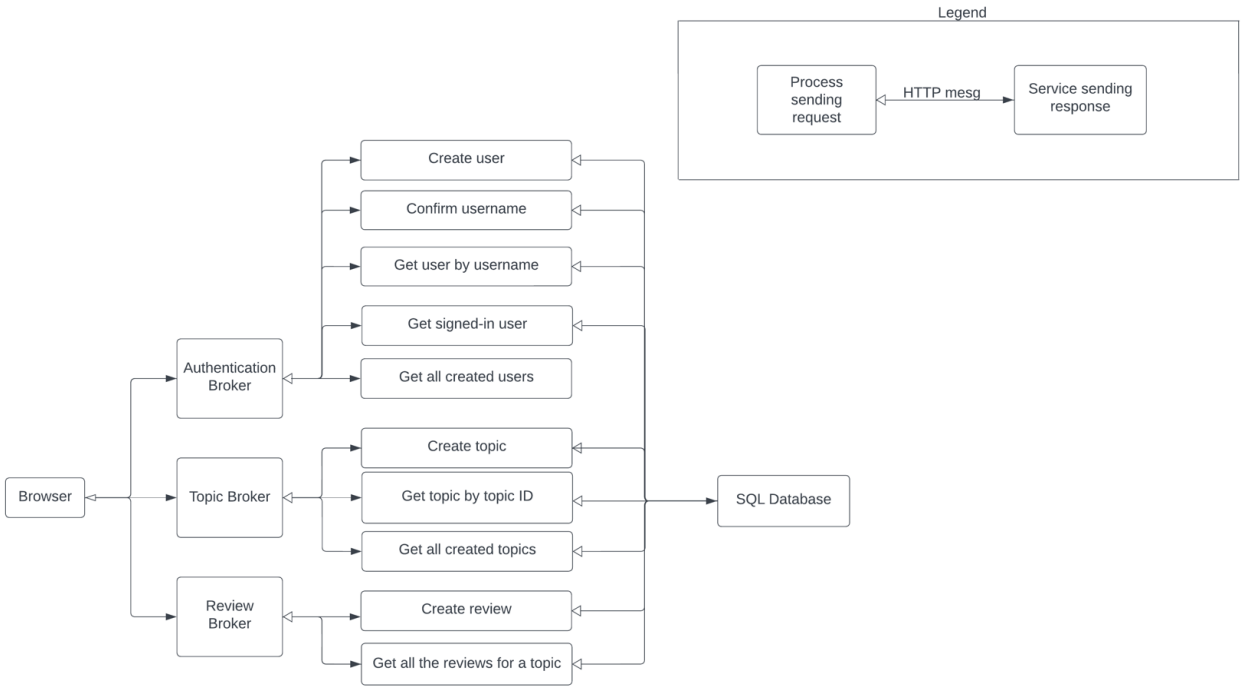
Microservices Architecture Document

Possible microservices identified



- Authentication broker
    - Create user.
    - Confirm username.
    - Get user by username.
    - Get signed-in user.
    - Get all created users.
- Topic broker
    - Create topic.
    - Get topic by topic ID.
    - Get all created topics.
- Review broker
    - Create review.
    - Get all reviews for a topic.

Comparison to component-based architecture

The component-based architecture had multiple components, each dependent on one another. In contrast, using a microservices architecture, the project is split up into

different services, each handling a specific domain of the project. Each service is small (hence the name ‘microservices’), independent of other services, and can be deployed to production separately.

Impact on team software process

Using a component-based architecture, the team members had meetings to discuss the connections between the components being built. Adopting a microservices architecture would change that. Since microservices don’t directly interact with other services, each team member working on a specific service would have more autonomy over the service’s implementation. Also, since service-to-service communication would be facilitated by certain tools (such as message brokers), team members would not need to worry about how the services connect.