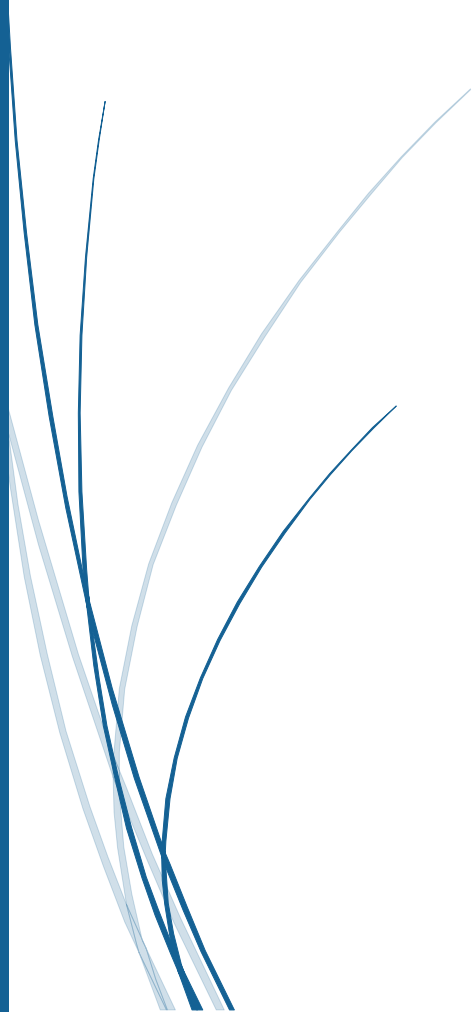
6/19/2022

S.A.D.

Kwetter-case



Samir Zalmay
FONTYS HOGESCHOLEN

CONTENTS

CH1. Introduction **2**

CH2. System context Diagram (C1) **3**

Ch3. Container Diagram **4**

GDPR-Compliant 5

Ch4. Component Diagram **6**

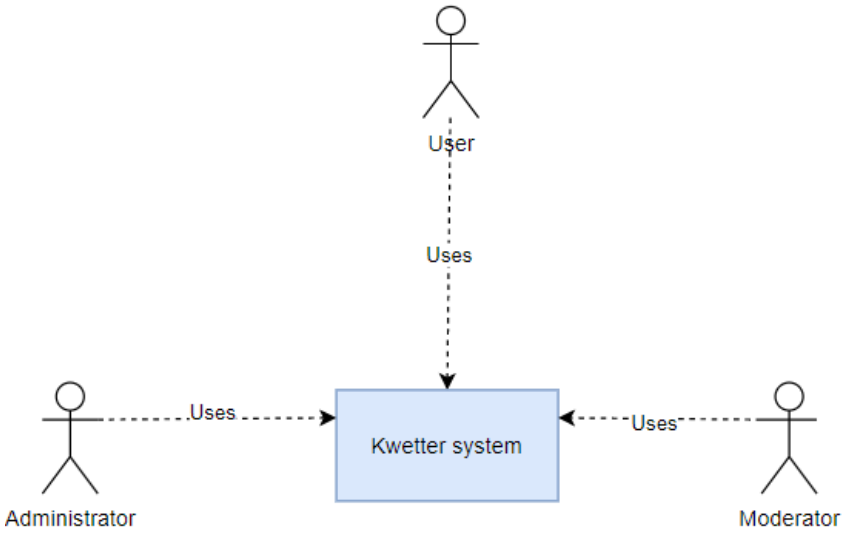
Ch6. ERD-MODel **7**

CH1. INTRODUCTION

The design and execution of Enterprise software is a major focus this semester. I'm going to demonstrate what I've learned through this project. Enterprise software must be able to support a high number of users doing actions on the program at the same time. It must, of course, be capable of handling a large amount of data transfer. The product's security, scalability, and maintainability are other significant considerations for this project. The plan of approach must be carefully examined in order to be able to implement all of this as efficiently as feasible.

This document demonstrates how I used the C4 model to create the architecture for this project. A picture of the finished object is made using architectural models. In this paper, all of the thoughts and facts gathered are organized in a logical manner.

