COMPSCI 497S

Scalable Web Systems

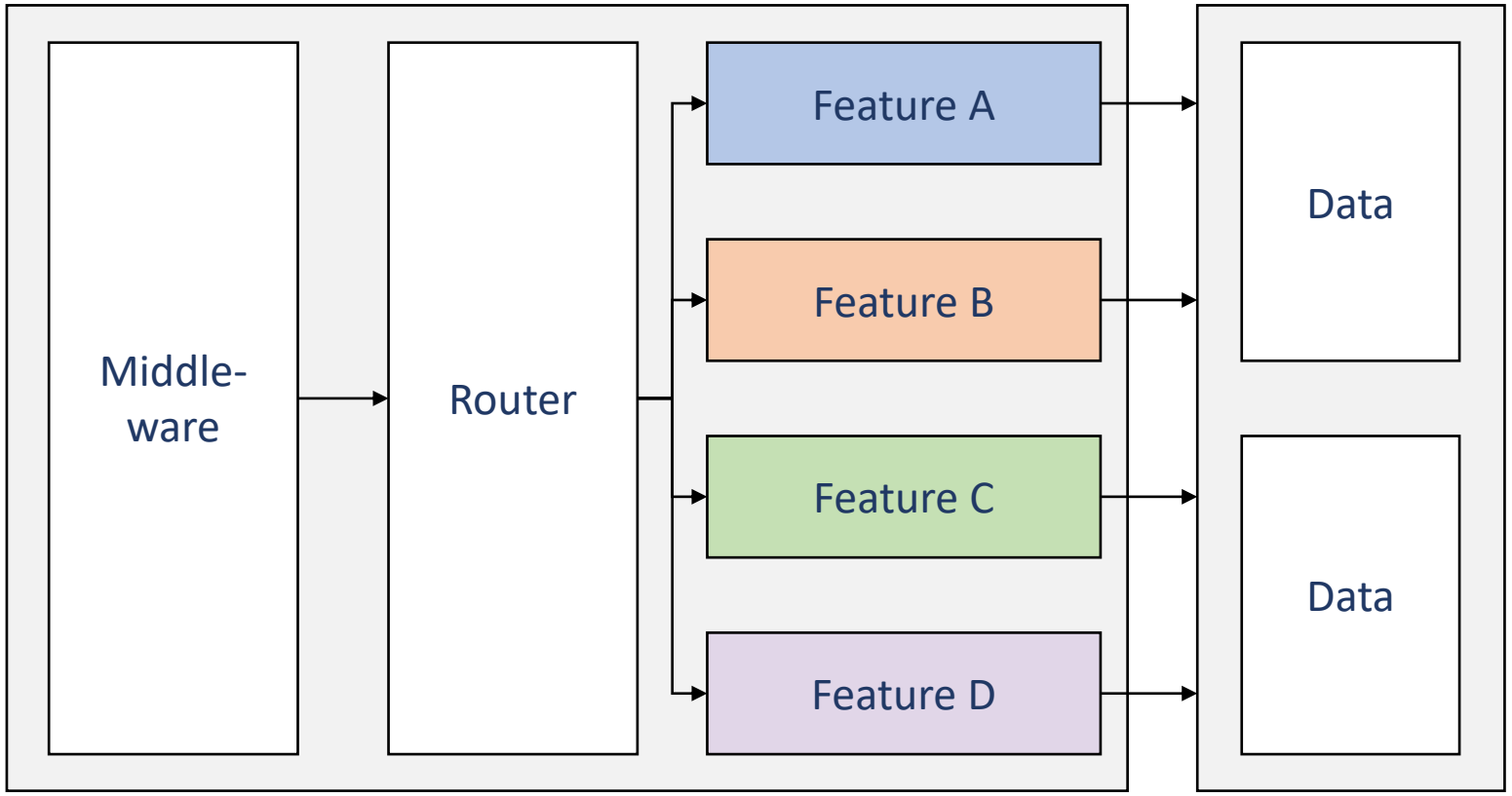
01 Micro-Services Fundamentals

Today

- Monolithic Architecture
- Micro-Service Architecture
- Data Management Between Services
- Database Per Service

Monolithic Architecture

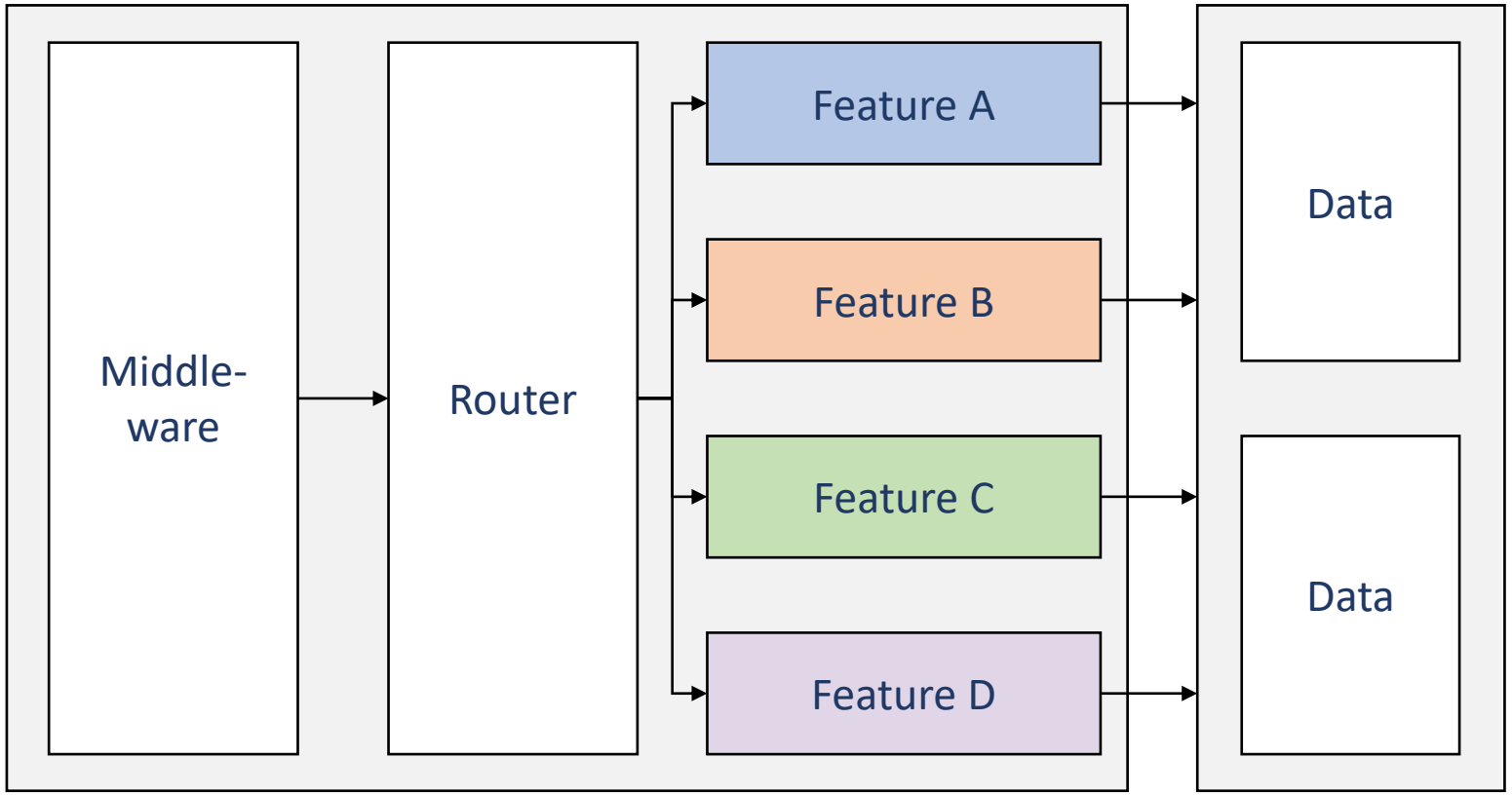
This is probably how you are building your systems right now.



Monolithic Architecture

A request arrives on some port given some route.

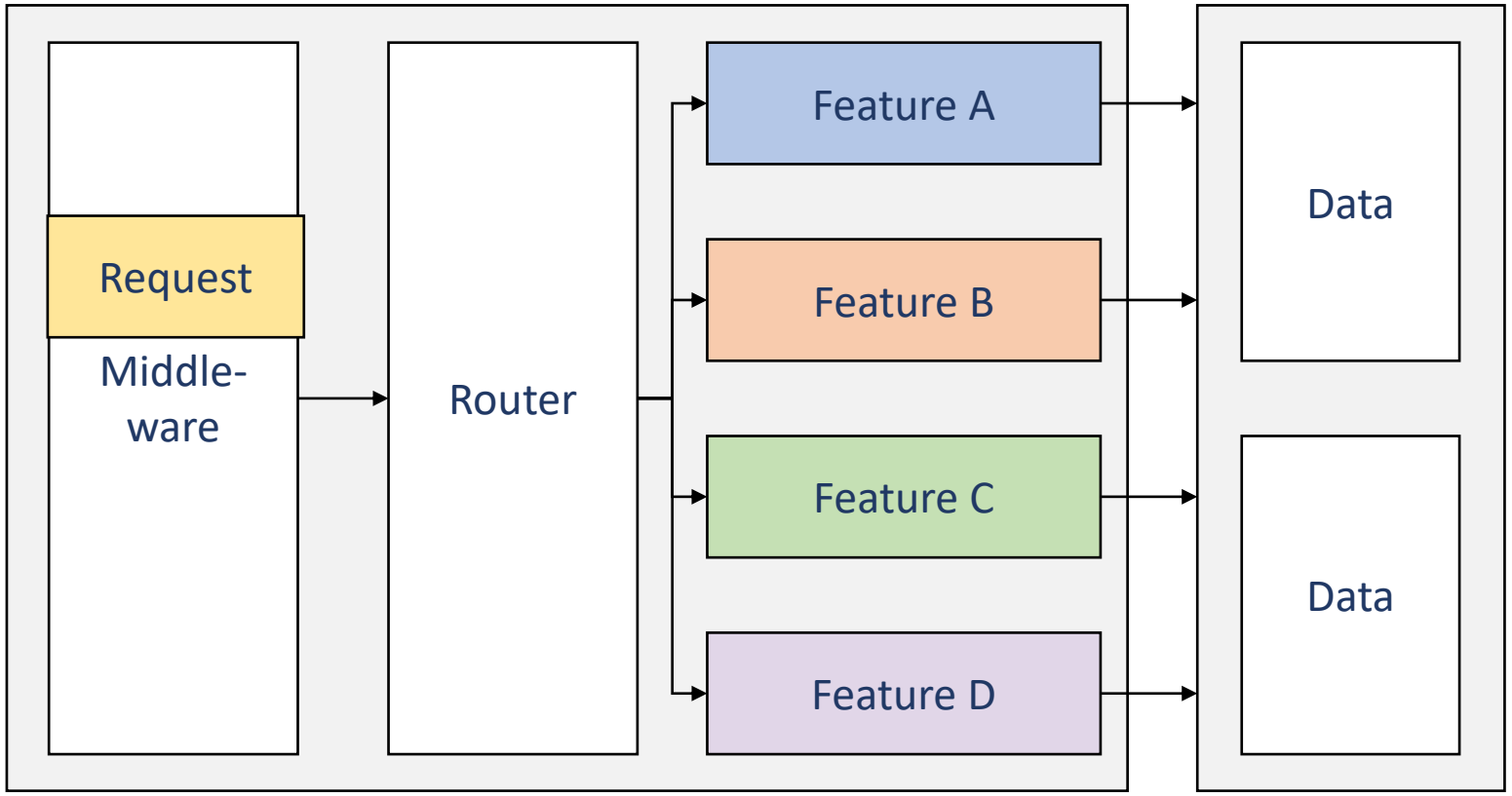
Request



Monolithic Architecture

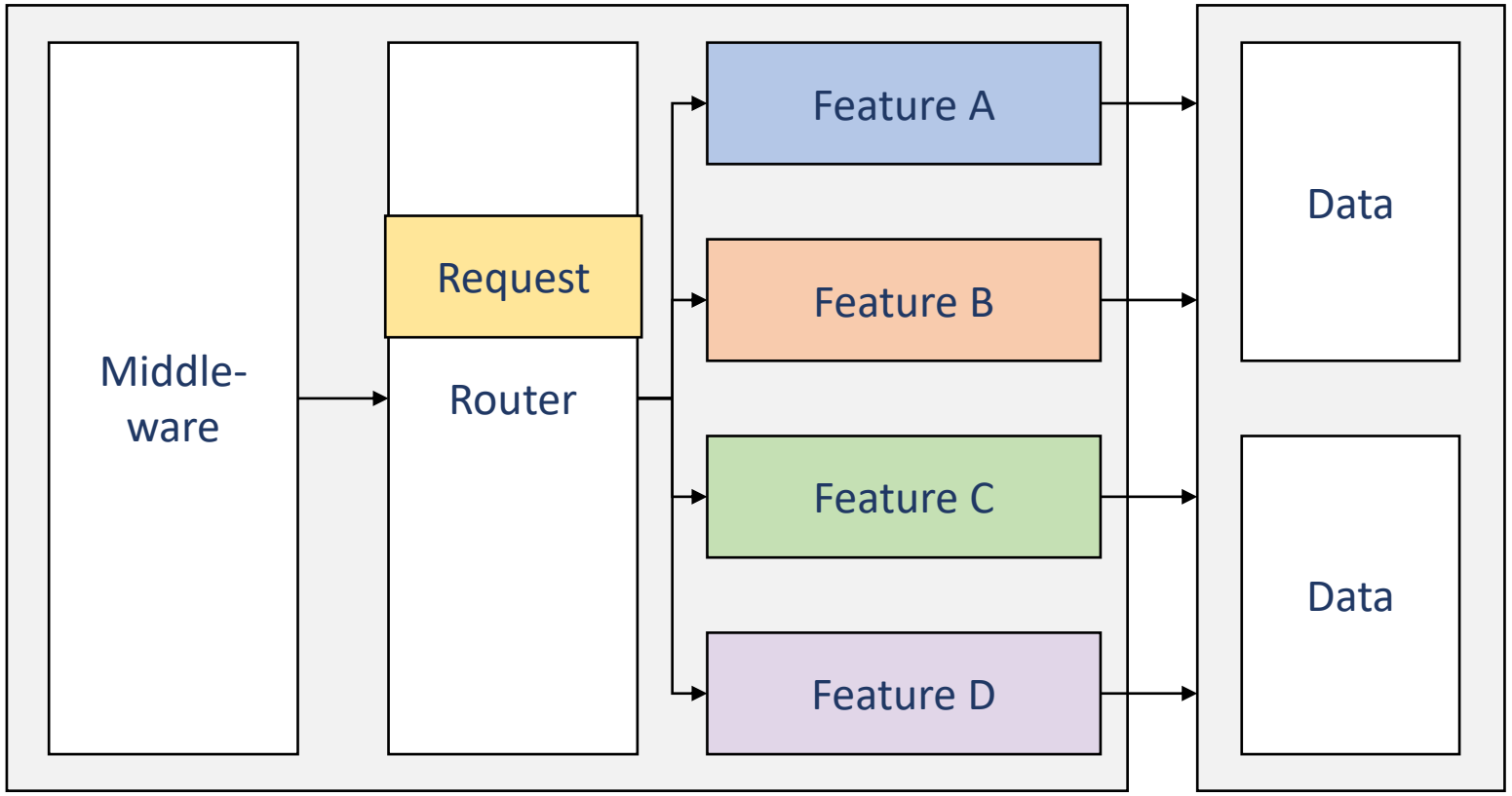
The various middleware processes the request.

