**Distributed**

**Overview**

I am planning to produce a forum type of application, where you are able to create new accounts, login to the created accounts, and if you are logged into an account, you are able to create new posts and comment on posts. Non-logged in users will still be able to see all the posts and comments but will not be able to create their own. I will use JsonWebTokens (JWTs) for authenticating that a user is logged in. When a user logs in to the application, they will receive a JWT-token from the authentication microservice that will be stored in the browser’s cookies, and it will be used to authenticate their actions. There are some situations where authentication is required, such as when the user wants to create a new post or post a comment, and JWTs are quite an easy option to check if the user is logged in. I have some experience with JWT-tokens, so I think they are a good option for this task. To store all of the user account, posts, and comments, I will use a mongoose database to save all the information.

**System requirements**

* All users can see all posts and comments
* Only logged in users can create new posts or post comments
* Users are able to create new accounts, account must have a username, an email, and a password
* Usernames must be unique, a new account with an existing username can’t be created
* Upon successfully logging in, the user gets a JsonWebToken into their cookies, which will be used for authentication
* If a user has a valid JWT-token, they can see an interactable form to create new posts and comments

With these requirements we can identify at least 3 different microservices that can be produced.

**The microservices**

One microservice is an **authentication** microservice that takes care of creating new user accounts, and logging into to these accounts with login credentials such as an email and a password. When a user logs in, they will receive a JWT-token into their cookies which will be used to authenticate them to do certain tasks.

**Forum posts** would also need its own microservice. The microservice is used to handle the posts and this microservice would take care of creating new posts and having the possibility to fetch all of the posts to be displayed on a webpage. Authentication will be necessary to create new posts, and JWTs will be used to check if the user has a valid token. Fetching all of the posts however doesn’t require authentication, as everyone should be allowed to see the posts, but not create new ones.

**Comments** also need their own microservice. The comments are linked to specific posts, as the comments are posted on the comment section of a post. The microservice allows logged in users to post new comments and get all of the comments of a certain post. Users that aren’t logged in can see all of the comments but are not able to post their own comments. In this situation as well, JWTs are used for authentication to check if the user is allowed to post a new comment.

**Demonstrate the architecture with UML diagrams of your choice and a single architecture diagram describing the whole system architecture with a strong focus on communication.**

**Clearly demonstrate the communication pattern used and describe the limitations around communication for microservices.**