CH2. SYSTEM CONTEXT DIAGRAM (C1)



There are three sorts of actors in this diagram: normal users (users), moderators, and administrators. The chatter system is used by ordinary users to send messages, read messages from others, follow persons, and so on. The moderators can do the same thing, but they have more authority than regular users. They can navigate through all users to see who has what role. They have the power to ban users who break the rules. Finally, moderators can delegate the function of moderator to other people. The administrators, on the other hand, have the most privileges of all categories of users. They have the same abilities as regular users and moderators. They can also add and remove other users' accounts. Finally, they can add/remove roles from other users.

CH3. CONTAINER DIAGRAM

In this chapter you can find the currently applied architectural design of the application and a short explanation of what each component stands for. The application is a microservice setup with Kafka for messaging and makes use of Ocelot framework in the API-Gateway.

The container diagram is shown in the picture below. I focused on the chatter system here. When consumers want to use the system, they communicate with the other services via the web application. Data is supplied and received from the API gateway via the web application. When a user logs in, for example, a message must be sent to the Login service over the API gateway. The Login service verifies the user's information. The Login service then sends a message back to the web application via the API gateway, granting the user access to rest of the application.

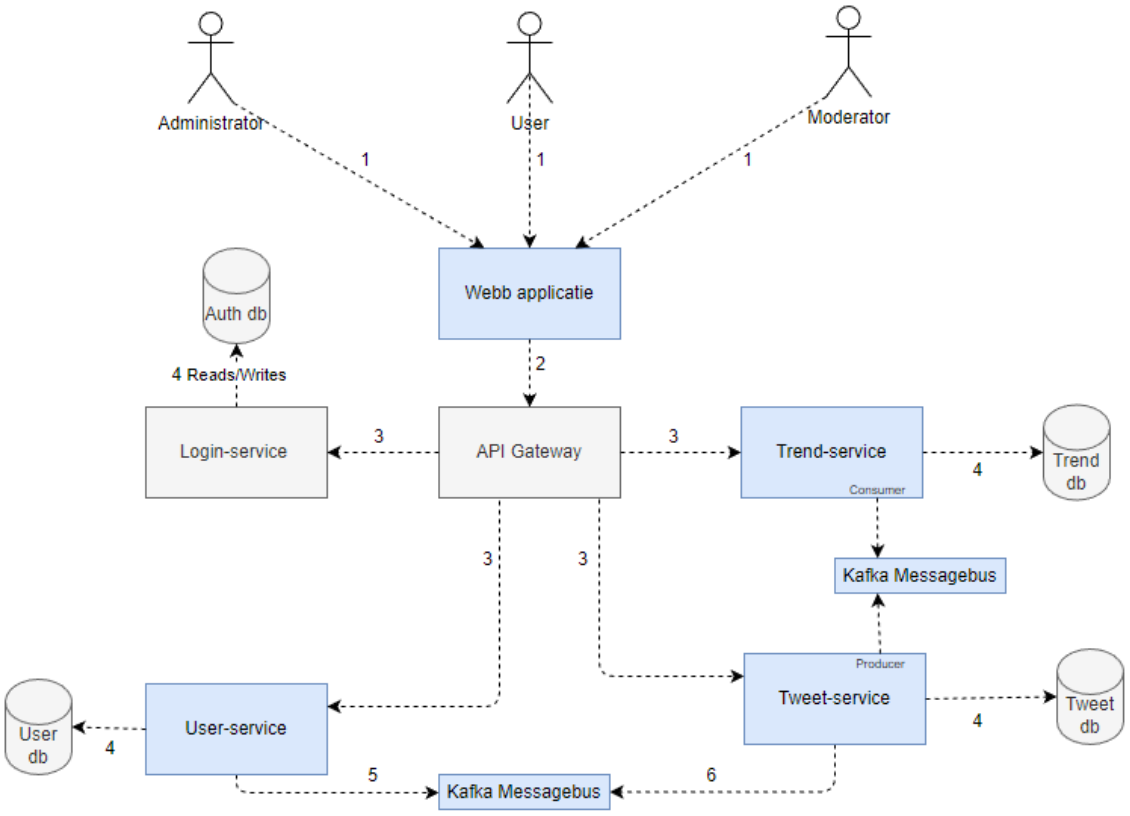


Figure 1, Container-Diagram, see table below for elaboration

Nr.	Data/service-call	Description
1	-	User makes use of the frontend of the Web-application to make use of the services.
2	Requests	Frontend (Vue-JS) makes request to API-Gateway (Ocelot) to either Post or request data.
3	Requests/Posts	API-Gateway (Ocelot) sends various requests and posts to all services. The user needs to be authenticated for all services, except for the Login-Service, for the API-Gateway to be able to make the requests/posts.
4	Requests/Posts	Persists/gets data to/from the database for each service.
5	Posts	Produces messages for the Message Bus.
6	Requests	Consumes messages from the Message Bus.

