

# Comparison of Available Web-System Design Methodologies

Arya Singh, Abhyuday Gupta

*Department of CSE, RIET, Jaipur, Rajasthan, India*

*Department of CSE, RIET, Jaipur, Rajasthan, India*

Submitted: 02-01-2022

Revised: 09-01-2022

Accepted: 12-01-2022

**ABSTRACT:** This paper discusses web design and development strategies. For examples can be taken existing methods, methods and products for the development and design of web applications, such as: monolithic and multi-layer architecture, microservice architecture, serverless architecture, BaaS (Backend as a service) method. Based on the analysis of these methods, a conclusion will be established and an understanding of the available web design approach in the current market.

The purpose of this research paper is to analyze major web application frameworks and to design and implement web-based web application applications for multimedia applications. web-based frameworks, which are described and compared on the basis of relevant terms and analysis.

The concrete part of the paper covers the selection of the appropriate framework for the use of both applications and describes their design, development process and timely implementation of projects. Despite this, these frameworks provide support for the development of one-page web applications. and enable blank HTML, JavaScript. Apart from that we are very focused on popular frameworks such as ReactJS, AngularJS and Vue JS the three most commonly used javascript frameworks in today's web development. . Both of these aforementioned frameworks are used to build interactive one-page applications (SPA) .This research paper will help to understand AngularJS, ReactJS and Vue JS and many other frameworks.

**KEYWORDS:** Baas, web-systems, development, architecture, microservices.

## I. INTRODUCTION

Nowadays, the problem of proper and

effective design is more important than the problem of successful development or support of any web system . This is largely due to the global spread of web applications or applications that operate in a certain way in sharing web applications. The burden on web systems is growing daily, in a ever-changing world, rapid changes in customer specifications are becoming the norm. With the spread of the internet around the world and access to any information around the world, skills that can be learned without graduating from specialized educational institutions are already flourishing. One such exception is software development. This includes the fullness of the market by many different experts. Thus, finding high-quality engineers for engineers is not a big deal. Using a variety of patterns, libraries and frameworks also contributes to development. However, the initial design of any program is not an easy task, the insight that comes with knowledge and cannot be taught by university classes or video lessons. This is why the function is so important to any system, mainly because changing the system settings during its development can be expensive or not possible at all. In this work, the popular methods used in web design and development will be reviewed, their main benefits and problems will be categorized, and usage cases will be explained. Based on them, by studying the development of design ideas, a conclusion will be made and the reader will be able to gain a similar understanding. Although many design architectures exist, not all are equally selective that is because some fit a particular context over others.

Therefore the analogy itself cannot be established but a line may be drawn about the popularity of buildings.

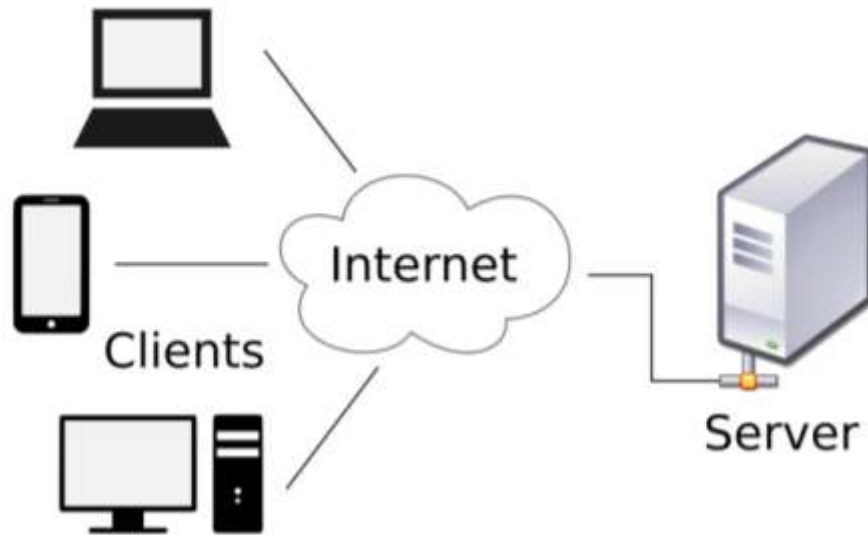


Fig 1: View of client-server architecture

## II. MONOLITHIC ARCHITECTURE

Over the past few years engineers have successfully applied Monolithic Architecture. A Monolithic application is a method by which different components (such as authorization, business logic notification module, etc.) are integrated into a single system from a single

location.

