

Traditional Monolithic Application Design

In traditional monolithic design, everything is bundled in the same archive such as presentation code, application logic and business logic code, and DAO and related code that interacts with the database files or another source.

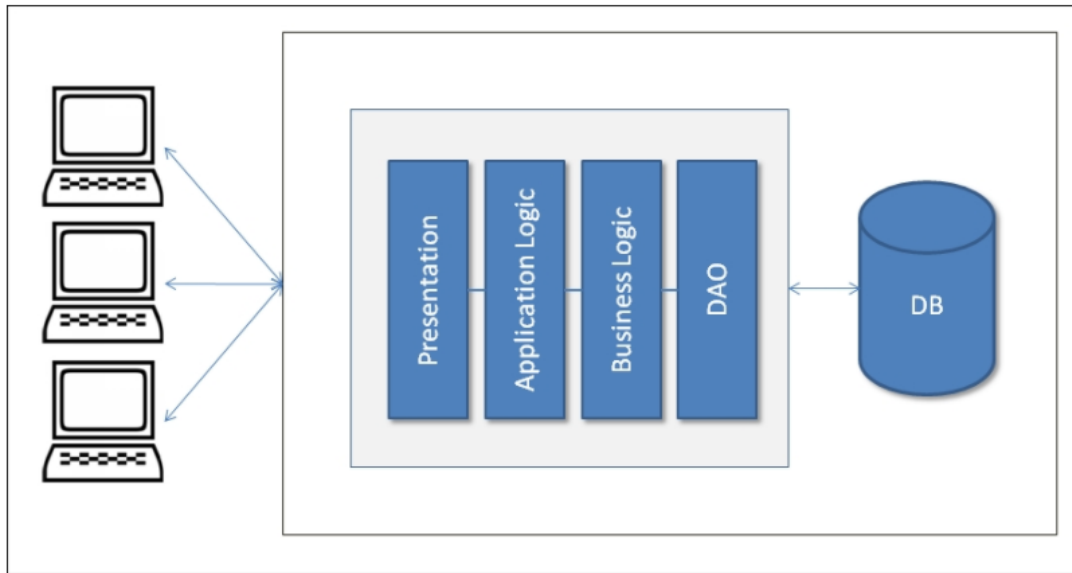


Figure: Traditional Monolithic Design

After SOA, applications started being developed based on services, where each component provides the services to other components or external entities.

The following diagram depicts the monolithic application with different services; here services are being used with a presentation component. All services, the presentation component, or any other components are bundled together:

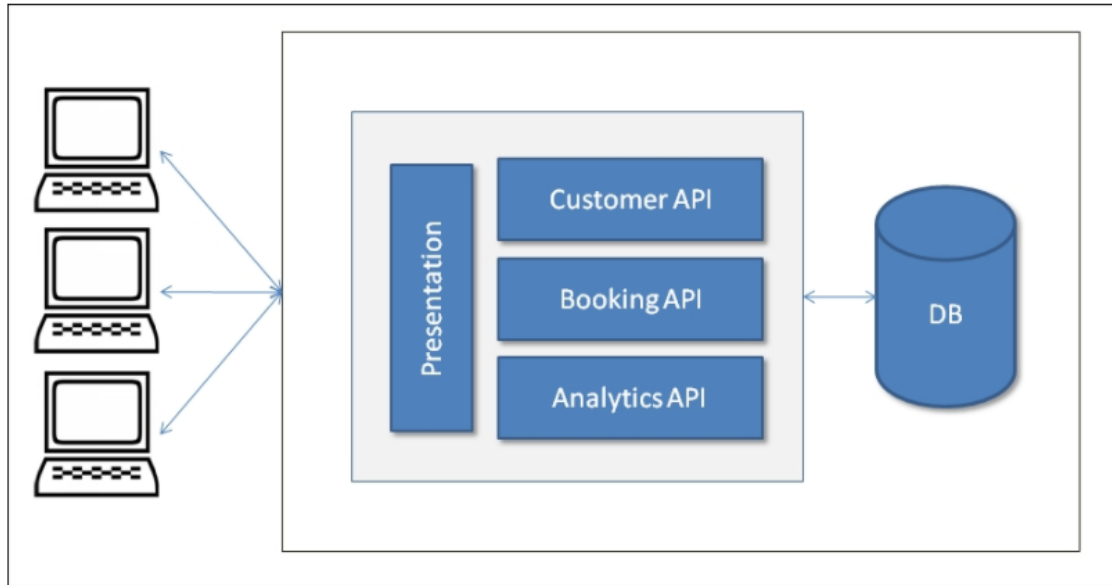


Figure: Monolithic with Services

The following third design depicts the Microservices. Here, each component represents autonomy. Each component could be developed, built, tested, and deployed independently.

