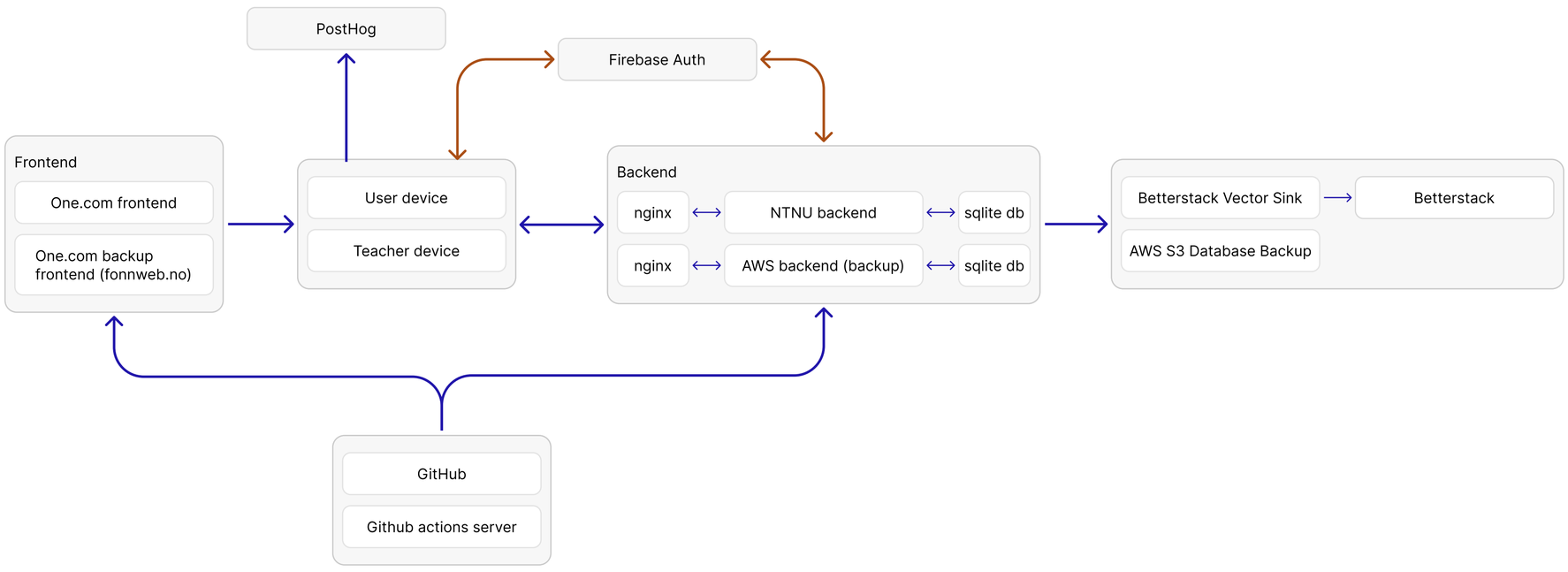

Load
Spike test
Above average
Average
Below average
Duration
Short
Medium
Long

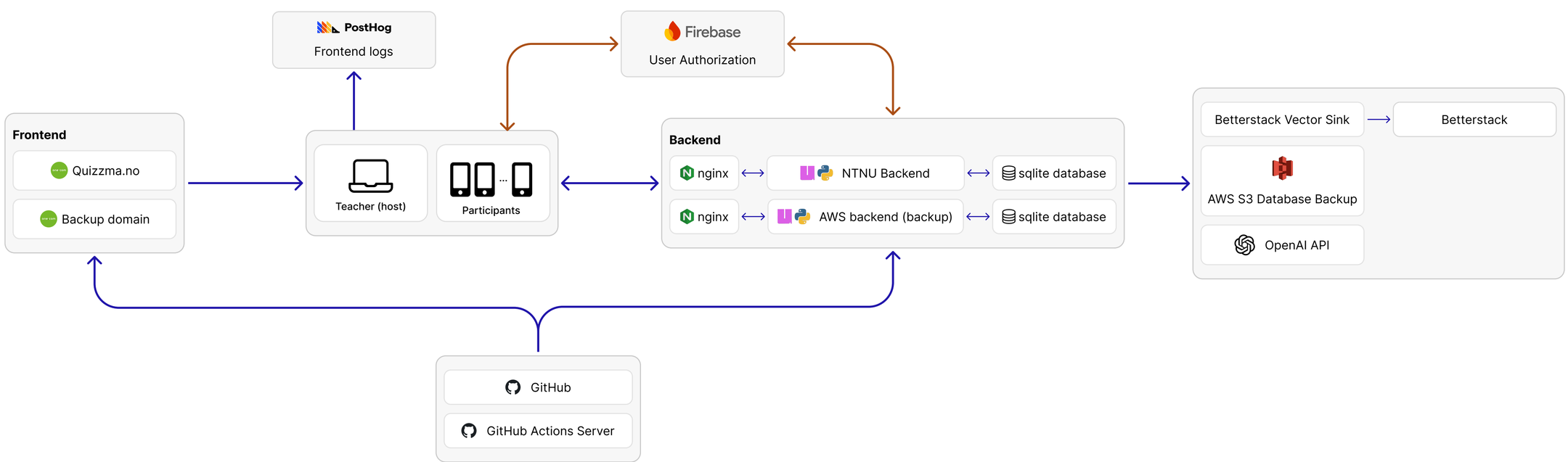

Linux Machine
k6
k6 test instance
Virtual Teacher
+
Virtual User 1
Virtual User 2
...
Virtual User n
Linux Machine
Backend API
Internet

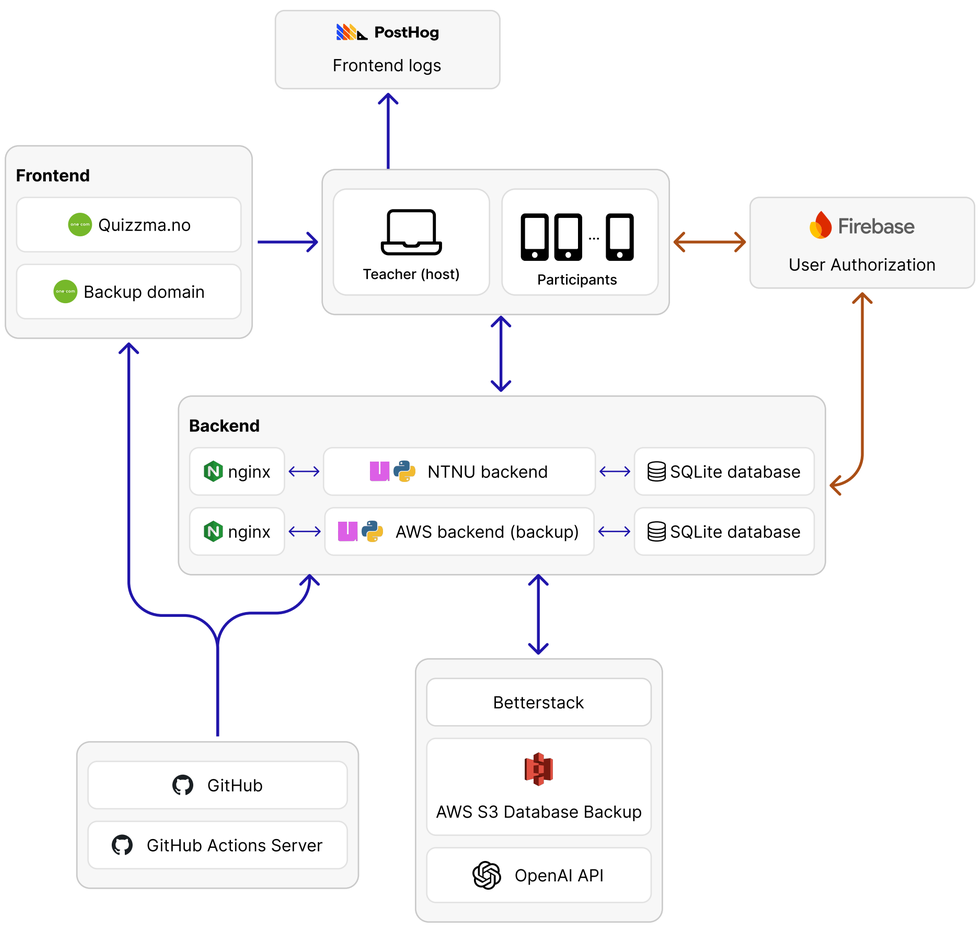

Load
Stress test
Above average
Average
Below average
Duration
Short
Medium
Long

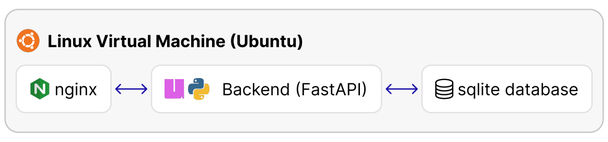

Linux Machine
k6
k6 test instance
Virtual User 1
Virtual User 2
...
Virtual User n
Backend API
localhost

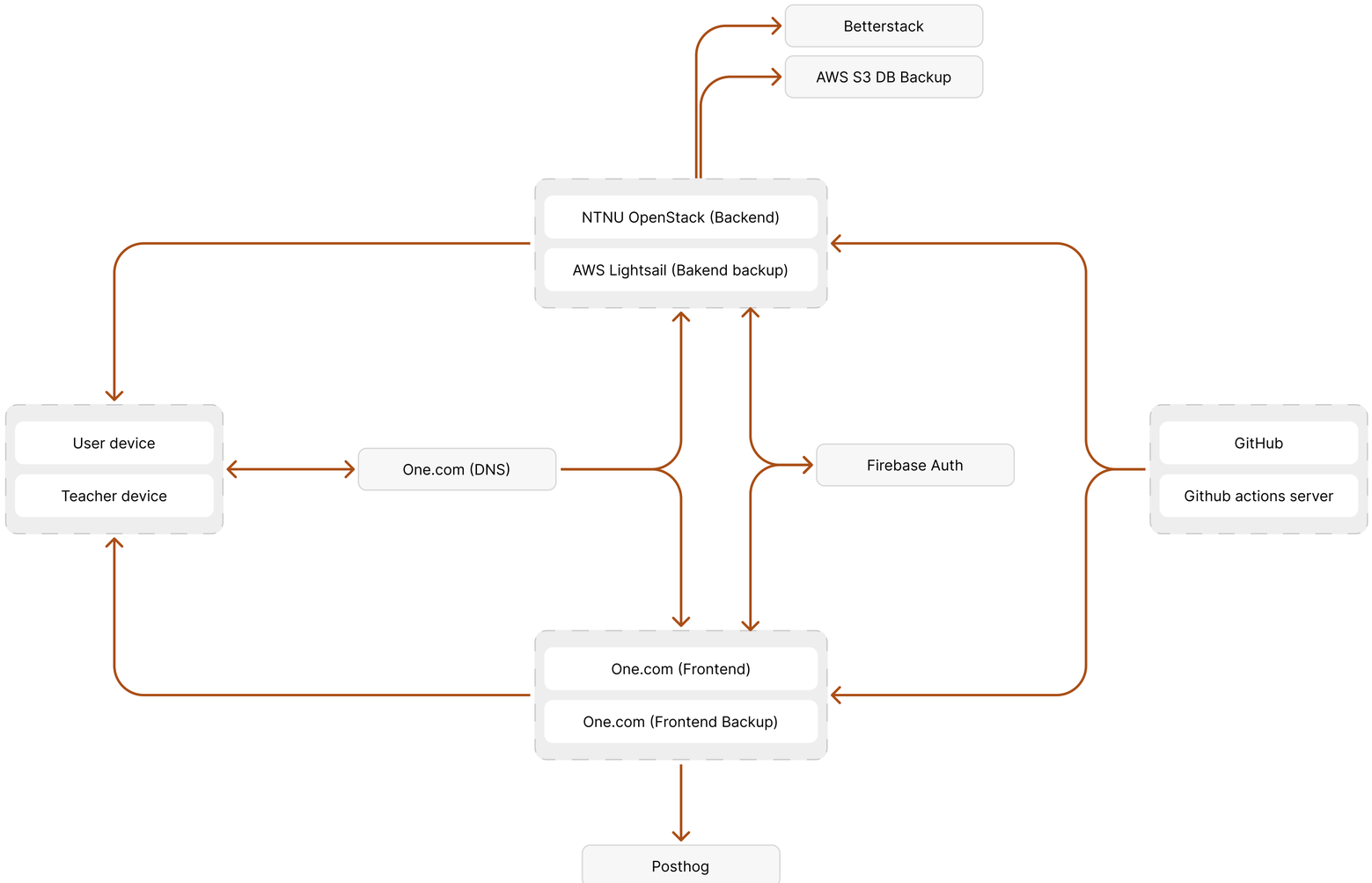

Ramp up
Sustain
Ramp down

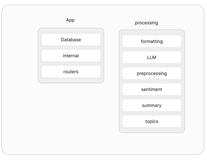

PostHog
Firebase Auth
Frontend
One.com frontend
One.com backup frontend (fonnweb.no)
User device
Teacher device
Backend
nginx
NTNU backend
sqlite db
nginx
AWS backend (backup)
sqlite db
Betterstack Vector Sink
Betterstack
AWS S3 Database Backup
GitHub
Github actions server