The picture above shows an example of a Monolithic app that gives an idea for an online business. Despite using many components of the whole software, the app is installed as a standalone application.

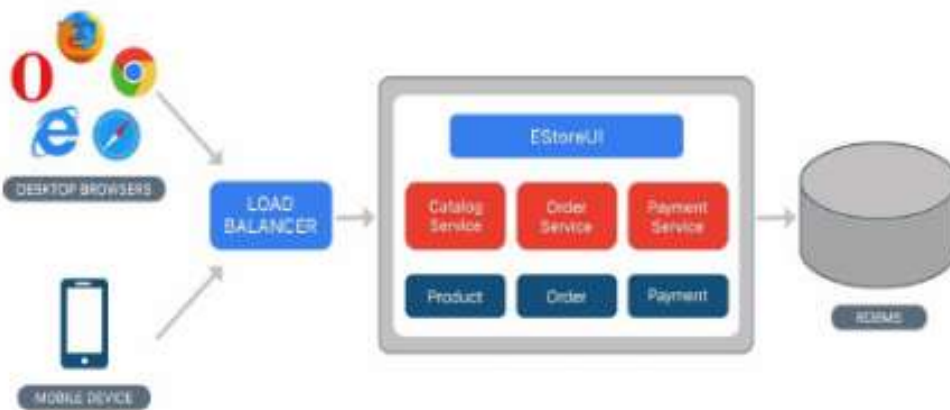


Fig 2: Monolithic flow on the example of E-commerce application

### III. MICROSERVICE ARCHITECTURE

Today, program planners find many new opportunities to create a program that will bring the development process to the next level. One of the most popular new solutions, about Google Trends, is Microservice Architecture. It is the architectural style that builds the app as a collection of services. Each small service should have provided one part of the business idea.



Fig. 2. Microservice flow on the example of E-commerce application

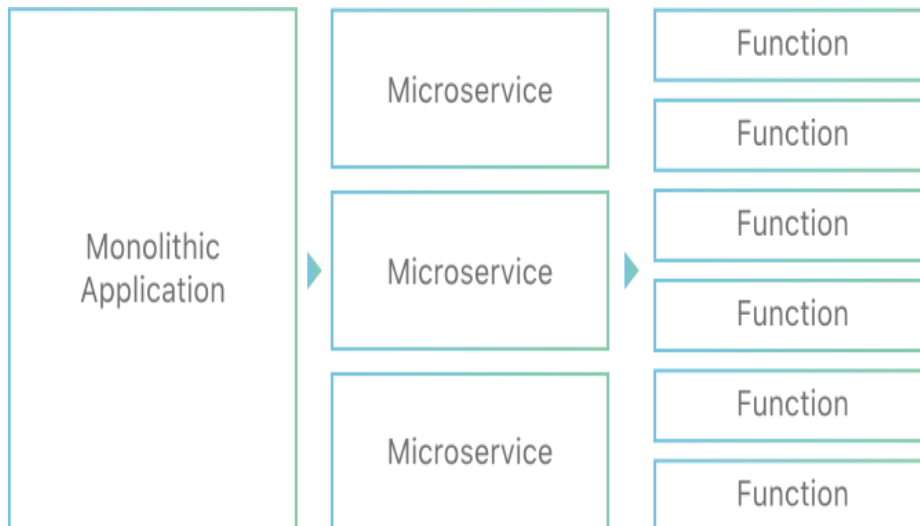
As you can see, there are four basic Microservices, which provide a business concept and for one additional service expose the functionality as it can be a single application.

