**Case study: Monolith vs Serverless**

Monolith is software architecture where all the components of application are bundle up together in one codebase. The entire functionality of the application is written in one single code and deploy. Whereas serverless also know cloud computing model where the application by a third party organization.

**Monolith:**

Monolith architecture is used in most of the early developing stages of software application. Where all the features, services and database are in a single codebase and are deployed as one unit. All features, services and database all are interconnected.

In monolith the management of a small/medium code is quite feasible. The testing, development and deployment of the software will be quick, we need not to look at multiple codebases at once just to find one bug. All the features, services and database the interaction between each other easily and smoothly with zero to none latency, as they are deployed a single unit. It is easy make one newly appointed employee about all the features and all the code base due to the presence of all the code in a single codebase.

Monolith structure is great for small to some medium software but large software is not chosen by most of the programmers. As all of the software features, service and database are present in a single code. The codebase will look like a garbage with all the different codes flooding the eyes of the coder. If the application lots of data is associated with each user record, then the company will run into performance issues with large datasets from too many database queries being executed too quickly. If anyone of the services breaks down the whole software will be affected by it and the software will not properly. And for updating a single service or adding new feature will need to redeploy the whole application. All services use the same resources such as memory and storage. And there is no way to restrict the interaction between any service, which is a very big no. As someone using this vulnerability to get into the database of the services.

**Serverless:**

Severless software application, it is deployed at others server. We know that managing server’s hardware, regularly doing security updates, security updates and configuring backup of it takes a lot of time. So deploying them in cloud will offload many duties from the company. And they will be able to be concentrate about the software product. In serverless all the services are decentralized and are not deploy as a single unit. And some of the companies now are using hybrid architecture where some of the servers are maintained by themselves and some of the services are hosted by could. As one company not like to use cloud for their confidential information to be hosted and stored at others servers. The cost operation will be low as compared to having our own servers as they cost a lot. One of the benefits of serverless model is that functions can scale automatically based on demand, ensuring optimal resource utilization

There are many problems in cloud services. One think is that any problem occurs in the server that your application is hosted. You can’t physically fix it yourself you have wait for services provider to fixed and you don’t know when they will. And if the vendor shut downs it operation, then you can’t retrieve your data that are available in the servers. Your data can be at compromised if shared server if improperly setup. As many clients may be using the servers at the same time.

**Conclusion:**

As monolith architecture can be used by people or company who are in the early stages of making their software application as it more cost effective.

Serverless should use by companies to deploy their services that are not that confidential and use deploy the services and databases on their own serve due to security reason