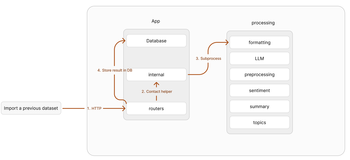

PostHog
Frontend logs
Firebase
User Authorization
Frontend
Quizzma.no
Backup domain
Teacher (host)
Participants
Backend
nginx
NTNU Backend
sqlite database
nginx
AWS backend (backup)
sqlite database
Betterstack Vector Sink
Betterstack
AWS S3 Database Backup
OpenAI API
GitHub
GitHub Actions Server


PostHog
Frontend logs
Frontend
Quizzma.no
Backup domain
Teacher (host)
Participants
Firebase
User Authorization
Backend
nginx
NTNU backend
SQLite database
nginx
AWS backend (backup)
SQLite database
Betterstack
AWS S3 Database Backup
OpenAI API
GitHub
GitHub Actions Server

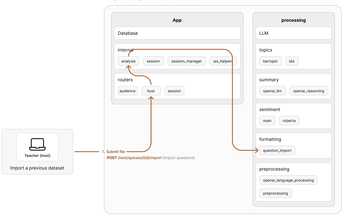

Linux Virtual Machine (Ubuntu)
nginx
Backend (FastAPI)
sqlite database


Betterstack
AWS S3 DB Backup
NTNU OpenStack (Backend)
AWS Lightsail (Bakend backup)
User device
Teacher device
One.com (DNS)
Firebase Auth
GitHub
Github actions server
One.com (Frontend)
One.com (Frontend Backup)
Posthog

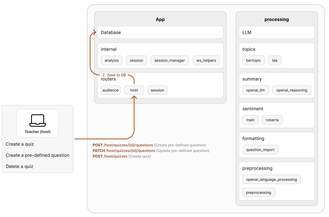




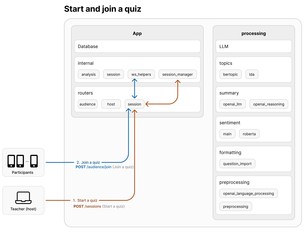
Scenario
Import a previous dataset



Scenario
Prepare a quiz

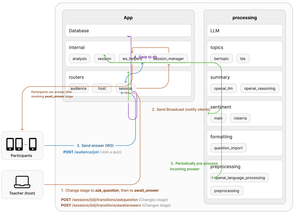


Scenario
Start and join a quiz

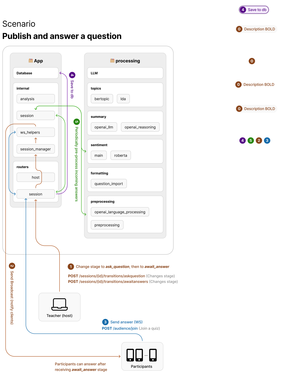


Scenario
Start and join a quiz

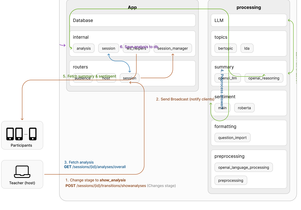
Scenario
Publish and answer a question



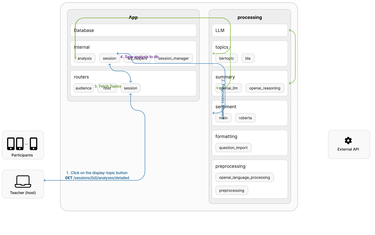
Scenario
Publish and answer a question

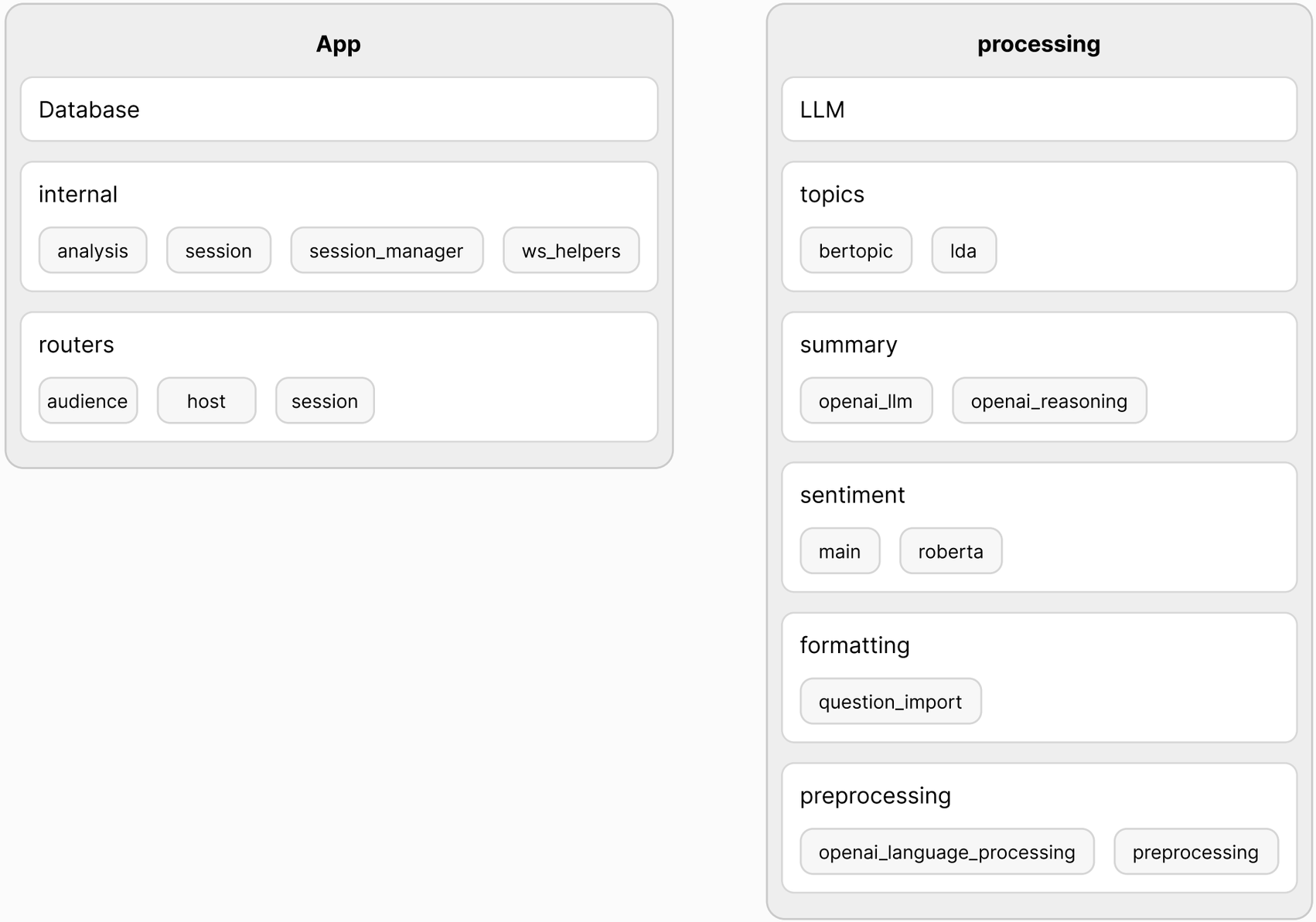


Scenario
Analyze answers



Scenario
Analyze answers (with topics)




App
Database
internal
analysis
session
session_manager
ws_helpers
routers
audience
host
session
processing
LLM
topics
bertopic
lda
summary
openai_llm
openai_reasoning
sentiment
main
roberta
formatting
question_import
preprocessing
openai_language_processing
preprocessing

