System Architectures

Monolithic architectures

In a monolithic architecture, the whole project is developed in one piece. In a monolith architecture all the components are interconnected and tightly coupled. Changing any aspect of the system requires an update to the whole system.

Distributed architectures

A distrusted architecture as system components distributed across various locations. Distributed architectures are more modular and the components are less tightly coupled.

Benefits of distributed architectures

Distributed architectures can be easier to develop with greater parallelization. Different teams can work of different components of the system. The modular nature allows different components to be updated independently of one another. Distributed systems can be more fault tolerant. If one component fails, the other may be able to continue.

Benefits of monolithic architectures

Monolithic systems can be easier to develop and debug. Less code will be required to integrate the systems. This code can be error prone. Running the system as one process allows easy debugging of the whole system. Monolithic systems can be more performant with less time spent making calls between the interfaces of different components.

In short distributed systems have advantages in scalability, reliability upgradability.

The increased number of interfaces can decrease performance however and increase the attack surface a hence vulnerability of the system however.

Microservices, Service orientated architecture and Common Object Request Broker Architecture

CORBA, Microservices and SOA's aim to break systems into smaller pieces. MS's and SOA's differ from CORBs such that CORBA's break systems into objects, and objects are

passed around whereas in a MS or SOA system the pieces are services where messages are passed about.

Both MS's and SOA approaches aim to create systems out of small components that are loosely coupled.

The SOA design became popular in the 1990s and leverage web interfaces and xml to tie applications together of the net. SOA's can be thought of as a response to monolithic design methodology.

Microservices are considered an evolution of SOA's (https://www.talend.com/blog/2017/06/02/microservices-lean-thinking-approach/). Both approaches break up applications into smaller components that run in the cloud. SOA's typically have larger services compared to the microservices.

Communications for microservices typically take place with 'light weight 'protocols such as HTTP. It is important that fast protocols are used otherwise the interconnected calls would slow the application.

Virtualization

The operating system can be virtualized so that applications are unaware of what platform they are operating upon. This allows developers to easily deploy their programs on platforms that is supported. Java for instance runs on a virtual machine.

Hackers will try to 'escape' the virtual machine that a program runs on to access data that should be hidden.