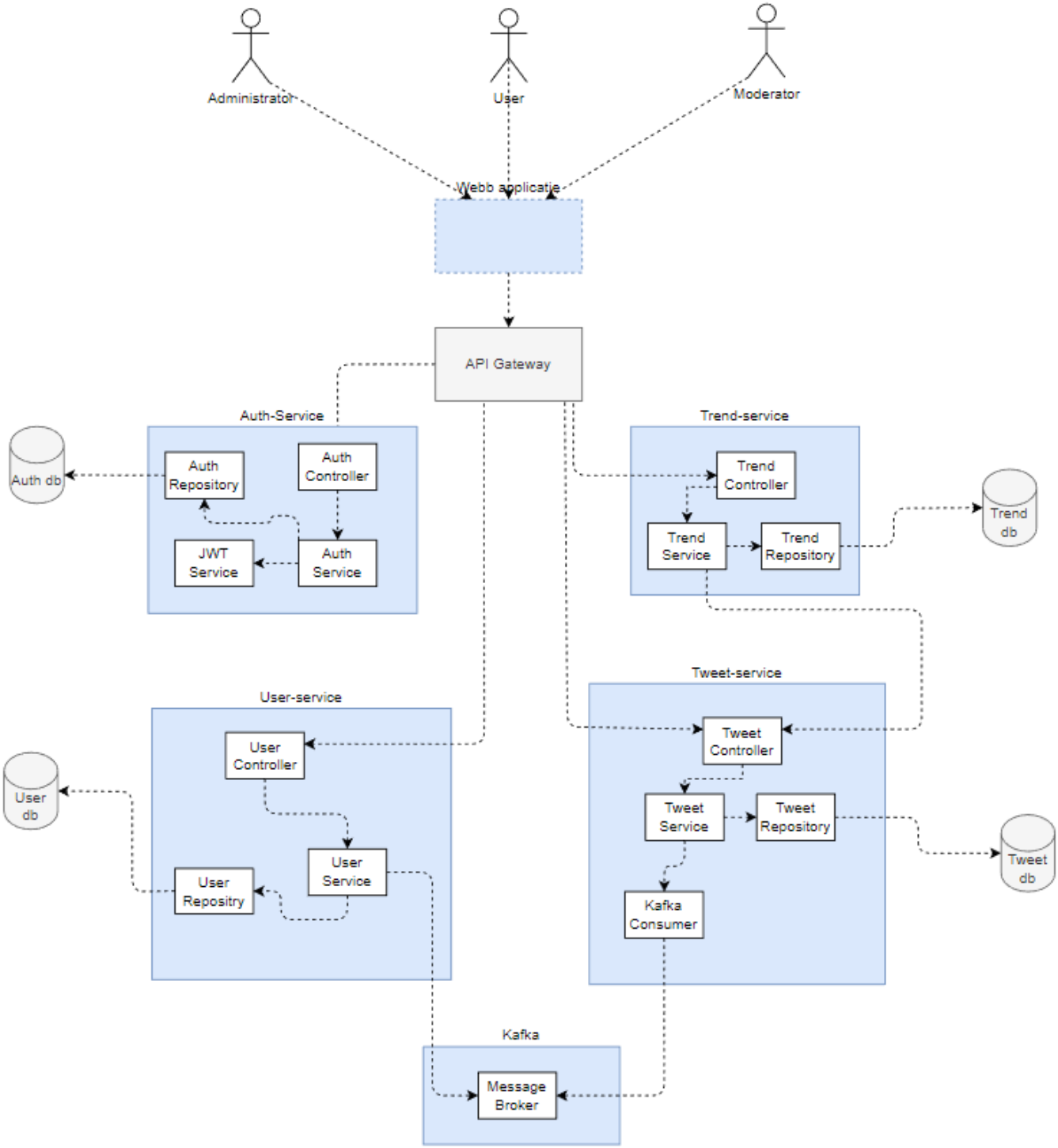
GDPR-COMPLIANT

As GDPR is of outmost important to be compliant to, a short description is given here. In the login service only the username, password and role are saved of the user and the user information in the User-Service. This to make sure the data is separated from each other, and the user will always have an option to re-activate his/her account.

