Atapkel Beibarys. Assignment1.

About services and their responsibility

- message service - receive one message from one user, saves it to database sends it another user. Sends notification to notification service.
- notification service - gets requests to send notification for users from stories viewers service and message service.
- stories viewers service - saves list of stories viewers, who likes to stories. When user upload stories it sends request to notification service. Not for all users, for users who likes to previous stories.

- sendMessage() - sends message to another user and saves database, sends to notification service.
- sendPhoto() - sends photo to another user and saves to object storage and link will be on sql, sends to notification service.
- sendAudio() - sends audio to another user and saves to object storage and link will be on sql, sends to notification service.

- listOfViewers() - returns list of viewers.
- listOfLikedViewers() - returns list of liked viewers.
- listWhoCannotView() - limits viewers.

- MessageNotification() - sends notification to user about message
- LikedToStoriesNotification() - sends notification to user about liked user.
- PinnedNotification() - sends notification to user about pinned message.

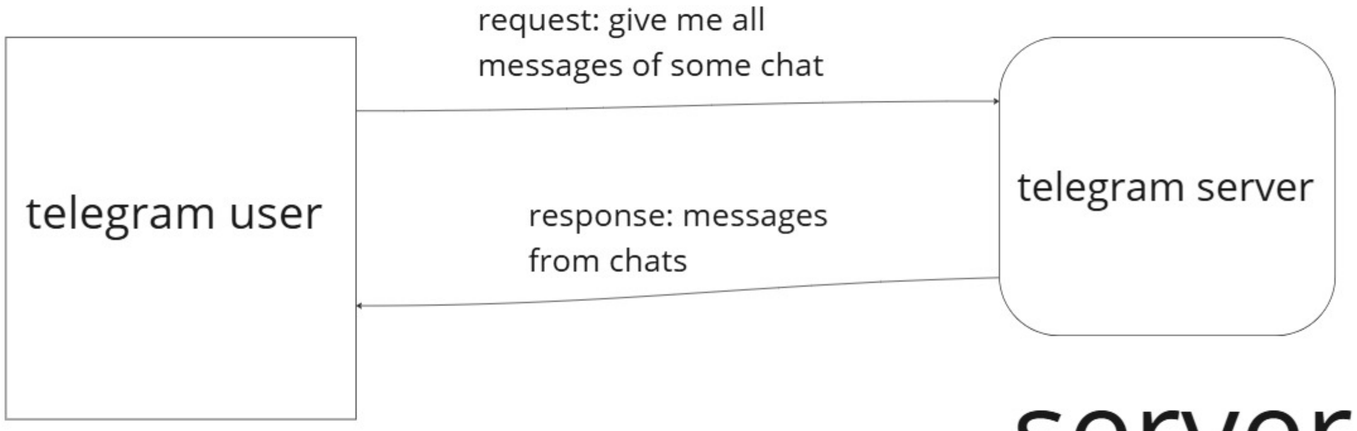
This app can be run on physical server or in cloud.
Cloud is more flexible. It gives resources by demand. Cloud more safe.

client

Calculator-calory.js

Run on User Device.
All logic and UI in
one file. Small app.
In this case we do
not need internet.

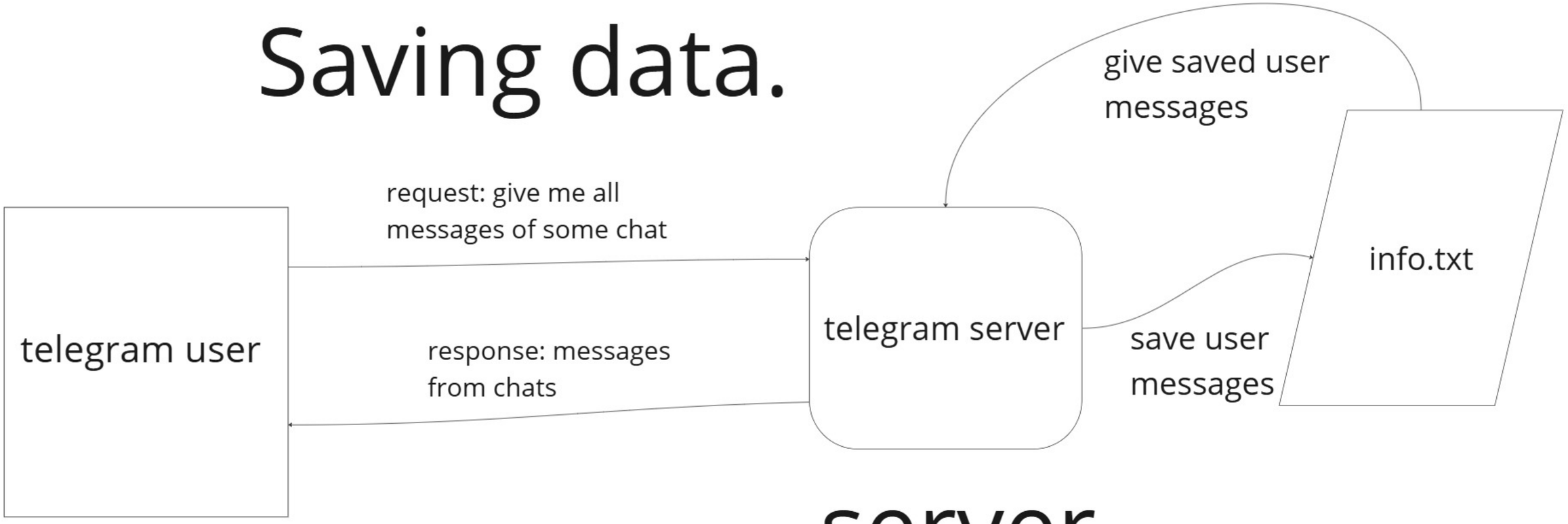
Using Server. Request/Response



We need server because.
Without server
we can not communicate
with each other. First we send our messages to server. Then server will give this messages to another users.
Server have:

- Powerful. More resources(cpu, storage, etc.).
- Good internet connection.