Body parsing, sessions, authentication, etc.



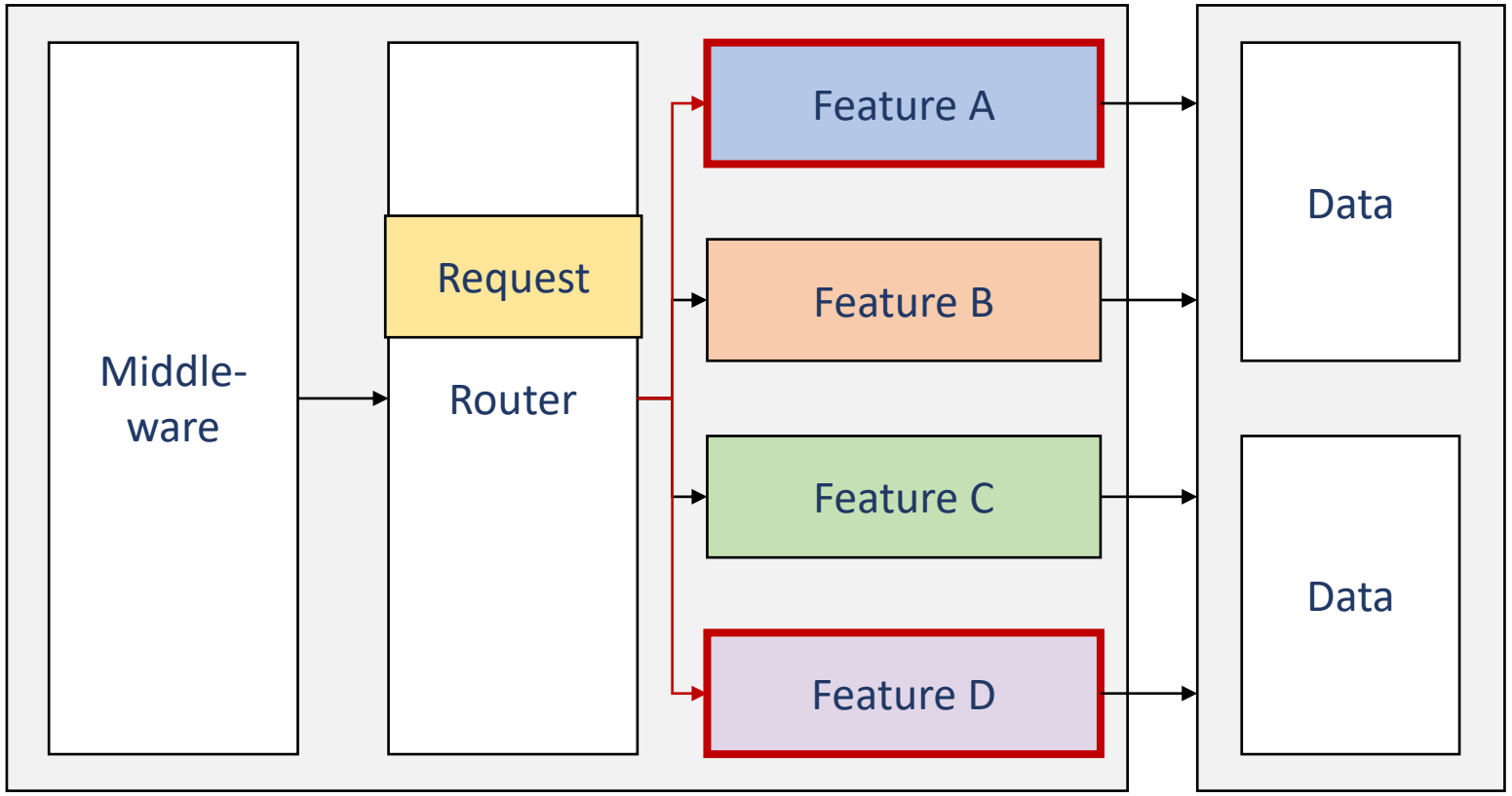
Monolithic Architecture

The router looks at the route to determine what the request and invokes the appropriate controller.



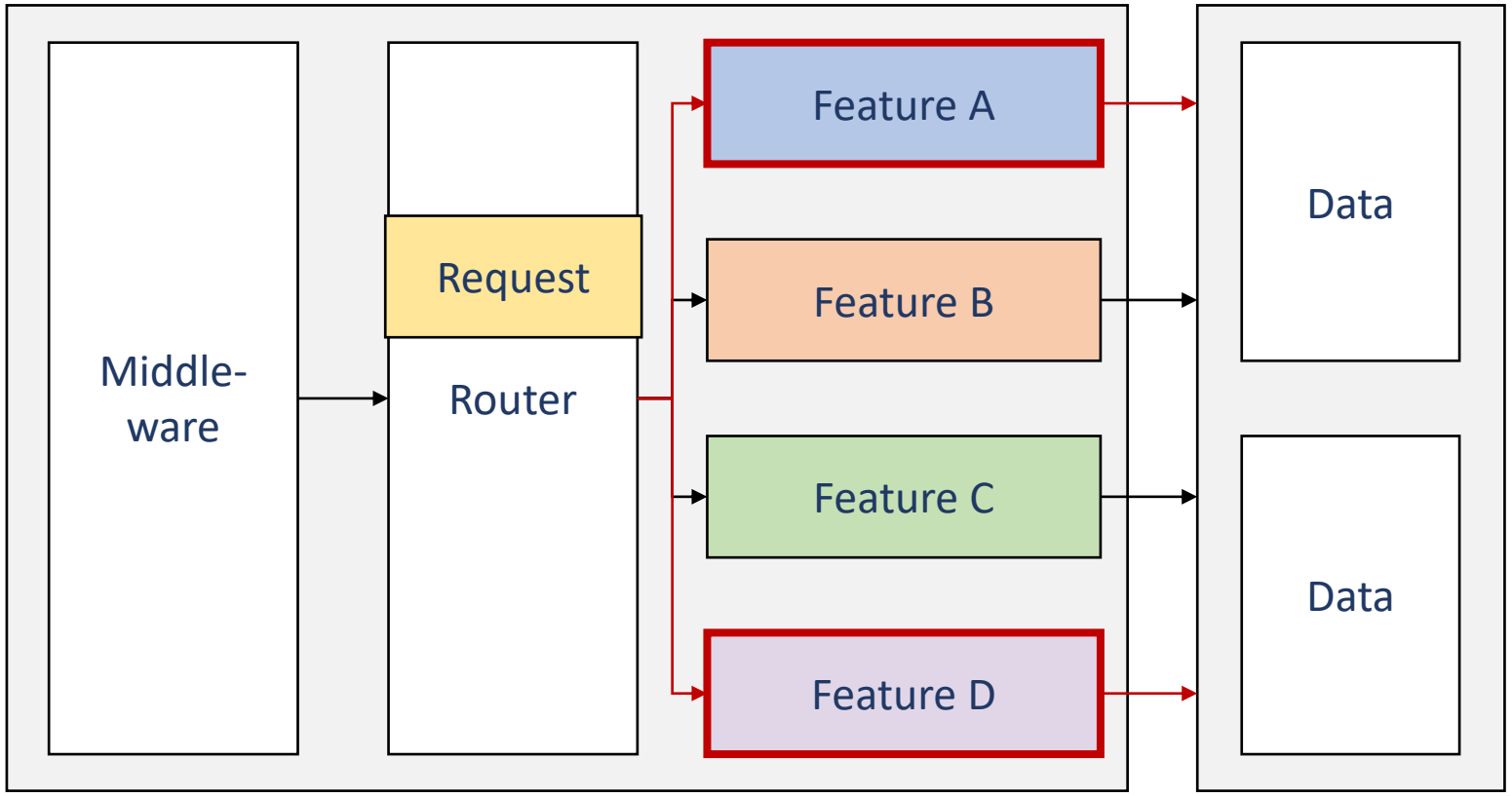
Monolithic Architecture

This, in turn, causes various features to be activated.



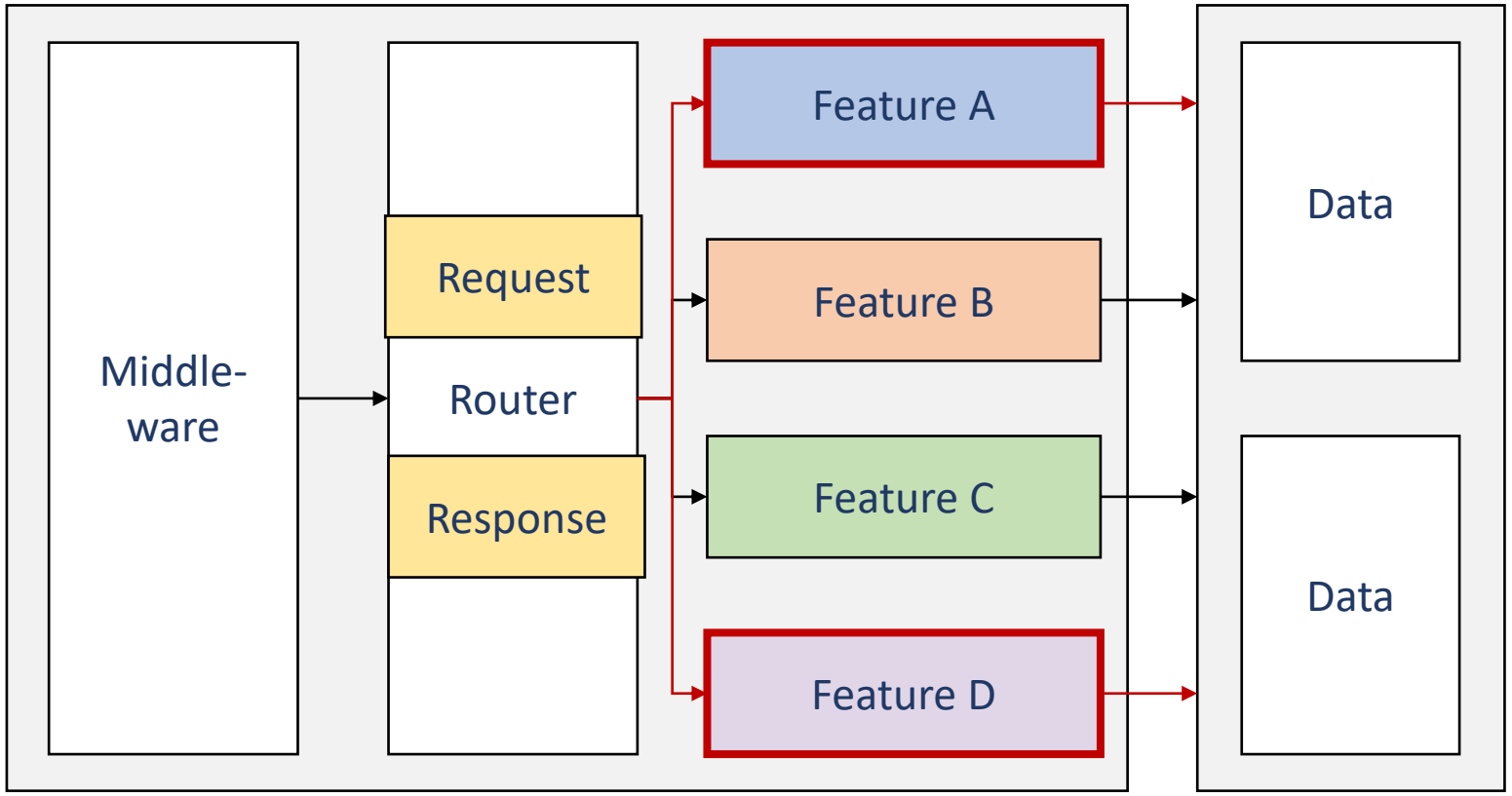
Monolithic Architecture

Data is fetched/stored through various CRUD operations.



