6/19/2022

GDPR

Kwetter-case

Samir Zalmay
FONTYS HOGESCHOLEN

CONTENTS

CH1. INTRODUCTION

In this document I will explore what distributed data really stands for regarding GDPR and I will try to make my application as viable as I can for GDPR.

CH2. WHAT IS GDPR?

The general data protection system is an EU statute that protects the personal information of European internet users. This law is also known as the AVG, or the Algemene verordening gegevensbescherming wet, in Dutch. There have been a lot of various conceptions described. For a company, in order to be GDPR-compliant, these conceptions are a must to hold on to. In this chapter I will describe three factors that are important, and what to look for when making a project.

2.1 PERSONAL DATA

Almost every engagement with an online organization necessitates the submission of personal information. Consider your name, phone number, and address. Frequently, one of these data sets is insufficiently valuable. However, if an organization has gathered additional information and data from you, this information and data might become quite useful. Personal Identifiable Information, or PII, is another term for this. It is possible to make this data anonymous in order to render it useless. But it must be irreversible in order to be genuinely anonymous. As a result, material that is anonymous but may be linked to a specific person is still considered personal data.

Companies must follow numerous laws when it comes to personal data, according to the GDPR. Companies, for example, are no longer permitted to request information that they do not require. When buying a mattress, it is no longer necessary for a company to know whether you are a male or a woman. As a result, it is no longer permissible for a firm to request this. Furthermore, businesses are only permitted to keep personal data for as long as they require it.

Companies can ensure that the information they store cannot be linked to a specific individual. Encryption or the use of pseudonyms are two ways they can do this. Data is modified with pseudonyms so that it can no longer be traced back to a specific person. Consider updating your friends list or your saved IP addresses.

2.2 PERMISSION

Companies must give permission to store their data under the GDPR. This privilege can be revoked at any moment by the user. As a result, users must take action to grant permission while storing data. Furthermore, the information relevant to storage must be explained in a way that all users can comprehend. Users' data may not be used in any way other than what is indicated in these terms by the organization.

Users have the right to access their own data at any time because it involves their personal information. The business must next make sure that all of the data they have on this user is made available to them. Users also have the right to have all data associated with them at the organization removed at any time. If a user no longer requires the organization's services, the company is required to erase all of the user's data. This includes not only the user's profile on a social networking platform, but also friendships, posts, likes, comments, and the like. There should be nothing left for this user to suffer.

