Just Your KidieZ

Applied Research Document

Date	:	05/12/2023
Version	:	3
State	:	Released
Author	:	Aya Shikh Suliman

Version history

Version	Date	Author(s)	Changes	State
0.1	3/23/2023	Aya Shikh Suliman	Main question and sub questions	Released
0.2	3/24/2023	Aya Shikh Suliman	Questions changed after Felipe’s feedback	Released
1	3/24/2023	Aya Shikh Suliman	Questions changed after Marcio’s feedback	Released
1.1	4/12/2023	Aya Shikh Suliman	Problem statement, chosen methods, and results	Released
1.2	4/14/2023	Aya Shikh Suliman	Results, conclusion, and recommendation	Released
2	4/14/2023	Aya Shikh Suliman	References	Released
3	5/12/2023	Aya Shikh Suliman	Questions	Released

Table of contents:

1. Problem statement:

There are numerous architectural patterns and approaches available to develop web applications, each with its own advantages and disadvantages. The challenge lies in identifying the most suitable architecture for a particular web application based on its functional and non-functional requirements, performance considerations, scalability, maintainability, security, and other factors. Thus, there is a need to explore and evaluate different architectural options and select the one that best aligns with the project goals and objectives.

2. Questions:

2.1 Main question:

2.1.1 How to choose an architecture for web applications?

2.2 Sub questions:

2.2.1 What are the common architectures?

2.2.2 What are the advantages of each architecture?

2.2.3 What are the disadvantages of each architecture?

2.2.4 What aspects should a developer put into consideration when choosing an architecture?

3. Chosen methods:

For this research, the DOT framework is going to be used.

The Design-Oriented Triangulation (DOT) framework is a mixed-methods research approach that combines qualitative and quantitative research methods to provide a comprehensive understanding of a research problem (The DOT framework).

For each sub question one or more research method is going to be used.

3.1 What are the common architectures?

A suitable research method for answering the question above would be a **literature** review.

It contains a systematic search and analysis of existing literature on a particular topic. In this case, the researcher would conduct a comprehensive search of academic articles, conference proceedings, technical reports, and other relevant sources to identify the most commonly used architectures in web development (Research Guides: Research Methods: Literature Reviews, n.d.).

The researcher would review and analyze the literature to identify the key characteristics, advantages, and limitations of each architecture. The findings from the literature review could be summarized and presented in a clear and concise manner, providing an overview of the common architectures in web development.

3.2 What are the advantages and disadvantages of each architecture?

A **comparative** analysis is a suitable research method for this question.

It involves comparing two or more variables or cases to identify similarities, differences, strengths, and weaknesses. In this case, the researcher would compare different architectures in web development to identify their advantages and disadvantages.

The comparative analysis could be conducted through a variety of research methods, such as case studies, surveys, or interviews with experts and stakeholders. For example, the researcher could select a set of case studies that use different architectures and compare their performance, scalability, maintainability, security, and other relevant factors (Indeed Editorial Team, 2022).

3.3 What aspects should a developer put into consideration when choosing an architecture?

An **expert** review or an **expert** survey is a good research method for this question.

An expert review pertains gathering opinions and insights from a panel of experts in the field of web development. The researcher could identify a group of experts who have significant experience in developing web applications and selecting architectures, and conduct a review of their opinions and insights on the factors that developers should consider when choosing an architecture (Expert Review in the usability.de Glossary – usability.de, n.d.).

An expert survey is similar to an expert review, but involves collecting data through a structured questionnaire. The questionnaire could be designed to gather information on the various factors that developers should consider when selecting an architecture, such as performance, scalability, maintainability, security, and other relevant factors. The survey could be distributed to a sample of experts in the field of web development, and the results could be analyzed to identify the most important factors to consider when selecting an architecture (M, 2016).

Both expert review and expert survey methods are valuable for identifying the key factors that developers should consider when selecting an architecture. They provide insights from experienced professionals who have significant knowledge and expertise in the field of web development. These methods can also be used to gather data quickly and efficiently, making them ideal for researchers who have limited time or resources to conduct a larger study.

4. Results

The result can be summarized as follows:

1. **Model-View-Controller (MVC) architecture:** MVC is a widely used architecture that separates the application into three interconnected components: Model, View, and Controller. The Model represents the data and business logic, the View is responsible for displaying the data to the user, and the Controller handles user input and interacts with the Model and View (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (Soni, 2021):

   Advantages:
   - Separation of concerns makes it easier to maintain and test code.
   - Code is more organized and easier to understand.
   - Supports multiple views for the same data.
   - Enables developers to work on different components independently.

   Disadvantages:
   - Can be complex to implement for smaller applications.
   - Can result in increased code duplication if not designed properly.
   - Changes to one component may require changes to other components.

2. **Microservices architecture:** Microservices architecture is an approach to software development that involves breaking down an application into smaller, independent services that can be developed and deployed separately. Each service is responsible for a specific functionality and communicates with other services through APIs (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (*Advantages and Disadvantages of Microservices - Javatpoint*, n.d.)**:**

   Advantages:
   - Allows for more efficient scaling of individual services.
   - Supports independent development and deployment of services.
   - Enables better fault isolation and easier fault recovery.
   - Can support multiple programming languages and technologies.

