BOSSCODER ACADEMY

System Design of

Quora

Step - By - **Step**

BOSSCODER ACADEMY

STEP 1

DESIGNING THE MVP (REQUIREMENT GATHERING)

Definition of MVP (Minimum Viable Product): When a company launches a product, it has only a limited core set of features. This initial product is called the Minimum Viable Product.

The MVP is then tested by users, who provide feedback and suggestions. They are then used to create new features and capabilities by the developers.

Thus when building a design for any product, you should find out what the product requirements are to form the core set of features.

BOSSCODER ACADEMY

What are the basic product requirements of an MVP of Quora?

- Users should be able to post questions
- Users should be able to answer questions
- Users should be able to upvote questions
- Users should have a personalised feed with relevant questions
- There should be different topics to sort questions and posts

However, these are not the only features that Quora has.

More complex features like spaces and communities, interaction through personal messages, searching for questions, and commenting on answers can be introduced after a product is launched, based on feedback received from users.

It is important even in interviews to understand the most important and core set of features, when asked to design a system.

BOSSCODER ACADEMY

STEP 2

Scale Estimation of Quora

You estimate the scale of a product/system through:

- Storage Requirements
- QPS (Queries per second Estimation)

Let's understand them one-by-one for Quora.

BOSSCODER ACADEMY

Storage requirements

Storage requirements can be analysed using Monthly Active Users (MAU).

Monthly Active Users (MAU) of Quora = **200 Million**

Daily Active Users (DAU) of Quora = **100 Million**

Assuming, 0.005% of these users actually post a question on the platform.

Number of questions posted each day on platform = **0.005% of 100 Million users**

= **5000 questions**

BOSSCODER ACADEMY

Questions may have features like

- question_id
- timestamp
- creator_id
- content
- topic

On an estimation, each question takes 1 Kb of space.

Then, space required for storing questions posted each day = **1 Kb * 5000 questions**

= **5 Mb**

BOSSCODER ACADEMY

Storage requirement per day to store questions = **5 Mb**

Storage requirement per year to store questions = **5 Mb * 365 = 1825 Mb = 1.8 Gb**

Storage requirement for 10 years to store questions = **18 Gb**

Now, let's also talk about answers as well. Questions will have answers also and those will also take space.

Questions posted each day = **5000**
Questions posted in 10 years
= **5000 * 365 * 10**
= **5 *365 * 10^4**
= **1825 * 10^4**
= **20 Million questions (around)**

So, questions posted in 10 years = 20 Million

On an average, let's say, if each question has 3 answers

Number of answers on platform (in 10 years)
= **20 Million * 3**
= **60 Million**

Answer may have features like

- answer_id
- timestamp
- creator_id
- content
- topic

On an estimation, each answer takes 10 Kb of space.

Then, space required for storing answers posted (in 10 years) = **60 Million * 10 Kb**

= **600 Million * Kb**

= **600 Gb**

Storage Required to store answers (for 10 years) = 600 Gb

Total Storage Requirement
= Requirement for Questions + Answers
= 18 Gb + 600 Gb
= 620 Gb (approx)

In comparison to other products like Microsoft Teams, Netflix, the storage requirement calculated is fairly less.

BOSSCODER ACADEMY

Queries Estimation

Queries can be defined as reads and writes.

Writes are those actions which add some data to the system, like posting questions, adding comments, upvoting questions etc.

Reads are those actions through which the user interacts with data, like reading questions and answers.

Let's estimate Reads in the system:

Let's say, on an estimate, number of pages visited by user each day = 10

Quora Daily Active Users = **100 Million**

Number of reads = **100 Million * 10**
= **1 Billion reads per day**

Let's now estimate Writes in the system:

Number of questions posted each day
= **5000**

Number of answer posted each day
= **5000*** (3 answers on an average everyday)
= **15000**

Number of writes
= **20000**
= **20K writes per day**

So, Quora is a read heavy system, where there are much more reads when compared to writes.

Queries are calculated using QPS (Queries Per Second)

Considering 1 billion queries per day,

QPS (Queries per Second) = **1 Billion/ 1 day**

Number of seconds in a day
= **24 hours * 60 mins * 60 seconds**
= **86400 seconds**
= **1 * 10^9 / 86400**
= **12K queries per second (approx)**

In a read heavy system, caching is an important parameter that can be used for faster reads.

We will discuss later in this pdf how to use caching in the Quora system.

BOSSCODER ACADEMY

STEP 3

Design Goals of Quora

There are two parameters to evaluate design : **Availability and Consistency**

Availability of a system is measured by its uptime, and are available to service requests most of the time.

This is very important for systems like live streaming of matches, personalised feeds for Instagram etc.

Consistency of a system is achieved by servicing the same data for different users without any mismatching.

It is very important for activities of a system like banking transactions, reservation systems through IRCTC etc

In the case of Quora, availability is comparatively more important than consistency.

Latency is basically the time delay between the start of the request from a user or client to delivery of the result to the user by the server.

The ultimate goal of building a good design would be to reduce latency for faster responses to the user.

Low latency is not a critical factor for services like Google Analytics, Emails and Notifications.

However in Quora, low latency is a design goal.

BOSSCODER ACADEMY

STEP 4

API’s requirement for Quora

API stands for Application Programming Interface. They let your product or service communicate with other products and services without having to know how they’re implemented.

The required APIs for your product will be obtained from the requirements as a part of the MVP.

For example few APIs of Quora
(from our MVP - Discussed in Step 1) can be

- Create Questions
- Submit Answers
- View Questions
- Fetch Newsfeed based on topics

BOSSCODER ACADEMY

STEP 5

Design Deepdive of APIs

Since we have a total of storage of 620GB we need a systematic way of storing the data.