Monolithic Architecture

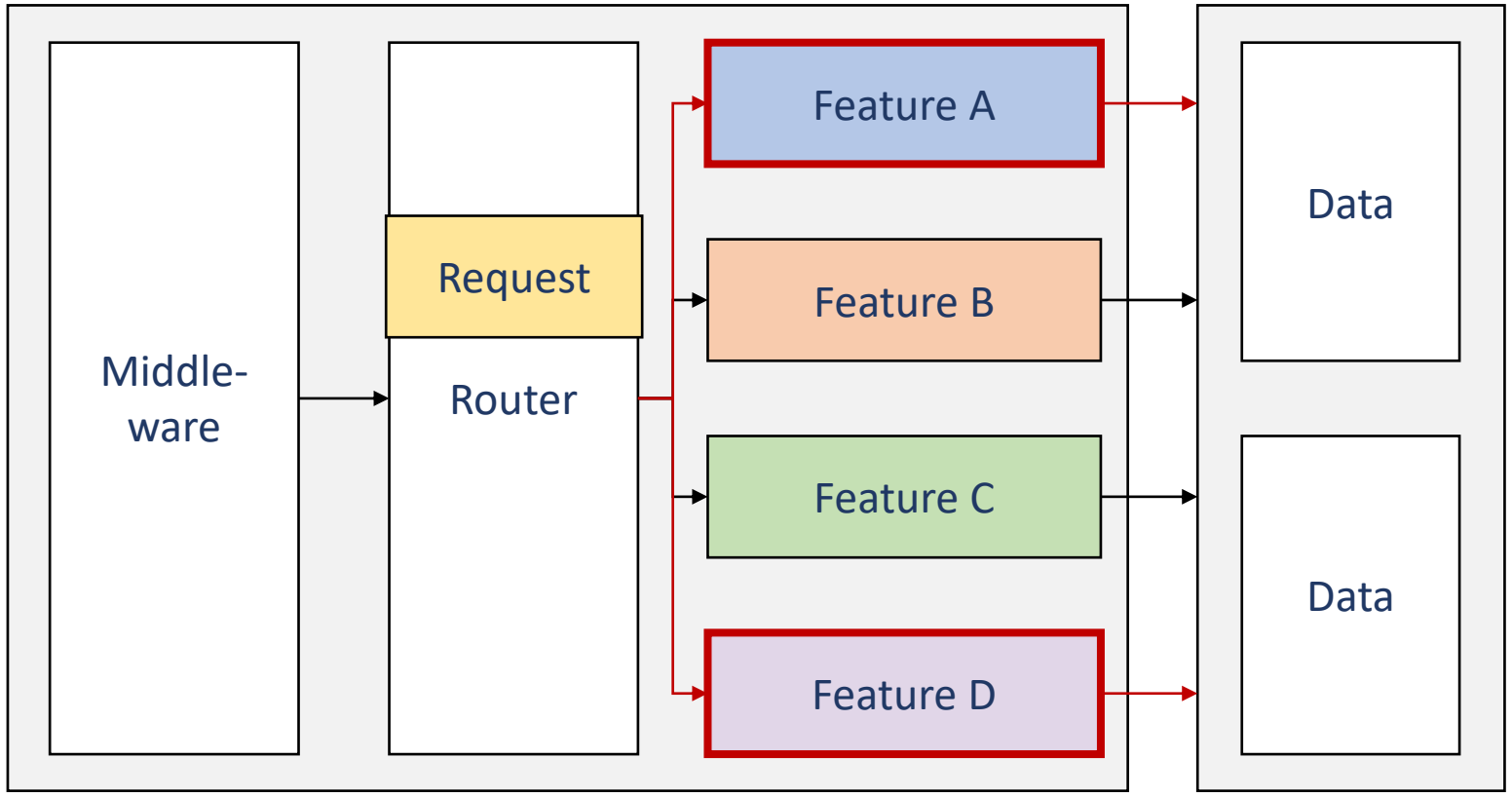
The activated features complete and a response is constructed.



Monolithic Architecture

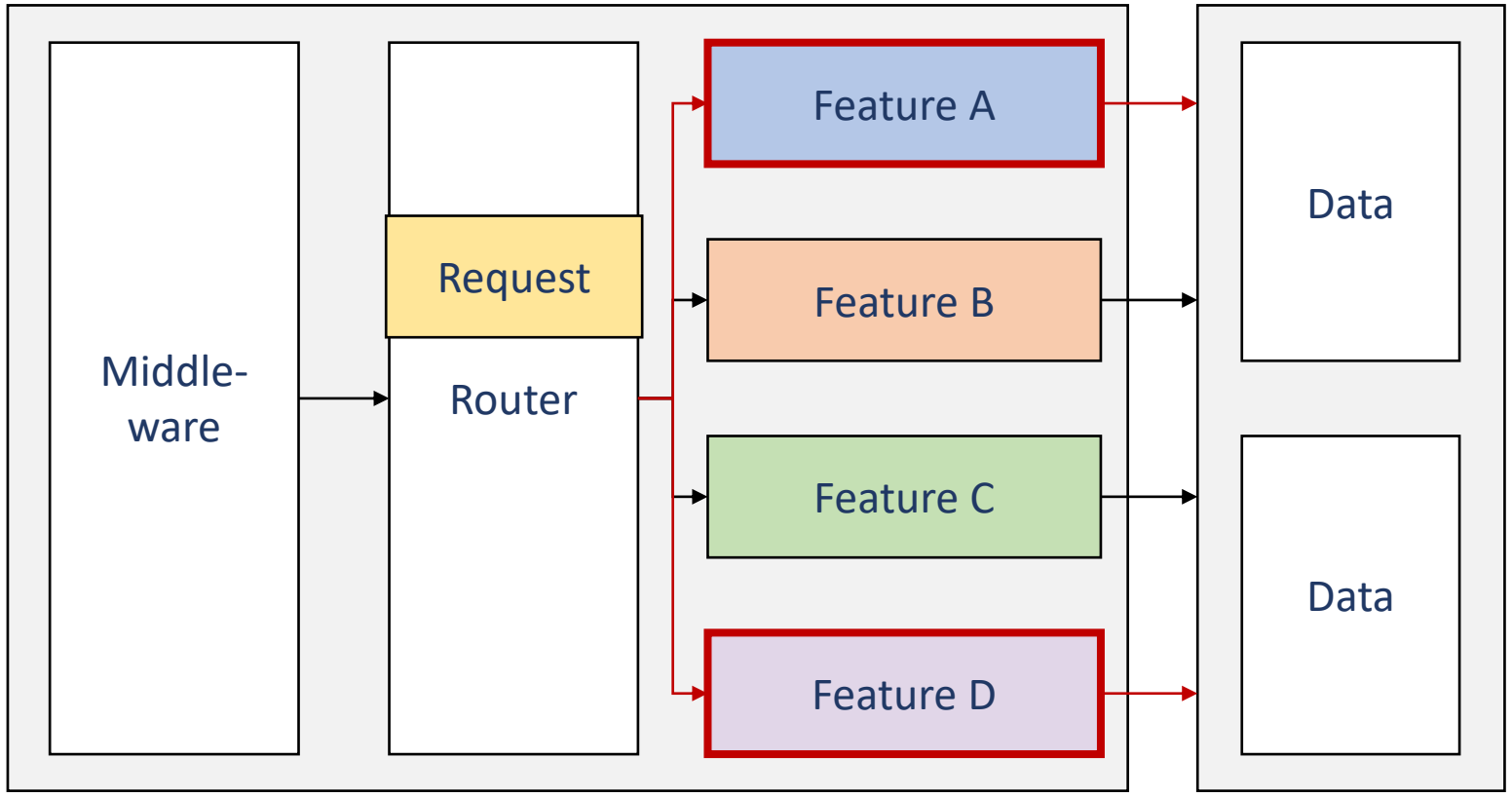
The response is then sent back to the requesting client.

Response



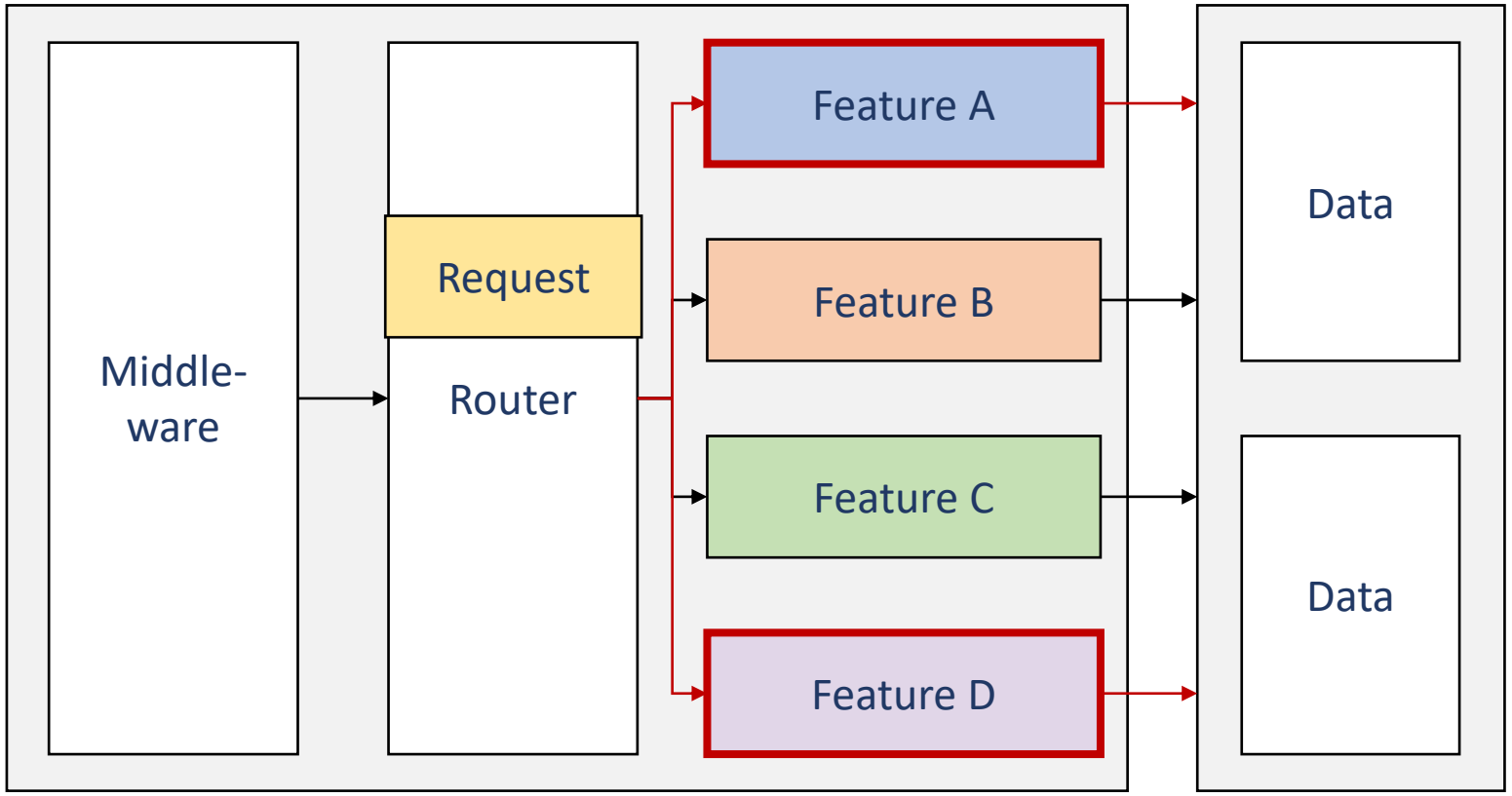
Monolithic Architecture

This is not an unreasonable approach to building a web system.



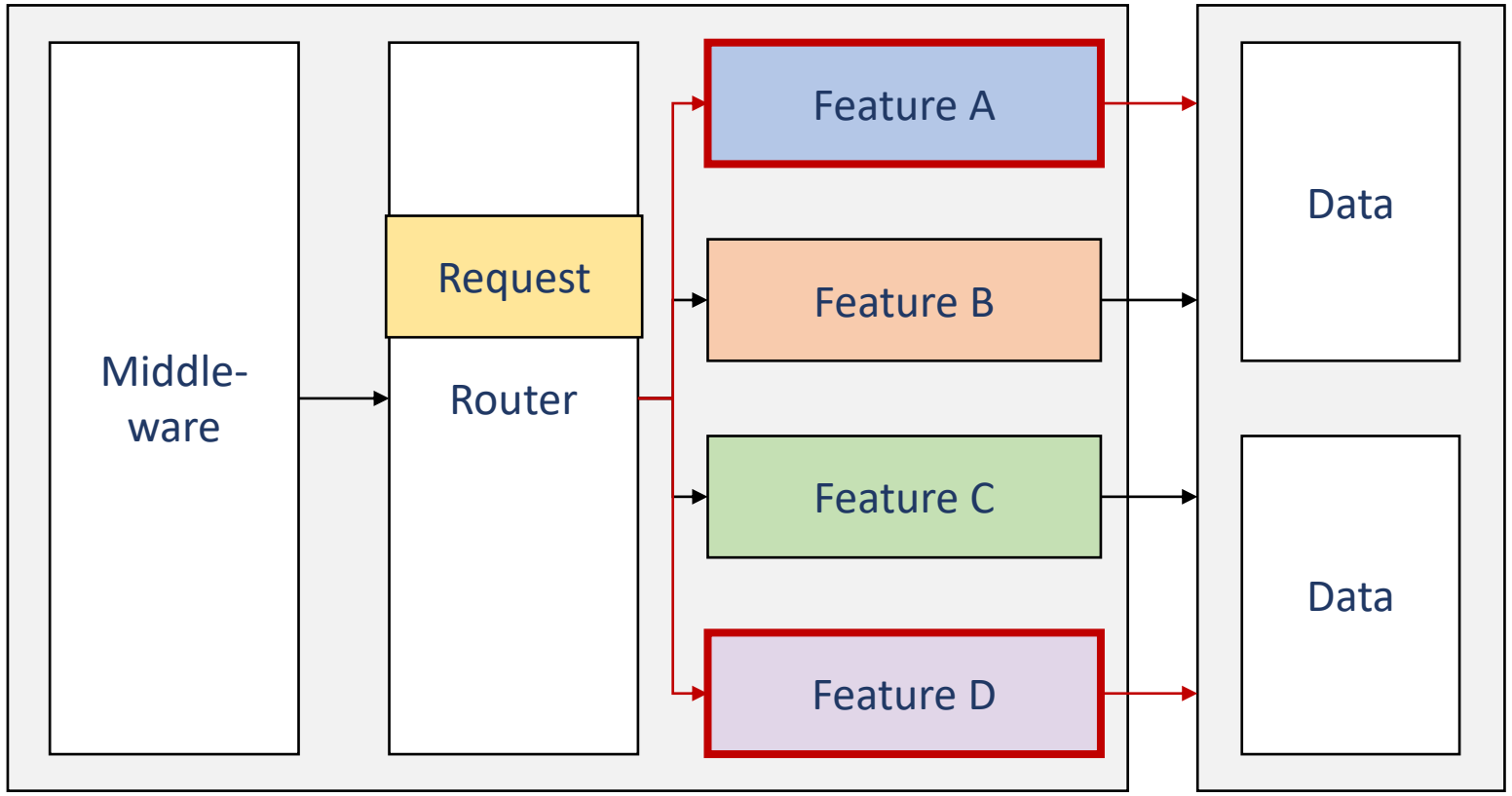
Monolithic Architecture

Do other architectures exist for doing this?



