Chapter 6
The Art of Breaking Down:
Mastering Microservice Architecture and Data Modeling Strategy

Tapan Kumar Behera
https://orcid.org/0000-0003-2524-9171
Department of Technology Management, Forrester Research, USA

ABSTRACT

Microservice architecture (MSA) is a popular software architecture style for developing scalable and resilient applications. However, designing data models for MSA presents unique challenges that require careful consideration. This chapter explores the relationship between MSA and data modeling and provides insights into best practices for designing data models that are optimized for MSA. It defines MSA and its key principles, examines the implications of MSA on data modeling, and discusses strategies for designing data models that are modular, decoupled, and flexible. The chapter also presents several case studies of organizations that have implemented MSA and data modeling strategies and discusses future trends in MSA and data modeling strategy, including the use of artificial intelligence and machine learning to automate data modeling. By following the best practices outlined in this chapter, organizations can realize the benefits of MSA while ensuring data consistency, scalability, and maintainability.

INTRODUCTION

In today's world of software design and development data modeling plays an important role. There are several factors that affect the design of a data model, such as the type and volume of data as well as the specific requirements of the application or system being developed. The type of data refers to its structure and format, such as whether it is structured or unstructured, or whether it requires complex relationships or hierarchies. For example, a data model for a relational database would typically be structured differently than a data model for a document-oriented NoSQL database.

DOI: 10.4018/978-1-6684-9809-5.ch006

Data models are also determined by the volume of data. In the case of large datasets, it may be necessary to store and process the data in a distributed manner, which may affect the design of the data model. Finally, the specific requirements of the application or system can influence the data model. For example, an e-commerce application may require a data model that is optimized for fast retrieval and processing of customer transactions, while a scientific research application may require a data model that supports complex data analysis and visualization.

Data modeling is an important aspect of developing applications using Microservice Architecture (MSA). MSA is a software development approach that involves building a system as a set of small, independent services that can be deployed and scaled independently (Behera, 2023). Here are some factors to consider when designing data models for MSA:

1. **Decoupling of Data Models:** In MSA, each microservice is responsible for its own data storage, which means that data models should be designed to be decoupled from each other. This allows for greater flexibility and scalability, as each microservice can evolve its own data model independently.
2. **Modular Design:** Data models should be designed to support a modular architecture, with each microservice having its own data model that is focused on the specific needs of that service. This allows for greater flexibility and makes it easier to make changes to the system over time.
3. **Data Consistency:** In a distributed system like MSA, maintaining data consistency across microservices can be challenging. Data models should be designed to ensure consistency across microservices, either using distributed transactions or other techniques such as eventual consistency.
4. **Data Partitioning:** In MSA, it is common for data to be partitioned across multiple microservices. Data models should be designed to support this partitioning, with each microservice responsible for its own partition of the data.
5. **API Design:** The design of the API that exposes the data should be considered when designing data models. The API should be designed to be simple and easy to use, with clear documentation and well-defined contracts between microservices.

MICROSERVICE ARCHITECTURE (MSA)

Microservice Architecture (MSA) is a software development approach that involves building a system as a collection of small, independent services that can be deployed and scaled independently. In MSA, each service is designed to perform a specific business function and can communicate with other services through well-defined APIs (Fresno et al., 2023). This approach contrasts with traditional monolithic architectures, where the entire system is built as a single, tightly-coupled application.

Monolithic vs. Microservices Architecture

The monolithic and microservices architectures are two different approaches to developing software applications. In a monolithic architecture, the entire application is designed as a single, tightly coupled unit, with all of its components running within the same process and on the same hardware. In this architecture, a single codebase, a single database, and a single user interface are typically used. A Monolithic Architecture requires the entire application to be modified and re-deployed to make changes to the system.