In a SQL database, you can consider the following entities or tables with their attributes:

BOSSCODER ACADEMY

USER

user
user id
name
topics followed

QUESTION

question
question id
creation id
timestamp
content
topics

ANSWER

answer
answer id
question id
creator id
content
timestamp

TOPIC

topic
topic id
name
description

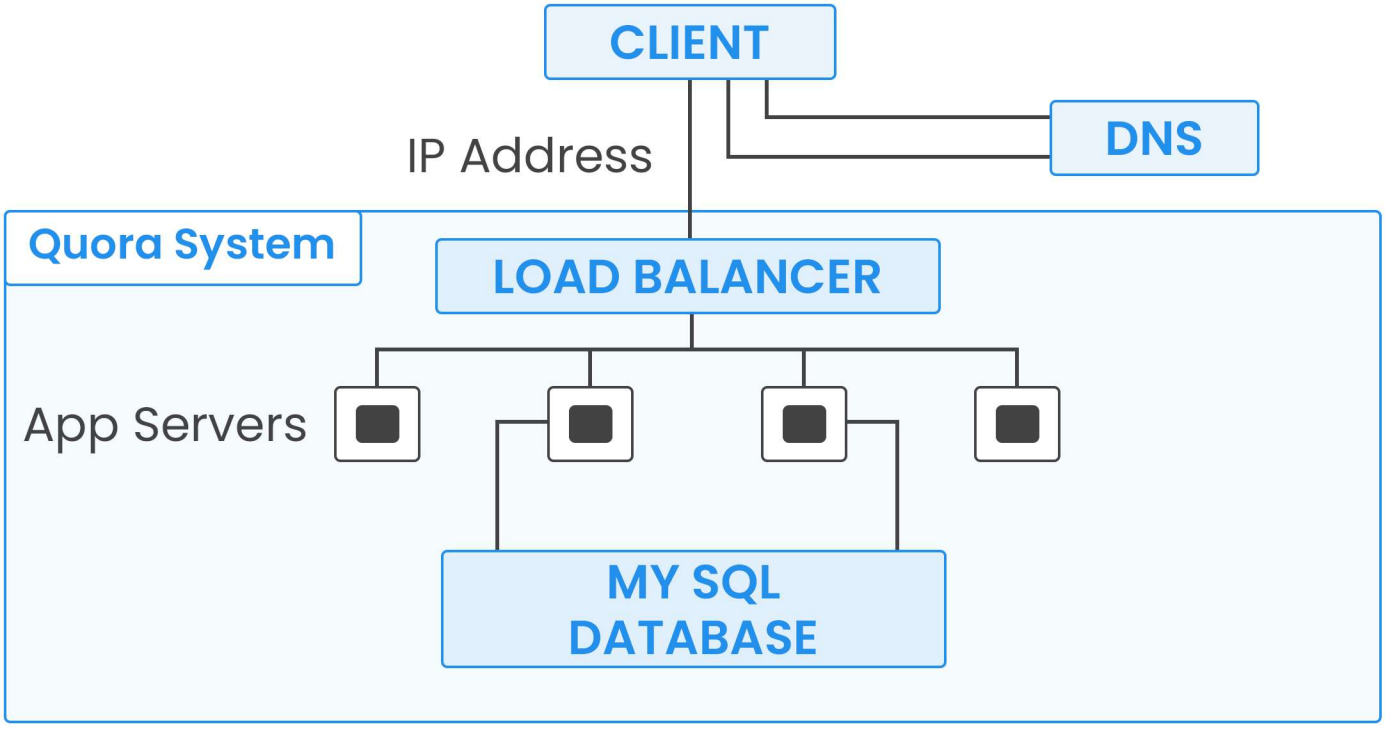
UPVOTES

upvotes
answer id
question id
creator id

BOSSCODER ACADEMY

The client will first go to the DNS Server, which sends the IP address of the load balancer.

The overall Quora system begins from the load balancer, which is connected to many app servers. These app servers have features like caching (will discuss caching later in this doc), and access data from the MYSQL database.



BOSSCODER ACADEMY

STEP 6

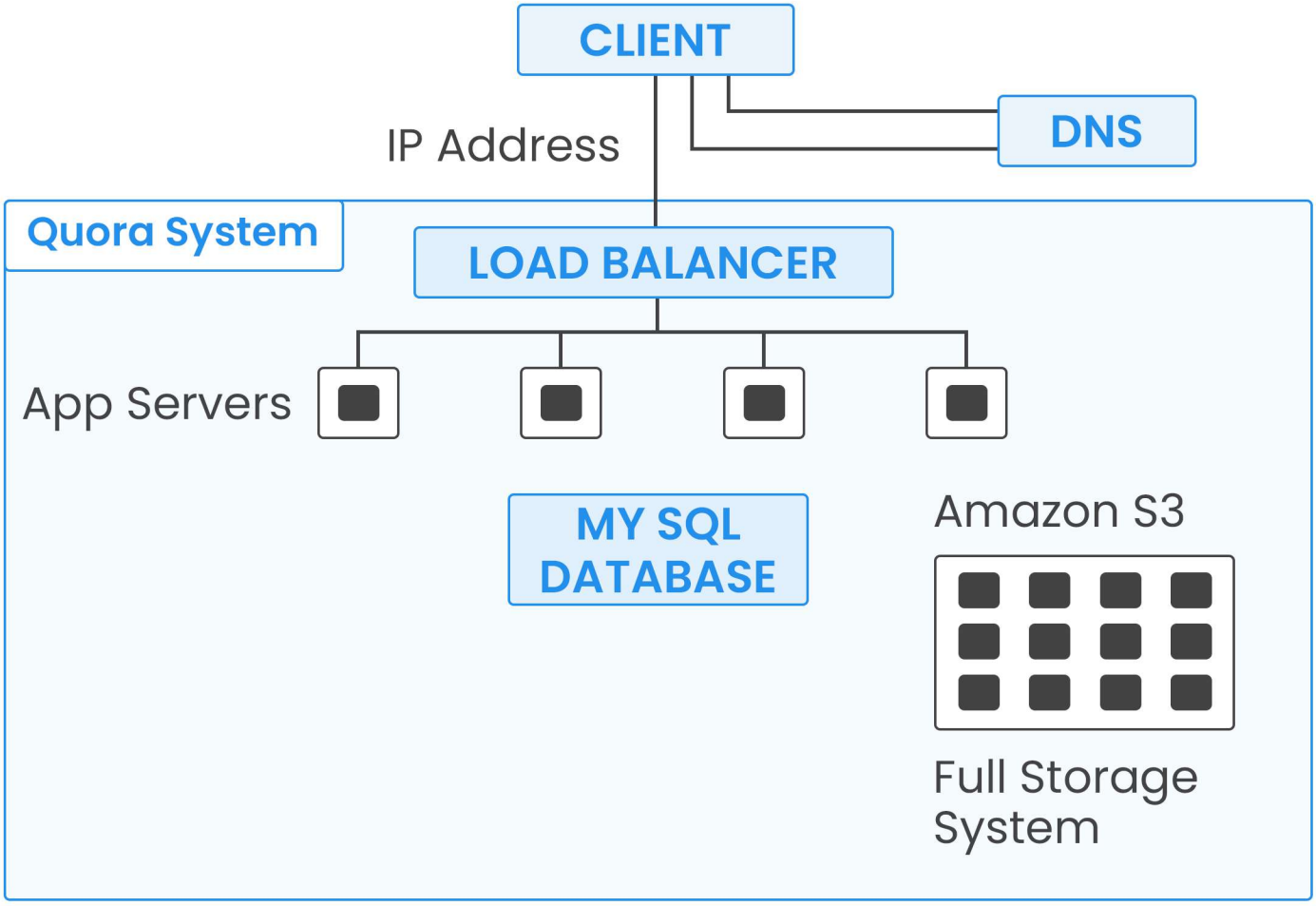
What about images and other data types used in Quora?