   Disadvantages:
   - Increased complexity due to the need for service discovery and management.
   - Can result in increased network latency due to communication between services.
   - Difficult to maintain consistency across services.
   - Can result in increased costs due to the need for additional infrastructure.

3. **RESTful API architecture:** REST (Representational State Transfer) is a style of software architecture that is used to create scalable web services. RESTful APIs are designed to be lightweight, scalable, and flexible, and they use standard HTTP methods such as GET, POST, PUT, and DELETE (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (Vathslya, 2021):

   Advantages:
   - Lightweight and flexible, suitable for a wide range of applications.
   - Supports multiple data formats, including JSON and XML.
   - Easy to cache and scale.
   - Well-suited for building web services and APIs.

   Disadvantages:
   - Limited functionality compared to other architectures.
   - Can result in increased complexity due to the need for managing URLs and resources.
   - Security risks if not designed properly.

4. **Single-page Application (SPA) architecture:** SPA architecture is a modern approach to web development that involves creating a single web page that dynamically updates as the user interacts with it. SPA architecture is often used in conjunction with frameworks such as Angular, React, and Vue (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (iTechArt Group, Custom Software Development, 2022):

   Advantages:
   - Faster and more responsive than traditional web applications.
   - Offers a more seamless user experience.
   - Can be developed using a wide range of frameworks and libraries.
   - Can be used to build mobile and desktop applications.

   Disadvantages:
   - Limited SEO and accessibility due to the reliance on JavaScript.
   - Increased complexity due to the need for managing state.
   - May require more testing due to the increased client-side code.

5. **Serverless architecture:** Serverless architecture is a cloud computing model that allows developers to build and run applications without having to manage infrastructure. In serverless architecture, the cloud provider takes care of the infrastructure, and developers can focus on writing and deploying code (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (Serverless Computing: Uses, Advantages, and Disadvantages | Okta, n.d.):

   Advantages:
   - Reduced infrastructure costs and management overhead.
   - Automatically scales based on demand.
   - Supports a wide range of programming languages and technologies.
   - Enables faster development and deployment.

   Disadvantages:
   - Limited control over the underlying infrastructure.
   - Higher latency due to the need to spin up resources.
   - Limited support for long-running tasks.
   - Higher costs for applications with constant traffic.

6. **Event-driven architecture:** Event-driven architecture is an approach to software development that focuses on the flow of events or messages between components. In event-driven architecture, components are loosely coupled and communicate through events, which are triggered by specific actions or changes in the system (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (Kawle, n.d.):

   Advantages:

   - Scalable and flexible.
   - Loosely coupled, which enables independent development and deployment of components.
   - Supports a wide range of use cases, including IoT and real-time applications.
   - Enables better fault tolerance and recovery.

   Disadvantages:

   - Can be difficult to design and implement.
   - Increased complexity due to the need for managing events and event sources.
   - Can result in increased costs due to the need for additional infrastructure.
   - May require more testing due to the asynchronous nature of events.

7. **Progressive Web Application (PWA) architecture:** PWA architecture is an approach to web development that aims to create web applications that have the same look and feel as native mobile apps. PWAs are designed to be fast, responsive, and reliable, and they use features such as service workers and web app manifests to provide a native app-like experience (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (Kvartalnyi & Kvartalnyi, 2023):

   Advantages:

   - Faster and more responsive than traditional web applications.
   - Offers a more seamless user experience.
   - Can be installed on a user's device for offline access.
   - Can be developed using a wide range of frameworks and libraries.

   Disadvantages:

   - Limited support for older browsers and operating systems.
   - May require more development effort than traditional web applications.
   - Limited access to device features compared to native applications.
   - Increased complexity due to the need for managing state and service workers.

8. **Three-tier architecture:** Three-tier architecture is a software design pattern that separates an application into three layers: presentation, application, and data. The presentation layer is responsible for displaying data to the user, the application layer contains the business logic, and the data layer manages the application data (Ardalis, 2023).

   These are the advantages and disadvantages of using this architecture (GeeksforGeeks, 2021):

   Advantages:

   - Separation of concerns makes it easier to maintain and test code.
   - Supports multiple presentation layers, including web and mobile applications.
   - Can be scaled horizontally by adding more application or database servers.
   - Offers better security and fault tolerance.

   Disadvantages:

   - Increased complexity due to the need for managing.

The result of the question "What aspects should a developer put into consideration when choosing an architecture?" using expert review or survey research method could include the following aspects:

a. **Performance**: The architecture should be able to handle the expected load and scale as the application grows.
b. **Security**: The architecture should ensure the safety of data and prevent unauthorized access.
c. **Flexibility**: The architecture should allow for changes and updates without causing disruption to the application.
d. **Cost**: The architecture should be cost-effective and provide a good return on investment.
e. **Complexity**: The architecture should not be overly complex, which can result in difficulty in development and maintenance.
f. **Availability**: The architecture should ensure high availability and uptime for the application.
g. **Integration**: The architecture should allow for easy integration with other systems and tools.
h. **Development speed**: The architecture should enable quick development and deployment of the application.