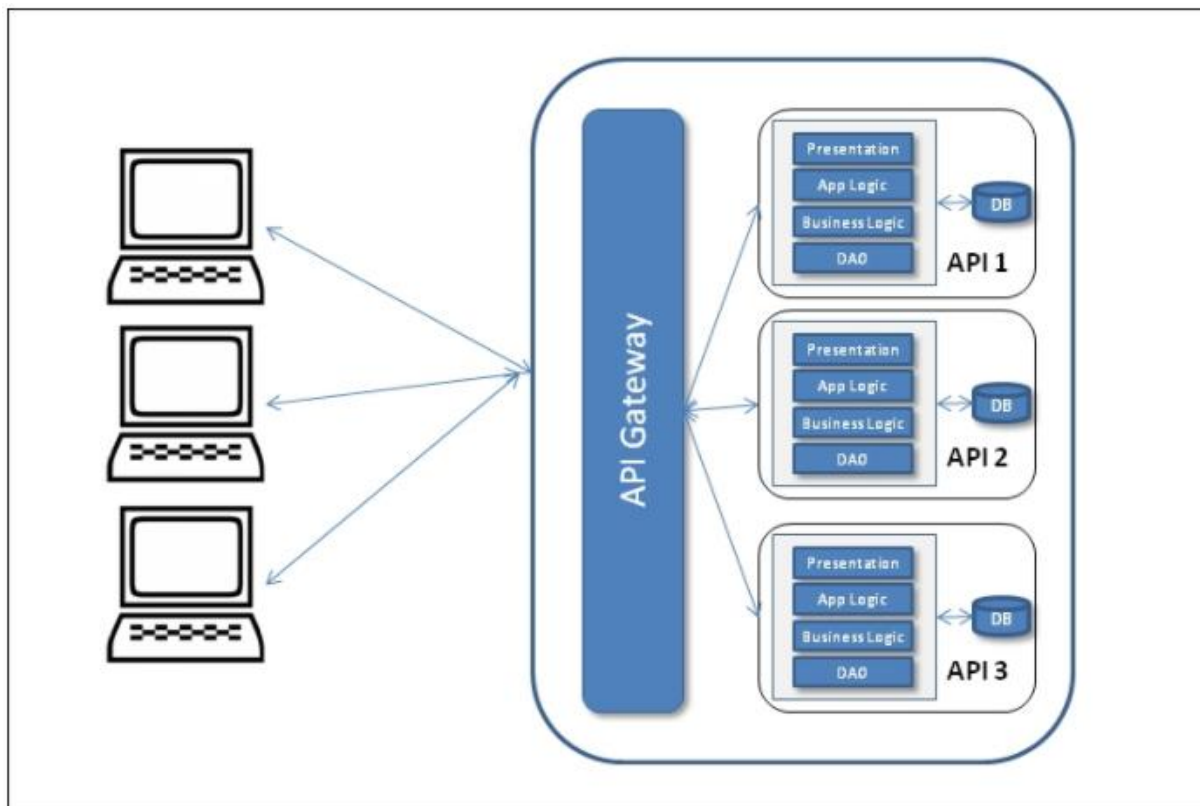


Figure: Microservices Design

The API gateway provides the interface where different clients can access the individual services and solve the following problems:

- What to do when you want to send different responses to different clients for the same service. For example, a booking service could send different responses to a mobile client (minimal information) and a desktop client (detailed information) providing different details and something different again to a third-party client.
- A response may require fetching information from two or more services:

Observations:

- In monolithic design, the components are bundled together and tightly coupled.
- In SOA services, all the services are part of the same bundle. All services could have their own layers and form different APIs, but, as shown in the figure, these are also all bundled together.
- In Microservices, design components are not bundled together and have loose coupling. Each service has its own layers and DB and is bundled in a separate archive. All these deployed services provide their specific API such as Customers, Bookings, or Customer. These APIs are ready to consume.