### IV. SERVERLESS ARCHITECTURE

Mike Roberts believes Serverless is a combination of Back-end-as-a-Service (BaaS) and Function-as-a-Service (FaaS). The main form of server representation is FaaS, so non-server computing is considered a "service as a service

(FaaS)" or a "functionally operated event" [3]. It is based on a code written by an accurate resource allocation engineer, and platform resources are started when a predefined event is launched.

Function-as-a-Service (FaaS) is a server-free way to apply code-cutting pieces. FaaS allows developers to write and update a piece of code on a fly, which can be used in response to an event, such as a user clicking on an object in a web application. This makes it easy to measure code and is a cost-effective way to use micro.



Developers should build relationships with a non-server provider to enable FaaS functionality in the web application. Since the integration of FaaS means that a specific application code will be delivered from the edge, the availability and location distribution of edge servers is an important consideration. An Italian user accessing a site that relies on the FaaS edge code provided in Brazil's largest data center will experience a type of delay that leads to higher jump rates. Cloudflare Workers is a FaaS solution that takes advantage of Cloudflare's global network of data centers in more than 250 cities, making it a popular choice.

The most obvious advantage of no server is that there is no need to maintain a server, no server can focus on application development, no need to worry about infrastructure services.

(1) Reduce operating costs

A useless architecture is actually an external solution and the infrastructure is not wasted. It only needs to pay the required amount of calculations depending on the size and nature of the traffic, which can significantly save on operating costs, especially the pre-existing and flexible application load requirements with different variables.

(2) Unlimited measurement

The highest ratings are not new to cloud services, but the construction of a no-service takes it to a new level. Do not use the server, there is no need to explicitly add and remove conditions from the server list, and allow resellers to expand the application. As the cloud computing provider makes extensions on the basis of each application, there is no need to consider the issue of how many relevant applications can be processed before the memory goes down.

(3) Problem with separation

Serverless divides the application into separate sections into separate sections so that each component solves one problem.

(4) The process of isolation

In a server-free environment, each Lambda function is completely isolated. If one feature is turned off, it does not affect other features, it does not cause the server to crash.

## V. BACKEND AS A SERVICE(BAAS)

Backend-as-a-Service (BaaS) is a cloud service model in which developers extract all the features in the back of a web site or mobile application to record and maintain only the front end. BaaS vendors provide pre-programmed software for server tasks, such as user authentication, site management, remote updates, and application (for mobile applications), and cloud storage and hosting.

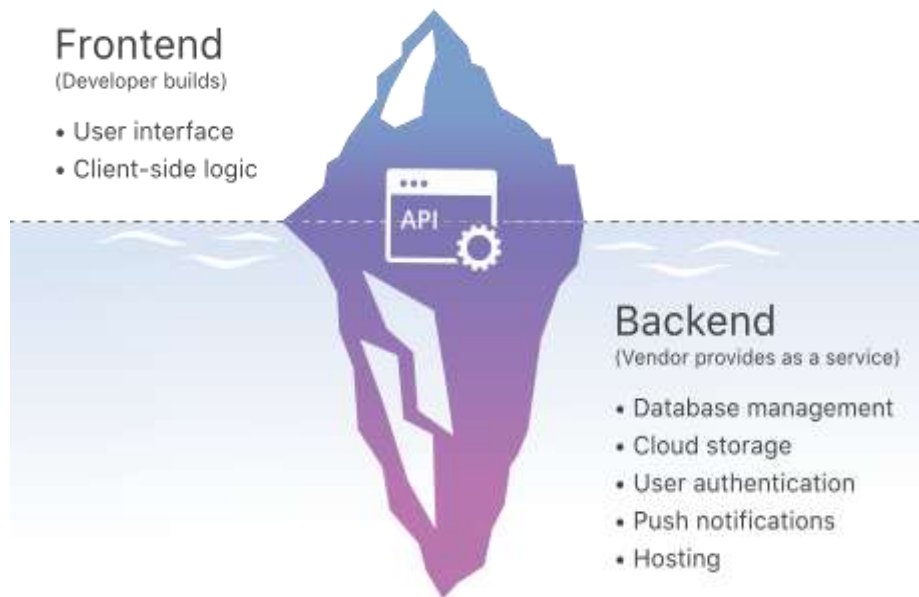
BaaS enables developers to focus on pre-application code. By using the APIs (which is another application application system) and SDKs (which are building software kits) provided by a BaaS vendor, they are able to integrate all the backend functionality they need, without having to build a backend. themselves. And they do not need to carry servers, virtual machines, or containers to keep the app running. As a result, they can build and launch mobile applications and web applications (including one-page applications) very quickly.

