International Journal For Technological Research In Engineering
Volume 10, Issue 8, April-2023 *ISSN (Online): 2347 - 4718*

DESIGN PATTERN FOR MULTITENANCY IN MICROSERVICES ARCHITECTURE

[1]Manoj Kumar Thapliyal
[1]Student
Department of Computer Science & Engineering
SAMALKHA GROUP OF INSTITUTION

Abstract: Multitenancy is an architectural practice to store and manage data for multiple tenancy, as single tenants per DB, single DB for multiple tenants or multiple DBs for every tenant. Database Sharding is process of storing a large database across multiple machines. This paper explains research on a Design Pattern (ShardAdapter Factory) for implementation of Sharding with in Microservices. Microsoft Azure SaaS SQL server Visual studio with C# has been used to research this Pattern.

Keywords- ShardAdapter Factory, Database Sharding, Elastic Pools, ShardLet, ShardMap Manager, Elastic Library

I. INTRODUCTION

In modern architecture systems, an efficient handling of multitenancy is a challenging approach. Key goal is high availability, performance telemetry, low latency, and accuracy. The challenge become harder when the middle layer of system realize the Microservices pattern. This paper explains the design and implementation of Multi tenancy with Microservices pattern.

II. RELATED WORK

The main motive of present work is to propose an efficient, generic architecture pattern design and implementation of multi-tenancy with Microservices. The application instance can have any number of multi-tenant databases. System can choose to store multiple tenant data with a shared database or one database per tenant. Elastic pool is used to store shared multi-tenant database for cost efficiency.

Shard key used to identify the tenant in a database. Shard keys are managed in Shardlet and Shard Map maintain the mapping of Shrdlet.

Elastic library is used to resolve the appropriate mapping at runtime.

ShardAdapter is a Factory, has been invented in this research which is responsible to inject the appropriate tenant identifier in the service JWT token and pass it to Elastic library.

III. TECHNOLOGIES USED

Azure SaaS Database –This research has been done in Microsoftf Azure cloud, SaaS SQL server database.
Microsoft Visual Studio- Microservices has been created using MS Visual Studio.
Patterns- Database Sharding, ShardLet, Elastic Pool, Microservices.

IV. IDENTIFY, RESEARCH AND COLLECT IDEA

There are several steps we can take to identify, research, and collect ideas for inventing a design pattern for Multitenancy Pattern with in Microservices, using Database Sharding, Elastic Pool, ShardAdapter Factory.

Identify the problem or challenge I am trying to solve:
Clear problem definition, this helped me to focus my research and ensure the ideas are relevant and meaningful for this research.

Research existing approaches and techniques: Once I defined the problem, I started researching existing solutions available so far. Source was academic papers and articles, personal industry experience. reviewing existing 3rd party software, architectures, and searching online for relevant resources and examples.

Collect ideas and potential solutions: As I research existing approaches and techniques, make a list of potential ideas and solutions that we think might be relevant and useful for our project. This can include specific algorithms or techniques, as well as potential tools and frameworks that we might use to implement our solution.

Evaluate and prioritize the ideas: among all possible and potential ideas/solutions, I evaluated and prioritize them based on their feasibility, efficiency and other accuracy. This helped me to identify the most efficient and relevant ideas to pursue further in my project.

Overall, identifying, researching, and collecting ideas for a Design Pattern for Multitenancy with in Microservices pattern using Database Sharding, Elastic pool, ShardAdapter Factory within SaaS SQL server database in MS Azure infrastructure is an efficient but continuous learning and exploration paradigm, it promises to be a primary choice to find the best

solution for our specific needs and goals.

V. GET PEER REVIEWED

Share our work with colleagues or peers: simply share our work with colleagues or peers who are interested in face recognition or related topics. This can be a more informal way to get feedback and review, but can still be valuable in helping us improve and refine our work.

VI. IMPROVEMENTS AS PER REVIEW COMMENTS

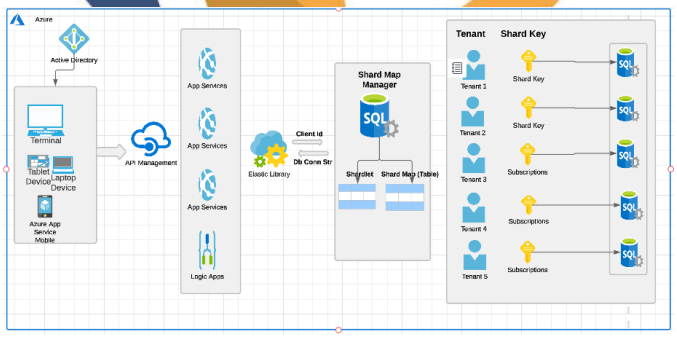
To improve our face recognition design and implementation using peer comments, here are a few key points:
Suggestions or critiques from my peers are most important to improve the efficiency anc accuracy of this system as I have missed some aspects of this design and implementation

Consideration for betterment: - I have planned to implement some more design ideas and tools to improve the accuracy of our Design Pattern. as follows:-

CosmosDB a NoSQL – This pattern can be used in NoSQL, having the reqork in ShardAdapter Factory, ShardManager which will be a revolutionary stem in NoSQL paradigm.
IoC- Depenndency Injection (DI) :- is a subject to implement ShardAdapter Factory. I will provide a higher level of Abstraction and Isolation with Concrete implementation.

Being open to new approaches or techniques, ShardAdapter Factory can be implemented in many different ways. By continuously experimenting and learning, you can improve the performance and efficiency.

VII. MODELS



VIII. CONCLUSION

Multi tenancy has been an attractive field of research for DB Architects and computer scientists. We suggest a proven pattern for managing multi tenancy with Microservices by evolving ShardAdapter, Elastic Pool and ShardLet, and ShardManager. The goal of this research is to ivolve a design pattern which will eliminate the need of architecting the Multitenancy systems with Microservices. The identify faces from a single photograph. Future consideration may include more and other criteria.

REFERENCES

[1] Dominic Betts, “Developing multitenant application for the cloud on Windows Azure”, March 2013, Microsoft Patterns & pracitices, SBN:-9781621140221

[2] Ahmad Osama , 2018, Professional Azure SQL database administration 31 July 2018, ISBN:-9781789535679,