When an image is added to Quora, it comes down to simply as a link to a file storage system or essentially text with special keywords. When browsers come across these keywords, the images are rendered.

A file storage system is kind of like an Amazon S3 or Azure blob. When some content like videos, images are inserted into the file system, it returns a URL of that data.

BOSSCODER ACADEMY

Example: http://s3.amazon.us.com Is an URL



Quora also uses CDNs which stand for Content Delivery Networks, for fast transmission of data over a large user

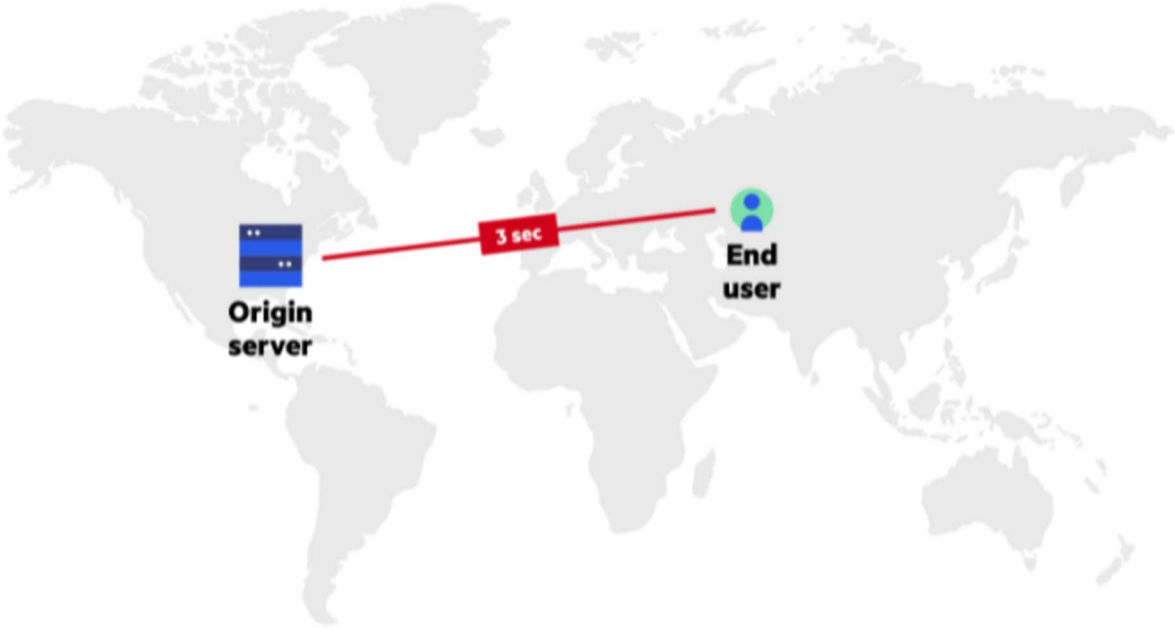
In India, Amazon has one datacenter, in Mumbai. Consider a user from Delhi, for the data to be sent to him, it has to travel a large distance which can cause latency issues.

CDNs are small edge servers, when compared to large data centres, which are distributed all over the world. The data is replicated in these servers and users get fast access through data by a CDN which is geographically closer to them.

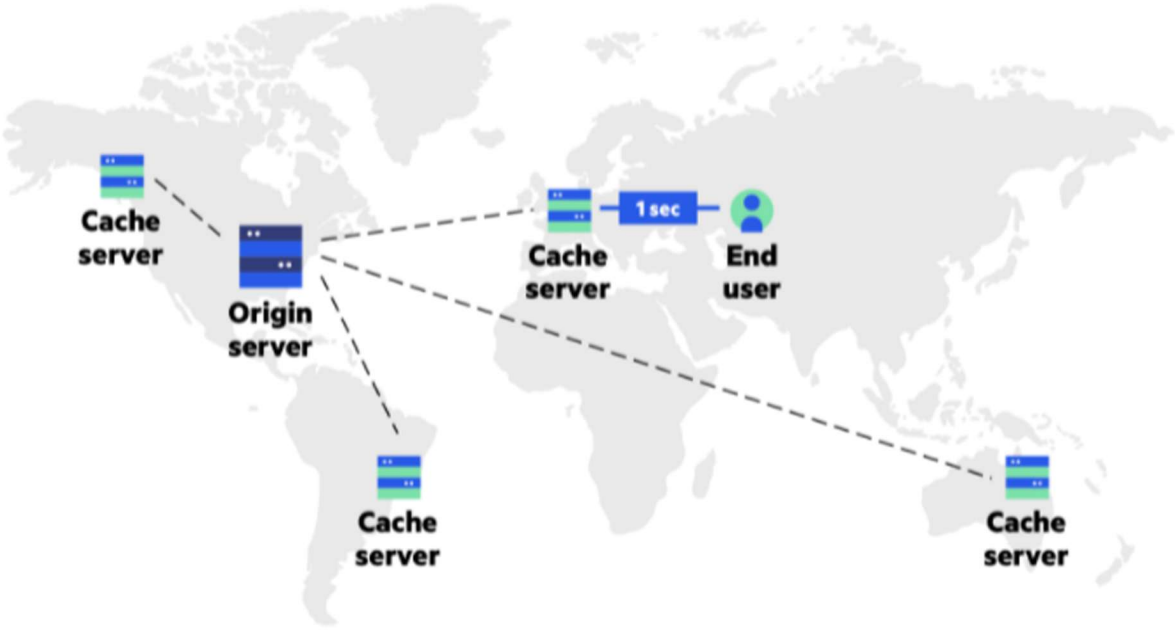
Images are usually replicated over CDNs for low latency and are synced through origin, that is the Amazon S3 instance as an example.