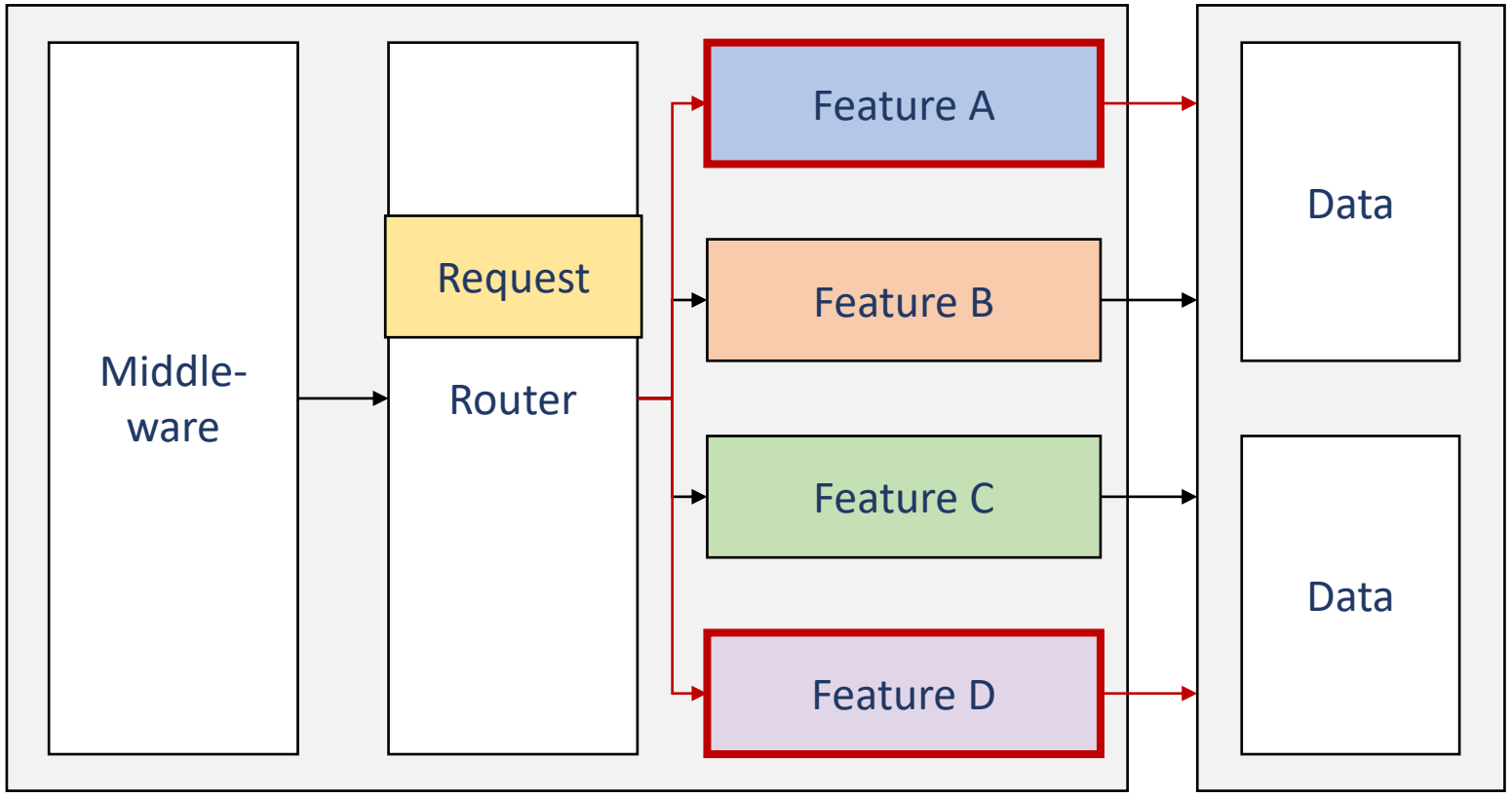
Monolithic Architecture

Are there ways to make this more efficient?



Monolithic Architecture

What if we scale this to a million requests per second?

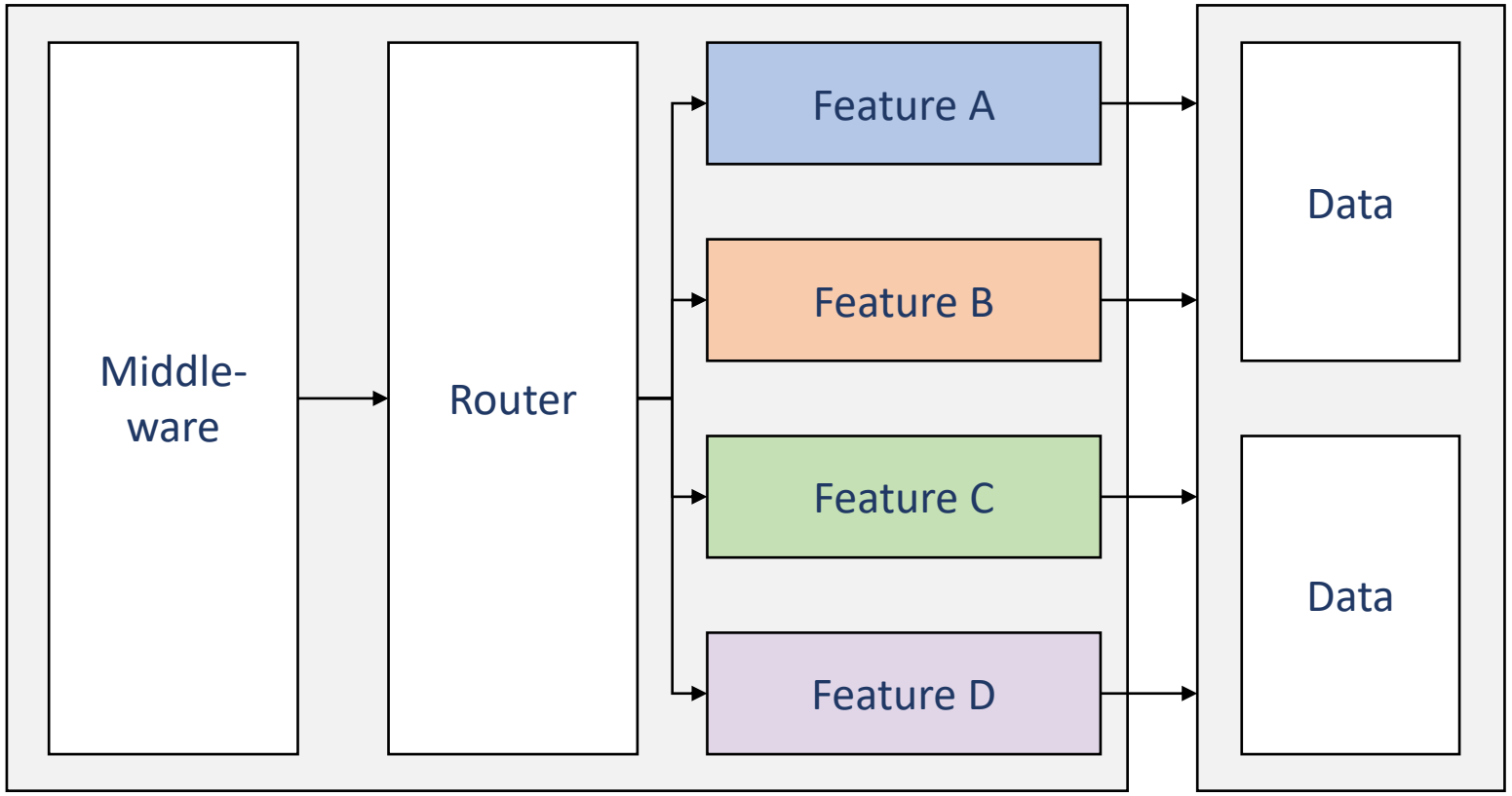


Code Example: 01-monolith

What is a micro-service?

Monolithic Architecture

Again, here is what a monolithic architecture.

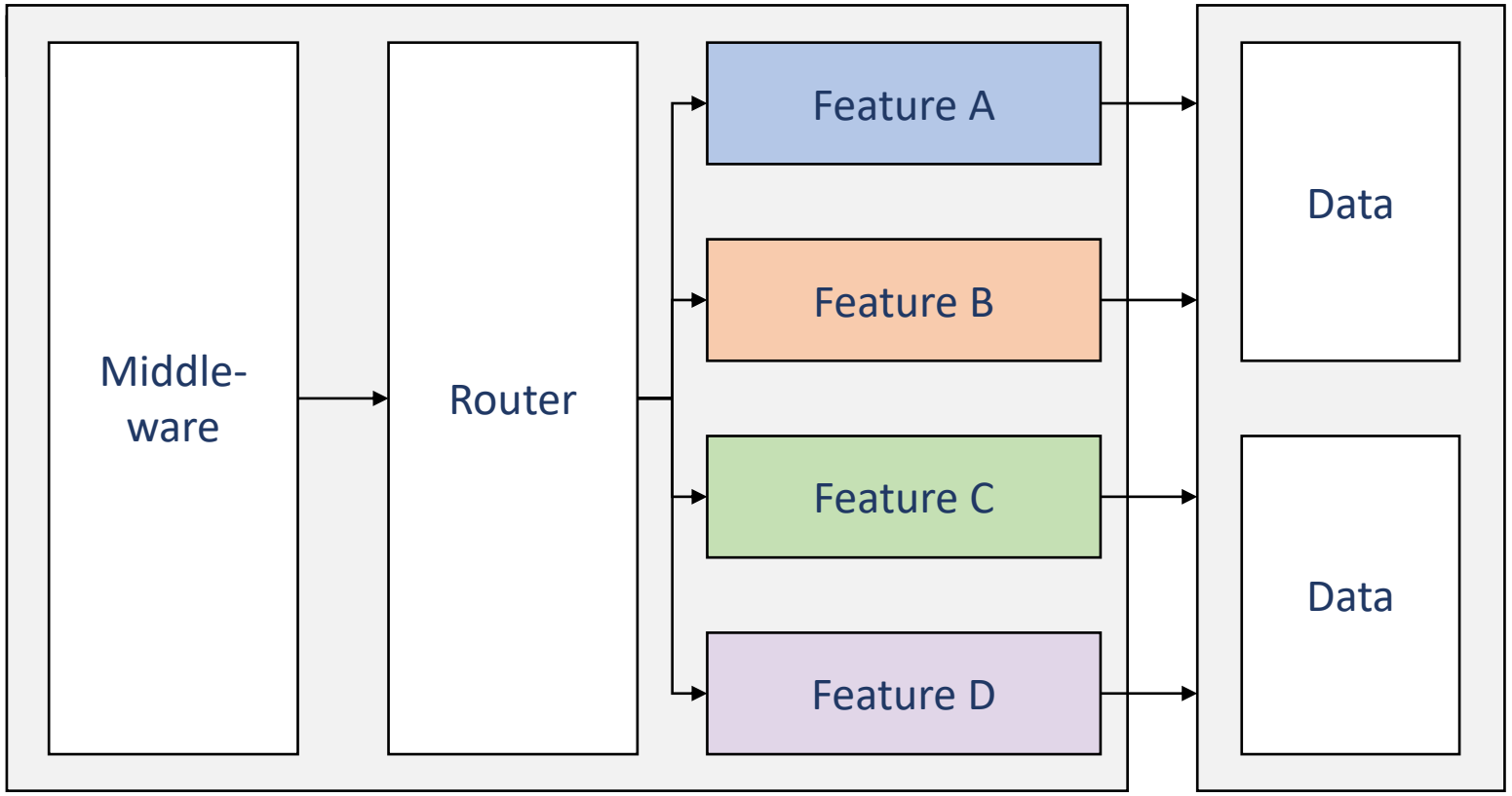


Monolithic Architecture

A **monolith** contains:

- Routing
- Middlewares
- Business Logic
- Database Access

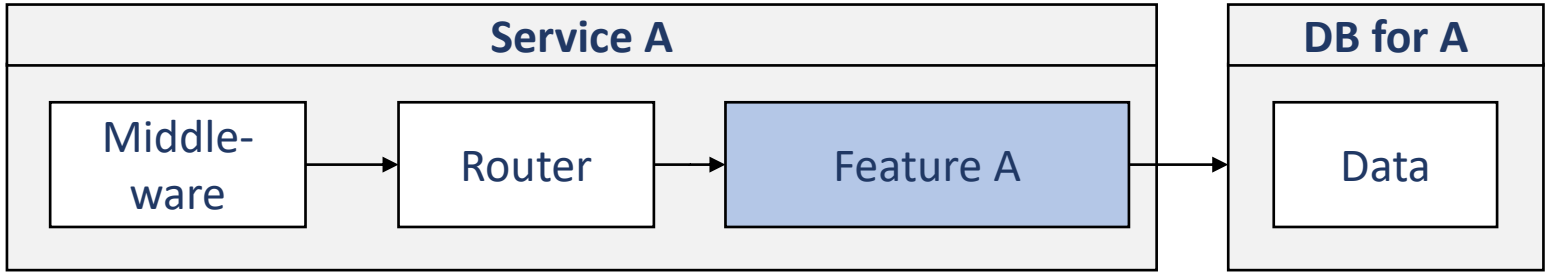
For **all** features.