Flights

id	int	PK
username	varchar	

id	timestamp	NULL	PK

varchar

text

int

date

timestamp

decimal

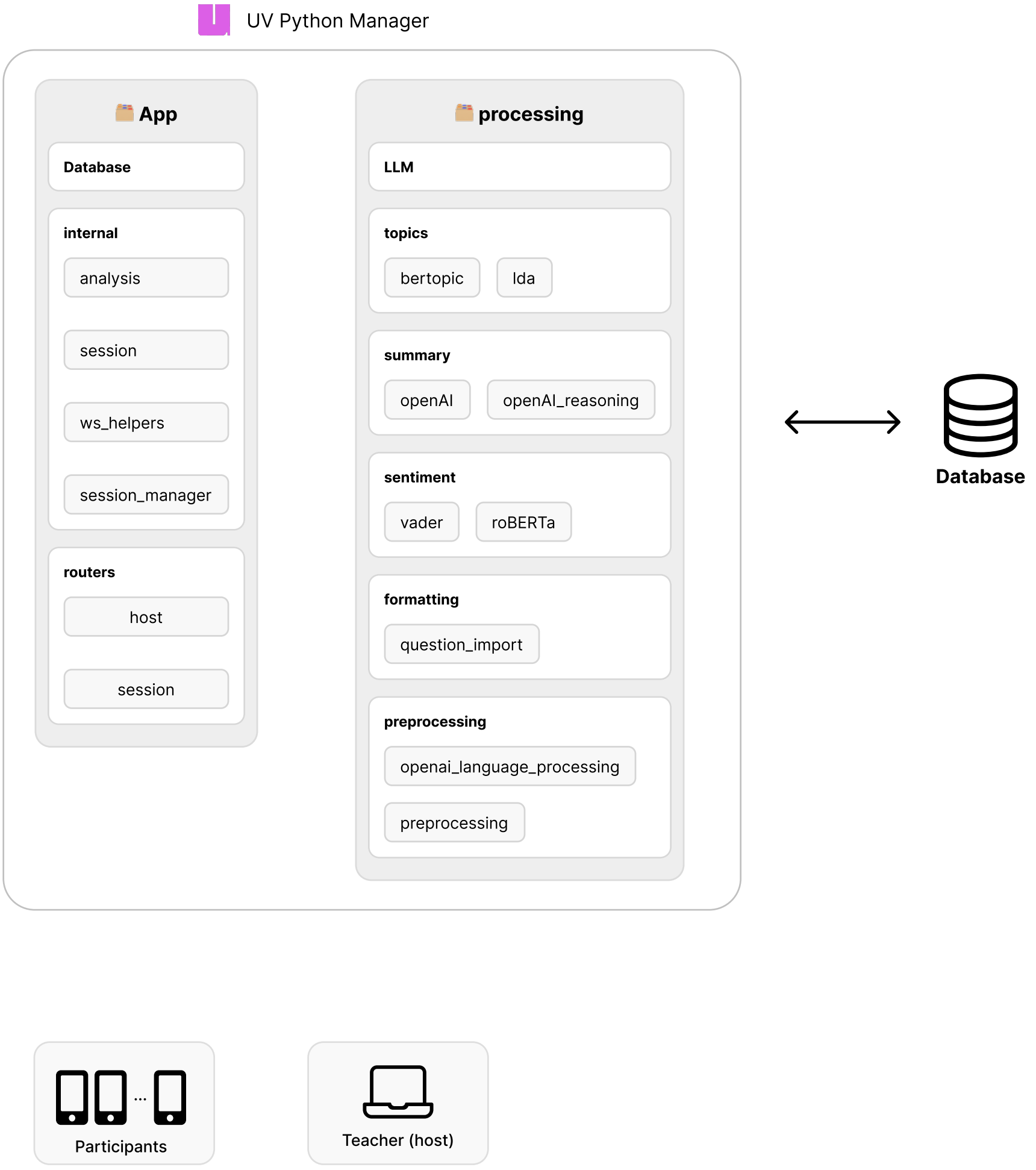
PK

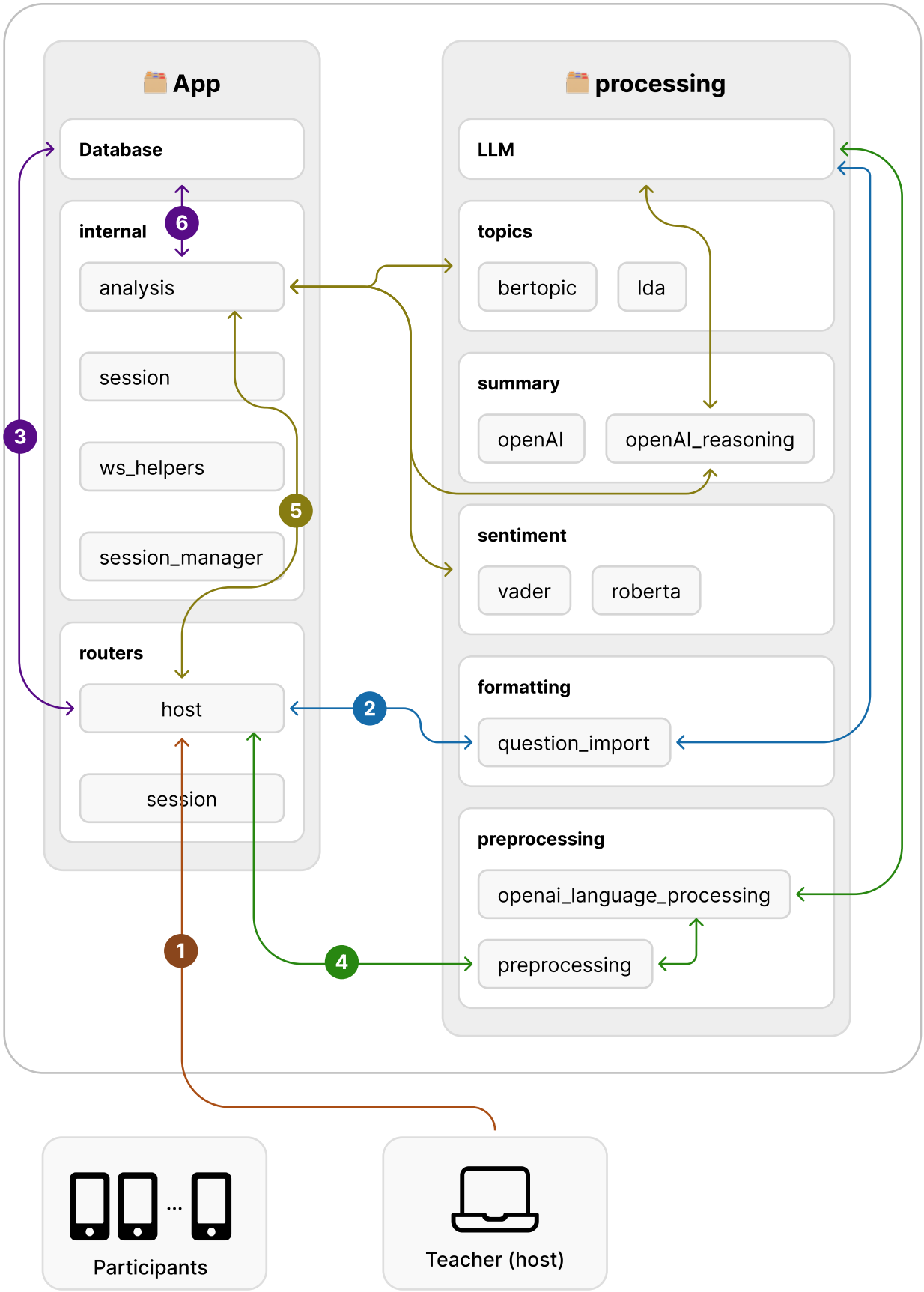
FK

NULL

NOT NULL

U


UV Python Manager
App
Database
internal
analysis
session
ws_helpers
session_manager
routers
host
session
processing
LLM
topics
bertopic
lda
summary
openAI
openAI_reasoning
sentiment
vader
roBERTa
formatting
question_import
preprocessing
openai_language_processing
preprocessing
Database
Participants
Teacher (host)



1 Submit file **POST** /host/quizzes/{id}/import (Import questions)

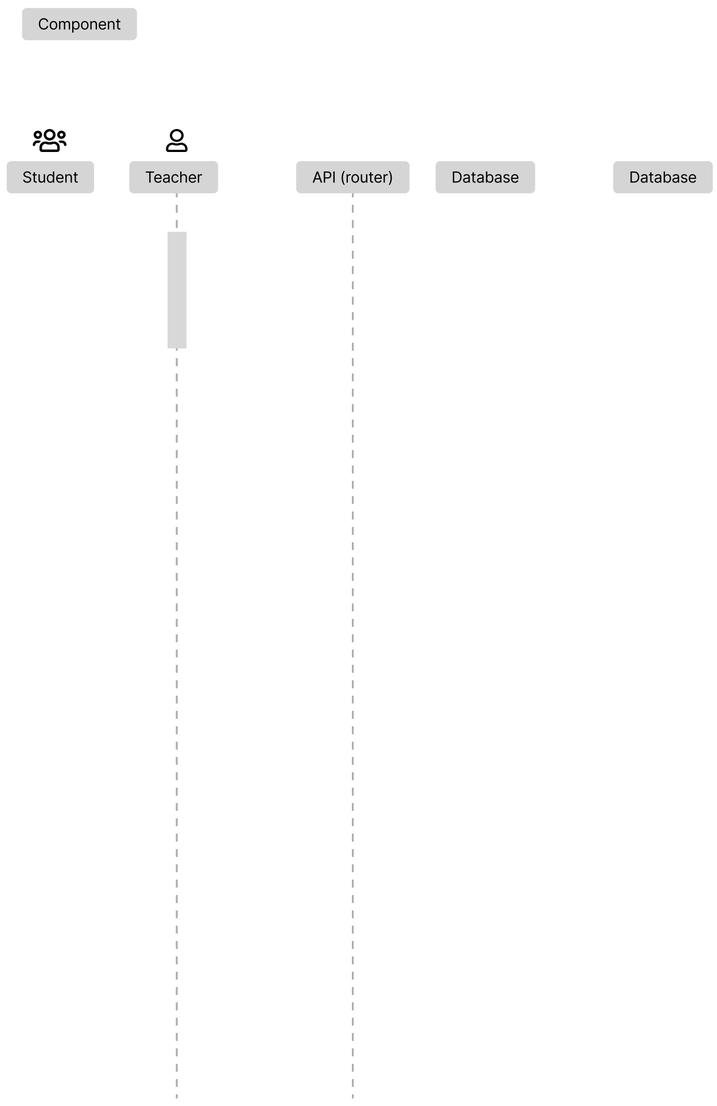
2 Perform import formatting

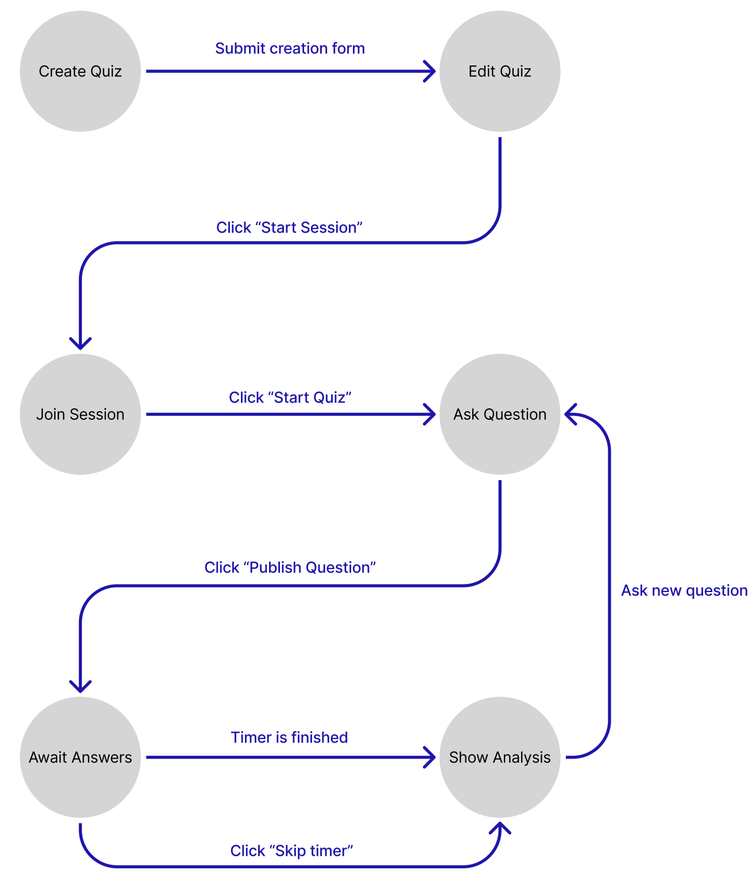
3 Save formatted questions and responses to database