BOSSCODER ACADEMY

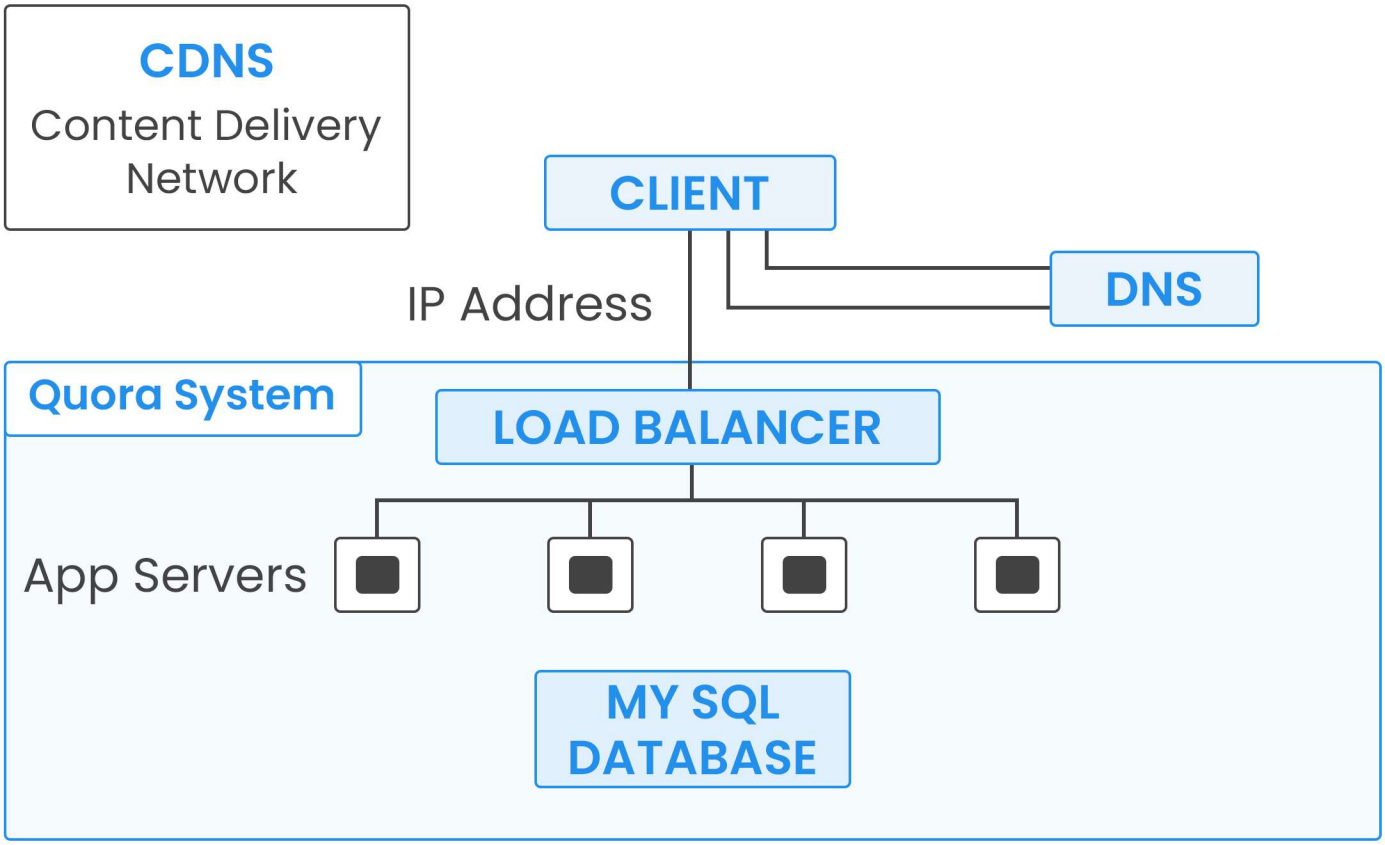
Without CDN



With CDN



BOSSCODER ACADEMY

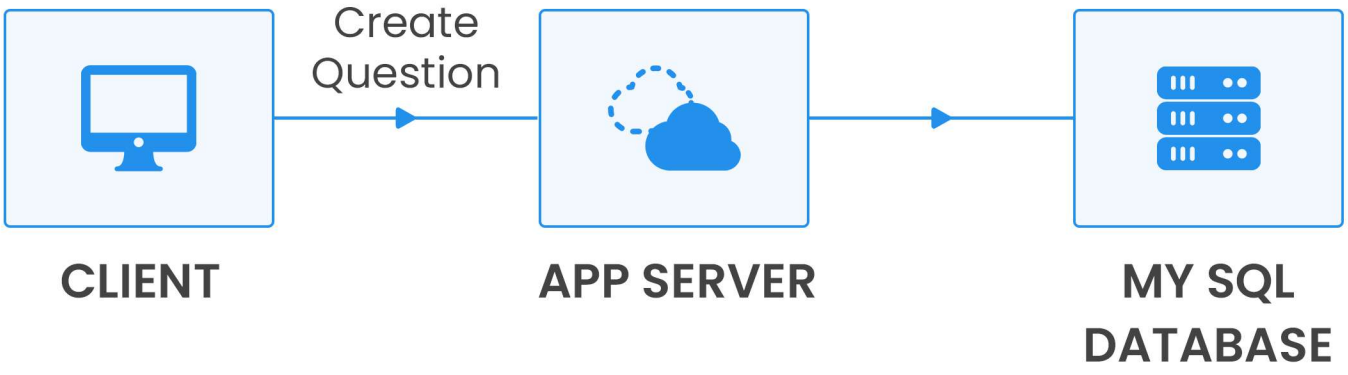


BOSSCODER ACADEMY

Checking the API speed on our above design

Create Question API

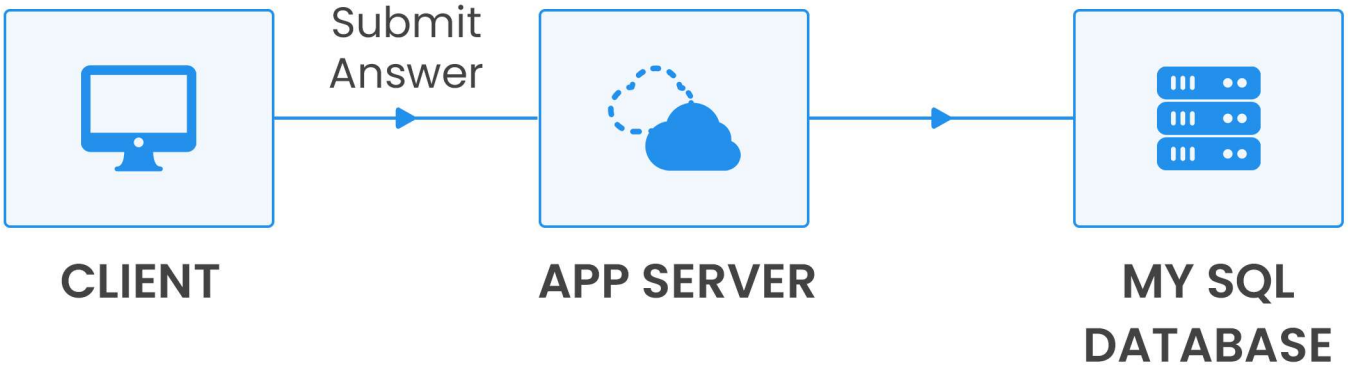
When a user wants to create a question, a server is used to simply add the question to the database. It is not that expensive.



BOSSCODER ACADEMY

Submit Answer API

Similarly when a user wants to submit answers, a server can be used and the answer is added to the database.