5. Conclusion

We can conclude that:

1. There are several common architectures in web development, including client-server, monolithic, microservices, and serverless architectures.
2. Each architecture has its advantages and disadvantages, such as simplicity, flexibility, complexity, and scalability.
3. When choosing an architecture, developers should consider several aspects, such as performance, security, flexibility, cost, complexity, availability, integration, and development speed.
4. The choice of architecture should depend on the specific needs and requirements of the application, the development team's expertise, and the project's budget and timeline.
5. To make an informed decision about the architecture, developers should thoroughly research and compare each architecture's advantages and disadvantages and consider the project's unique requirements and constraints.

In **summary**, choosing an architecture for web applications requires careful consideration of several factors, and developers must make an informed decision based on their research and understanding of each architecture's benefits and drawbacks.

6. Recommendation

Based on the research conducted, the following recommendations can be made:

1. **Evaluate your application's requirements**: Before choosing an architecture, it is important to evaluate your application's requirements, such as performance, scalability, security, and integration needs.
2. **Consider your team's expertise**: Consider your development team's skills and expertise when choosing an architecture. Ensure that the chosen architecture aligns with your team's skills and capabilities.
3. **Analyze the available architectures**: Thoroughly research and analyze the available architectures, such as client-server, monolithic, microservices, and serverless architectures. Consider each architecture's advantages and disadvantages and evaluate how well it aligns with your application's requirements.
4. **Consider cost and timeline**: Consider the project's budget and timeline when choosing an architecture. Some architectures may require more resources and time to develop and maintain than others.
5. **Prototype and test**: Before finalizing an architecture, it is recommended to prototype and test it to ensure that it meets your application's requirements and is scalable, secure, and maintainable.
6. **Stay informed**: Stay up to date with the latest trends and best practices in web development architecture. Attend webinars, read articles, and follow industry experts to stay informed and continuously improve your development practices.

By following these recommendations, developers can make an informed decision when choosing an architecture for their web applications that aligns with their requirements, team's expertise, budget, and timeline.

7. References

“The DOT Framework.” *The DOT Framework - ICT Research Methods*,
https://ictresearchmethods.nl/The_DOT_Framework.
Date: 4/12/2023

Research Guides: Research Methods: Literature Reviews. (n.d.).
https://guides.auraria.edu/researchmethods/literaturereviews#:~:text=Definition,%2C%20dissertation%2C%20or%20grant%20proposal.
Date: 4/12/2023

Indeed Editorial Team. (2022, June 24). *What Is Comparative Analysis and How Is It Used?* Indeed Career Guide.
https://www.indeed.com/career-advice/career-development/comparative-analysis
Date: 4/12/2023

M, M. (2016, July 9). *Survey of Expert’s Opinions*. Business Jargons.
https://businessjargons.com/surveyofexpertsopinions.html#:~:text=Definition%3A%20In%20Survey%20of%20Expert's,their%20opinions%20on%20sales%20trend.
Date: 4/12/2023

Ardalis. (2023, March 7). *Common web application architectures*. Microsoft Learn.
https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures
Date: 4/12/2023

Soni, N. (2021, May 17). *Benefits and Drawbacks of MVC Architecture*. S2 Labs by Shrey Sharma.
https://shreysharma.com/benefits-and-drawbacks-of-mvc-architecture/
Date: 4/12/2023

Advantages and Disadvantages of Microservices - javatpoint. (n.d.).
www.javatpoint.com.
https://www.javatpoint.com/advantges-and-disadvantages-of-microservices
Date: 4/12/2023

Vathslya. (2021, November 23). *Advantages and Disadvantages of REST API*. Krify - Web and Mobile App Design & Development Company in India & UK. https://krify.co/advantages-and-disadvantages-of-rest-api/
Date: 4/12/2023

iTechArt Group, Custom Software Development. (2022, July 18). *The pros and cons of single page applications (SPAs)*. iTechArt.
https://www.itechart.com/blog/pros-cons-of-single-page-applications/
Date: 4/14/2023

Serverless Computing: Uses, Advantages, and Disadvantages | *Okta*. (n.d.). Okta, Inc.
https://www.okta.com/identity-101/serverless-computing/
Date: 4/14/2023

Kawle, S. (n.d.). *Advantages And Disadvantages Of Event Driven Architecture*.
https://www.neebal.com/blog/advantages-and-disadvantages-of-event-driven-architecture

Kvartalnyi, N., & Kvartalnyi, N. (2023, January 16). Benefits of Progressive Web Apps (PWA) – Advantages and Disadvantages | Inoxoft. *Inoxoft* |. https://inoxoft.com/blog/benefits-of-progressive-web-apps-pwa-advantages-and-disadvantages/
Date: 4/14/2023

GeeksforGeeks. (2021). Advantages and Disadvantages of Three Tier Architecture in DBMS. *GeeksforGeeks*. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-three-tier-architecture-in-dbms/
Date: 4/14/2023