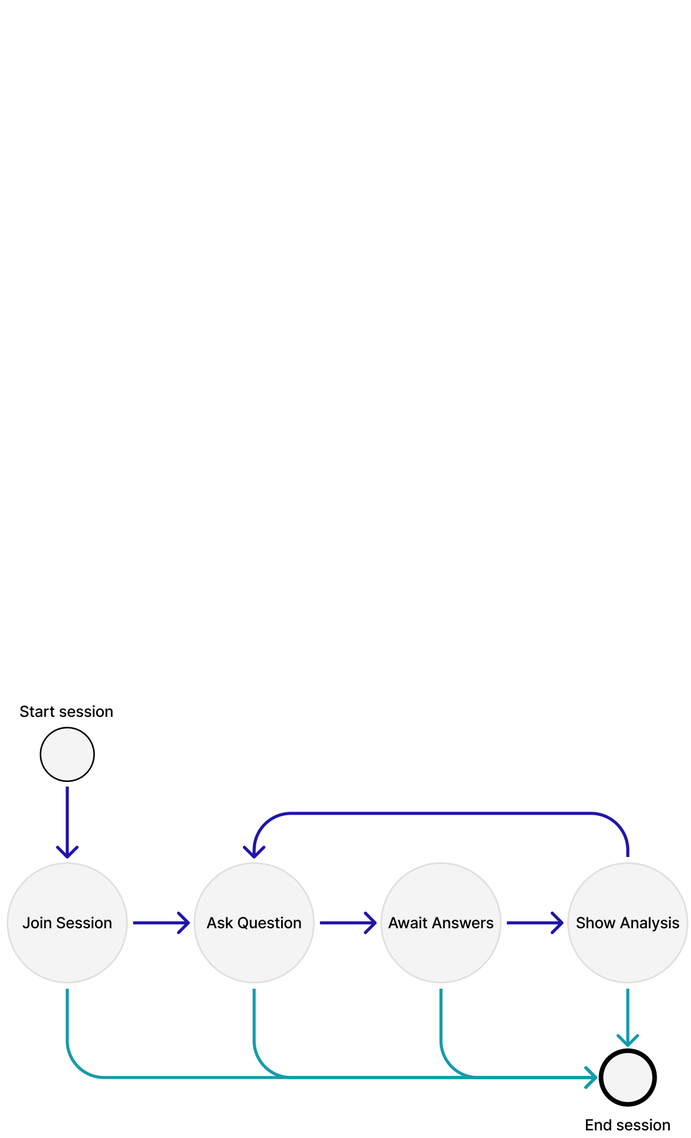
4 Preprocess answers

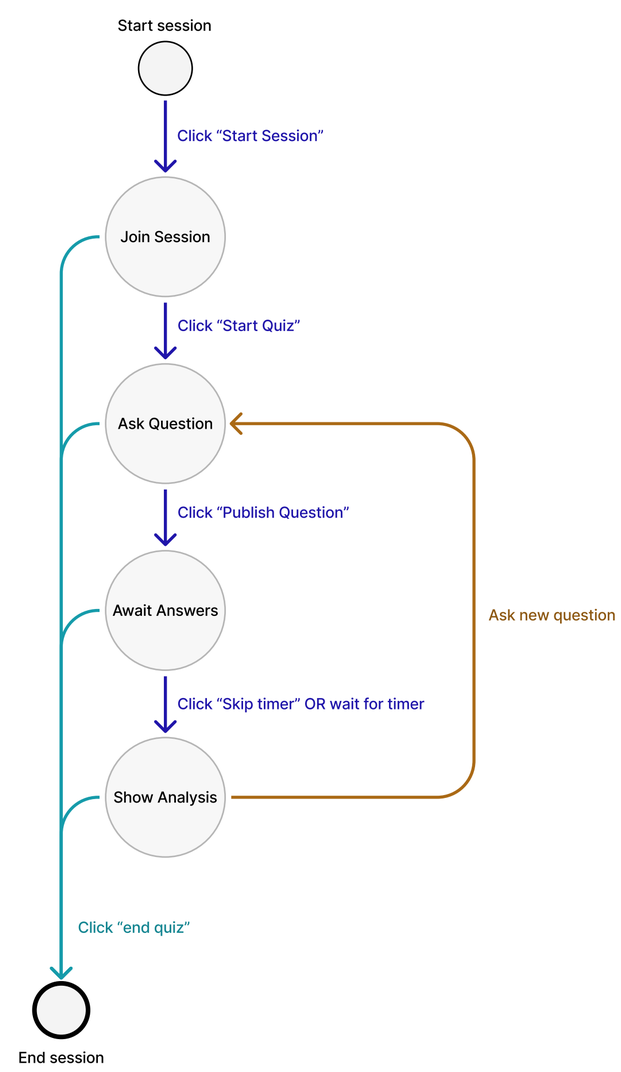
5 Analyze answers

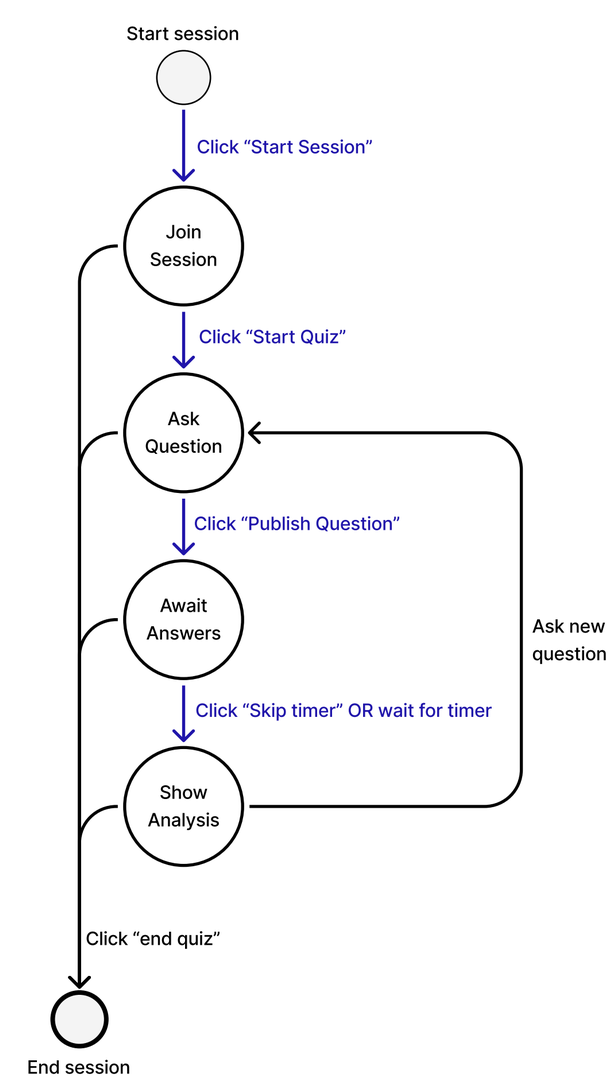
6 Save analysis to database

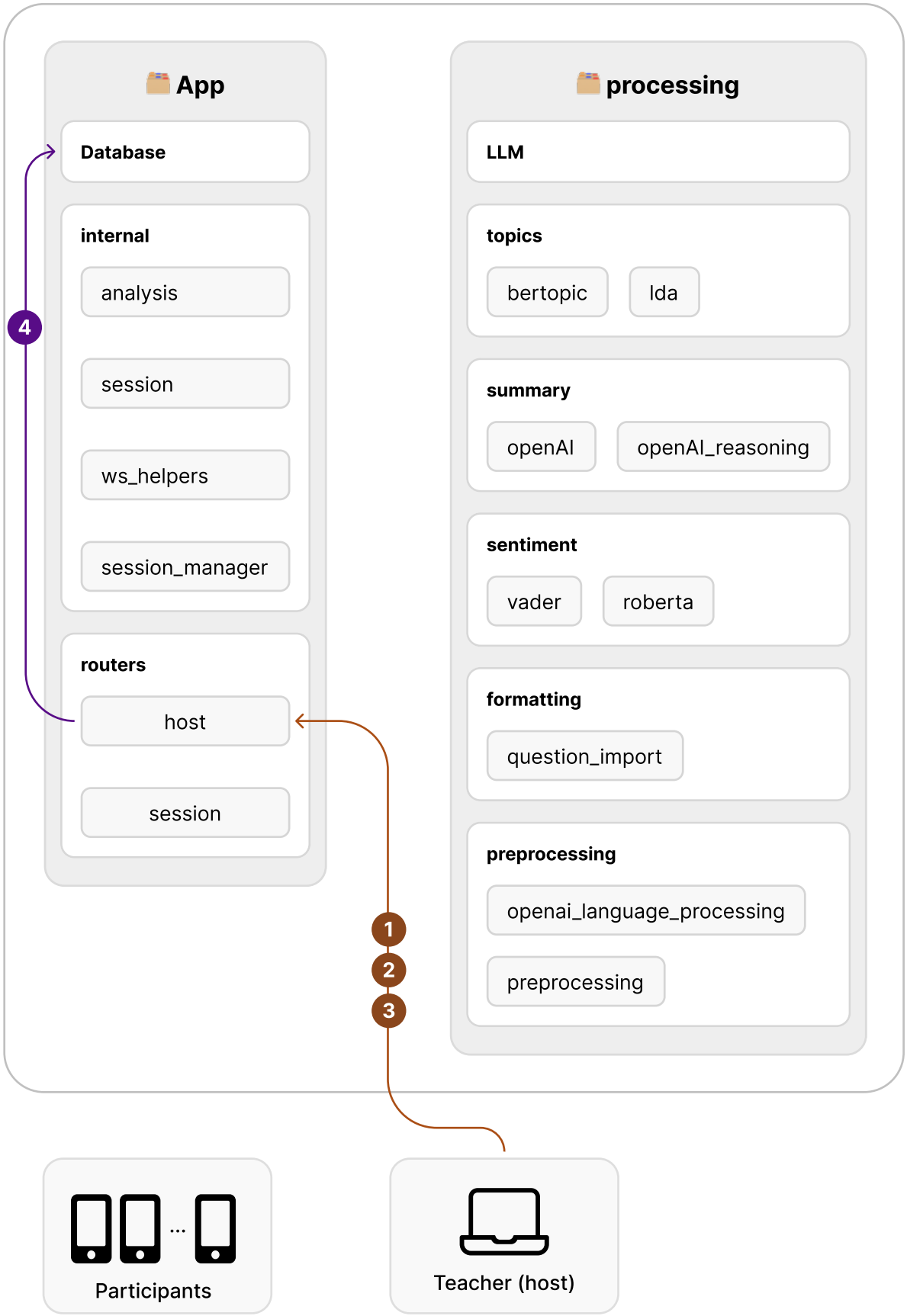

Component
Student
Teacher
API (router)
Database
Database


Create Quiz
Submit creation form
Edit Quiz
Click “Start Session”
Join Session
Click “Start Quiz”
Ask Question
Click “Publish Question”
Ask new question
Await Answers
Timer is finished
Show Analysis
Click “Skip timer”


Start session
Join Session
Ask Question
Await Answers
Show Analysis
End session


Start session
Click “Start Session”
Join Session
Click “Start Quiz”
Ask Question
Click “Publish Question”
Ask new question
Await Answers
Click “Skip timer” OR wait for timer
Show Analysis
Click “end quiz”
End session


Start session
Click “Start Session”
Join Session
Click “Start Quiz”
Ask Question
Click “Publish Question”
Ask new question
Await Answers
Click “Skip timer” OR wait for timer
Show Analysis
Click “end quiz”
End session