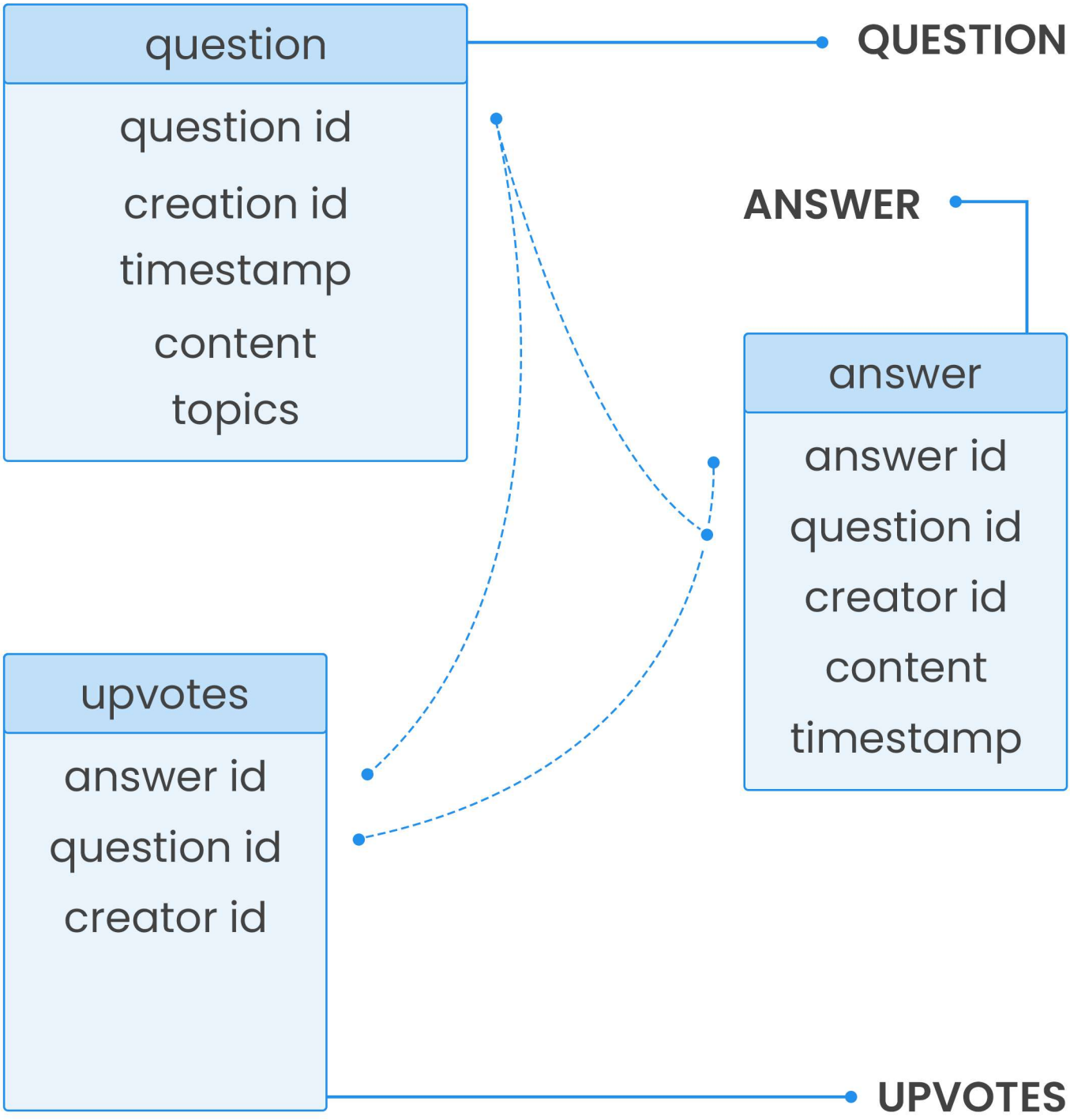
View Questions API

Whenever the page is loaded, all the content including questions, answers to the question, upvotes have to be displayed.

Since the question_id attribute is used to access the question to be displayed, indexing is done on the basis of the question_id attribute to get all the answers to be shown. Similarly upvotes related to the question and answers are also obtained by indexing.

Getting the upvotes related to a question or answer is an expensive query.