Micro-Service Architecture

A **micro-service** contains:

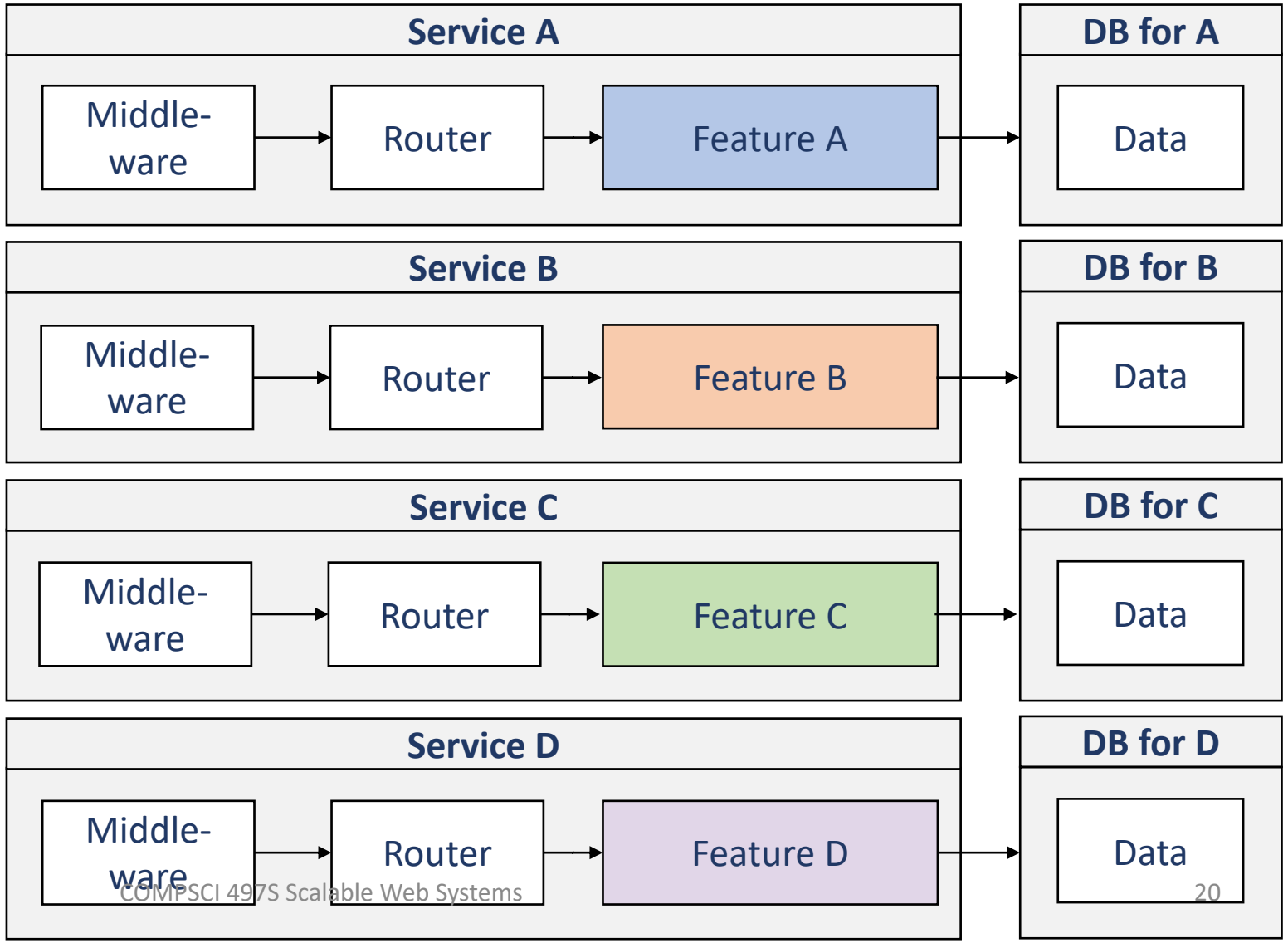
- Routing
- Middlewares
- Business Logic
- Database Access



For a **single** feature!

Micro-Service Architecture

A **micro-service architecture** is composed of many micro-services.



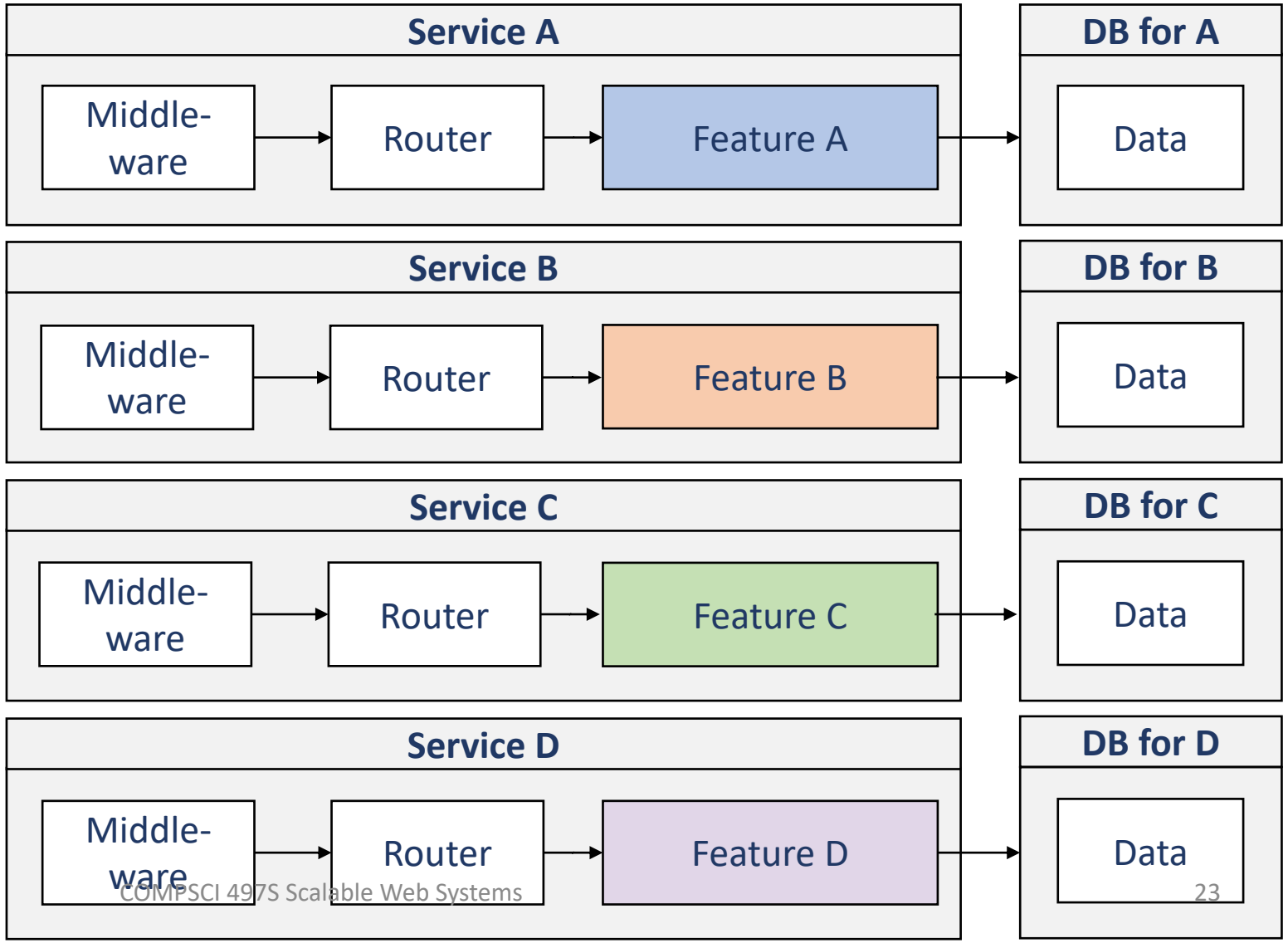
What is the biggest challenge with micro-services?

Data management between services.

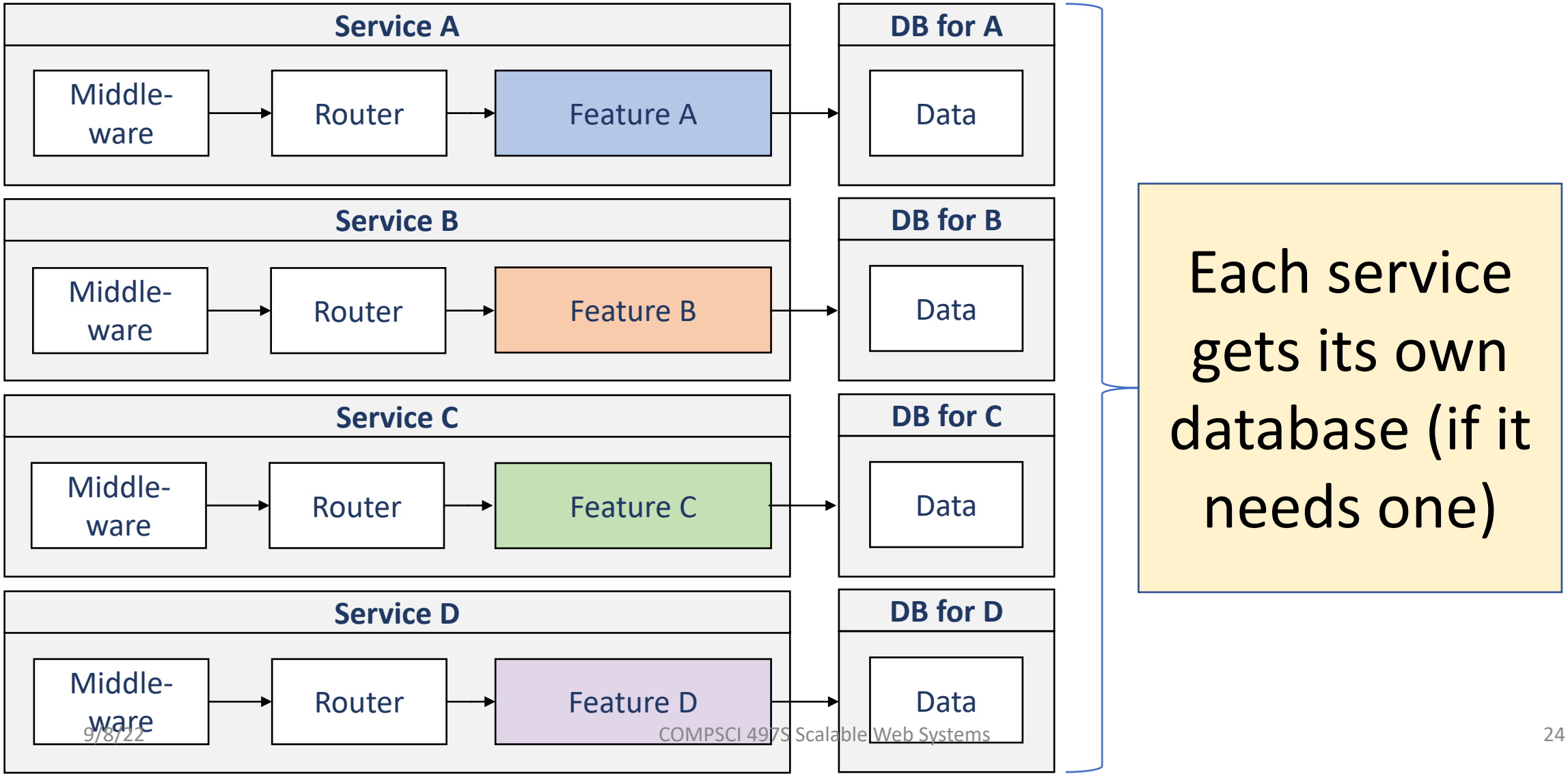
Micro-Service Architecture

A **micro-service architecture** works with data that is fundamentally different than a monolith.

