**Monolithic vs Microservices Architecture**

**Monolithic applications**:

Diagram

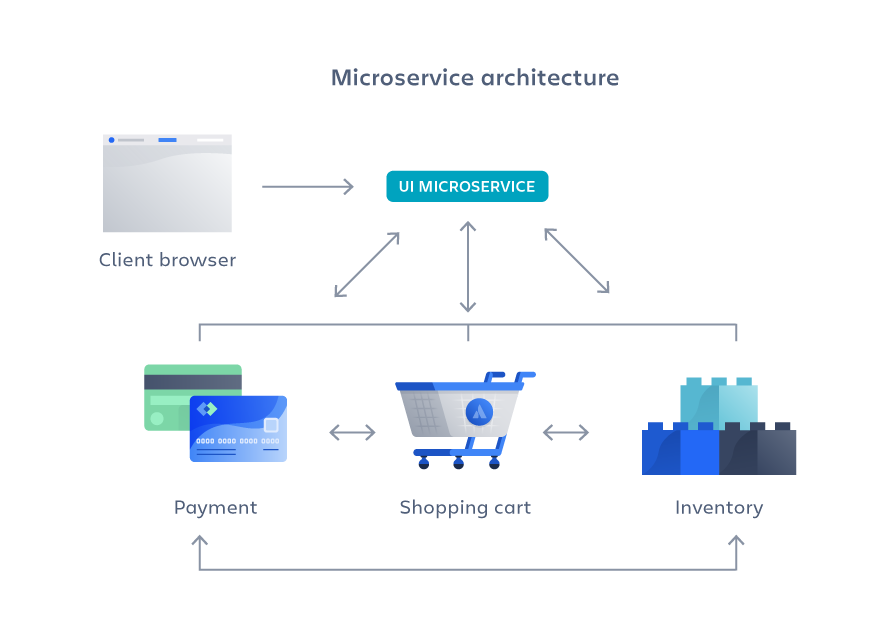
Description automatically generated

* If all the functionalities of a project exist in a single codebase, then that application is known as a monolithic application.
* We all must have designed a monolithic application in our lives in which we were given a problem statement and were asked to design a system with various functionalities. We design our application in various layers like presentation, service, and persistence and then deploy that codebase as a single jar/war file.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Simple to develop relative to microservices, where skilled developers are required in order to identify and develop the services. | becomes too large with time and hence, difficult to manage. |
| Easier to deploy as only a single jar/war file is deployed. | need to redeploy the whole application, even for a small change. |
| Relatively easier and simple to develop in comparison to microservices architecture. | As the size of the application increases, its start-up and deployment time also increases. |
| Developers need not learn different applications, they can keep their focus on one application. | very difficult to adopt any new technology which is well suited for a particular functionality as it affects the entire application, both in terms of time and cost. |

**Microservices applications**:

Diagram

Description automatically generated

* It is an architectural development style in which the application is made up of smaller services that handle a small portion of the functionality and data by communicating with each other directly using lightweight protocols like HTTP. According to Sam Newman, “Microservices are the small services that work together.”
* Instead of sharing a single database with other microservices, each microservice has its own database. It often results in duplication of some data, but having a database per microservice is essential if you want to benefit from this architecture, as it ensures loose coupling.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| If there’s any update in one of the microservices, then we need to redeploy only that microservice. | Being a distributed system, it is much more complex than monolithic applications. Its complexity increases with the increase in a number of microservices. |
| Microservices are self-contained and, hence, deployed independently. Their start-up and deployment times are relatively less. | Microservices are costly in terms of network usage as they need to interact with each other and all these remote calls result in network latency. |
| It is very easy for a new developer to onboard the project as he needs to understand only a particular microservice providing the functionality he will be working on and not the whole system. | Microservices are less secure relative to monolithic applications due to the inter-services communication over the network. |
| Each microservice can use different technology based on the business requirements. | Debugging is difficult as the control flows over many microservices and to point out why and where exactly the error occurred is a difficult task. |
| If a particular microservice goes down due to some bug, then it doesn’t affect other microservices and the whole system remains intact and continues providing other functionalities to the users. | Skilled developers are required to work with microservices architecture, which can identify the microservices and manage their inter-communications. |

Taken From:

* <https://www.geeksforgeeks.org/monolithic-vs-microservices-architecture/>

* <https://www.n-ix.com/microservices-vs-monolith-which-architecture-best-choice-your-business/>
* [https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith#:~:text=A%20monolithic%20application%20is%20built,of%20smaller%2C%20independently%20deployable%20services](https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith" \l ":~:text=A%20monolithic%20application%20is%20built,of%20smaller%2C%20independently%20deployable%20services).

Video Recommendations:

* <https://www.youtube.com/watch?v=RJkn9VHM7lc>
* <https://www.youtube.com/watch?v=0PR7fWZt27I>