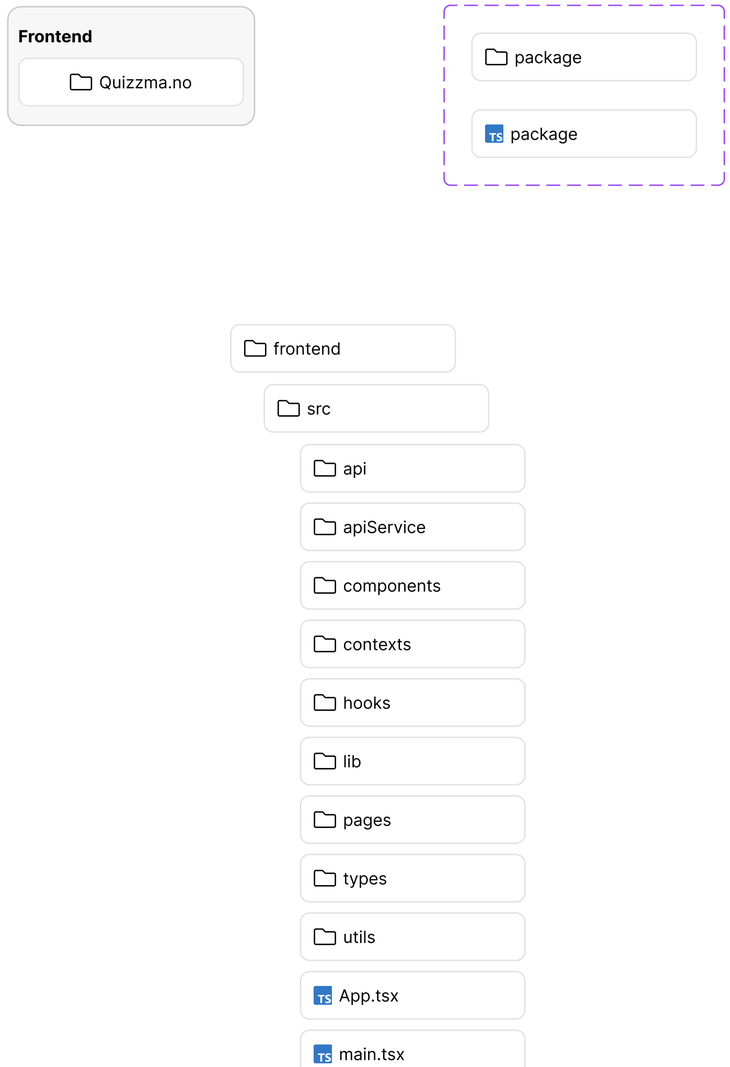


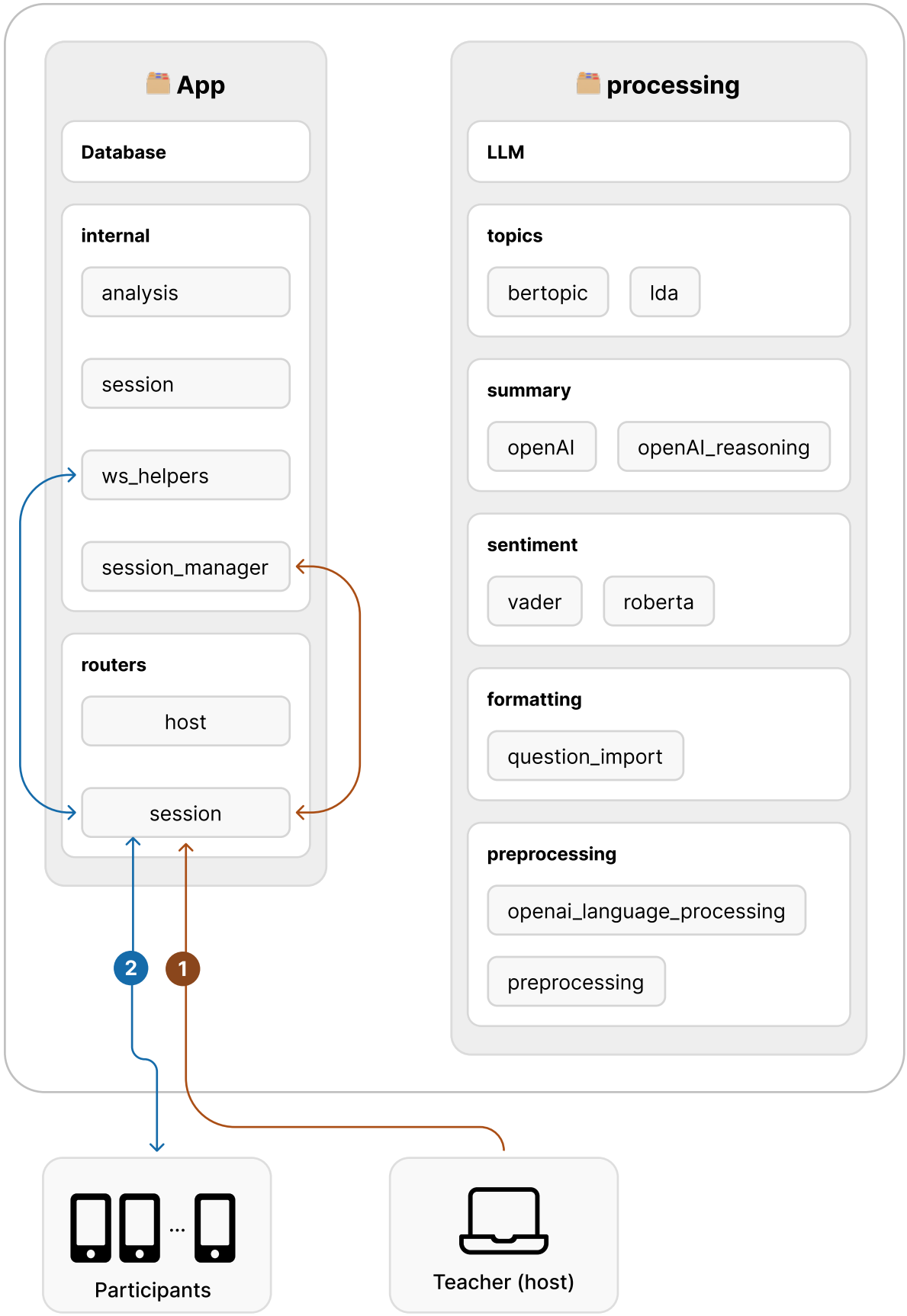
1 Create a quiz **POST** /host/quizzes

2 Create pre-defined question **POST** /host/quizzes/{id}/questions

3 Update pre-defined question **PATCH** /host/quizzes/{id}/questions

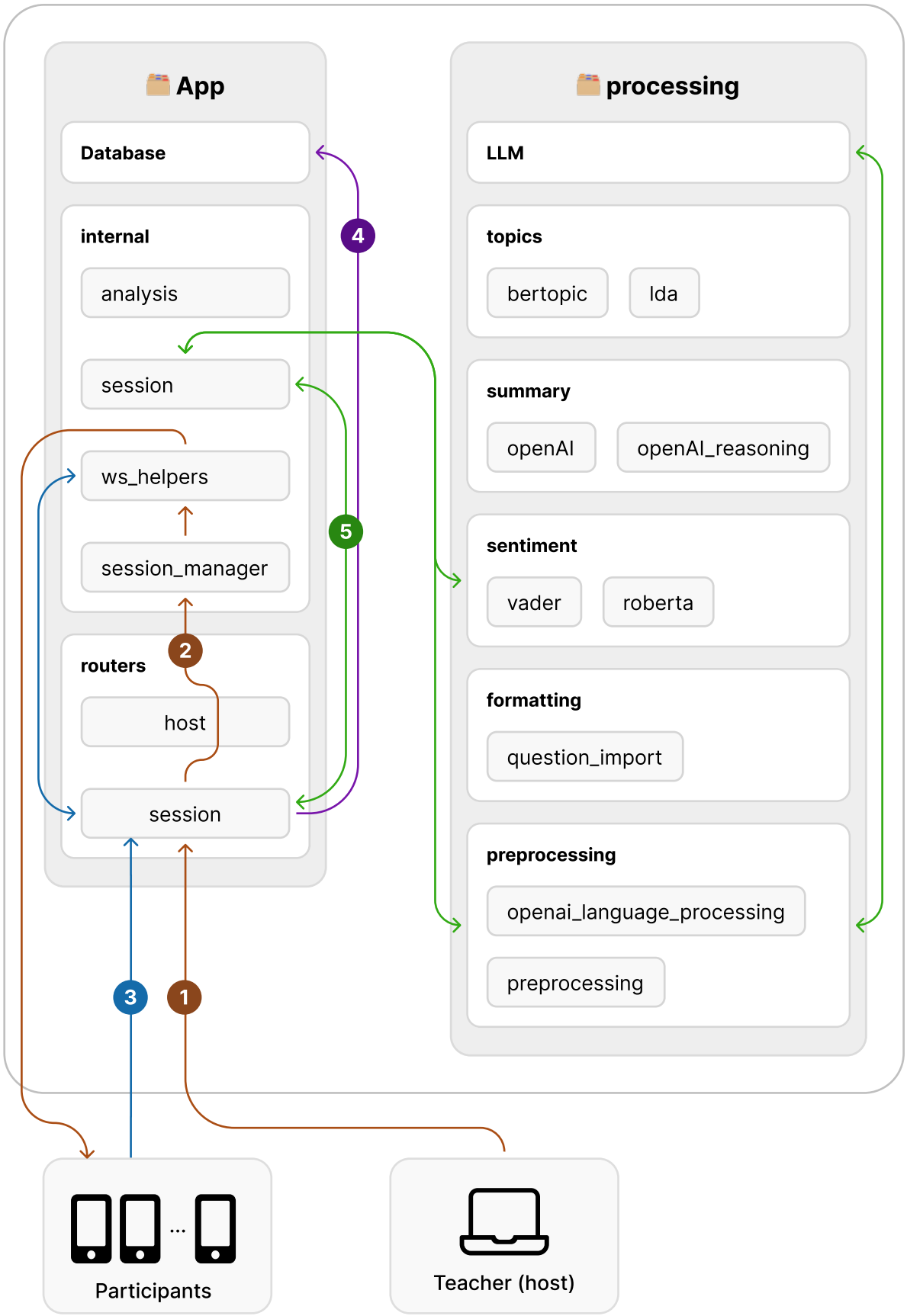
4 Save to database


Frontend
Quizzma.no
package
package
frontend
src
api
apiService
components
contexts
hooks
lib
pages
types
utils
App.tsx
main.tsx



