

Architectural Design Justification and Evaluation

Kubernetes was selected as the container orchestration tool for its dynamic approach in managing containerized applications. Moreover, Kubernetes' adeptness in managing containerized applications in different deployment environments, such as physical machines, virtual machines, and cloud (even hybrid) environments, make it a suitable general-purpose solution. The available requirements for this DORIS-app application could, in practical terms, require the use of many microservices. Given this reality, a monolith solution is not desirable. If we choose to utilize containers to represent, possibly, hundreds of microservices used in the execution of an application, we can circumvent or mitigate the need for resource intensive scripts or self-made tools used to manage the cloud environments. Kubernetes is strategically positioned to offer certain key advantages. Kubernetes offers high availability or no downtime, scalability or high performance, and disaster recovery or backup and restoration. The third point mentioned is of material concern because it means that if an infrastructure has some problems like data loss resulting from a power outage at the server center, Kubernetes permits for the backup of data and subsequent restoration to the latest state. Kubernetes, as a container orchestration tool, also allows for the containerized application to run against the latest state after the recovery.

When considering the design of DORIS-app itself, a layered architectural design pattern was selected for its key advantage of high ease of development and high testability. These are key advantages when we consider Agile and DevOps process management frameworks—the Developer and Operations team require an architecture that lends itself to a swift cadence of development and testing, so that adjustments and pivots can be made to the codebase and technical infrastructure in a data-driven and rapid manner. The fact that the layered architectural design pattern, in isolation, has notable downsides, such as low ease of deployment, low performance, and low scalability, is largely a null point, since the architecture's overall implementation against a Kubernetes cluster makes it quite advantageous. So, when we consider the layered design pattern used for DORIS-app in conjunction with the Kubernetes cluster, a reasonable argument could be made that the overall architecture of the system is, in fact, or akin to a Space-Based (Cloud) Pattern. If we consider the architecture from this lens, then there are more defined advantages as compared to when we evaluate the relative strengths and weaknesses of an application purely based on a Layered architectural design pattern. Namely, an architecture based on Space-Based (Cloud) Pattern offers high responsiveness to change, high ease of deployment, high performance, and high scalability.