Saving data.

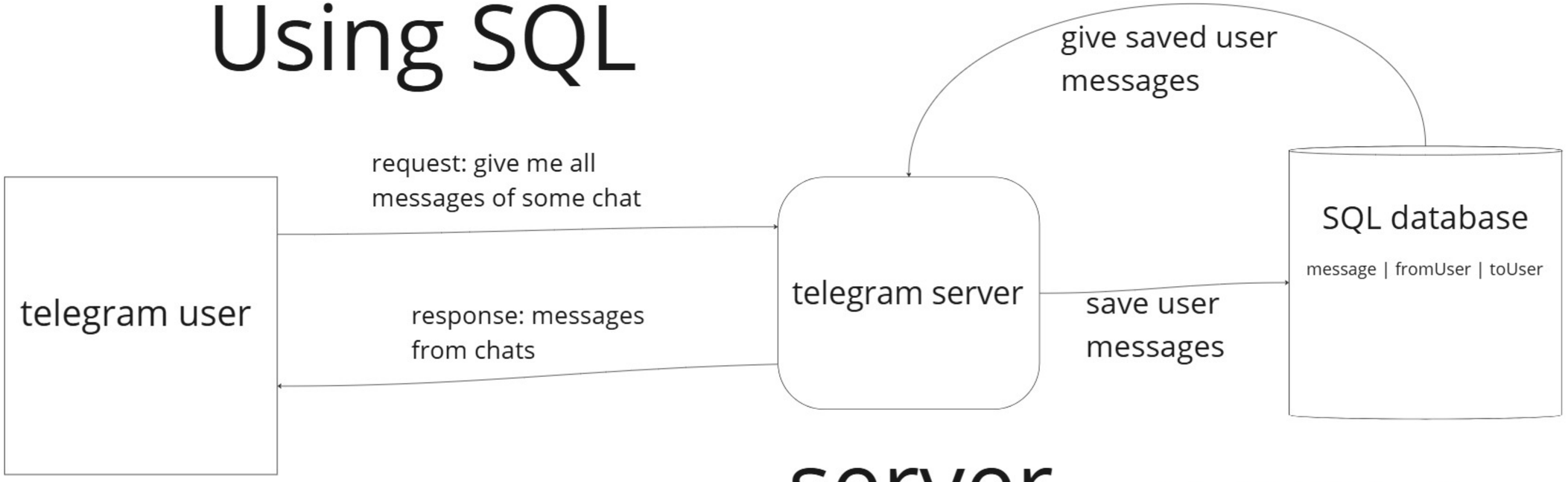


client

server

Without storing data.
All messages, user info, etc.
will be deleted when we
restart server.

