International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Architectural Design and Scalability: A Hybrid Approach with Kotlin, Mongodb And PostgreSQL

Hareeshkumar.rapolu
Nikhitha Gundeti
Navya Punna
Hareeshkumar.rapolu@gmail.com

Abstract

This research paper has been based on the topic: Architecture Design and Scalability - A hybrid approach with Kotlin, MongoDB, and PostgreSQL, for a modern web service. This research paper has discussed the importance of architectural design and scalability, and how it builds up a system and helps it stay reliable and is able to grow. This maintains its costs and the speed of the system. There has been an analysis of three very important components or approaches: Kotlin, MongoDB and PostgreSQL that are often used in developing software. They act as some of the best tools for resolving any commonly occurring challenges in the development of modern applications.

I. INTRODUCTION

Architectural design and scalability are two concepts that are interconnected in engineering software. The choices in architecture tend to determine the scale of one's system without a huge drop in the system's performance. Scalability is when a system is able to handle or manage the growing amount of work in a graceful way. The choices of design that are made in architecture are very important, especially when it comes to loose coupling and horizontal scaling. There are various design techniques, like caching and load balancing, that could be used so that a scalable architecture performs well enough.

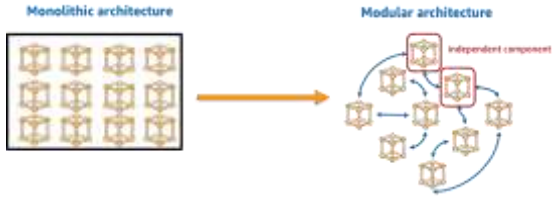

Figure 1: Archtectural Design and Scalability

II. ARCHITECTURAL DESIGN AND SCALABILITY

Architectural Design - This acts as the process of explaining a structure, its elements and interfaces of a system, analysing its behaviour as well. The design helps systems to fix bugs, if any, in addition to new features and understanding the code from time to time. It allows the system to be reliable and maintain its stability over a period of time. The system can adapt to the requirements of businesses that do not stay the same all the time without additional costs[1]. The systems can perform in an efficient way to process all the data and make responses to requests made by the users. The design has to be made in such a way that it is able to accommodate the workload that increases from time to time. Designing architecture is not just about allowing the systems to work, but to ensure it is stable for a very long time. Some qualities that may help individuals to make design decisions are modularity, high cohesion, loose coupling, and fault tolerance and resilience. Some components of design are monolith, microservices and an architecture that is event-driven.

Architectural Scalability - The Scalability of architecture is the capabilities and abilities of systems that tend to handle an amount of work that is growing gracefully. This could be achieved either by adding more resources and upgrading the ones that are already there, scaling out and scaling up, respectively. To help the system grow, horizontal scaling is important. This tends to allow the system to manage large growing spikes[2]. To control the costs of the system, modular design could be used. This would allow one to scale the elements that are under great load. This allows one to save costs by reducing the provision of resources for the overall system. To enhance the experiences for the users, caching and load balancing is necessary. This makes sure that the performances are fast and stay consistent to help with user experience. This may reduce the loading times that are slow and prevent users from leaving. For future proofing, loose coupling is important. This allows the system to upgrade or even change important components without actually changing or rebuilding the entire application. This and the design need to be managed from the beginning[3]. This could not be changed later on.

Figure 2: Kotlin

III. KOTLIN APPROACH

- This approach is more of a modern take on developing software that is focused more on the productivity of the developers, code of safety, and efficient interoperability in the Java ecosystem that already exists.
- This could be seen as just a language, but it is not. It is a choice of design and adds digis on writing, fewer boilerplate codes, and building up more applications that are resilient[4]. It has many technical features, and they tend to resolve many challenges as well.
- It helps to decrease the quantity of code that is required for tasks by using features like classes of days and lambda. Developers will not have to spend much time typing out their code repetitively and will be able to focus more on business logic, leading to cycles of development. This is known to be Kotlin's most important design.
- It acts as a difference between the variables which can and cannot hold a null variable over a period of time. If there are any infamous NPEs, they are effectively eliminated. This leads to a lot of crashes in Java. If they are prevented, then the applications are made more stable and reliable.