BOSSCODER ACADEMY



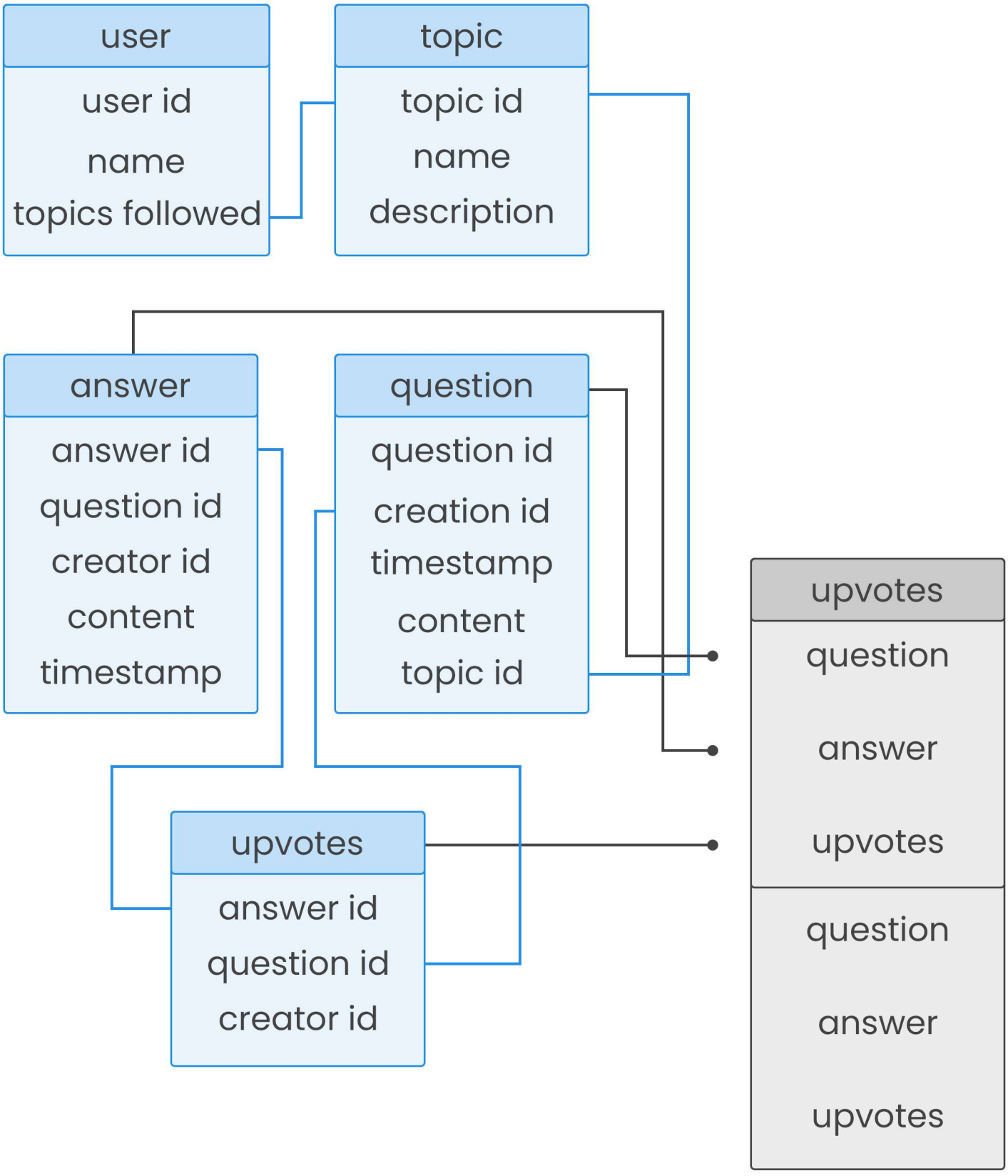
BOSSCODER ACADEMY

Fetch Newsfeed API

Based on the topics that a user is subscribed to, you have to find out questions for each topic.

In turn you also have to fetch the answers to each question, along with upvotes for each of these questions and answers. Finally you have to show a certain set of questions and answers in the feed, and divide the content into multiple pages. This process is known as Pagination.

BOSSCODER ACADEMY



BOSSCODER ACADEMY

STEP 7

Caching for Efficient APIs

Both the View Questions API and Fetch Newsfeed API are very expensive and require a lot of computation time to carry out the data transfer.

You can reduce this latency by using Caching techniques.

View Question API

Given a question_id, getting the content of the question is fairly easy. Getting the answers related to the question is also easy, however the expensive part of the API is getting the upvotes for the question and each answer to it.

Maintaining a simple cache, containing question_id, answer_id and upvote_count as attributes seems to be a plausible solution. However the problem arises due to the stale data present in cache, which may not be consistent with the actual data.

For example: The actual no of upvotes of a particular answer may be 15, but the cache may not be updated yet and may still show 13 upvotes.

Thus for improving latency and for faster data access we can compromise on the consistency of data.