Using SQL

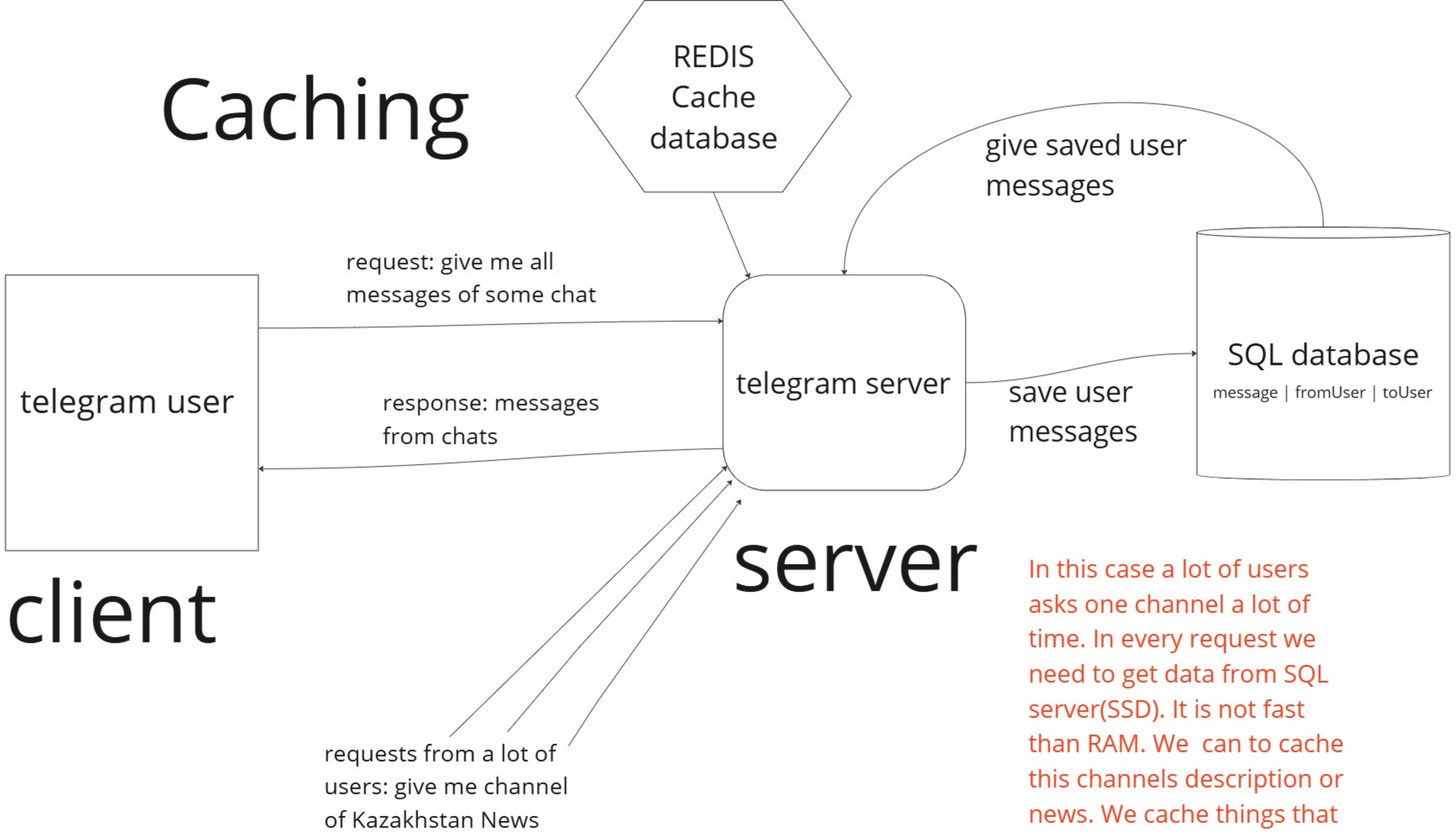


client

server

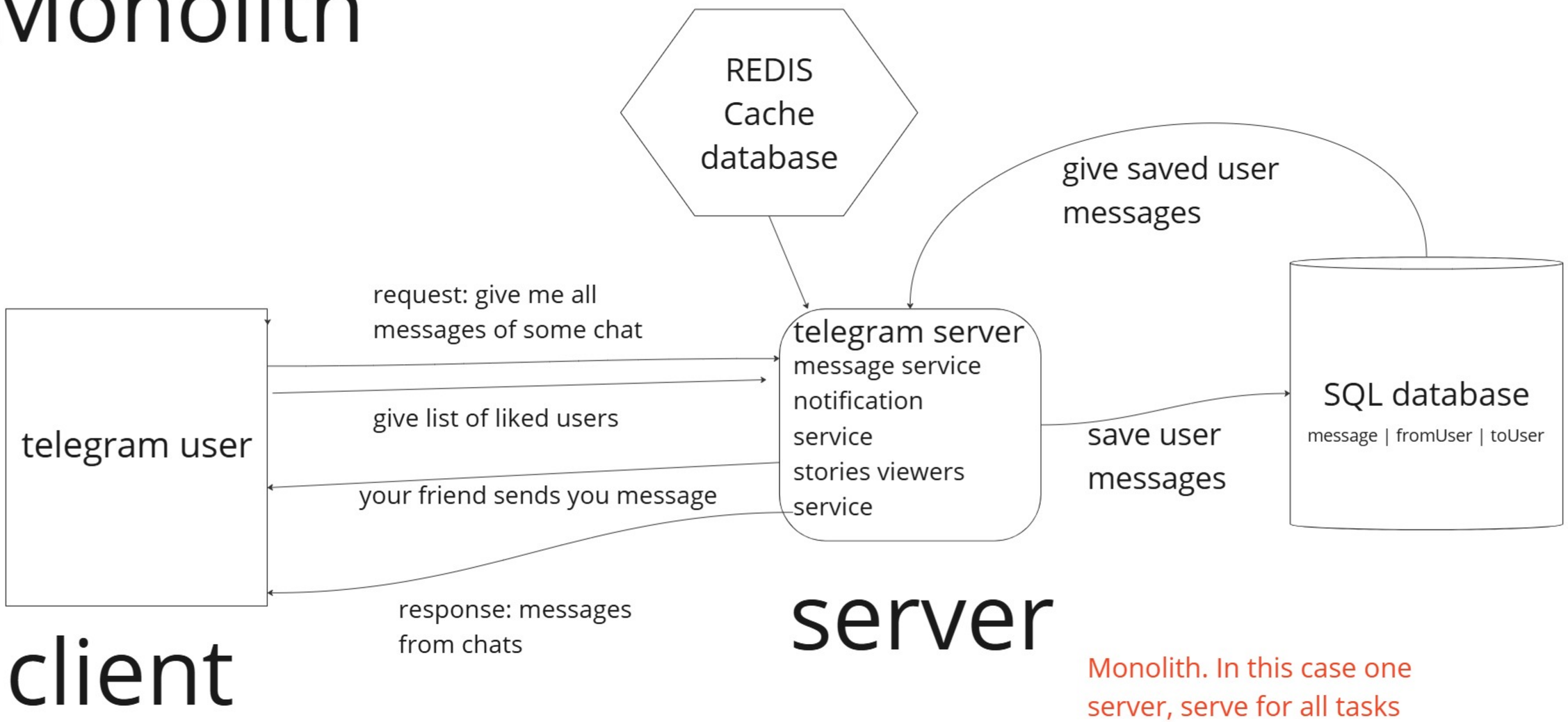
txt not fast if data will be big. Time complexity of the txt will be O(n).
In SQL we can get in constant time.