As for the second point, all user data that is only needed for comparison (like password), will be hashed and salted by using BCrypt.

As final point, the user is able to completely delete his/her account, which means all personal information will be deleted, but also all tweets, as this might have personal information in them. Kafka message bus is used to make sure once a user deleted his/her account, all the tweets will be deleted as well.

CH4. COMPONENT DIAGRAM



CH6. ERD-MODEL

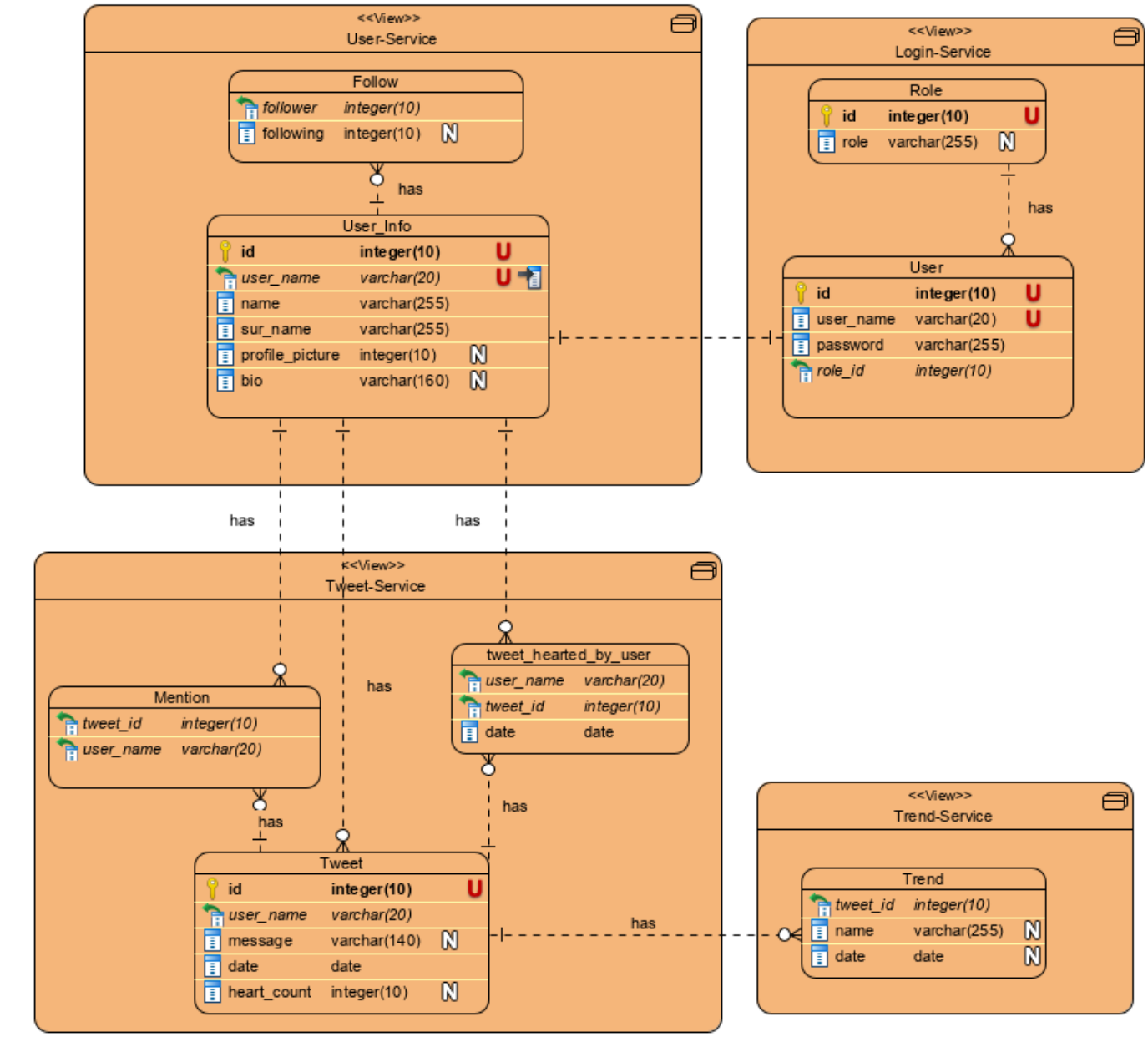


Figure 3, ERD-model