- **How it is stored**
- **How it is accessed**

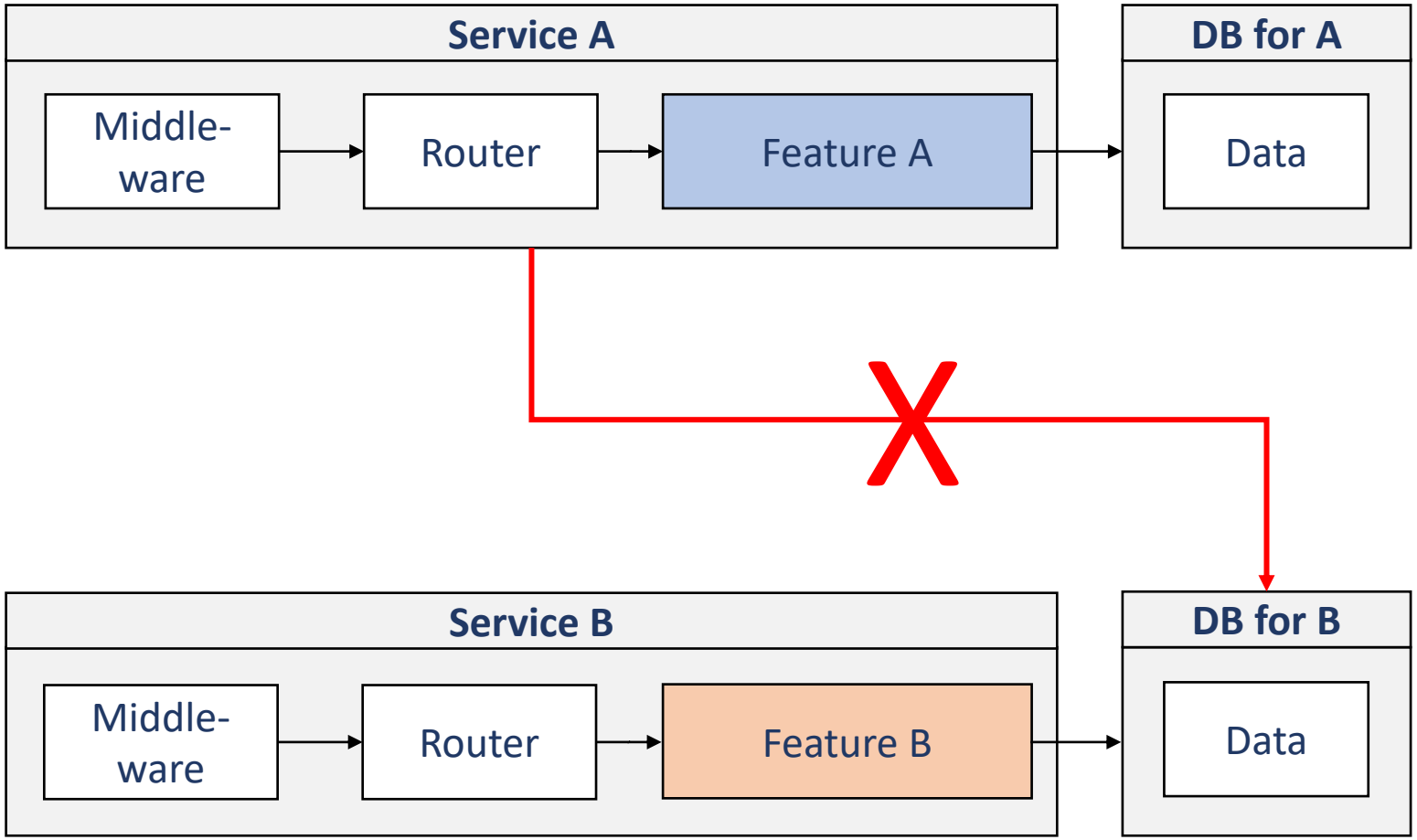


Micro-Service Architecture



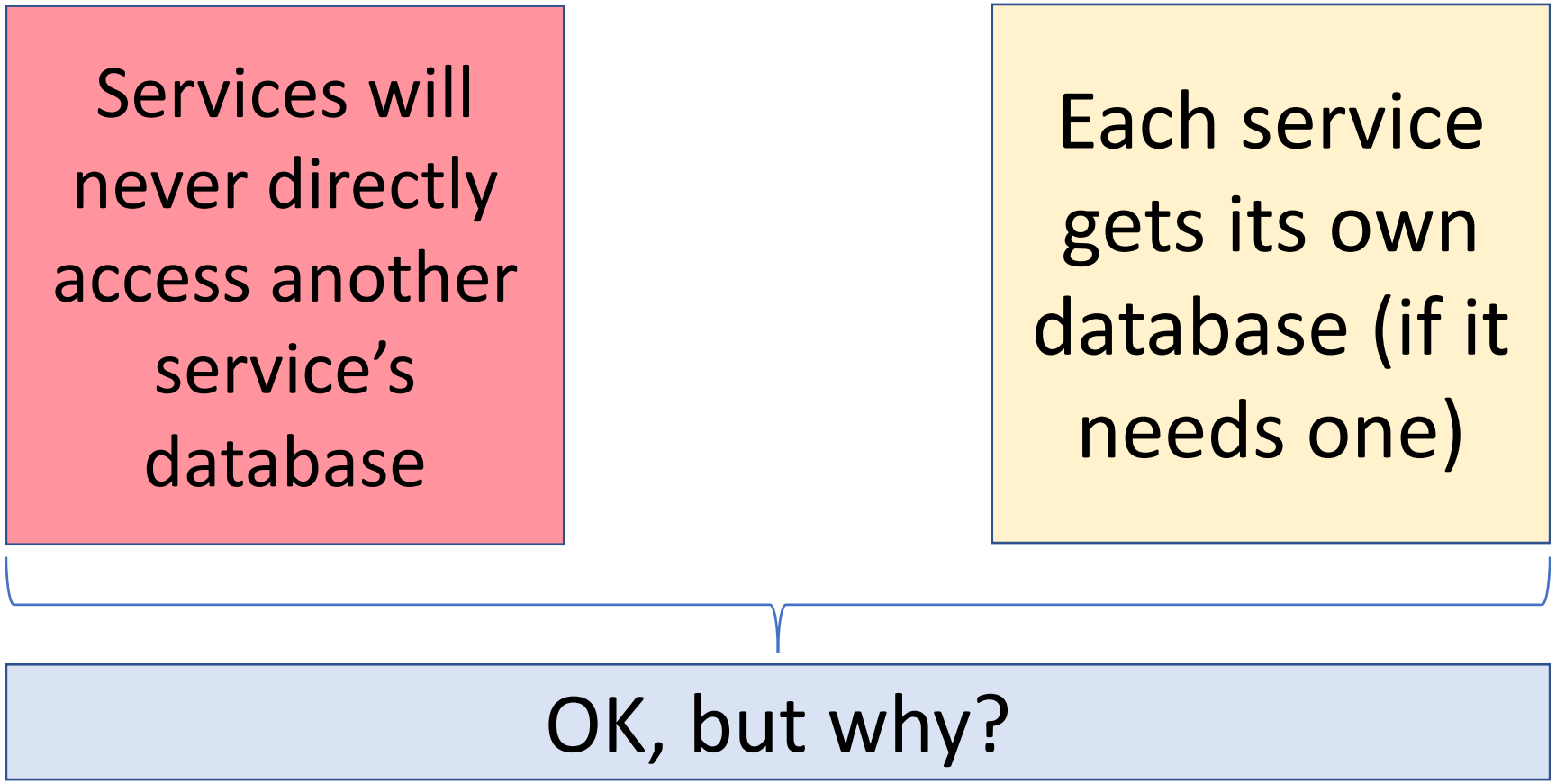
Each service gets its own database (if it needs one)

Micro-Service Architecture



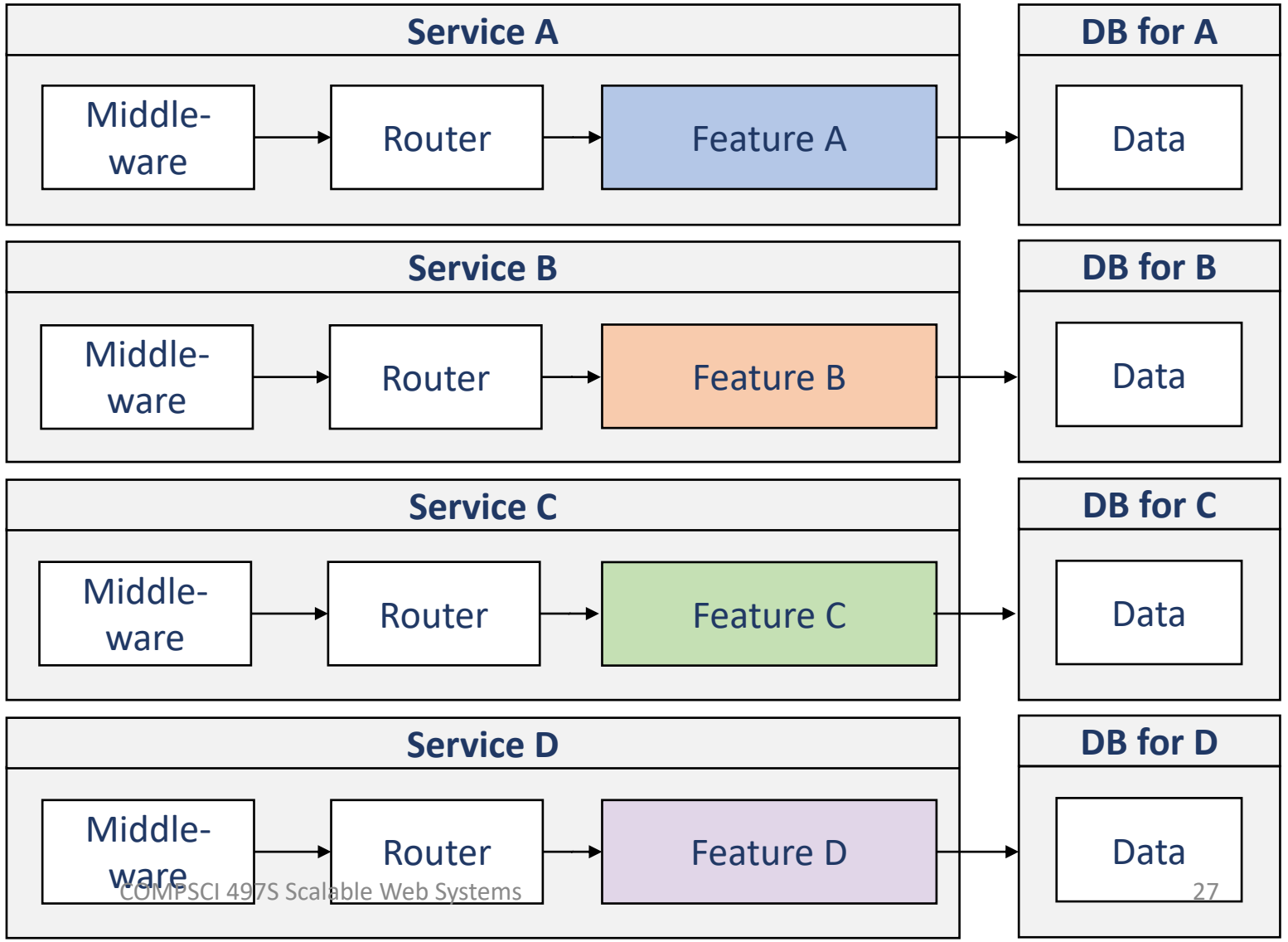
Services will never directly access another service's database

Micro-Service Architecture



Micro-Service Architecture

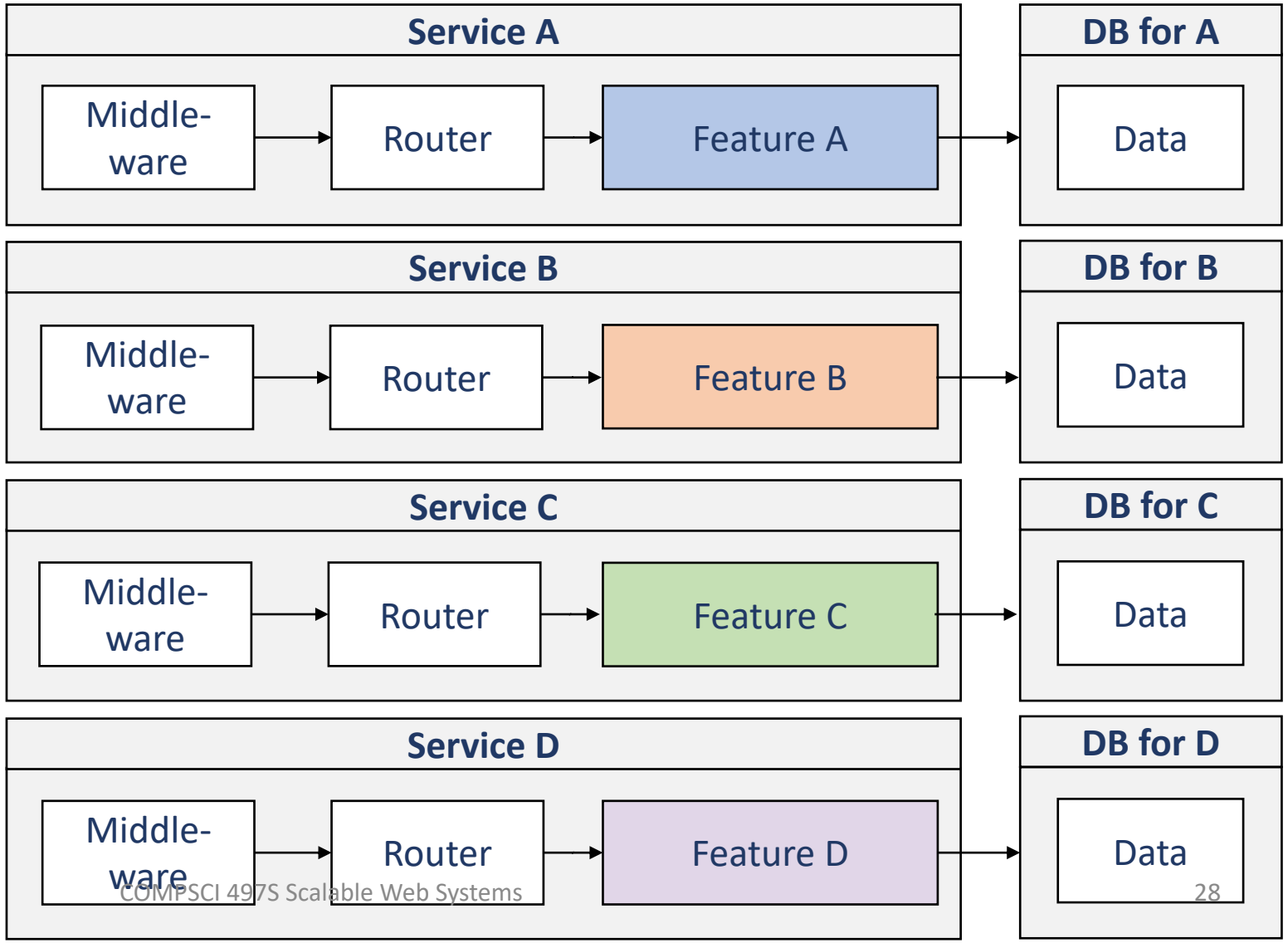
Why a database per service?



Micro-Service Architecture

Why a database per service?