Caching



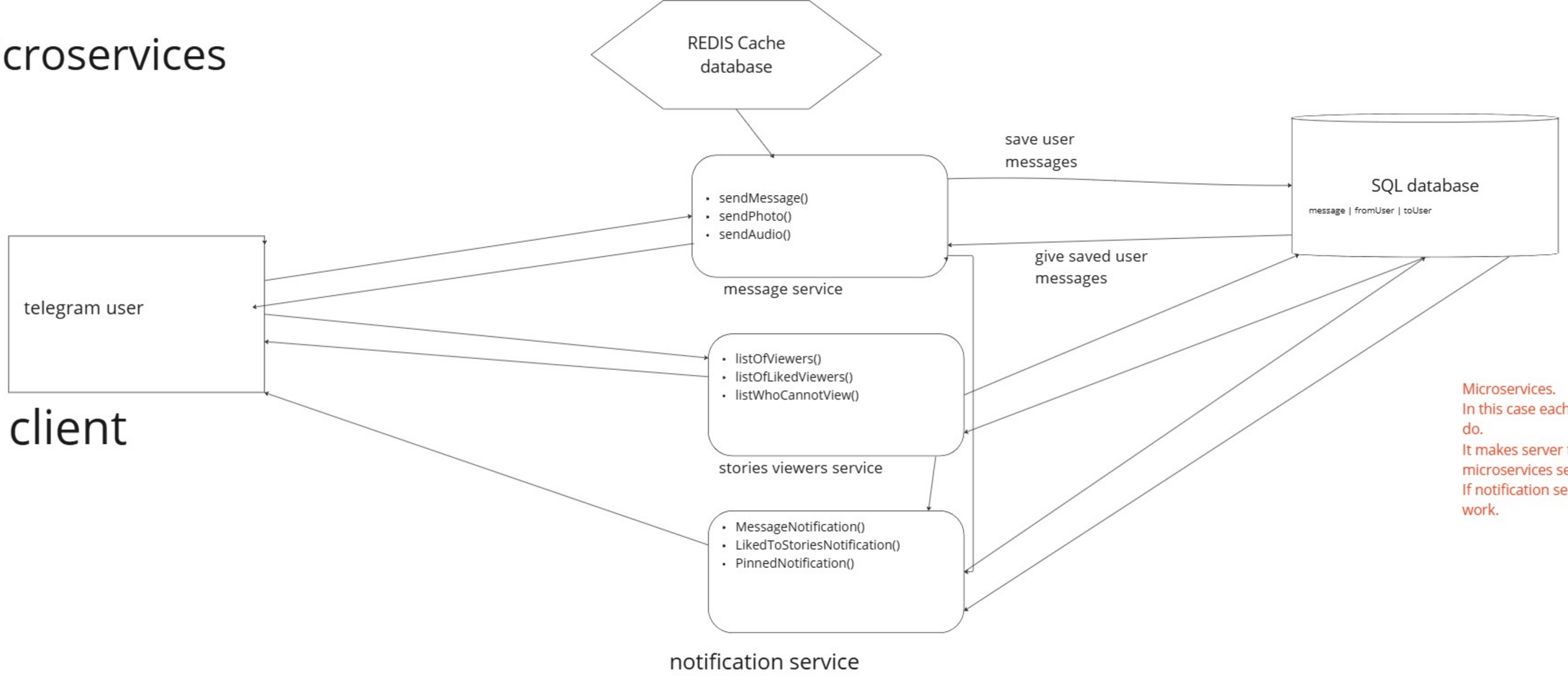
In this case a lot of users asks one channel a lot of time. In every request we need to get data from SQL server(SSD). It is not fast than RAM. We can to cache this channels description or news. We cache things that requested a lot of time to make server faster. Next time news will be send from redis database.