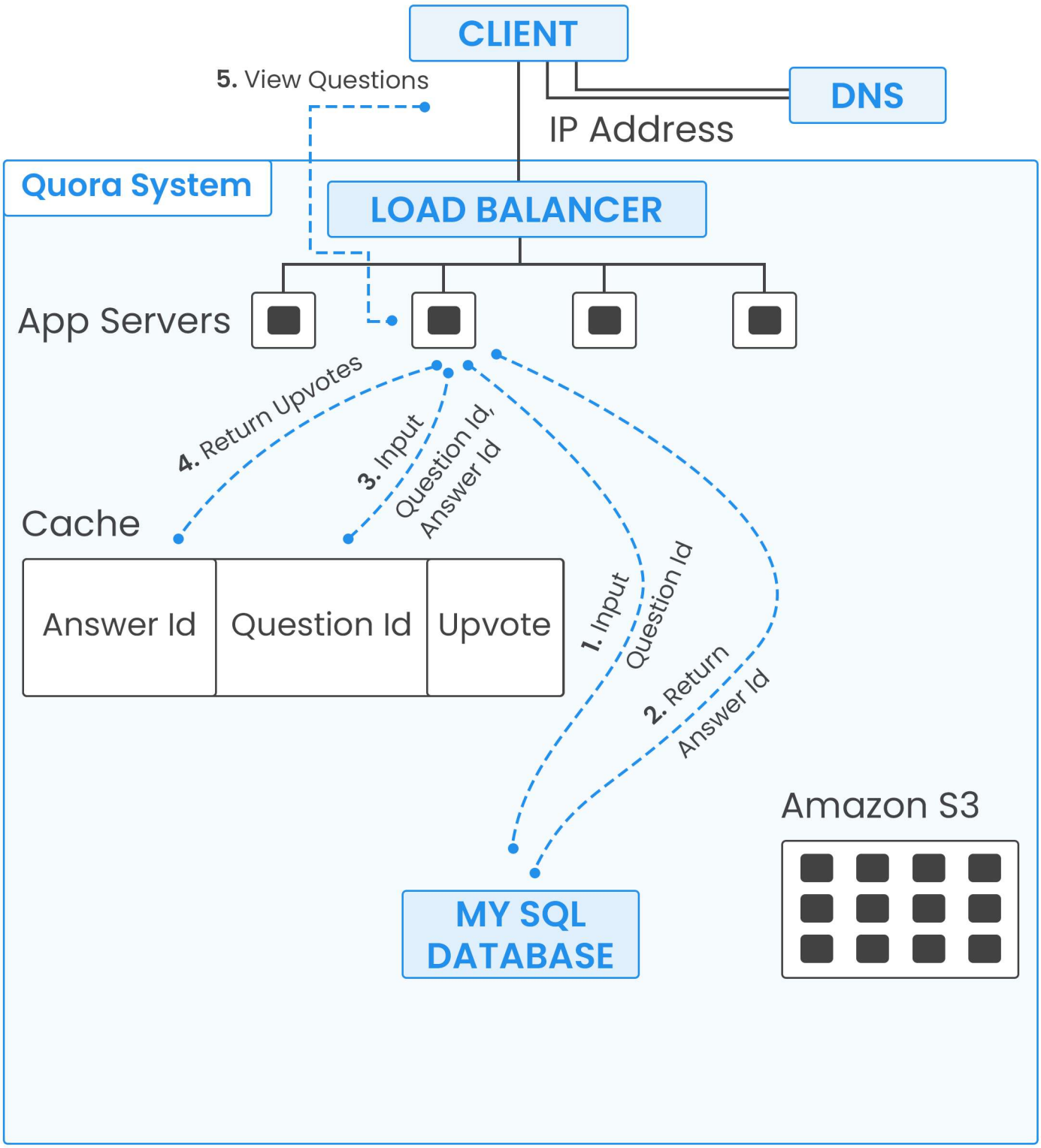
Cache can either be updated by the MYSQL database (PUSH MODEL) or fetch data from the MYSQL database itself (PULL MODEL).

Working of View Question Caching:

- The question_id is used as input to access the different answer_id in the MYSQL DATABASE

- Corresponding answer_id is returned by the database through indexing
- Both question_id and answer_id are used as input to the cache
- The corresponding upvotes are returned by the cache and combined data is relayed to the client

BOSSCODER ACADEMY



BOSSCODER ACADEMY

Fetch Newsfeed API

For displaying a personalised newsfeed to each of the estimated 100 million users active every day, who are subscribed to many topics, you will need to display appropriate questions, answers and the upvotes.

A naive solution would be to cache details of each and every user. This is not feasible as the number of users is very large.

Topic based caching is a better solution, since the number of topics are limited. Caching the top 50 or so questions for a topic is feasible as compared to caching all the questions of a topic.

BOSSCODER ACADEMY

Working of Fetch Newsfeed Caching:

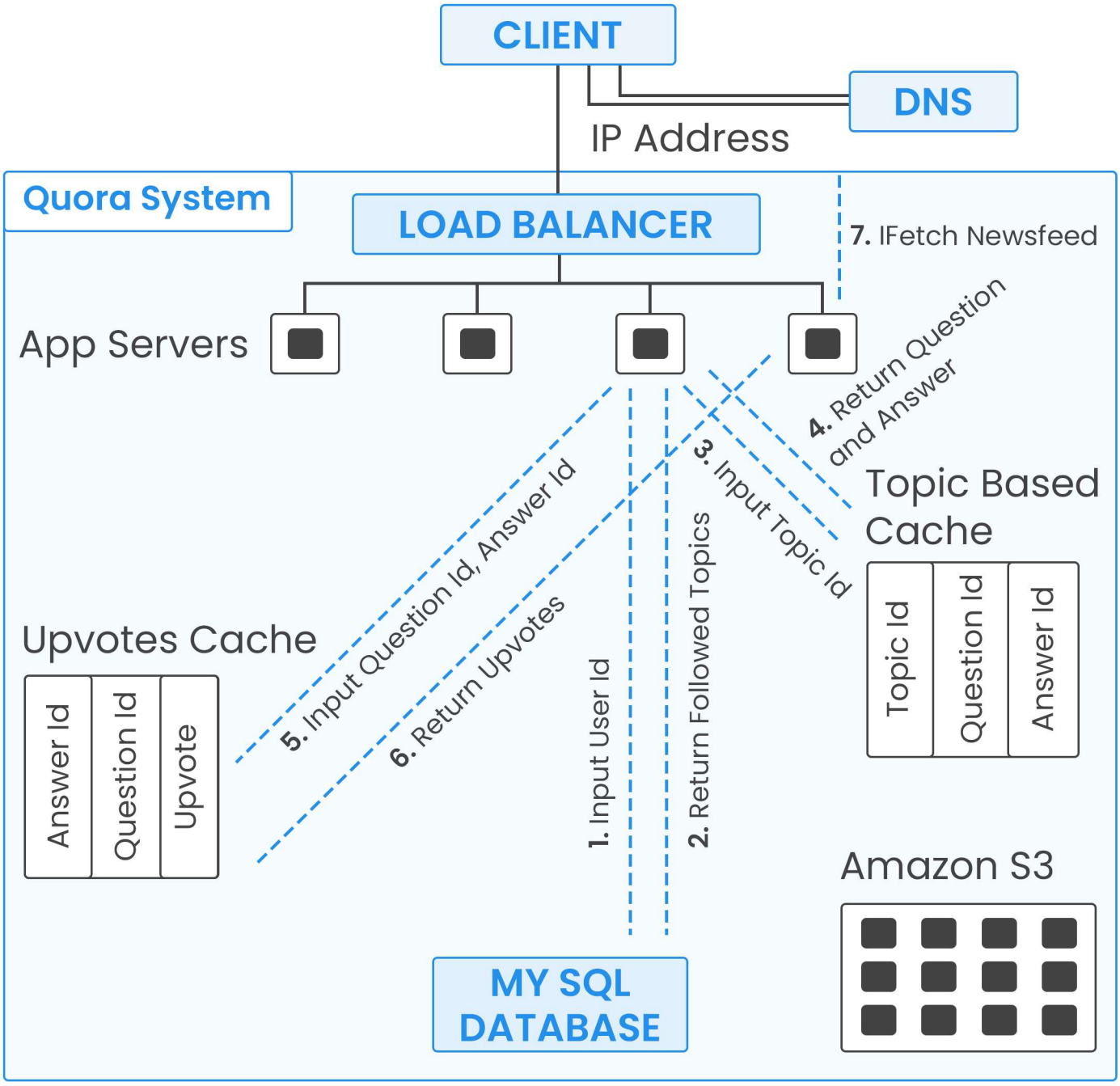
- The user_id is used as input to access the different topics followed in the MYSQL database.
- Corresponding topic_id followed by the user are returned
- Topic_id is used as an input to the Topic Based Cache to access top 10/50 question_id and answer_id
- The corresponding question_id and answer_id are returned by the Topic Based Cache

