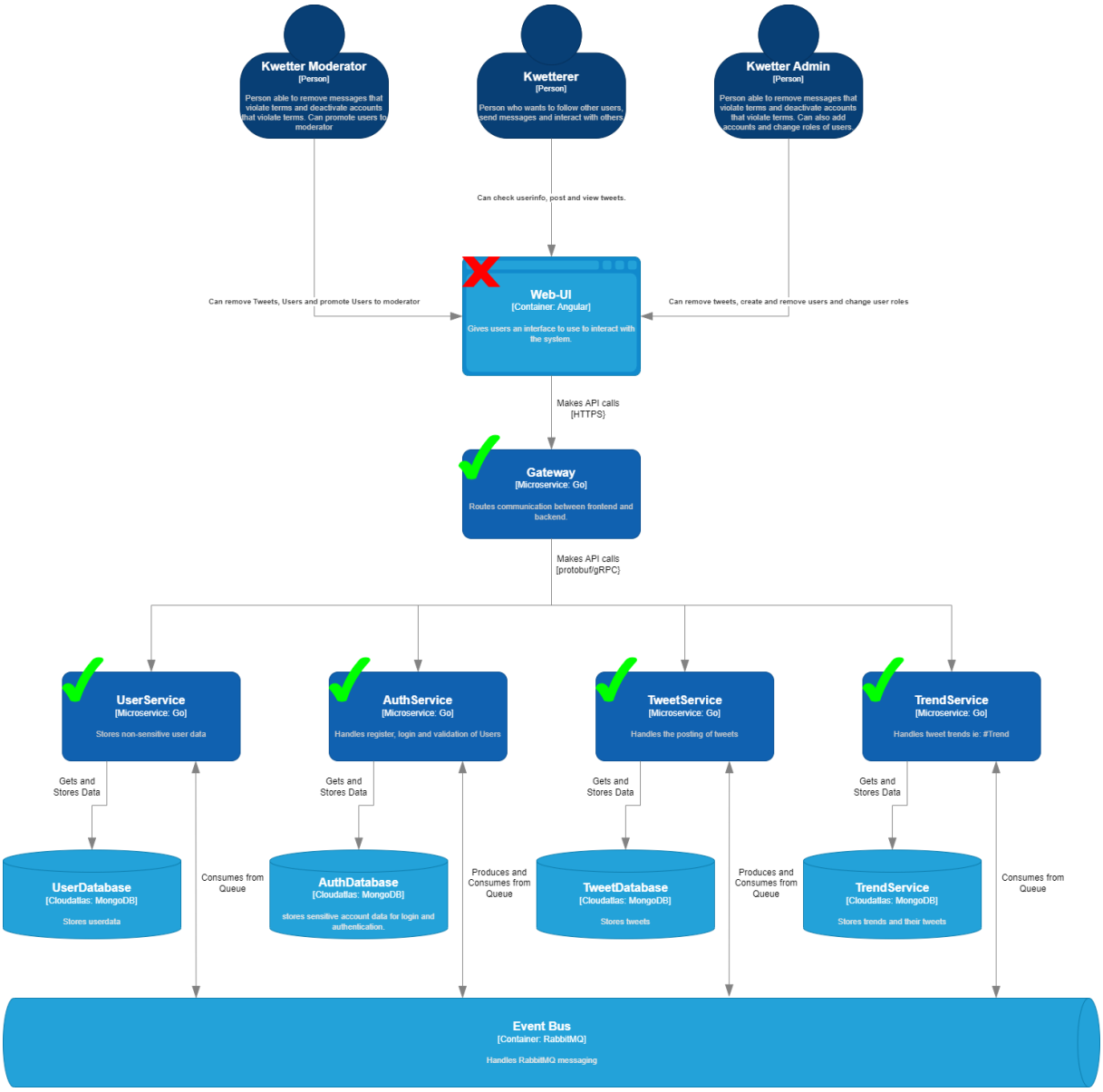
Distributed Data

In this document I want to explain how I handle data through my microservices and how this data gets stored. In addition I want to explain the extra steps I took to be GDPR compliant.

Microservices



I have the following microservices with the following exposed actions:

Gateway

- GetAllUserData: endpoint that asks the TweetService, UserService and AuthService to provide all data for the given UserID. The idea is that this data can later be downloaded to give full insight into UserData. This endpoint does not store data.

AuthService

- Register: Asks a user if they give permission to store data. If not the operation will abort. Otherwise the service will hash the password and store email, password and datapermission in its database.
- Login: Checks given email and password against the database to see if valid. If yes it generates a JWT token.
- Validate: Checks if token is valid and if so returns the userid linked to the token.

UserService

- UpdateUser: Not implemented.
- DeleteUser: endpoint that sends a message to the event bus which causes the AuthService and TweetService to start deleting all data found with the appropiate userid. Once all data has been deleted from these services the UserService will delete all data associated with the ID.

TrendService

- GetTrends: Not implemented.

TweetService

- ReturnTweet: Not fully implemented. Gets a tweet based on the objectid. Intended for use with an interface.
- PostTweet: Takes the UserID from the authorised usertoken in the gateway and stores the Tweet into it's database with the userid.

These are all endpoints exposed through the gateway. There are ofcourse other endpoints used by the services themselves. I will describe a few common actions a user will perform in my application and what data gets stored along the way.

User actions

Register

- User gives permission to store data (datapermission = true.) sends an email and password.
- If data permission = false, abort.
- If email is found in the database, abort
- Password gets hashed and email, password and datapermission gets stored in authdatabase.
- Event gets sent to event bus and gets consumed by UserService
- UserService has received email and userid from AuthService and stores this in UserDatabase.