Monolith



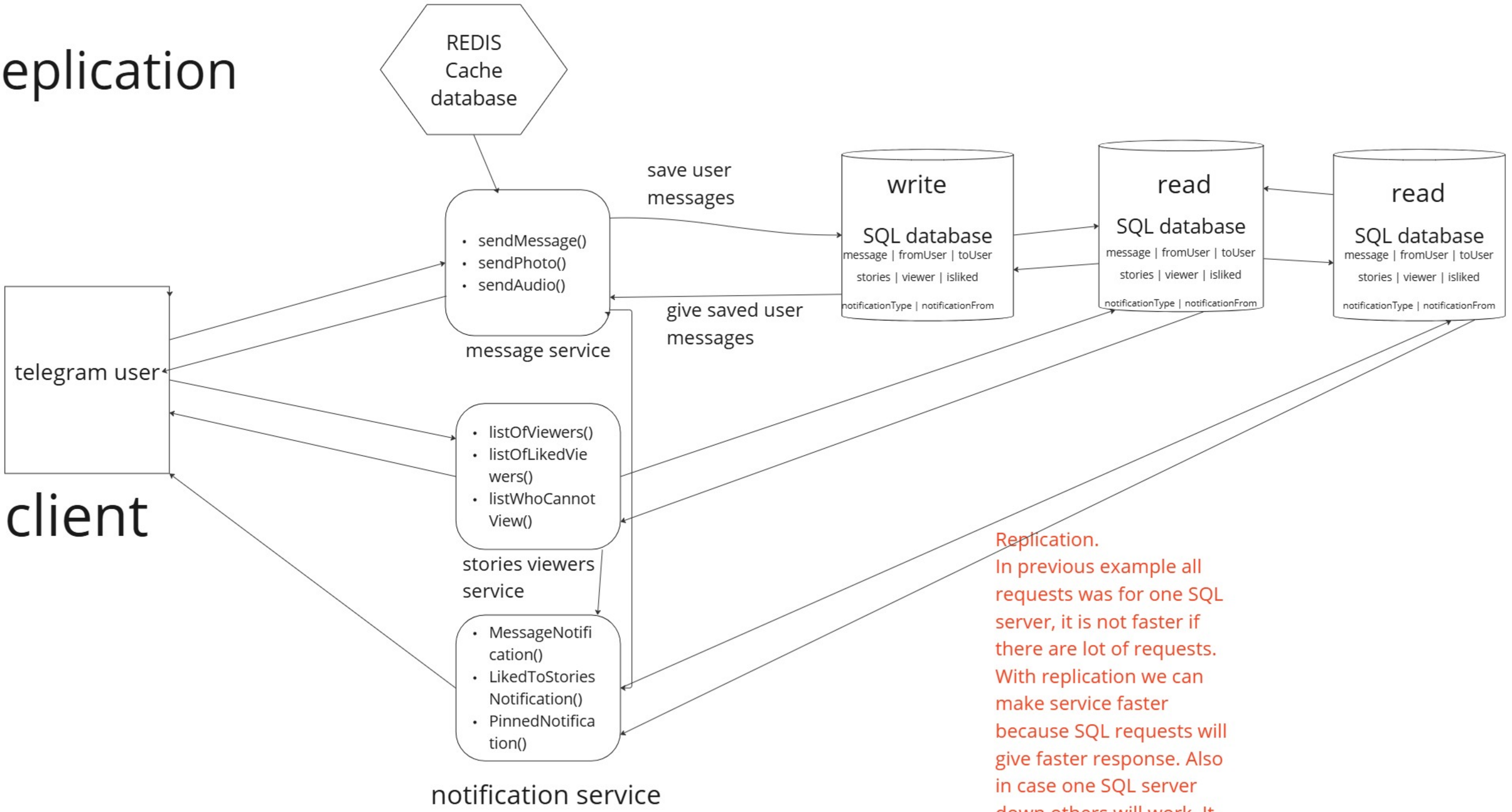
Monolith. In this case one server, serve for all tasks this server sends messages, gives list of viewers, sends notification. Frontend + Backend in one server.

Microservices



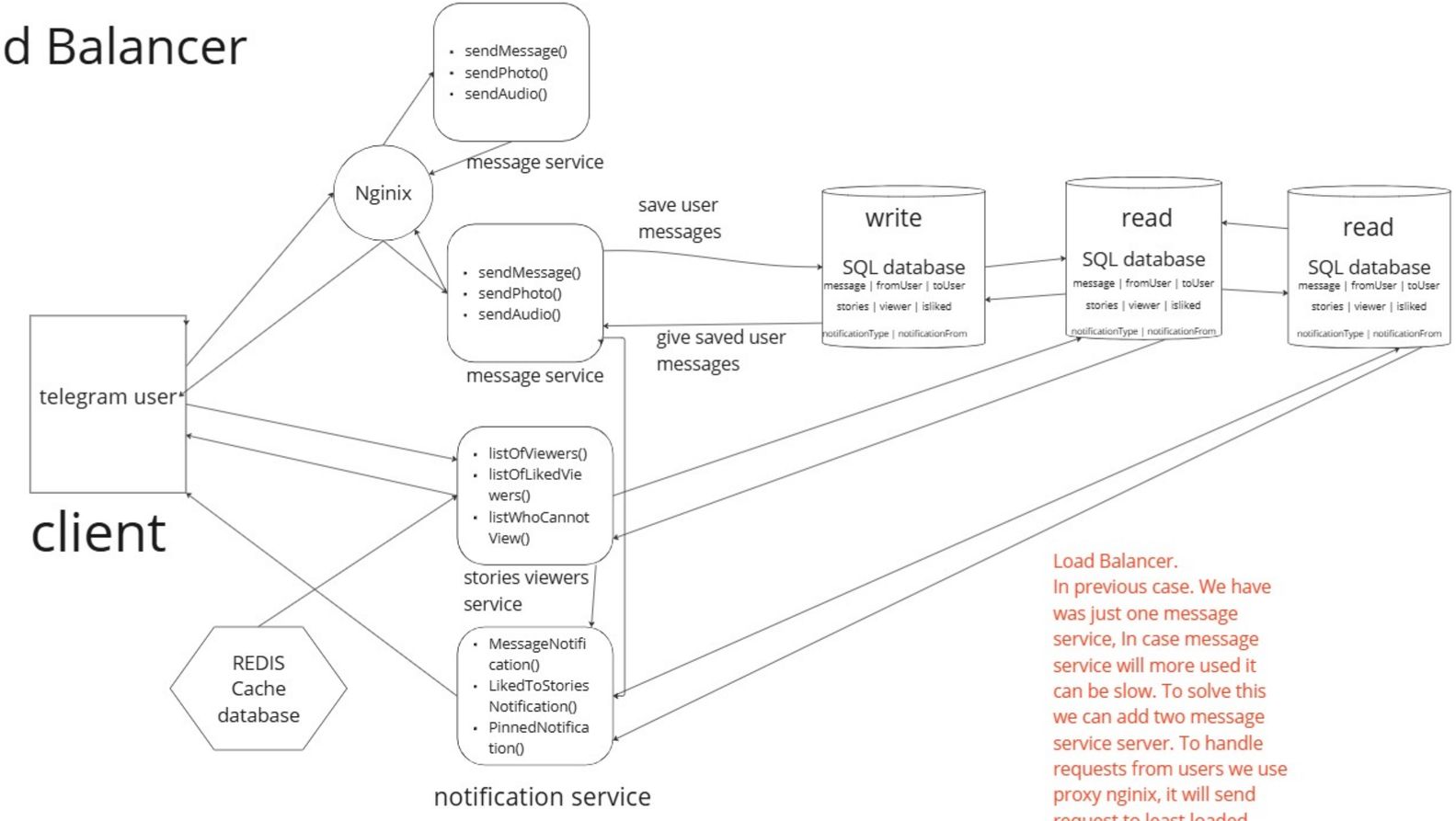
Microservices.
In this case each server has own responsibility to do.
It makes server faster, less cost. Than monolith in microservices services independent to each other.
If notification server will crash other servers will work.