BOSSCODER ACADEMY

Working of Fetch Newsfeed Caching:

- Both question_id and answer_id is used as input to Upvotes Cache
- Upvotes cache returns corresponding upvotes
- The combined information is relayed to the client.

BOSSCODER ACADEMY



1. Input User Id

2. Return Followed Topics

3. Input Topic Id

4. Return Question and Answer

5. Input Question Id, Answer Id

6. Return Upvotes

7. IFetch Newsfeed

BOSSCODER ACADEMY

STEP 8

Replication for solving Single Point Failures

If the MYSQL database or Amazon S3 file storage system or the caches fail in our design, the entire application will collapse. This is called Single Point Failure, where failure of any part of a system will stop the entire system from working.

Replication: Storing multiple copies of the main database, called master database into different databases, called slave databases.

Thus failure of any one database will not affect the whole system as another database can service the query or request.

- **Read**
  Read operations can be distributed among different slave databases to avoid latency and faster access.

- **Write**
  However we have to synchronise the data changes among all the databases when any write operation is done.

  We can either make changes to all the slave databases and then return the result to the user

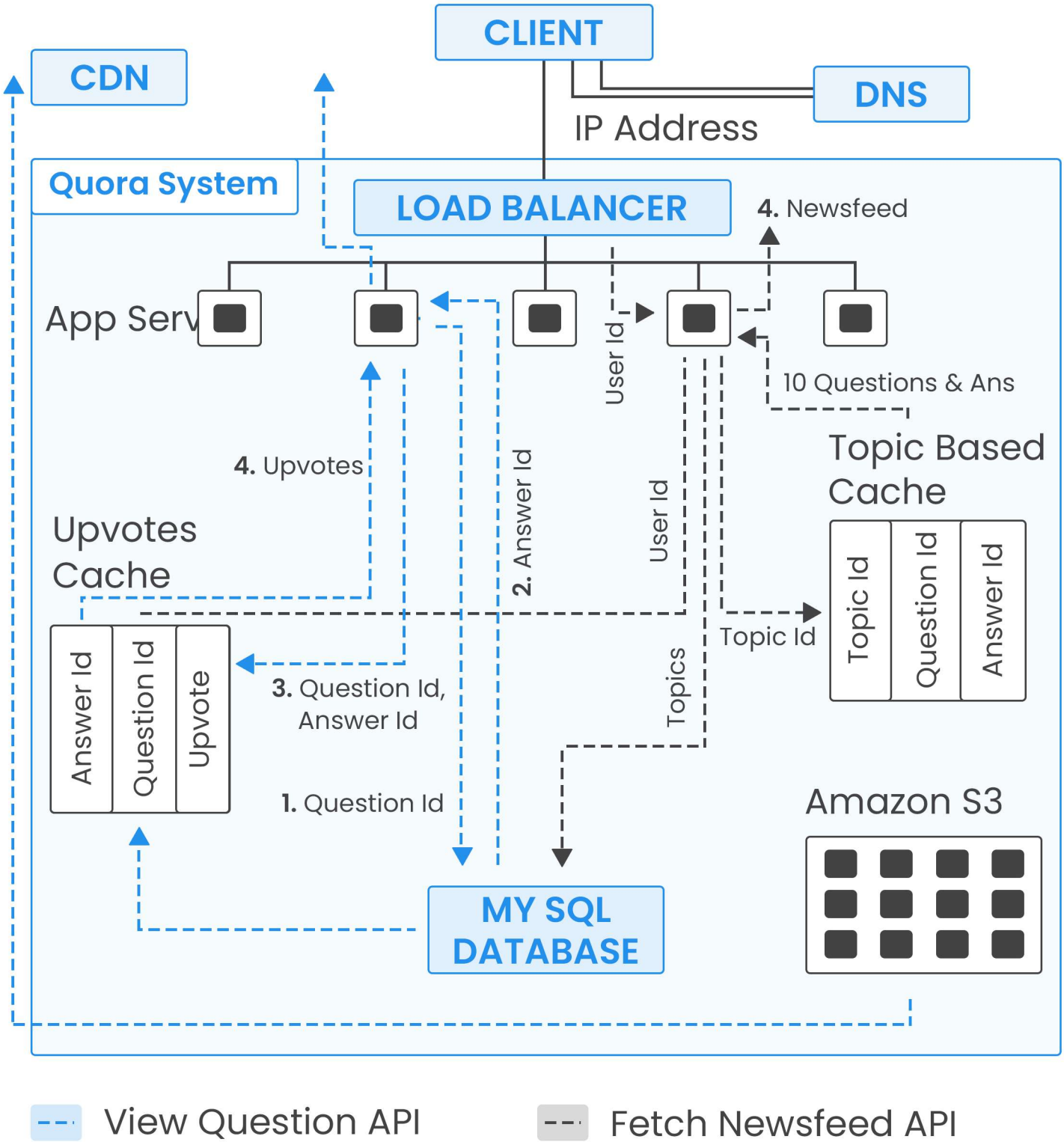
Or

as soon as write query has arrived, result is returned and then data is modified in the slave databases as a background operation.

This operation may lead to data loss.

BOSSCODER ACADEMY

Final system design



BOSSCODER ACADEMY

Bosscoderacademy

Why Bosscoder ?

- **400+** alumni placed at Top **Product-based** companies
- More than **136% hike** for every 2 out of 3 working professional
- Avg package of **22 LPA**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

We had teaching & mentoring sessions from Industry experts from **top notch companies.**

Ujesh
Google