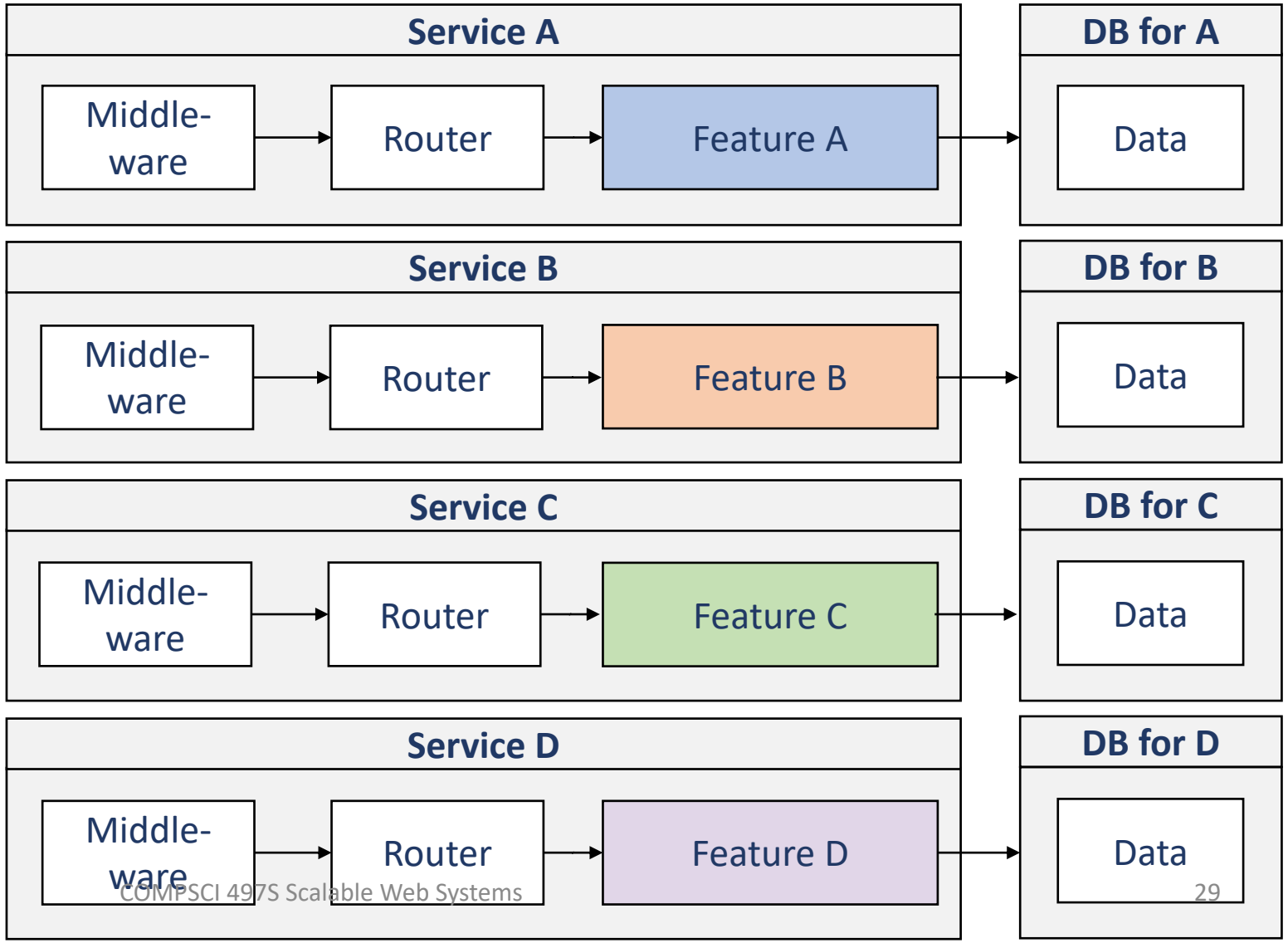
We want each service to run independently of other services.



Micro-Service Architecture

Why a database per service?

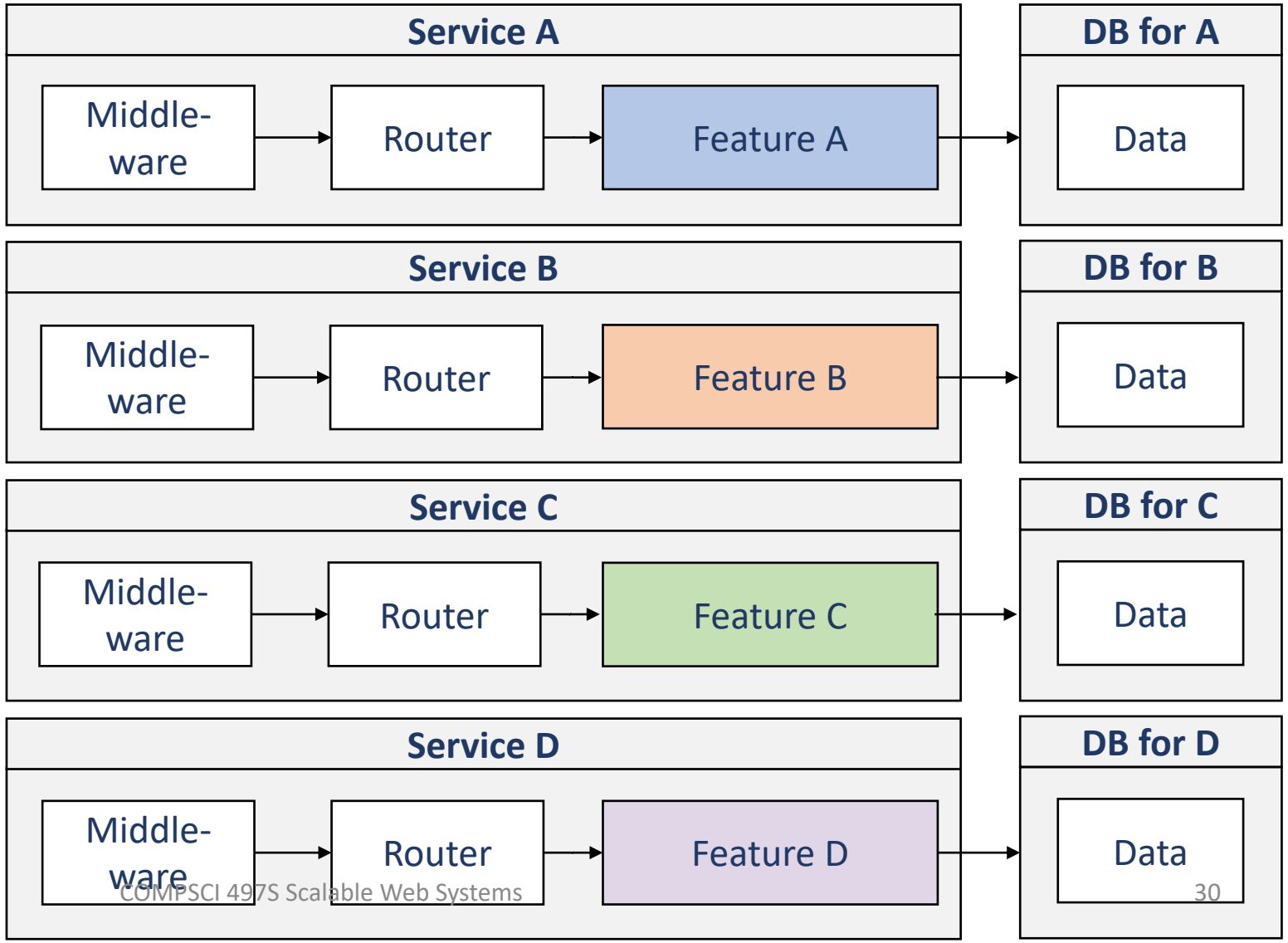
Database schema/structure might change unexpectedly.



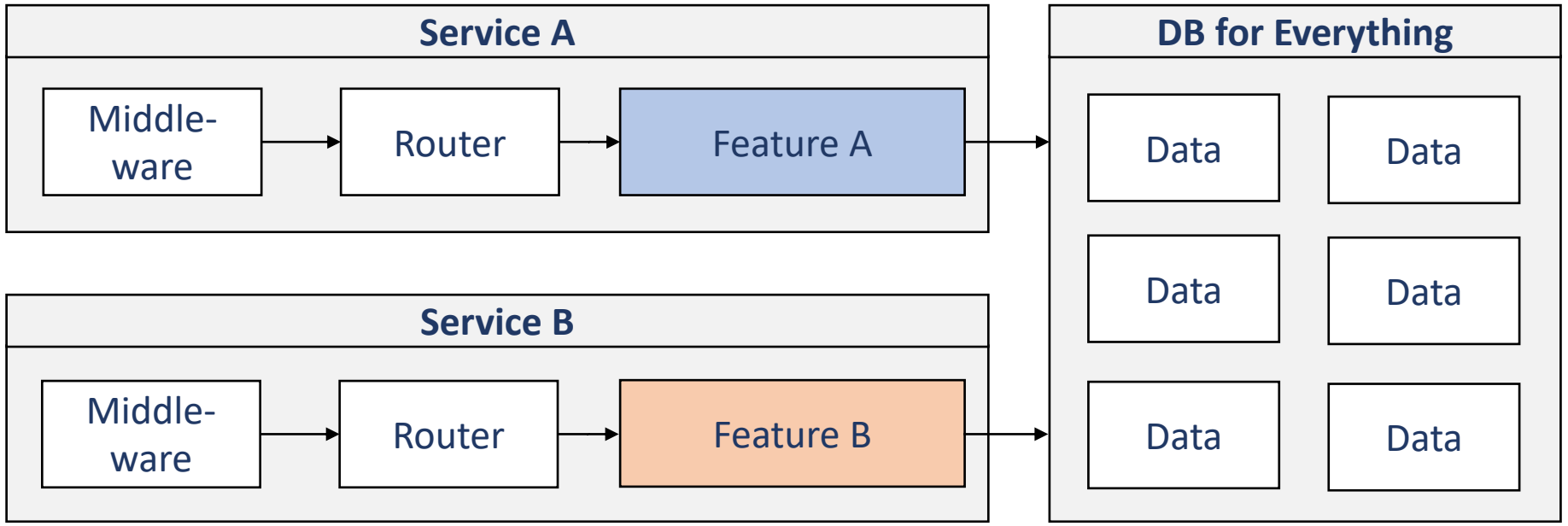
Micro-Service Architecture

Why a database per service?

Some services might function more efficiently with different types of DB's (sql vs nosql)

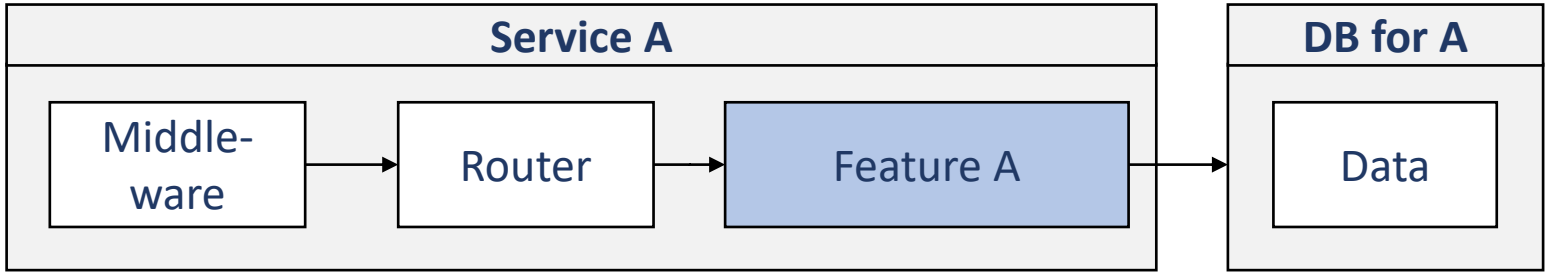


Micro-Service Architecture



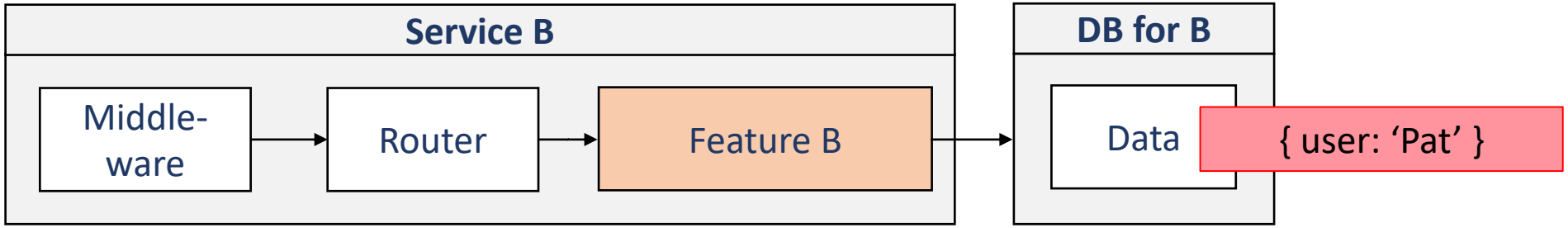
What could go wrong with this architecture?