Replication



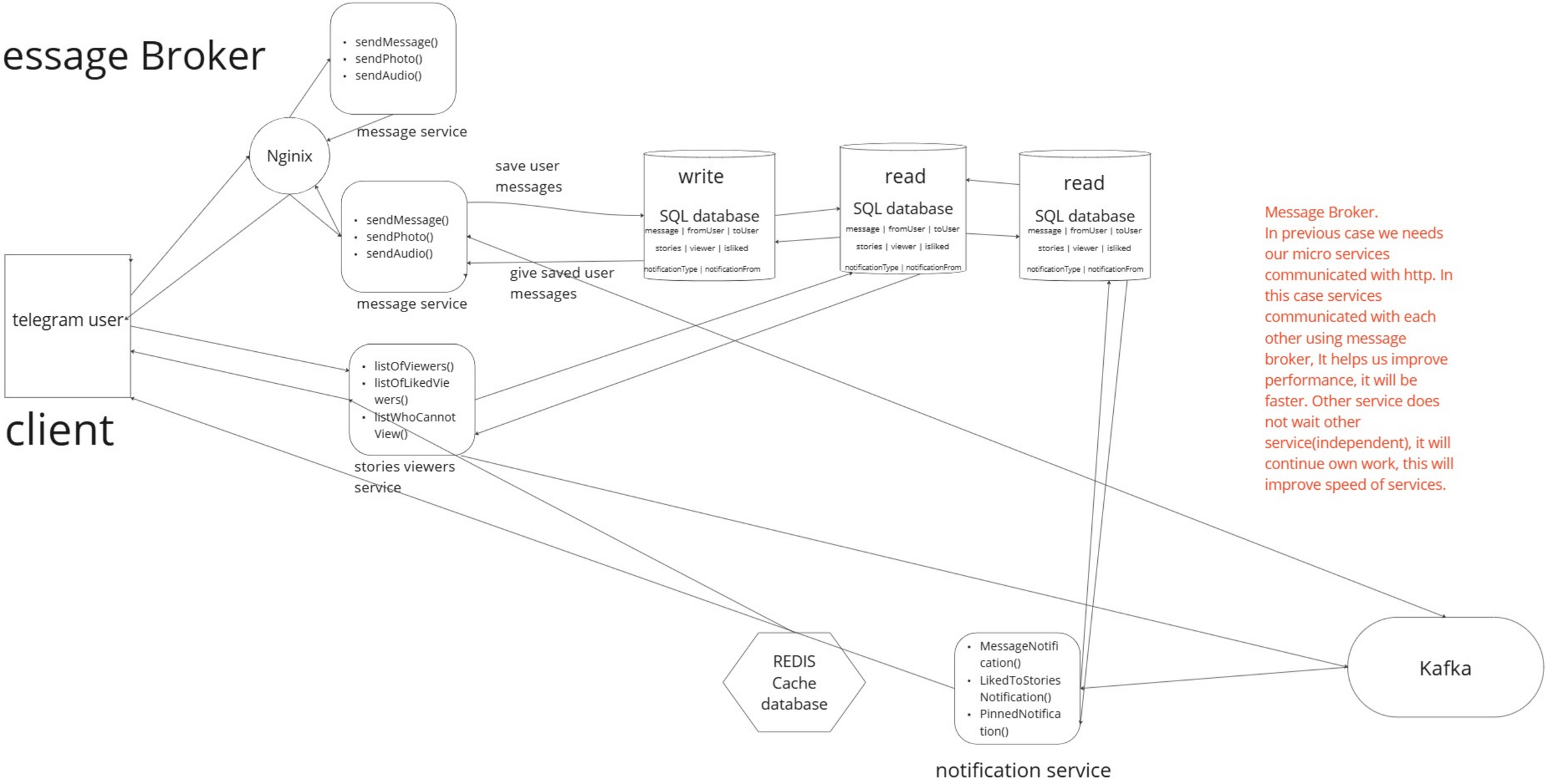
Replication.
In previous example all requests was for one SQL server, it is not faster if there are lot of requests. With replication we can make service faster because SQL requests will give faster response. Also in case one SQL server down others will work. It will be faster also safety, but there are also own cost $$$

