Global Science Research Journals

Available online at www.globalscienceresearchjournals.org

Global Journal of Engineering & Architecture

Editorial 2021 Open Access

ISSN: XXXX-XXXX Vol. 2 (1). Pp.12-13
February, 2021

Application Architecture

Magyar P[1]

[1] Professor, Seaton Hall Manhattan, KS 66506. T: (785) 532-5953.
*Corresponding author. E-mail: pmagyar@k-state.edu

APPLICATION ARCHITECTURE

An application architecture describes the patterns and techniques used to design and build an application. The architecture gives you a roadmap and best practices to follow when building an application, so that you end up with a well-structured app.

Software design patterns can help you to build an application. A pattern describes a repeatable solution to a problem.Patterns can be linked together to create more generic application architectures. Instead of completely creating the architecture yourself, you can use existing design patterns, which also ensure that things will work the way they're supposed to.

As part of an application architecture, there will be both front-end and back-end services. Front-end development is concerned with the user experience of the app, while back-end development focuses on providing access to the data, services, and other existing systems that make the app work.

The architecture is a starting point or roadmap for building an application, but you'll need to make implementation choices not captured in an architecture. For example, a first step is to choose a programming language in which to write the application.

There are many programming languages used for software development. Certain languages may be used to build certain types of applications, such as Swift for mobile apps or JavaScript for front-end development. JavaScript used with HTML and CSS is currently 1 of the more popular programming languages for web application development.

Other popular programming languages include Ruby, Python, Swift, TypeScript, Java, PHP, and SQL, among others. The language used when building an application will depend on the type of application, available development resources, and the requirements.

Historically, applications were written as a single unit of code, where the components all share the same resources and memory space. This style of architecture is referred to as a monolith.

Modern application architectures are more often loosely coupled, using microservices and application programming interfaces (APIs) to connect services, which provide the foundation for cloud-native applications.

Cloud-native development is a way to speed up how you build new applications, optimize existing ones, and provide a consistent development and automated management experience across private, public, and hybrid clouds.

TYPES OF APPLICATION ARCHITECTURE

Layered or N-Tier Architecture

A layered or N-tier architecture is a traditional architecture often used to build on-premise and enterprise apps, and is frequently associated with legacy apps.In a layered architecture, there are several layers or tiers, often 3, but there can be more, that make up the application, each with their own responsibility.

Monolithic Architecture

A monolith, another architecture type associated with legacy systems, is a single application stack that contains all functionality within that 1 application. This is tightly coupled, both in the interaction between the services and how they are developed and delivered. Updating or scaling a single aspect of a monolithic application has implications for the entire application and its underlying infrastructure.

Microservices Architecture

Microservices are both an architecture and an approach to writing software. With microservices, apps are broken down into their smallest components, independent from each other. Each of these components, or processes, is a microservice.The goal of using a microservices architecture is to deliver quality software faster. You can develop multiple microservices concurrently. And because services are deployed independently, you don’t have to rebuild or redeploy the entire app when changes are made.

Event-Driven Architecture

With an event-driven system, the capture, communication, processing, and persistence of events are the core structure of the solution. This differs from a traditional request-driven model.An event is any significant occurrence or change in state for system hardware or software. The source of an event can be from internal or external inputs.Event-driven architecture enables minimal coupling, which makes it a good option for modern, distributed application architectures.