1. The following paper is describing what is a monolithic application and a micro-service based one. There is also an analysis of the benefits and disadvantages of both approaches.

2. As stated, a monolithic application is the classic type of an application, where it works as one organism. All the features and functionalities of the application are running as a single execution.

3. A microservice-based application has its workflow divided on smaller applications. There are several services that each do their own work, and as a whole they represent the running application.

4. The monolithic application is the classic way of developing an app, which means that it is deployed as a whole, afterwards it works as a whole as well. When looking at a microservice-based application, it has more flexibility, because each service can be deployed and updated on its own, while keeping the other services running. In a similar way, these services are living by their own reasons each, which implies an individual database with which the service is manipulating data.

5. One can migrate to a monolithic application in terms of easier deployment. When working with a single running unit it is easier to go on a fast pace of developing and deploying the project. While one can strive to a microservice architecture for more scalability. When working with multiple services, where each of them is given an individual functionality, it is much easier to add new features or update the old ones.

6. When starting a new project, the team must look at the complexity of it. For instance, if the app is an online shopping app, then it is better to go with the microservice-based architecture. The reason is because there would be many services, like a shopping cart, payment, delivery checkup and many more, which would be more suitable to work on their own, as separate services and not overload the workflow of the project. On the other hand, if the project is one less complicated, such as a culinary notebook, then there is no need of splitting the functionality by individual entities. Its workflow is rather simpler and it would be more convenient to test and deploy it as a whole.

7. These two architectures are different in the terms of maintenance. Given the fact that monolithic applications are a big whole, then its testing and bug hunting is done on the whole project. When working with small projects it is not a big deal, but in bigger ones, it could affect the overall work potential of the app. Here I could see the risk of it crashing much higher. In microservice-based architectures, the maintenance is rather safer, for the reason that each service works individually, so in case one of them crashes, the dev team can freely work on it, while not affecting all the other microservices. This