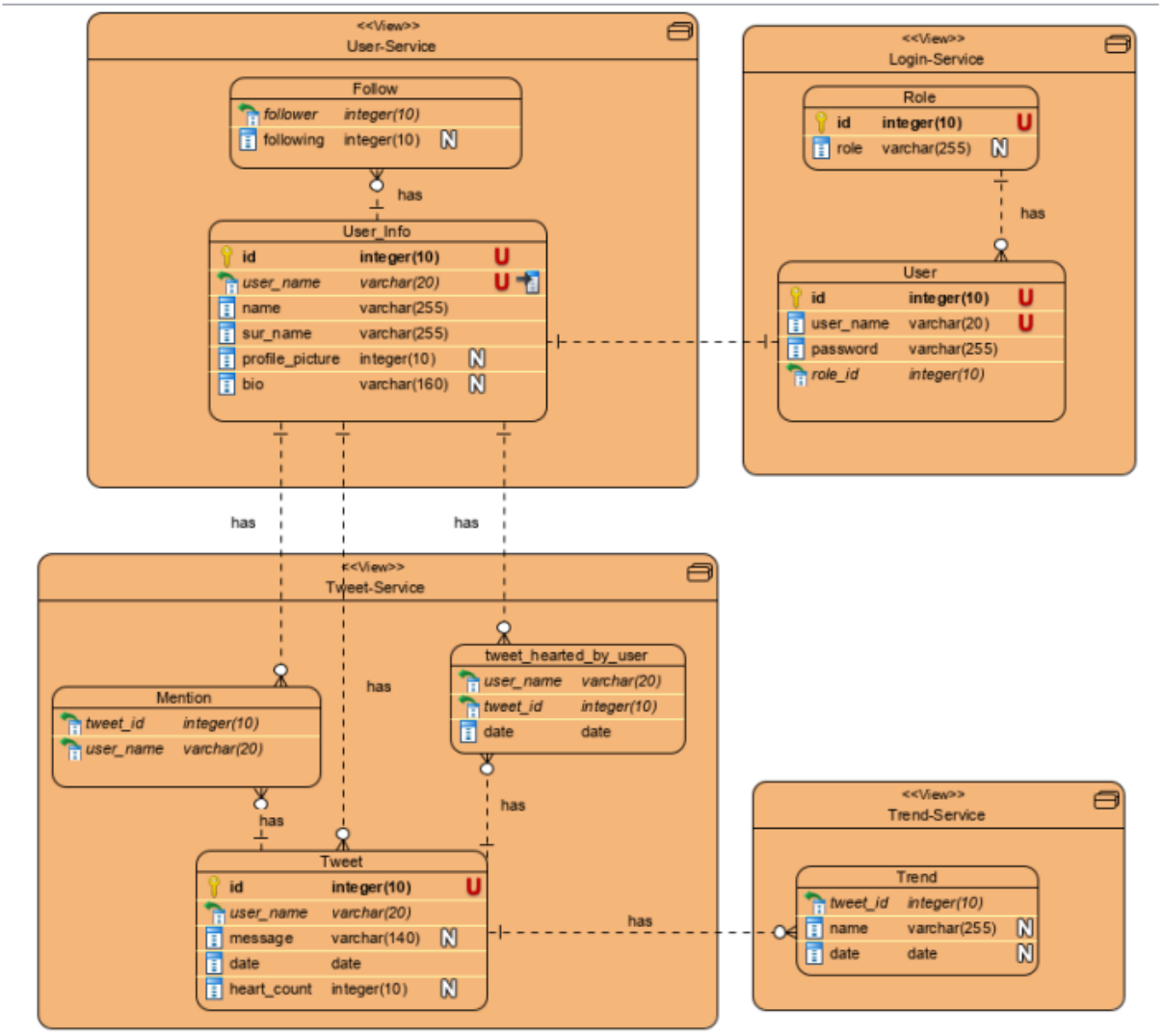
2.3 LEGITIMATE INTERESTS

Only data that is beneficial to the organization should be stored. According to the GDPR, there are several legitimate interests. This includes marketing, fraud prevention, and IT security, as well as the usage of consumer or employee data. When employing these interests, you might ask yourself three questions to see if they are truly important. The goal; why do I want this data; interest; is this data truly important for the process; and balance; whether it supports the user's or the organization's interests.

CH3. GDPR IMPLEMENTATION

3.1 STEPS TAKEN TO BE GDPR-COMPLIANT

Before the research, I already took some steps into being GDPR-compliant. The first goal for me was to setup my Microservices Architecture. This included in separating my database into the different Microservices. The ERD-model below shows how each database is setup. This to keep personal information separated from the Authentication (which is the one most likely to get attacked).



Furthermore, before saving the password in the database, the password is hashed and salted with BCrypt as you can see in the image below. This to ensure users that even when the passwords, are hacked, as long as your password is strong enough with multiple numbers and signs.

```
2 references
public void Register(User user)
{
    if (GetUser(user.UserName) != null)
    {
        throw new DuplicateNameException();
    }

    user.Id = new Guid();
    user.Password = BCrypt.Net.BCrypt.HashPassword(user.Password);

    userContext.Add(user);
    userContext.SaveChanges();
}
```

The most important implementation I did for GDPR is adding the delete me functionality in the backend. This endpoint will make sure that everything regarding the user's personal information will be deleted.

```
// DELETE: api/Users/5
[HttpDelete("{id}")]
0 references
public async Task<IActionResult> DeleteUser(Guid id)
{
    var user = await _context.User.FindAsync(id);
    if (user == null)
    {
        return NotFound();
    }

    _context.User.Remove(user);
    await _context.SaveChangesAsync();
    SendToKafka(topic, id.ToString());

    return NoContent();
}
```

As you can see in the image above, there is a function SendToKafka(), in which the userId will be send from the producer User-Service to Kafka on the topic "delete_user_tweets_topic". The Console.Writeline and GetResult are added to make sure the message is produced.

```
1 reference
private void SendToKafka(string topic, string message)
{
    using (var producer =
        new ProducerBuilder<Null, string>(config).Build())
    {
        try
        {
            var test = producer.ProduceAsync(topic, new Message<Null, string> { Value = message })
                .GetAwaiter()
                .GetResult();
            Console.WriteLine(test);
        }
        catch (Exception e)
        {
            Console.WriteLine($"Oops, something went wrong: {e}");
        }
    }
}
```

In the picture below you can see how Kafka consumes the message and makes sure that every tweet is deleted off that user.

```
1 reference
public class KafkaConsumerHandler : IHostedService
{
    private readonly IServiceScopeFactory factory;

    0 references
    public KafkaConsumerHandler(IServiceScopeFactory factory)
    {
        this.factory = factory;
    }
    private readonly string topic = "delete_user_tweets_topic";

    0 references
    public Task StartAsync(CancellationToken cancellationToken)
    {
        using var scope = factory.CreateScope();
        var tweetService = scope.ServiceProvider.GetRequiredService<ITweetService>();

        var conf = new ConsumerConfig
        {
            GroupId = "st_consumer_group",
            BootstrapServers = "localhost:9092",
            AutoOffsetReset = AutoOffsetReset.Earliest
        };
        using (var builder = new ConsumerBuilder<Ignore,
            string>(conf).Build())
        {
            builder.Subscribe(topic);
            var cancelToken = new CancellationTokenSource();
            try
            {
                while (true)
                {
                    var consumer = builder.Consume(cancelToken.Token);
                    Console.WriteLine($"Message: {consumer.Message.Value} received from {consumer.TopicPartitionOffset}");
                    tweetService.DeleteTweetsByUserId(consumer.Message.Value);
                }
            }
            catch (Exception)
            {
                builder.Close();
            }
        }
        return Task.CompletedTask;
    }
```

By activating the endpoint all user personal information is deleted, except for the auth-credentials. With these credentials the user is still able to visit the website and reactivate his/her account by filling in the missing fields.

3.2 FURTHER STEPS TO BE TAKEN

In chapter 2 you can read that there are multiple ways to keep consideration with GDPR. I have first handedly experienced that while working for Bull44 as a Software-developer. During this time I had to consider a lot of GDP-regulations as I was working with a Payment-system and lots of personal information of a user including creditcard number and all personal info. To make sure we are GDPR compliant all the steps in CH3.1 are already taken, but also the following that currently is missing from my individual project.

- **Frontend password checker.**
  In the frontend when registering, the password must be confirmed and at least consist out of 8 character with Capitals, numbers and signs. This will ensure that the password of the user is strong and not easily cracked. I would even suggest to make the password at least 10-12 characters long.
- **Delete user after 5 years of inactivity.**
  When a user is not active anymore, we are obligated to keep the records (with payments) of that user for 5 years. However, after 5 years, we will not be needing any more of the user personal data for bookkeeping/taxes and the user with all related personal information will be deleted (account stays safe). This will make sure that the company does not keep unnecessary personal information.
- **Terms-and agreement**
  As for the last point, the users need to be notified and accept that we will be saving their data in our database. If the user does not accept these terms, he/she will not have access to the website