1 Start a quiz **POST** /sessions

2 Join a quiz **POST** /sessions/ws/{session_id}



1 Set stage to ***ask_question***, then to ***await_answer*** **POST** /sessions/{id}/transitions/{stage}

2 Send Broadcast (notify clients) Participants can answer in ***await_answer*** stage

3 Participant submits answer **POST** /sessions/ws/{session_id} (Join a quiz, create WS)

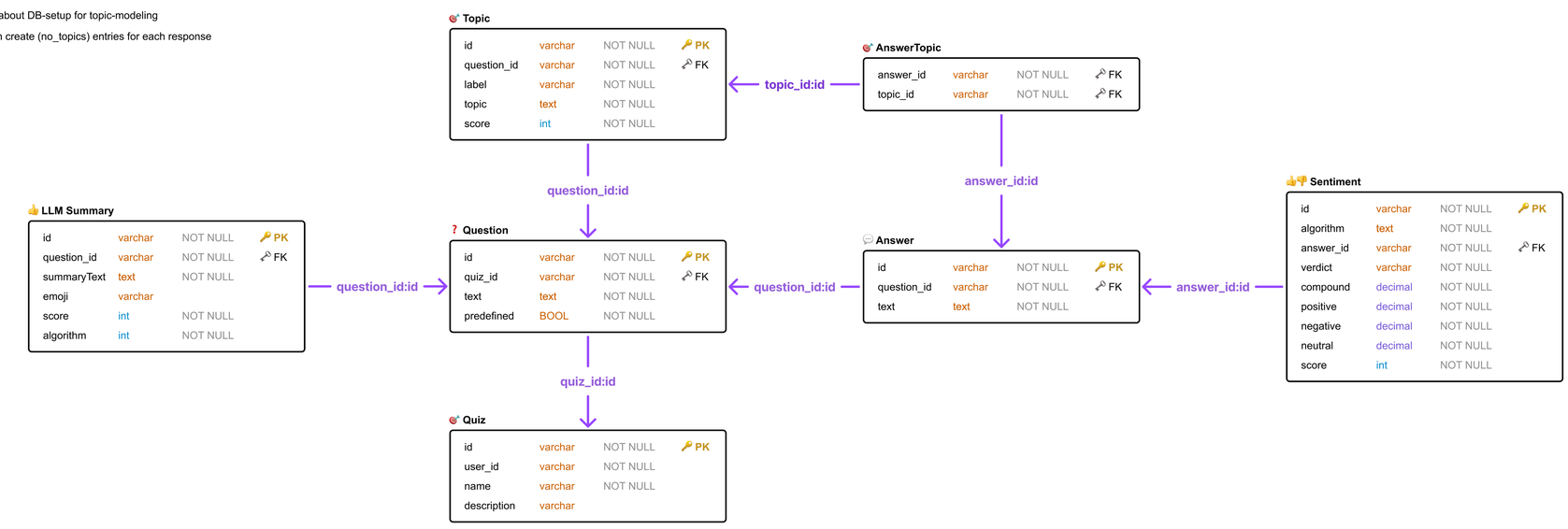
4 Save to database

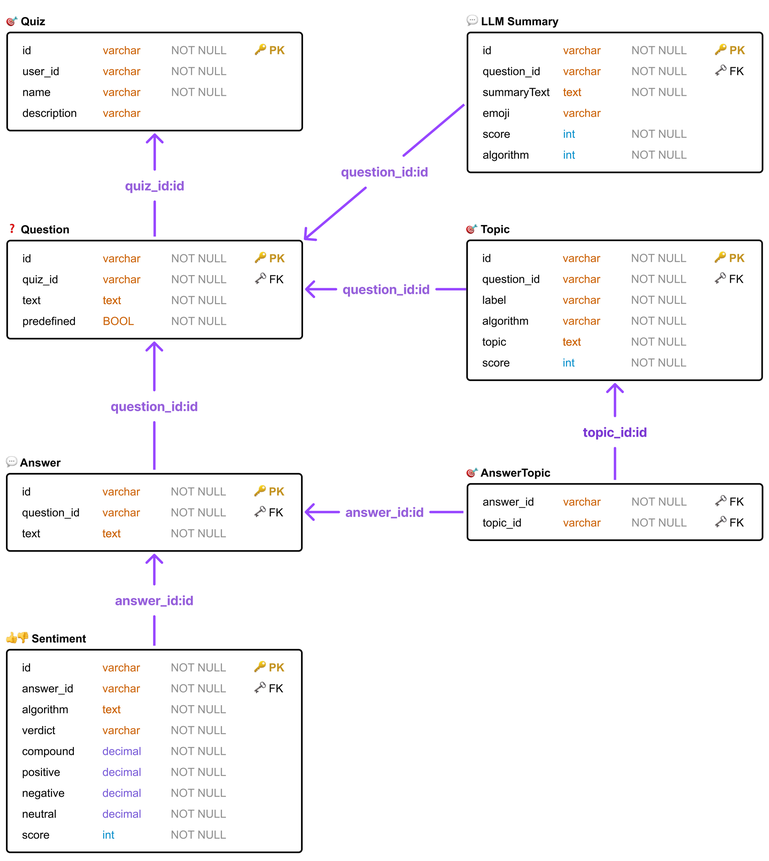
5 Periodically pre-process incoming answers

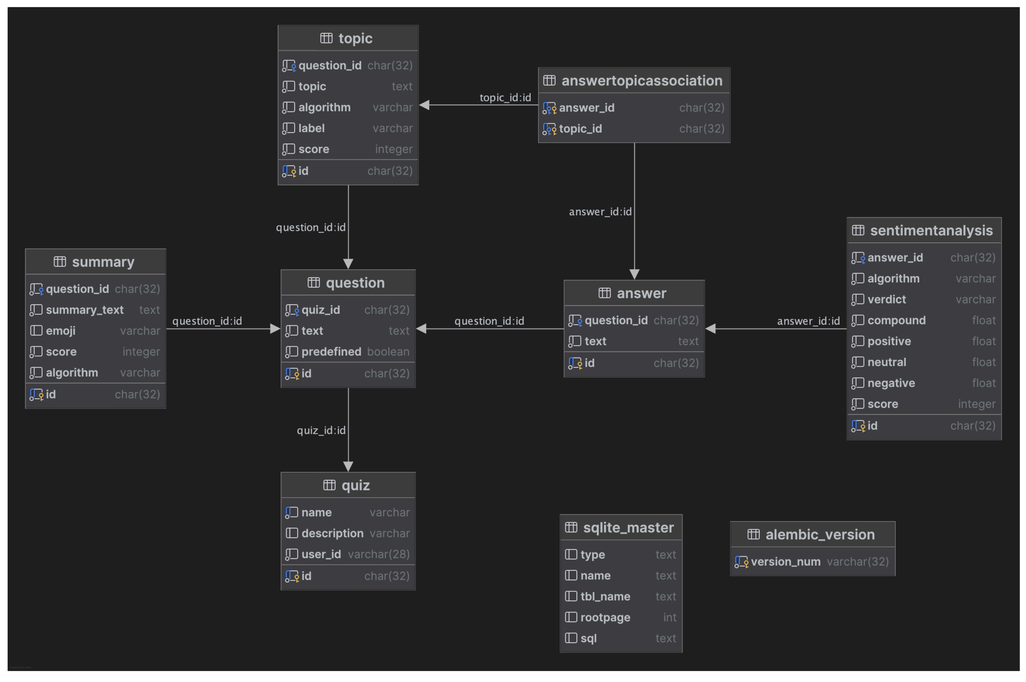
Topic is a string, formatted as comma-separated-values:
0 money, reducing, supply, printing, ...

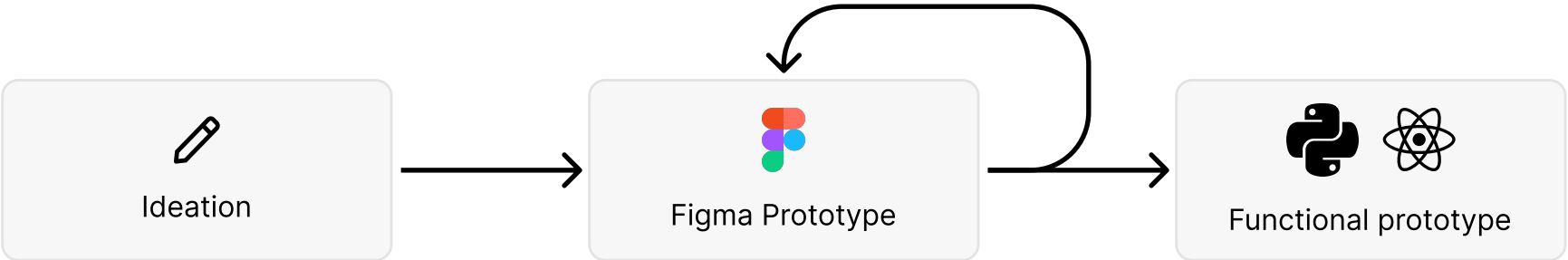
unsure about DB-setup for topic-modeling

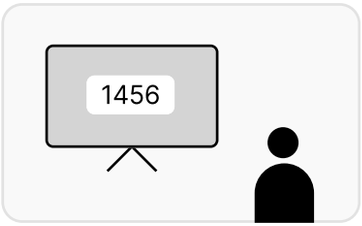
⚠️ This can create (no_topics) entries for each response