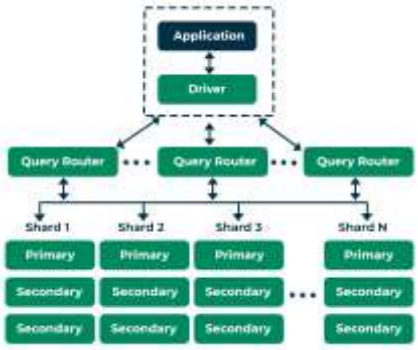

Figure 3: Mongo DB

IV. MONGODB APPROACH

- This is an approach used to manage data that does not want to accept the table-based and rigid structure of older types of databases. This model is highly agile and has horizontal scalability. It is known to be easy for developers to use.
- The two most important features of this approach would be the document model and schema flexibility[5]. The document model does not store data in tables but stores data in JSON-like documents. Each of these documents is filled with data structures, which may include sub-documents and arrays as well. It has been designed in a way that data that is accessed together should also be stored together.
- In this approach, there are no such restrictions, but it is quite flexible. Documents that are in the same collection tend to have different areas, types of data and structures as well.

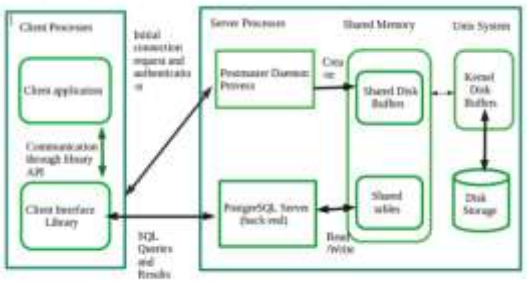

Figure 4: PostgreSQL

V. POSTGRESQL

- This approach is also used for the management of data that is focused more on robustness, advanced features, and adherence to different standards (SQL). This is based on the integrity of data and complexity, as well as the manipulation of data.
- PostgreSQL is also known as a powerful RDBMS or Relational database management System. Unlike the MongoDB Approach, PostgreSQL is quite strict when it comes to organising data into tables with columns that are defined and effective techniques. Here, the data needs to be organised in a way that makes the data reliable and accurate[6]. This acts as the foundation for most financial systems and inventory.
- It also follows the SQL standards while also staying highly extensible. This ensures that the system is stable while developers are also allowed to add on whenever deemed necessary. Custom data types, functions, and add on's are extended.

- It is important to follow the PostgreSQL approach as it promotes data integrity, reliability and complexity.

VI. CONCLUSION

The analysis of Architectural Design and Scalability with a choice of Kotlin, MongoDB and PostgreSQL has helped provide a clear understanding of modern software development. Architectural design only acts as a plan that needs to be followed. Most efficient and effective systems are created with Kotlin, MongoDB and PostgreSQL approaches and other tools are effective for design and scalability[7]. This makes the structure of the architecture quite resilient and well structured. This refers to the foundation of complexity and high performance, which builds up various digital products and systems as well.

Abbreviation

SQL: Structured Query Language
DB: Database
JSON: JavaScript Object Notation
NPE: Null Pointer Exception

ACKNOWLEDGEMENT

I would like to express my gratitude towards my professor for providing me with all the necessary support that has helped me to achieve the goals of the research. I would also like to thank my classmates and my peer members who have constantly supported me in gaining better understanding of the areas of the research.

REFERENCES

[1] U. K. Rapolu, "Optimizing Cloud Costs Through Serverless Computing in Google Cloud Platform," *Google.com*, 2024. https://scholar.google.com/citations?view_op=view_citation&hl=en&user=EQz5sccAAAAJ&citation_for_view=EQz5sccAAAAJ:Y0pCki6q_DkC (accessed Oct. 02, 2025).
[2] Upesh Kumar Rapolu, "Developing Custom Software Solutions for Compliance in Multi-Cloud Environments," *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 01, pp. 1–7, Jan. 2024, doi: https://doi.org/10.55041/ijsrem28015.
[3] A. Makris, K. Tserpes, G. Spiliopoulos, D. Zissis, and D. Anagnostopoulos, "MongoDB Vs PostgreSQL: A comparative study on performance aspects," *GeoInformatica*, vol. 25, Jun. 2020, doi: https://doi.org/10.1007/s10707-020-00407-w.
[4] Mukesh Reddy Dhanagari, "Scaling with MongoDB: Solutions for Handling Big Data in Real-Time," *Journal of Computer Science and Technology Studies*, vol. 6, no. 5, pp. 246–264, Dec. 2024, doi: https://doi.org/10.32996/jcsts.2024.6.5.20.
[5] C. Ashok Jahagirdar, "The Advantages of Using Multi - Dimensional Databases: A Comprehensive Analysis," *International Journal of Science and Research (IJSR)*, vol. 14, no. 1, pp. 1149–1154, Jan. 2025, doi: https://doi.org/10.21275/sr25126111728.
[6] D. Iovescu and C. Tudose, "Real-Time Document Collaboration—System Architecture and Design," *Applied Sciences*, vol. 14, no. 18, p. 8356, Sep. 2024, doi: https://doi.org/10.3390/app14188356.
[7] Marko Aleksendrić *et al.*, *Mastering MongoDB 7.0*. Packt Publishing Ltd, 2024.