

# Chat system design:

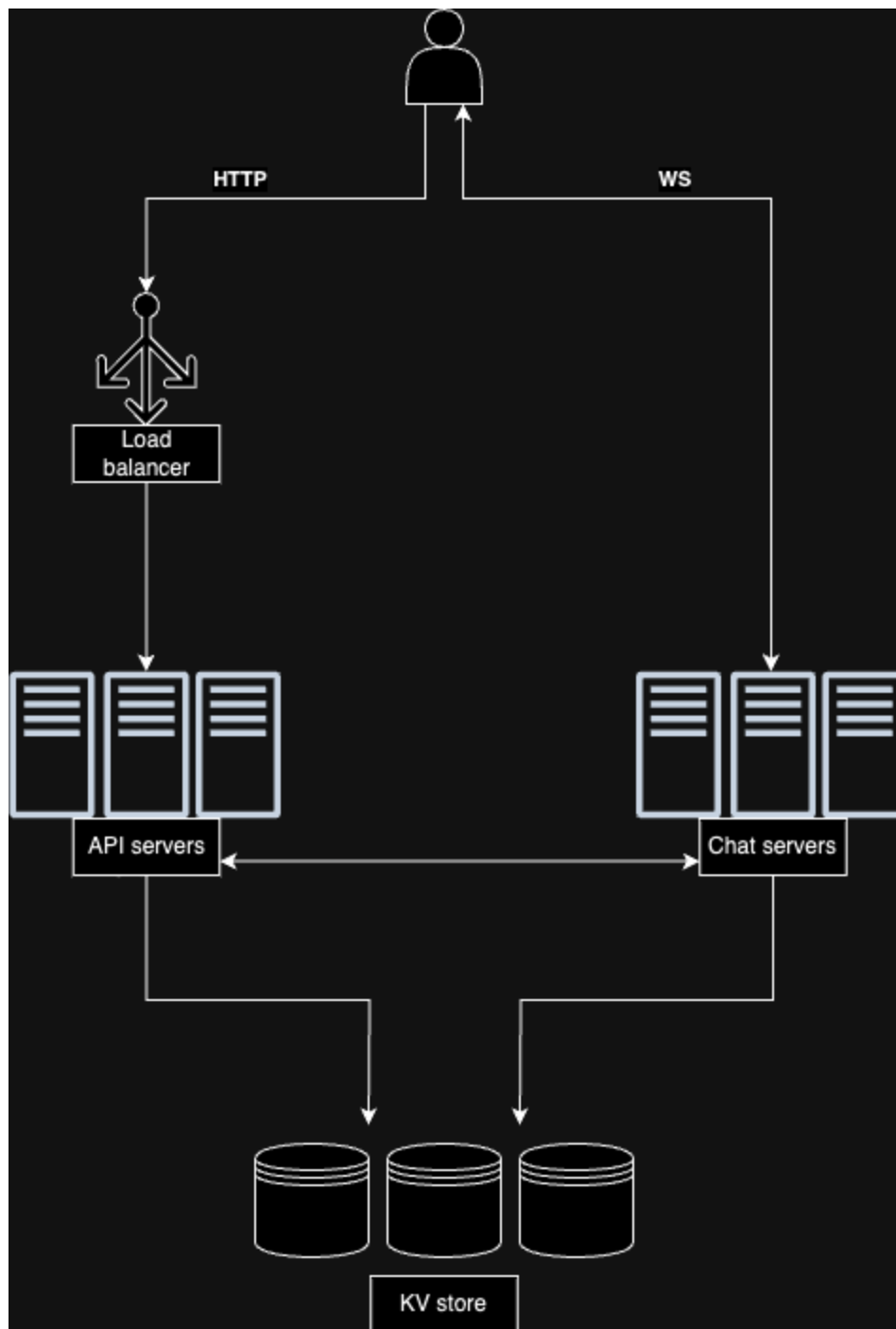
## Założenia funkcjonalne:

- czat grupowy umożliwia przesyłanie wiadomości tekstowych do innych użytkowników będących w tej samej grupie
- użytkownik jest automatycznie dodawany do danej grupy na podstawie jego zapisów na przedmioty