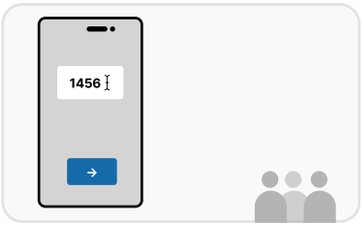




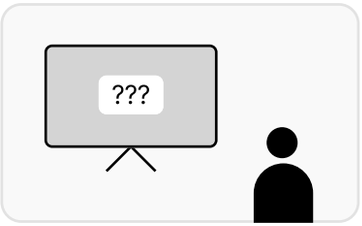

Ideation
Figma Prototype
Functional prototype



1 Create session code

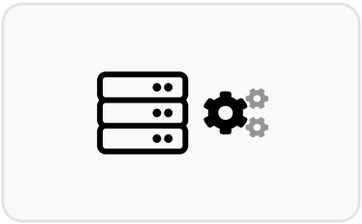


2 Participants join

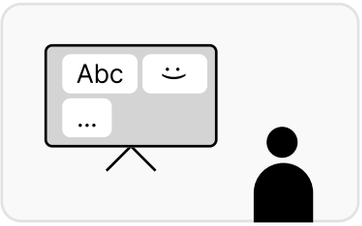


3 Educator publishes question

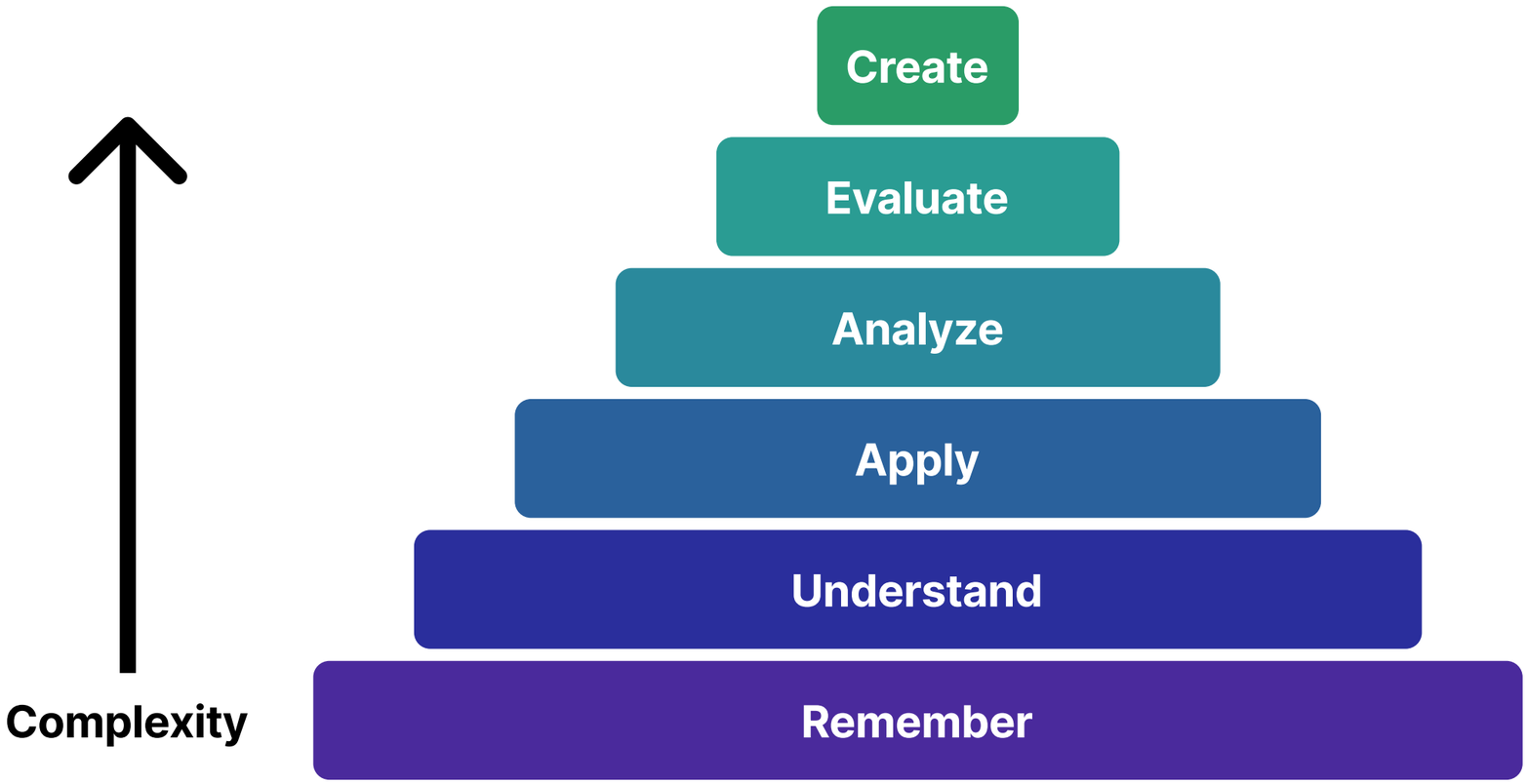
4 Participants submits answers

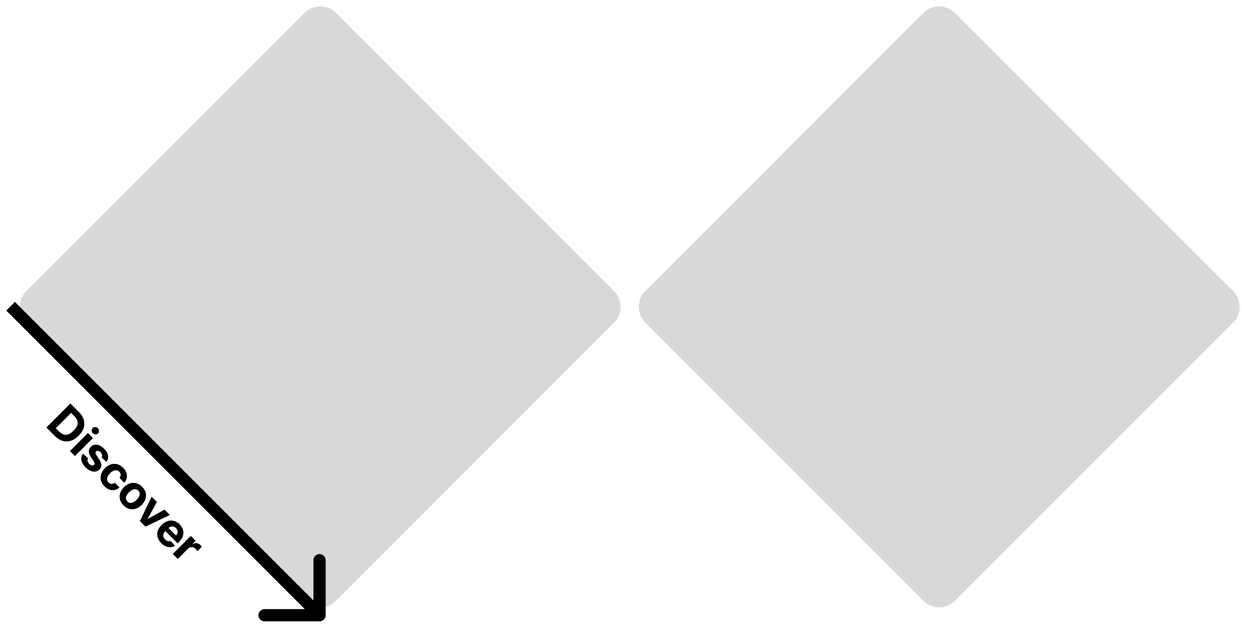


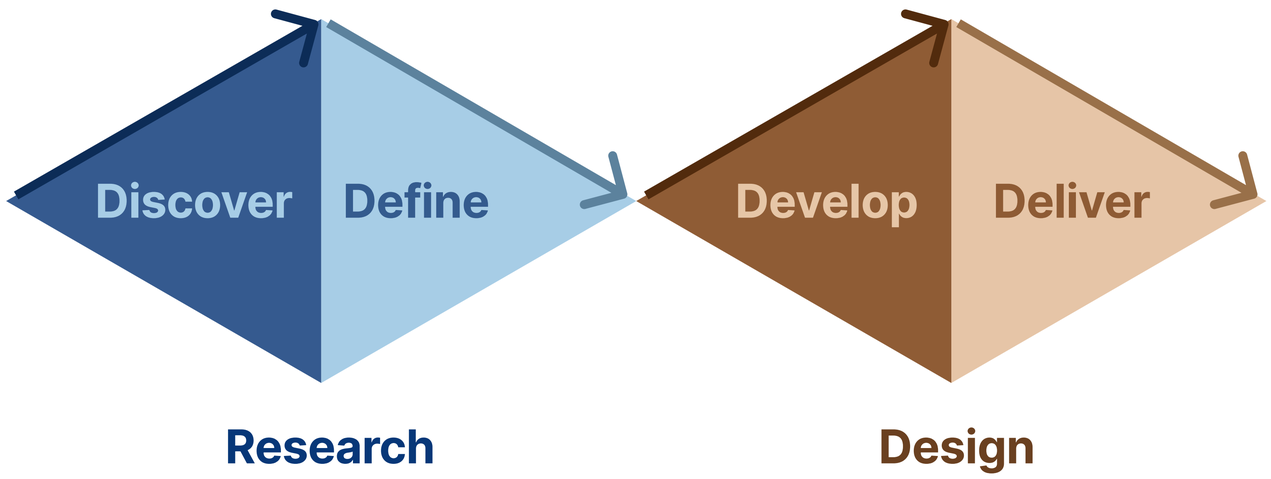
5 Backend processes answers

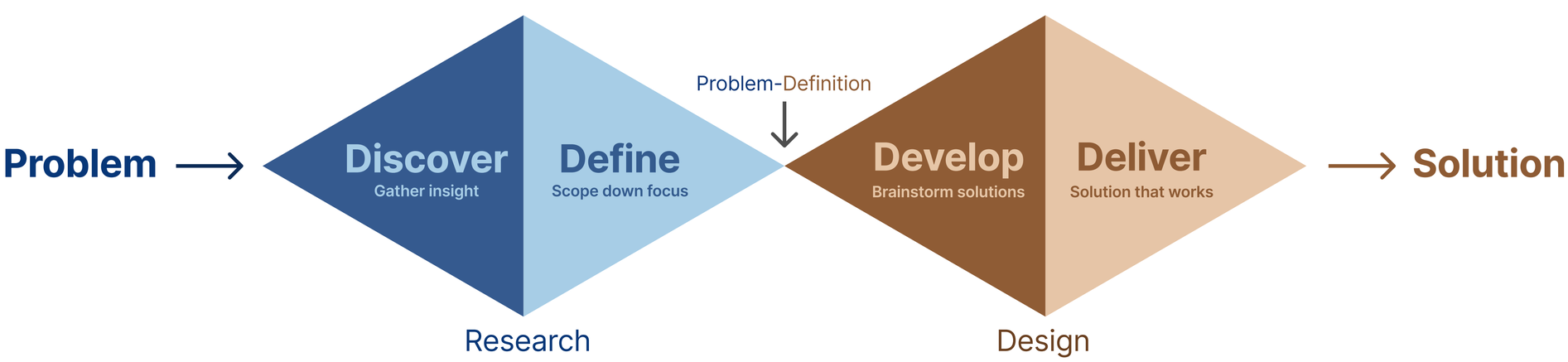


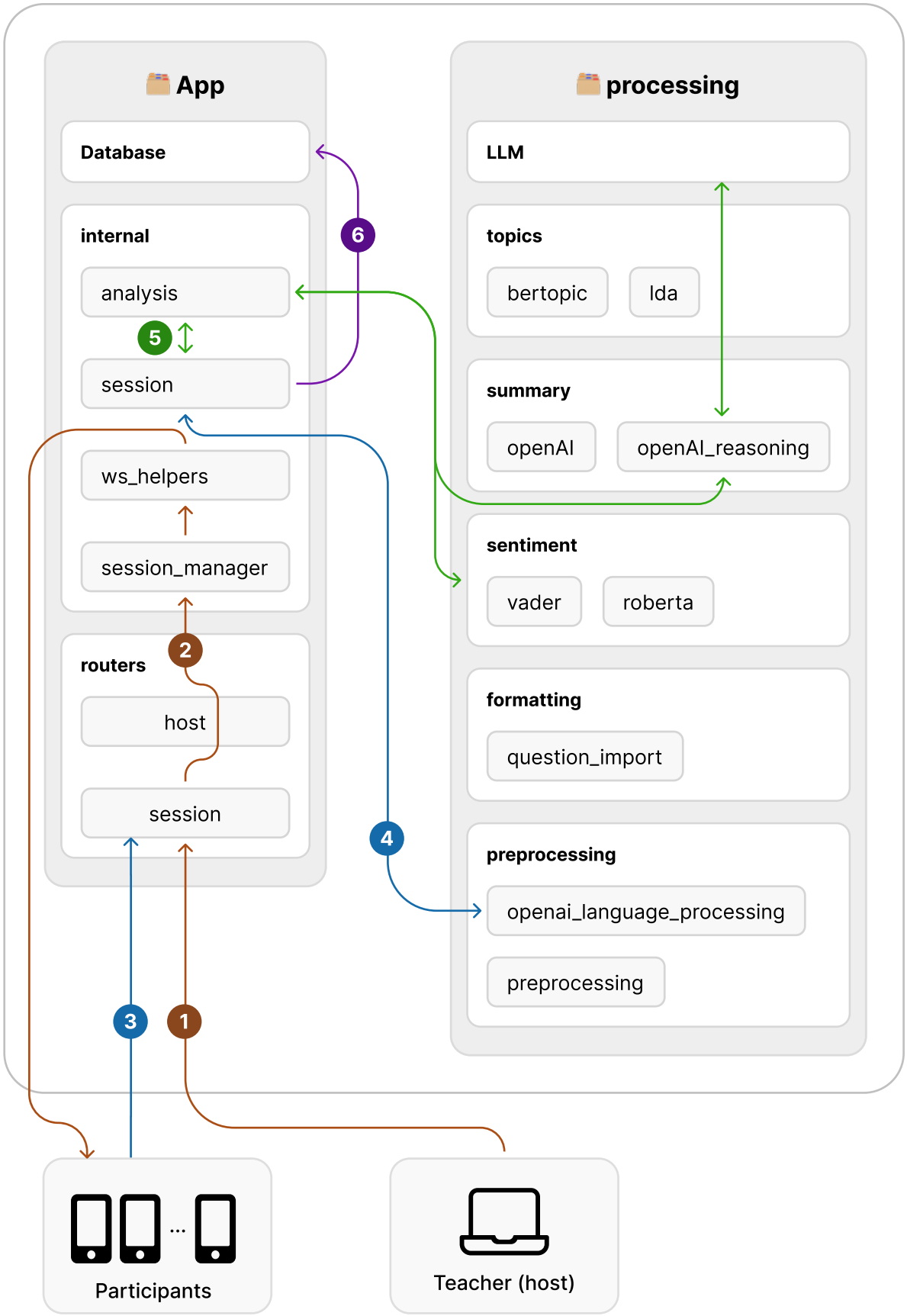
6 Educator reviews analysis


Create
Evaluate
Analyze
Apply
Understand
Remember
Complexity


Discover


Discover
Define
Develop
Deliver
Research
Design


Problem
Discover
Gather insight
Define
Scope down focus
Problem-Definition
Develop
Brainstorm solutions
Deliver
Solution that works
Solution
Research
Design