Backend-as-a-Service (BaaS) is a cloud service model in which developers extract all the features in the back of a web site or mobile application to record and maintain only the front end. BaaS vendors provide pre-programmed software for server tasks, such as user authentication, site management, remote updates, and application (for mobile applications), and cloud storage and hosting.

Consider building an app without using a BaaS provider as directing a movie. The film director is responsible for overseeing or supervising the crew of the camera, lighting, building sets, wardrobe, acting character, and production program, in addition to recording and directing scenes from the movie. Now imagine if there was a service that took care of all the activities behind the stadium so that all the director had to do was direct and shoot the venue. That's the BaaS concept: The vendor takes care of the 'lights' and the 'camera' (or, server-side functionality) so that the director (engineer) can focus on the 'action' - that ultimately the user sees and information.

BaaS enables developers to focus on pre-application code. By using the APIs (which is another application application system) and SDKs (which are building software kits) provided by a BaaS vendor, they are able to integrate all the backend functionality they need, without having to build a backend. themselves. And they do not need to carry servers, virtual machines, or containers to keep the app running. As a result, they can build and launch mobile applications and web applications (including one-page applications) very quickly.

There is a disconnect between BaaS and serverless computing, because in both cases the developer has to write his own application code only and not think about backend. In addition, many BaaS providers also offer server-free computer services. However, there is a significant difference in performance between applications built using BaaS and a real server without a server.



## VI. CONCLUSION

All tested properties have certain advantages and disadvantages. It all depends on the problem that needs to be solved. The load test performed showed that the Microservice configuration works best if the application has to handle a large number of applications. It has many advantages that allow you to build high quality software, easy to measure, reliable and ultimately easy to maintain. Although it has many advantages, the monolithic structure is not bad. It works very well during low load and is easy to upgrade. There are not many problems with integration, connection and configuration. The choice of suitable properties should be defined for business purposes so that the investor can find a product that will meet his expectations. As for non-server as well.

## REFERENCES

- [1]. <http://ceur-ws.org/Vol-2604/paper59.pdf>
- [2]. [https://www.researchgate.net/publication/341956559\\_The\\_Comparison\\_of\\_Microservice\\_and\\_Monolithic\\_Architecture](https://www.researchgate.net/publication/341956559_The_Comparison_of_Microservice_and_Monolithic_Architecture)
- [3]. [https://www.researchgate.net/publication/339680821\\_Overview\\_Of\\_Serverless\\_Architecture\\_Research](https://www.researchgate.net/publication/339680821_Overview_Of_Serverless_Architecture_Research)
- [4]. [https://www.researchgate.net/publication/341956559\\_The\\_Comparison\\_of\\_Microservice\\_and\\_Monolithic\\_Architecture](https://www.researchgate.net/publication/341956559_The_Comparison_of_Microservice_and_Monolithic_Architecture)
- [5]. <https://www.cloudflare.com/en-in/learning/serverless/glossary/backend-as-a-service-baas/>