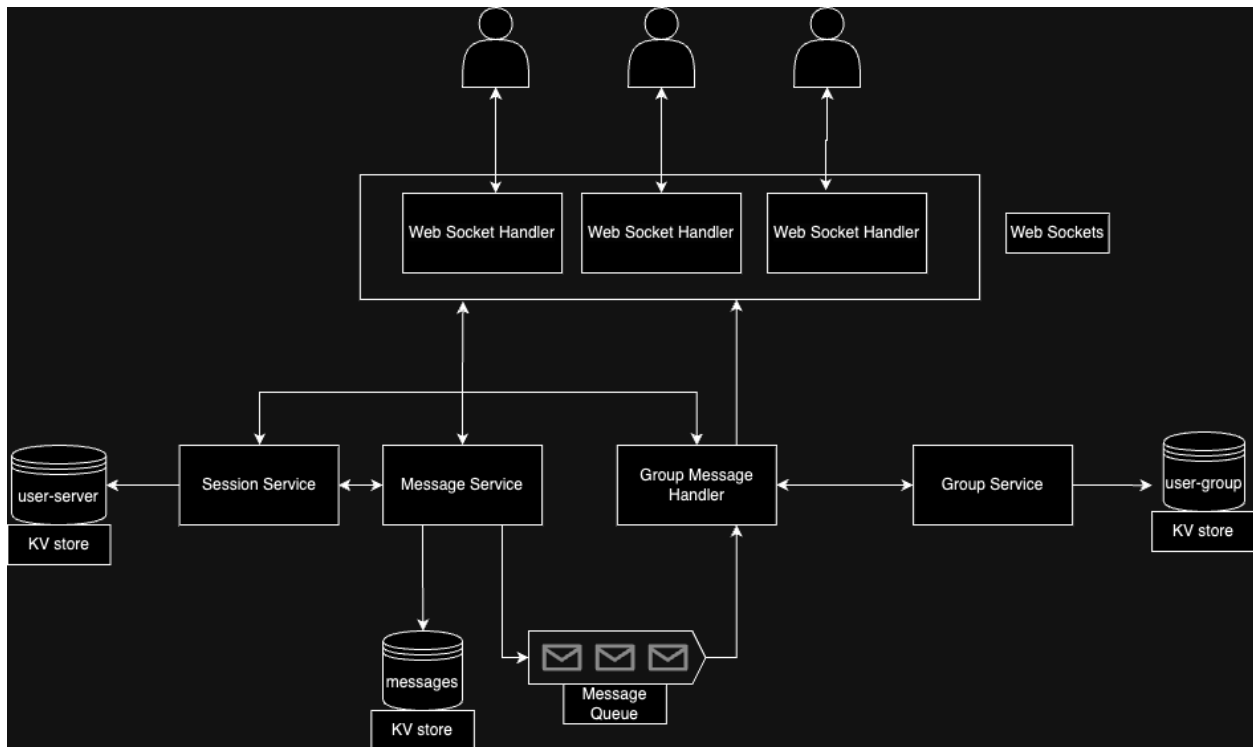
## Założenia niefunkcjonalne:

- limit 150 uczestników czatu
- bez załączników
- limit 400 znaków w jednej wiadomości (~3.125Kb)
- brak szyfrowania end-to-end
- brak synchronizacji pomiędzy różnymi urządzeniami
- brak monitorowania statusu użytkownika (presence service)
- wszystkie wiadomości są przechowywane w persistent storagu (stała baza)
- 500 daily active users (DAU) całego systemu
- skalowalny
- opcjonalny service discovery

## Diagram wysokiego poziomu:



**Diagram dla serwisów komunikatora (chat servers):**



**Message Service** - wysyłany jest request użytkownika z wiadomością i id grupy do tego serwisu, następnie za pomocą Session Service odczytuje, gdzie są docelowi użytkownicy. Przekazaną przez użytkownika wiadomość wrzuca do Message Queue (z informacją jaki użytkownik wysyła do której grupy, jaką wiadomość). Działa jako Publisher. Gdy użytkownik łączy się z chatem, Messaging Service przekazuje informacje do Session Service, do którego serwera jest podłączony ten użytkownik. Dodatkowo zapisuje wysłane wiadomości do trwałej bazy danych

**Session Service** - przechowuje informację o użytkownikach podłączonych do danego serwera/websocketa

**Group Message Handler** - nasłuchuje na nowe wiadomości pojawiające się w Message Queue. Za pomocą Group Service zbiera listę użytkowników należących do danej grupy będącej w wiadomości. Odczytuje informacje o serwerach/websocketach do którego są podłączeni wybrani użytkownicy. Na końcu przesyła wiadomość do określonych Web Socket Handlerów do których są podłączeni właściwi użytkownicy (odbiorcy wiadomości)

**Group Service** - przechowuje informację o użytkownikach należących do danej grupy

**Web Socket Handler** - odpowiada za komunikację użytkownika z chat serverem

## Endpointy API:

Od strony użytkownika:

1. **send\_message**(sender\_userID, receiver\_groupID, text)
2. **get\_message**(groupID, screen\_size) - zwraca listę x najnowszych wiadomości

## Schemat danych:

Wiadomości będą przechowywane w bazie Key Value, ze względu na ich łatwą skalowalność, małe opóźnienie przy dostępie do danych.

GroupMessages	
<u>channel_id</u>	<u>bigint</u>
<u>message_id</u>	<u>bigint</u>
user_id	<u>bigint</u>
content	<u>text</u>
created_at	<u>timestamp</u>

Struktura wiadomości  
przechowywana w bazie KV

Sessions	
user_id	<u>bigint</u>
server_id	<u>bigint</u>

Mapowanie użytkownika z  
serwerem do którego jest  
połączony

Groups	
group_id	<u>bigint</u>
user_id	<u>bigint</u>

Mapowanie użytkowników  
należących do danej grupy