Load Balancer



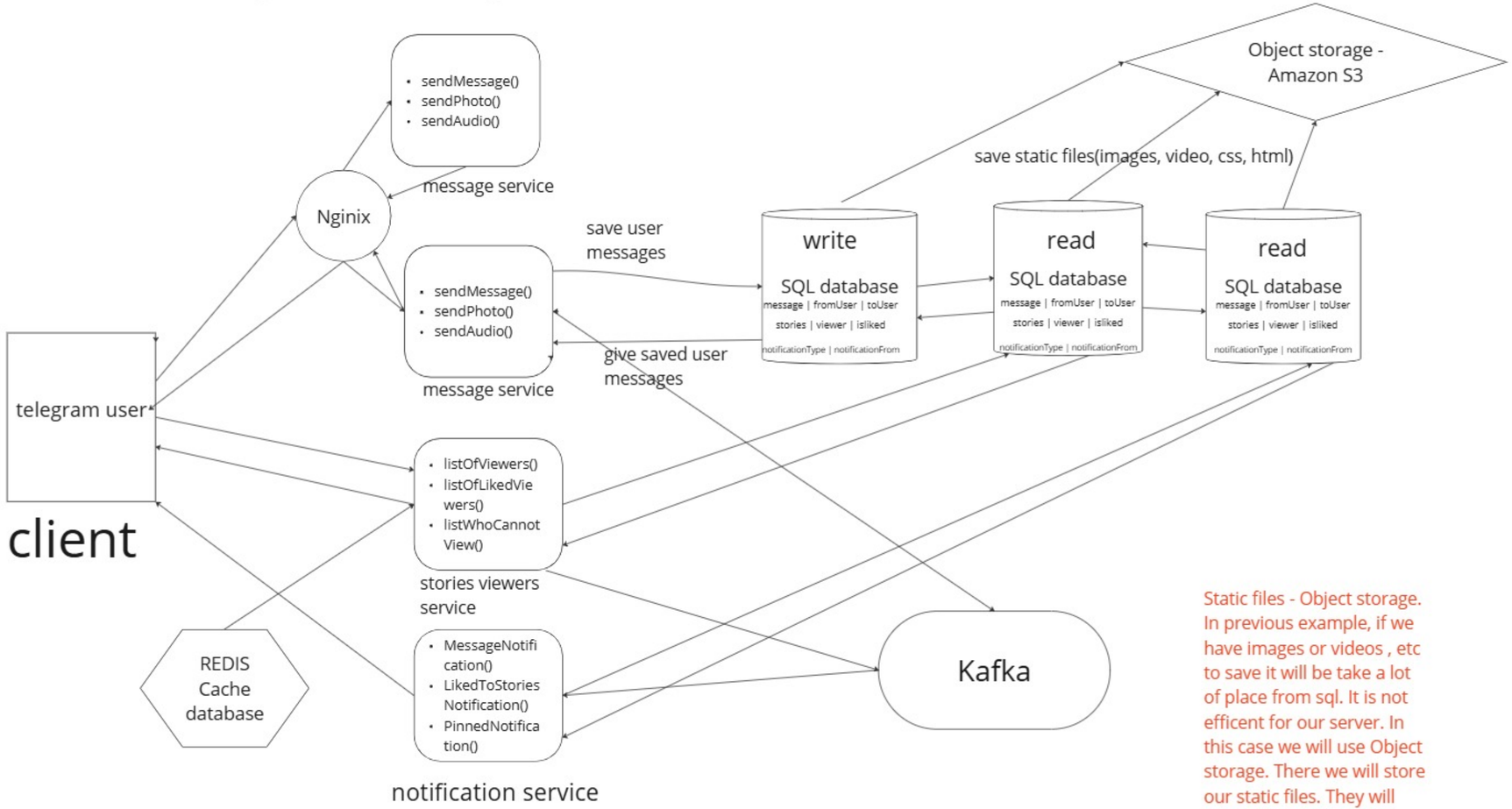
Load Balancer.
In previous case. We have was just one message service, In case message service will more used it can be slow. To solve this we can add two message service server. To handle requests from users we use proxy nginix, it will send request to least loaded server. It is increase performance of our service,