1 Change stage to ***show_analysis*** **POST** /sessions/{id}/transitions/showanalyses

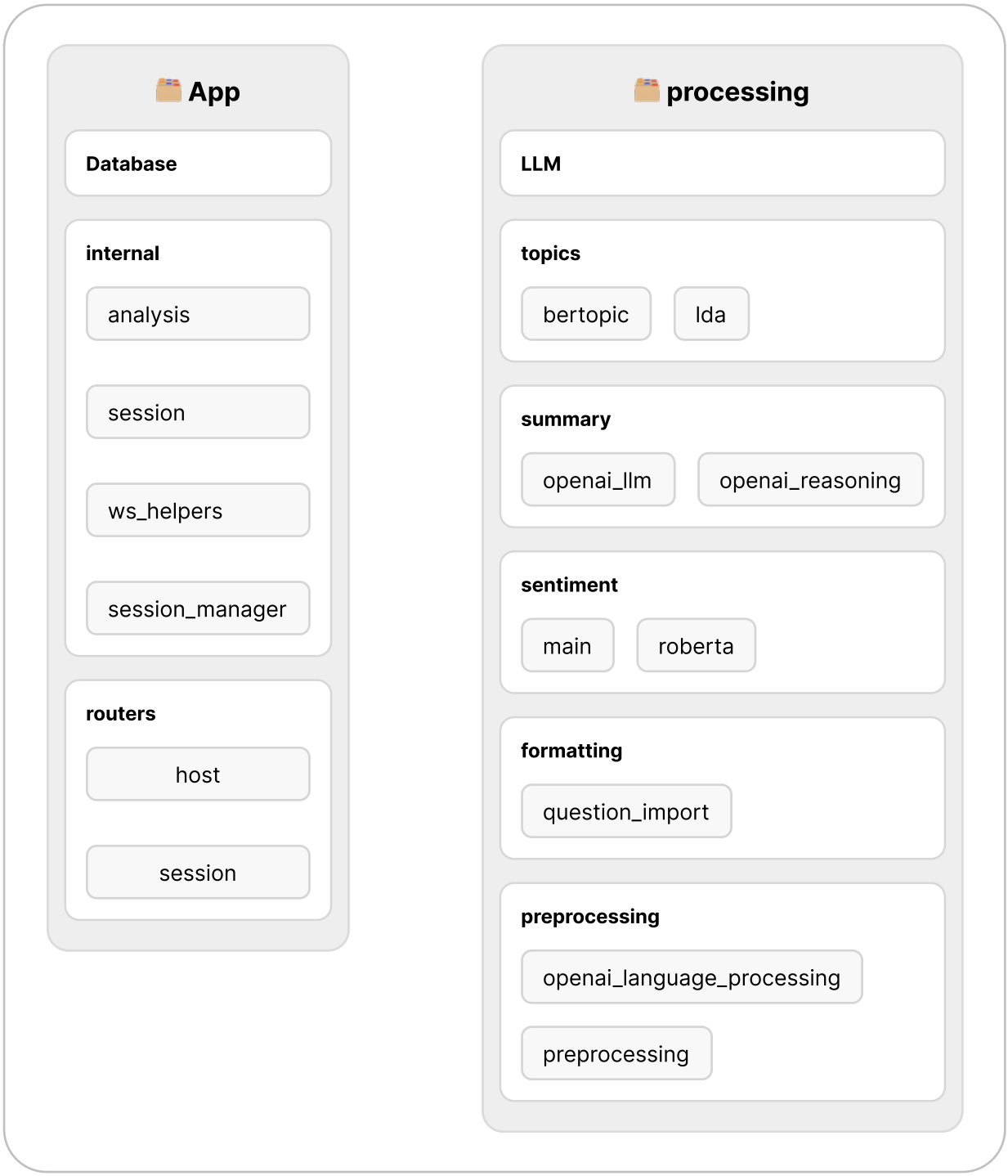
2 Send Broadcast (notify clients)

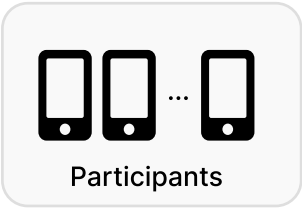
3 Fetch analysis **GET** /sessions/{id}/analyses/overall

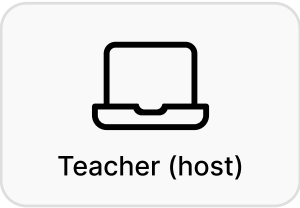
4 Preprocess answers

5 Fetch summary & sentiment

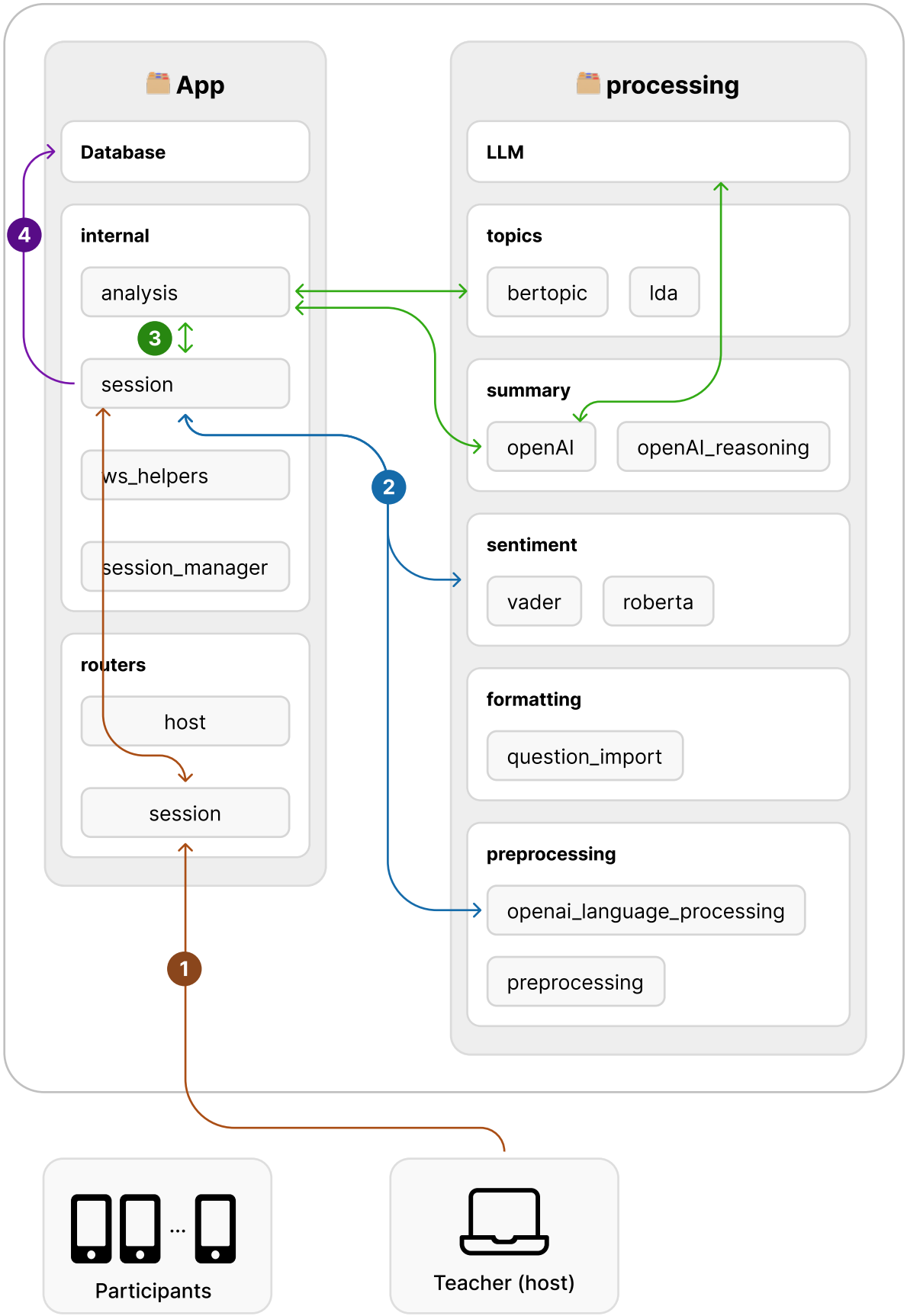
6 Save to database







1. Change stage to ***show_analysis*** — **POST** /sessions/{id}/transitions/showanalyses (Changes stage)
2. Send Broadcast (notify clients) — Participants can answer in ***await_answer*** stage
3. Participant submits answer — **POST** /audience/join (Join a quiz, create WS)
4. Save to database
5. Periodically pre-process incoming answers

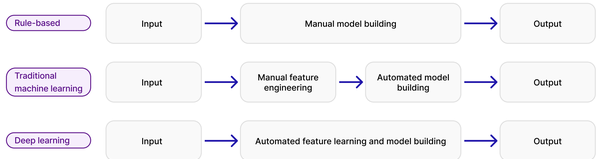


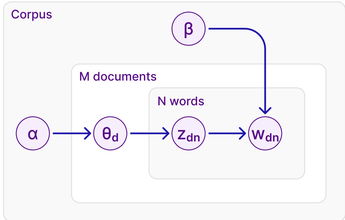
1 Click on the display-topic button **GET** /sessions/{id}/analyses/detailed

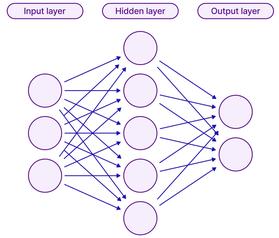
2 Preprocess answers

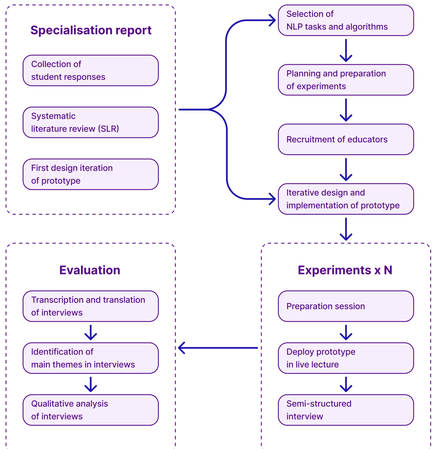
3 Fetch Topics **GET** /sessions/{id}/analyses/overall

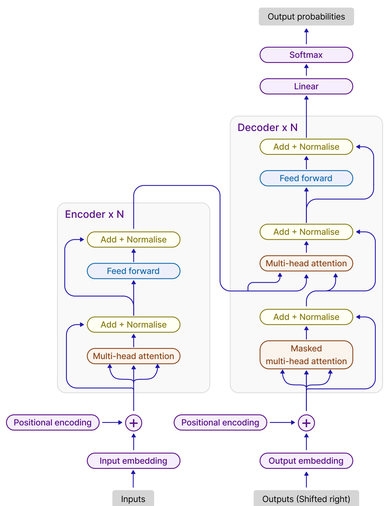
4 Save to database

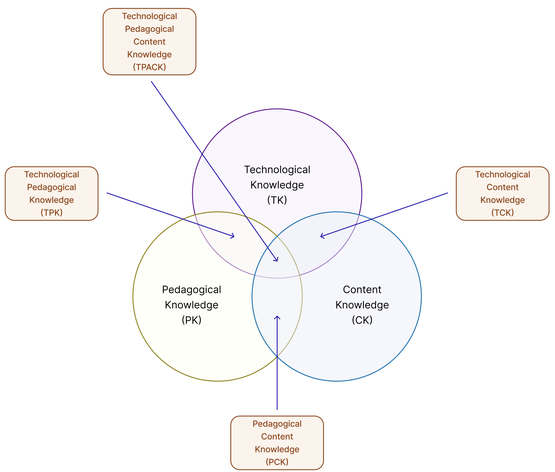

Rule-based
Input
Manual model building
Output
Traditional machine learning
Input
Manual feature engineering
Automated model building
Output
Deep learning
Input
Automated feature learning and model building
Output


Corpus
β
M documents
N words
α
θd
zdn
wdn


Input layer
Hidden layer
Output layer


Specialisation report
Collection of student responses
Systematic literature review (SLR)
First design iteration of prototype
Selection of NLP tasks and algorithms
Planning and preparation of experiments
Recruitment of educators
Iterative design and implementation of prototype
Experiments x N
Preparation session
Deploy prototype in live lecture
Semi-structured interview
Evaluation
Transcription and translation of interviews
Identification of main themes in interviews
Qualitative analysis of interviews


Output probabilities
Softmax
Linear
Decoder x N
Add + Normalise
Feed forward
Add + Normalise
Multi-head attention
Add + Normalise
Masked multi-head attention
Encoder x N
Add + Normalise
Feed forward
Add + Normalise
Multi-head attention
Positional encoding
Positional encoding
Input embedding
Output embedding
Inputs
Outputs (Shifted right)


Technological Pedagogical Content Knowledge (TPACK)
Technological Pedagogical Knowledge (TPK)
Technological Content Knowledge (TCK)
Technological Knowledge (TK)
Pedagogical Knowledge (PK)
Content Knowledge (CK)
Pedagogical Content Knowledge (PCK)

Join page

※ Quizzma

Session Pin

Enter the 4-digit pin displayed on the teacher's screen.

Join →

Waiting page

6630

You are all set

Wait for your host to ask a new question

Answer page

6630

What do you know about AI?

Type your answer

Submit