Micro-Service Architecture

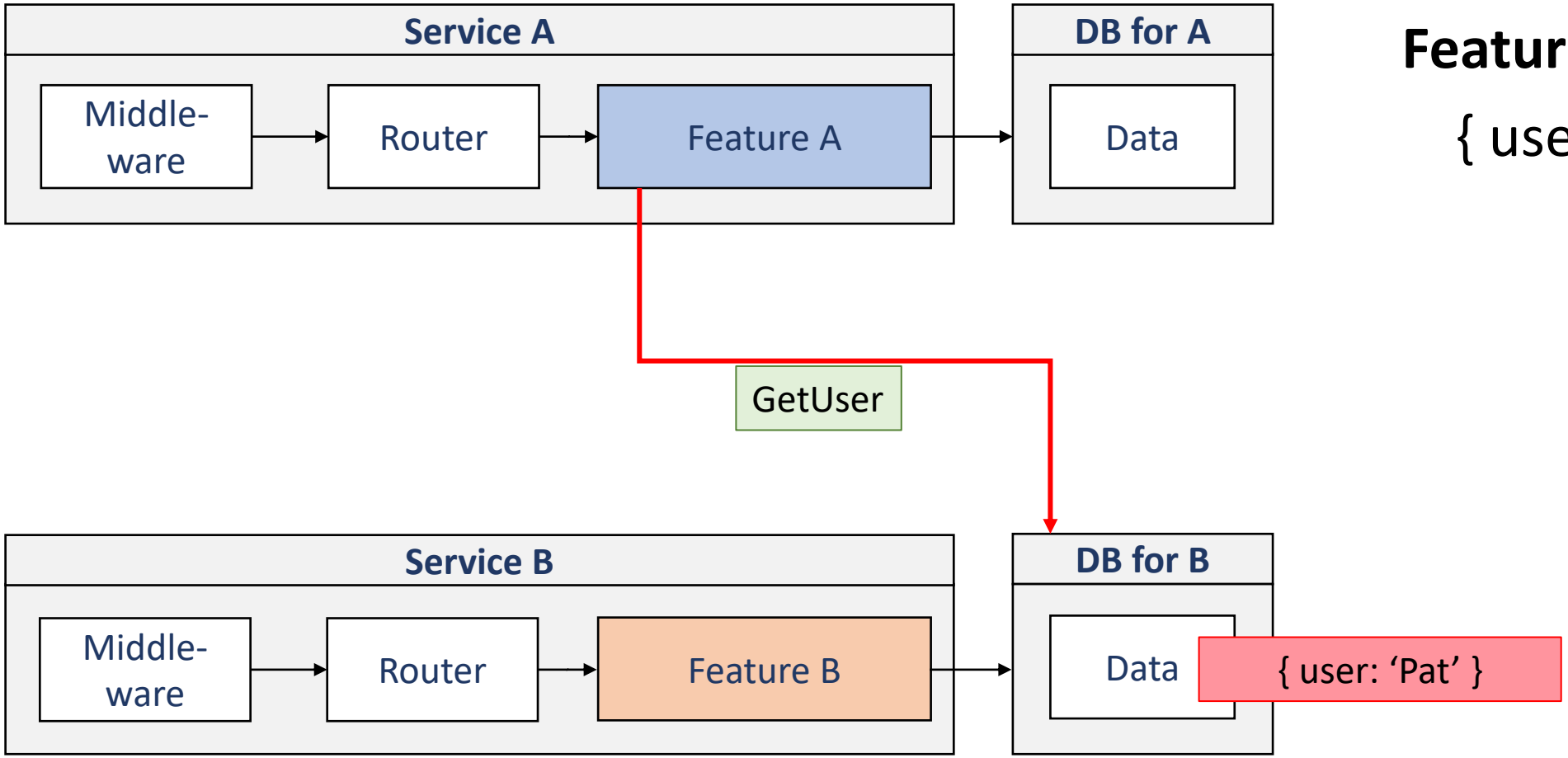


Feature A expects

{ user: *string* }



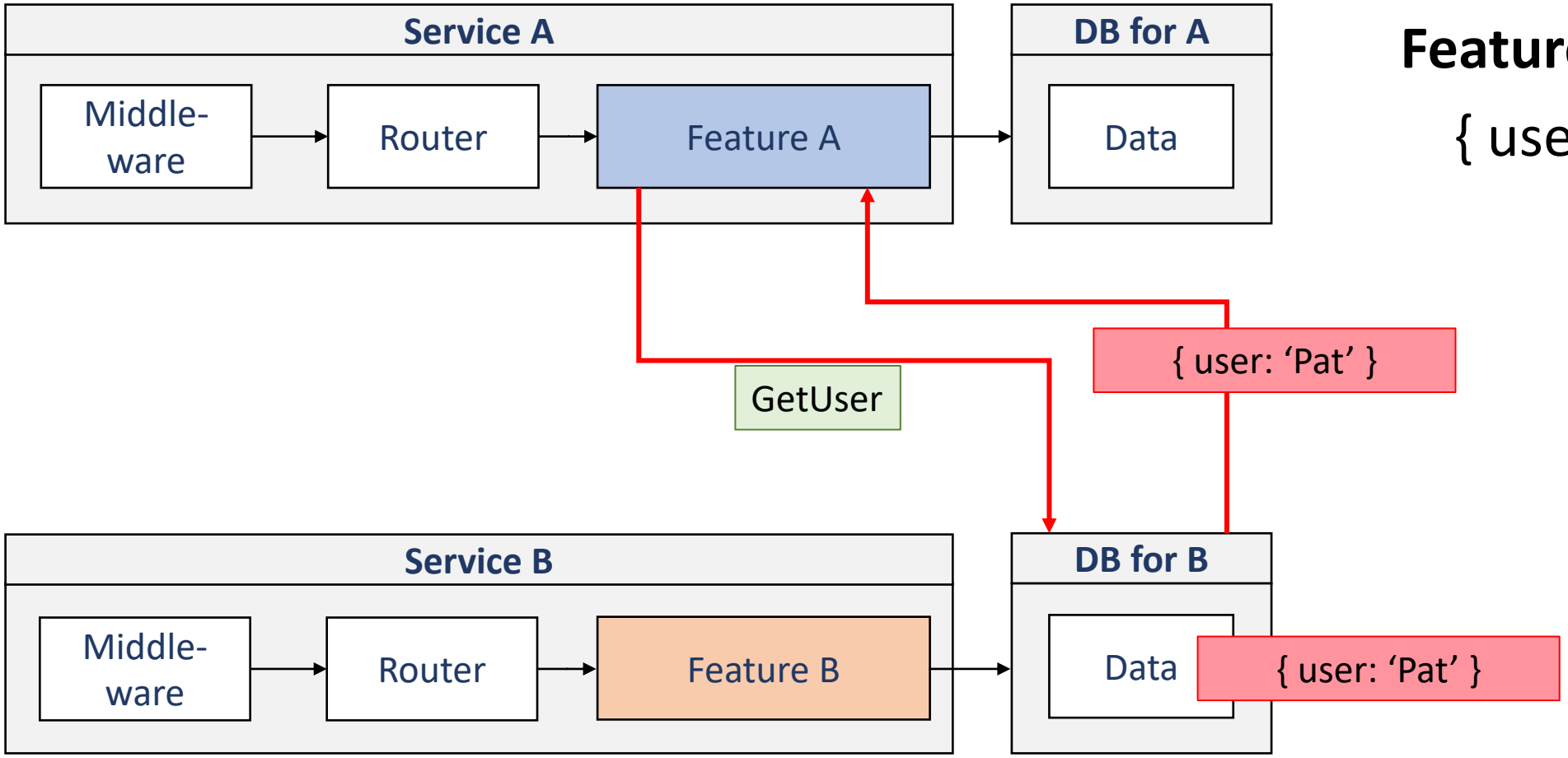
Micro-Service Architecture



Feature A expects

{ user: *string* }

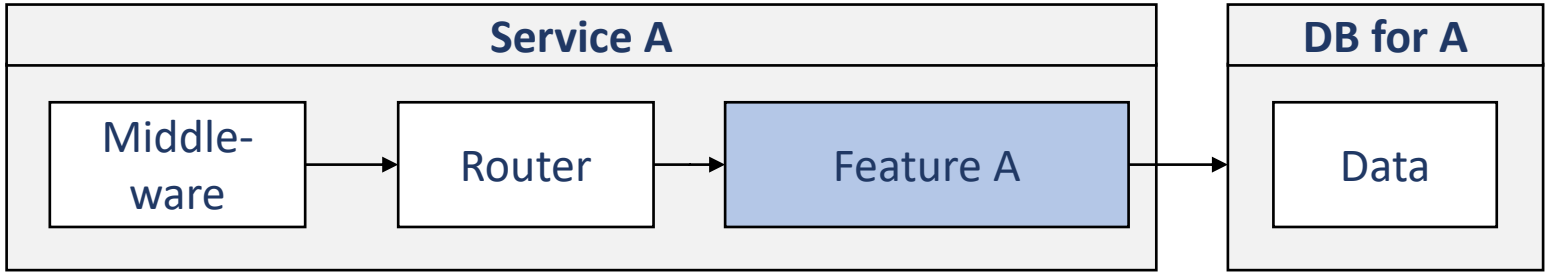
Micro-Service Architecture



Feature A expects

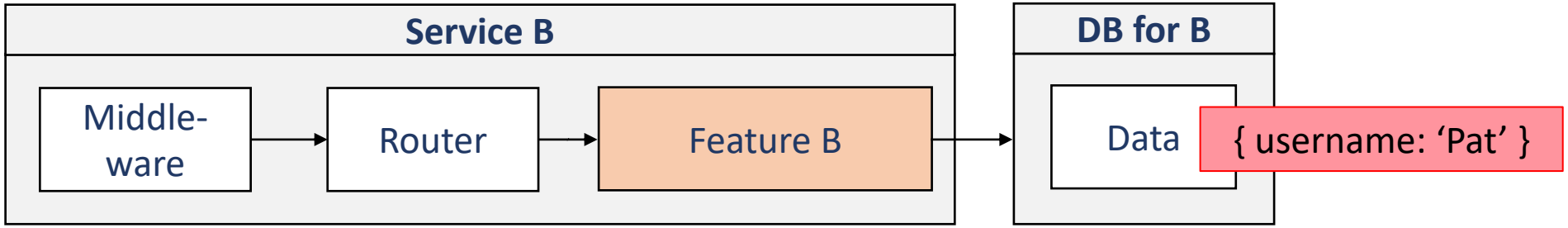
{ user: *string* }

Micro-Service Architecture

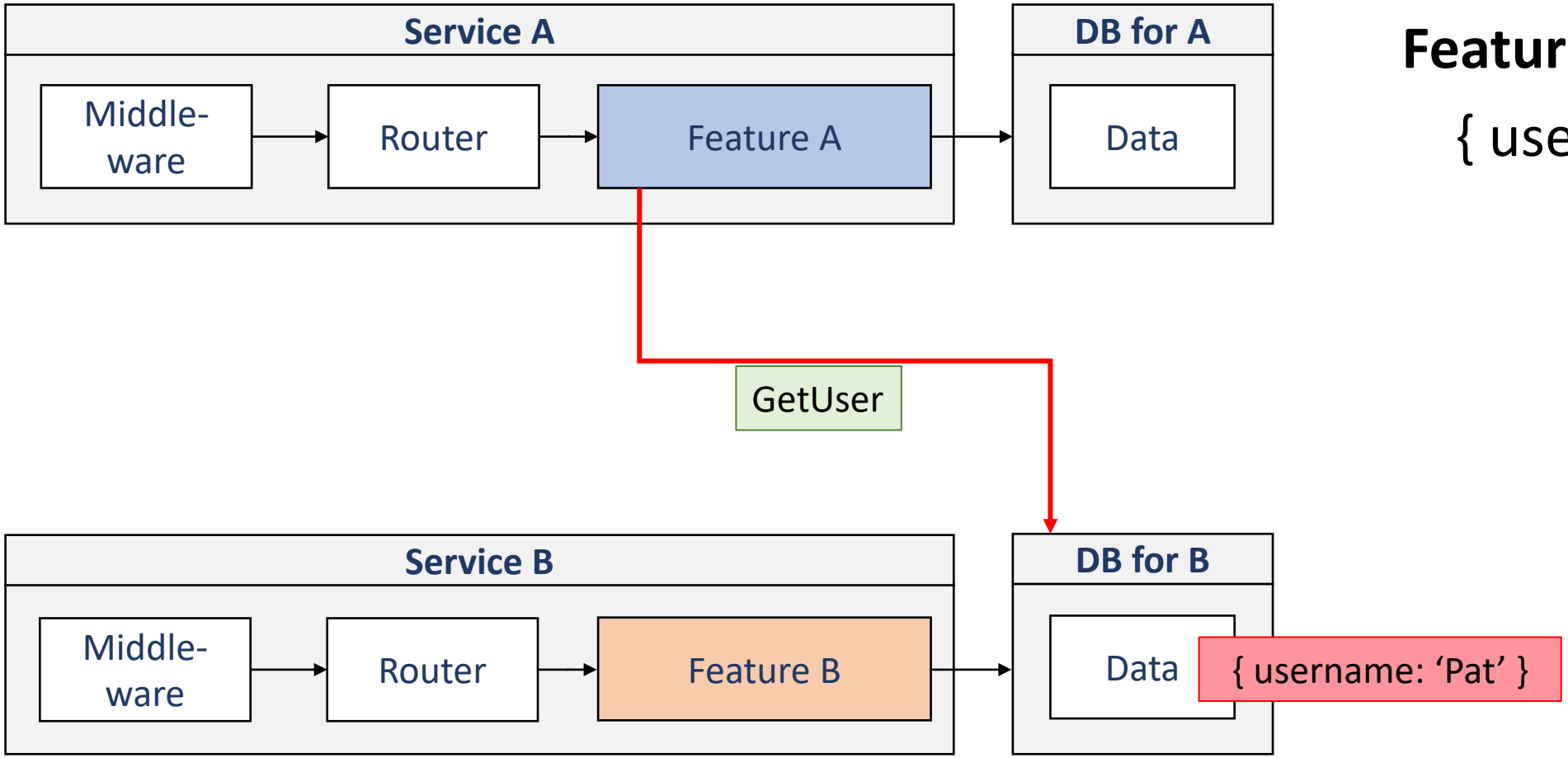


Feature A expects

{ user: *string* }



Micro-Service Architecture



Feature A expects

{ user: *string* }

Micro-Servi ture

Feature A expects

{ user: *string* }

Basic e-commerce Application

Now that we have all of that covered, let us look at an actual application.

For that, let us switch over into a tool that is great for diagramming.

Group-Based Exercise

- See GROUP.md in course material.