Message Broker

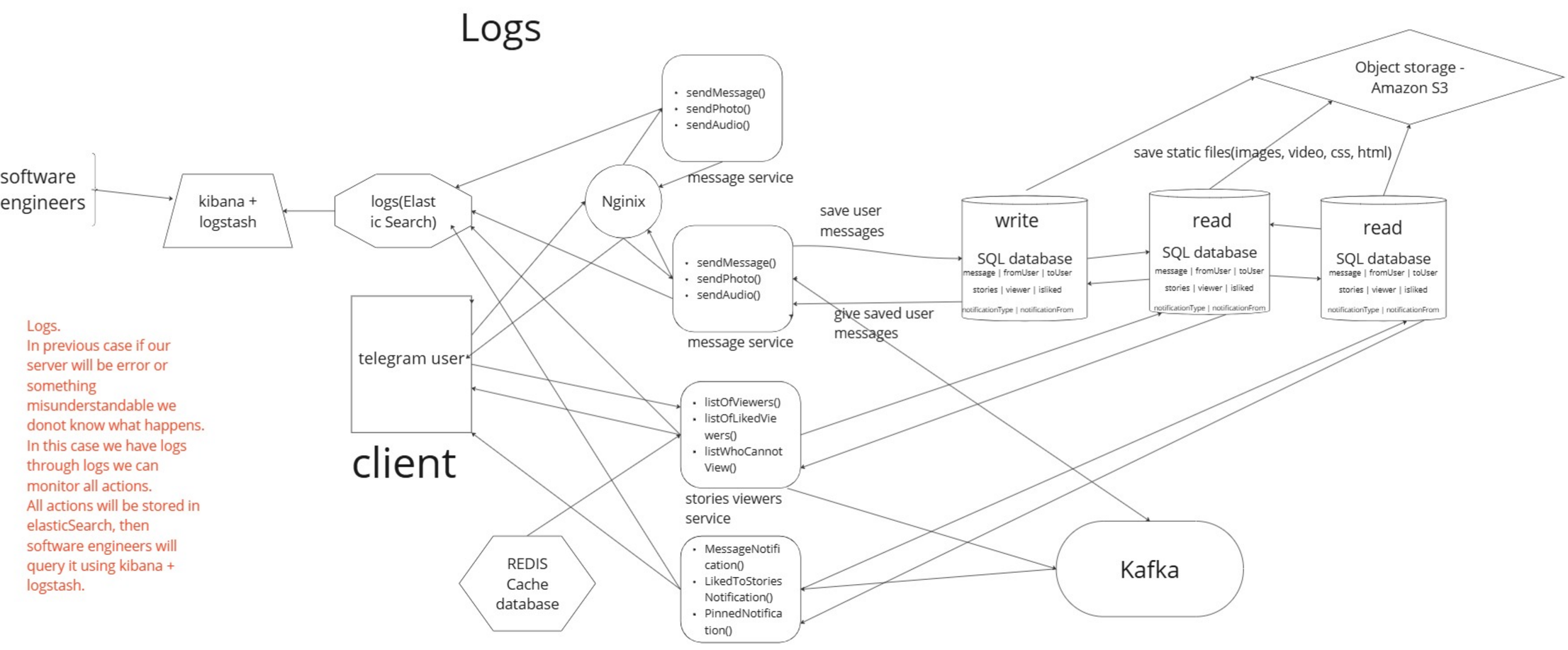


Message Broker.
In previous case we needs our micro services communicated with http. In this case services communicated with each other using message broker, It helps us improve performance, it will be faster. Other service does not wait other service(independent), it will continue own work, this will improve speed of services.

Static files - object storage

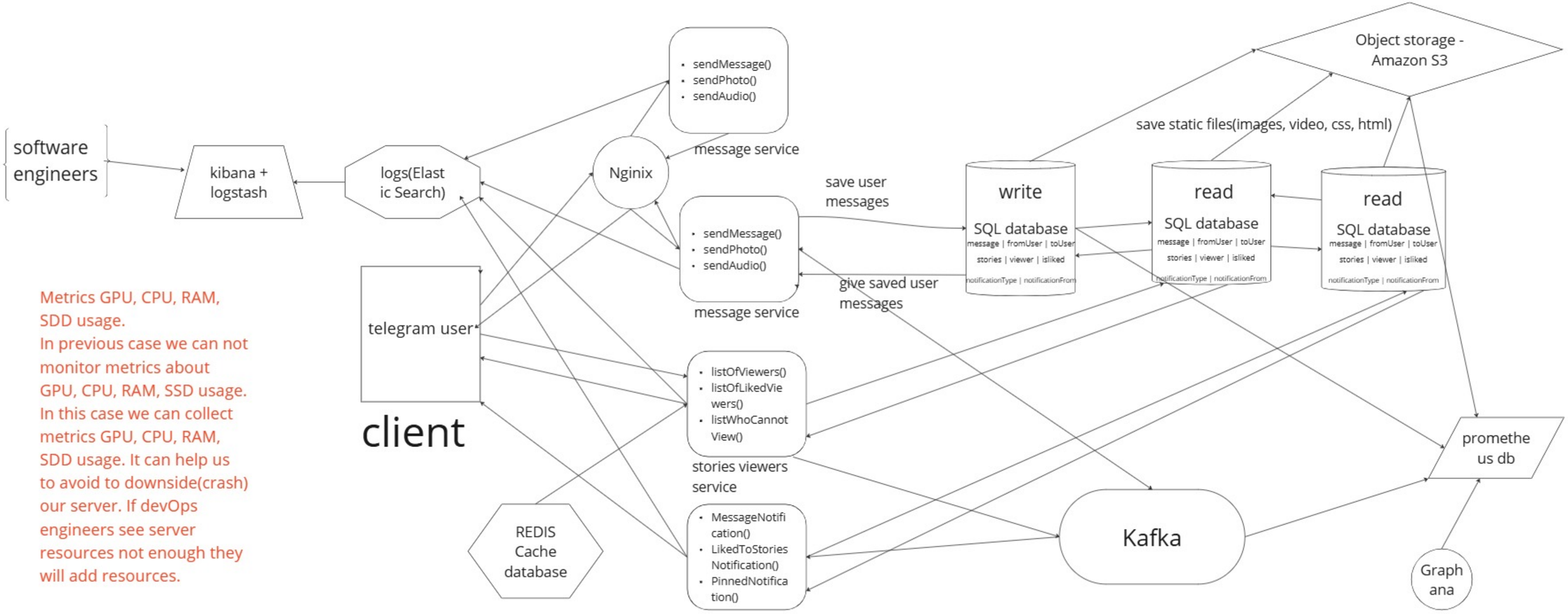


Static files - Object storage. In previous example, if we have images or videos , etc to save it will be take a lot of place from sql. It is not efficent for our server. In this case we will use Object storage. There we will store our static files. They will provide content faster using location.



Logs.
In previous case if our server will be error or something misunderstandable we donot know what happens. In this case we have logs through logs we can monitor all actions.
All actions will be stored in elasticSearch, then software engineers will query it using kibana + logstash.

Metrics GPU, CPU, RAM, SSD usage



Metrics GPU, CPU, RAM, SDD usage.
In previous case we can not monitor metrics about GPU, CPU, RAM, SSD usage. In this case we can collect metrics GPU, CPU, RAM, SDD usage. It can help us to avoid to downside(crash) our server. If devOps engineers see server resources not enough they will add resources.