PostTweet

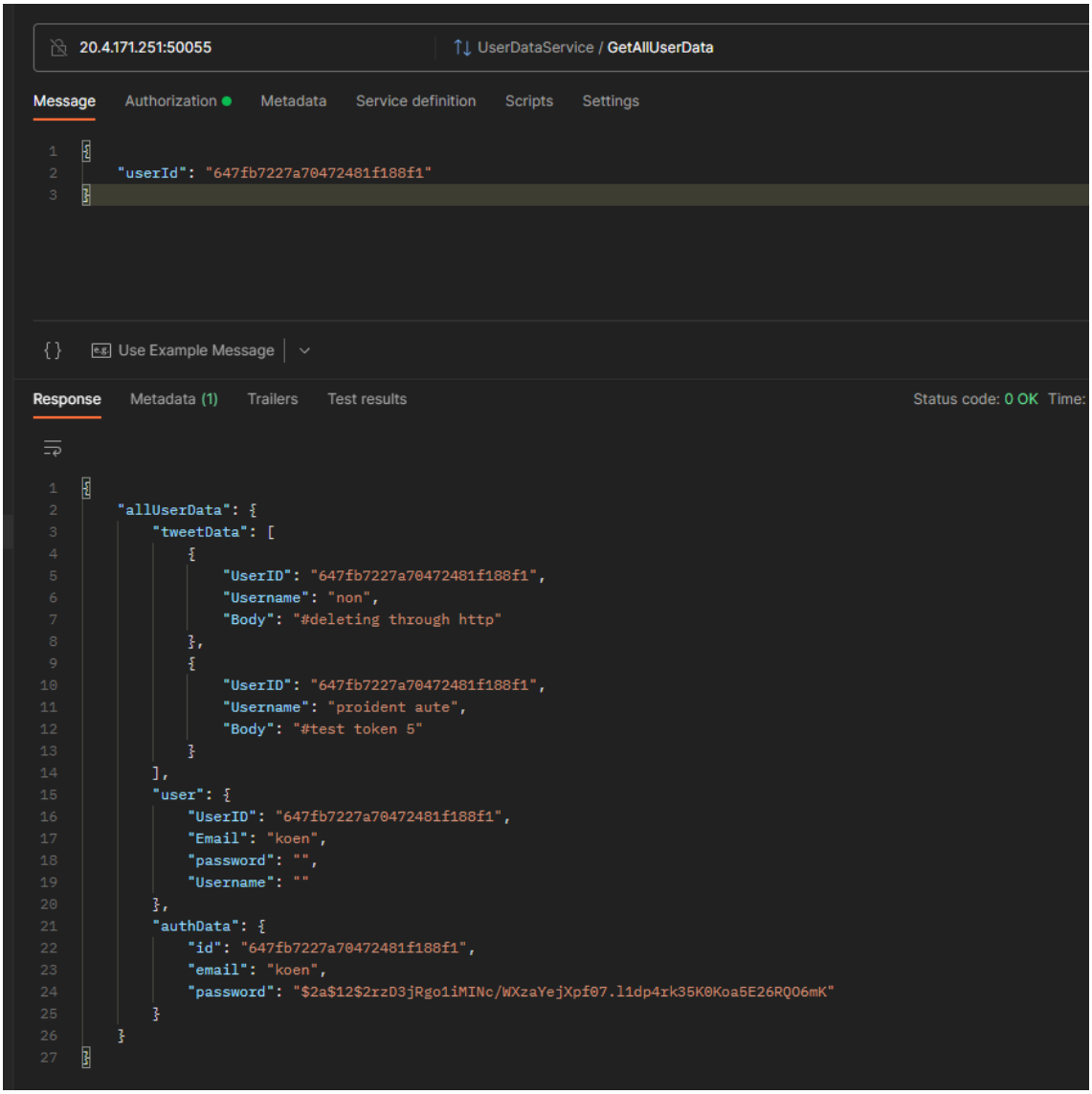
- User sends posttweet request with a body.
- Header gets checked for valid token by authservice/validate.
- If invalid, unauthenticated and user has to login.
- If valid, userid of posttweet request gets set by the validate response.
- Posttweet reaches the tweetservice.
- If tweet has a '#' in the body the tweet gets send to the event bus and gets consumed by the trendservice.
- Trendservice scans and extracts the hashtags and adds them to an array.
- Trendservice stores the same tweet in the trenddatabase but adds an arrayfield of hashtags to the record.

Hopefully this shows somewhat how I handle data. Due to the asynchronous nature of this program and microservices there can not be a single source of truth when it comes to data. Therefore a lot of data has to be stored in duplicate across seperate databases. And extra care has to be taken tot he ACID principle of databases to ensure that the source of truth is not in conflict.

GDPR

As hinted at in previous segments I took some special measurements to comply with GDPR. I will list them in bullet points below.

1. Users have to agree to have their data be stored/handled when registering. It is impossible to register if you do not agree to this.
2. Users have the ability to delete ALL of their data through the DeleteUser endpoint.
3. Users have the ability to gather all their data through the GetAllUserData endpoint. Due to the lack of frontend interface downloading is not possible at the moment.



Data formats

All my data formats can be found in the associated .proto files in my github. (https://github.com/Portfolio-Adv-Software/Kwetter)

These are found in the subdirectories microservice/internal/proto/.proto