

Decomposing Data Within Distributed Systems



Mark Richards

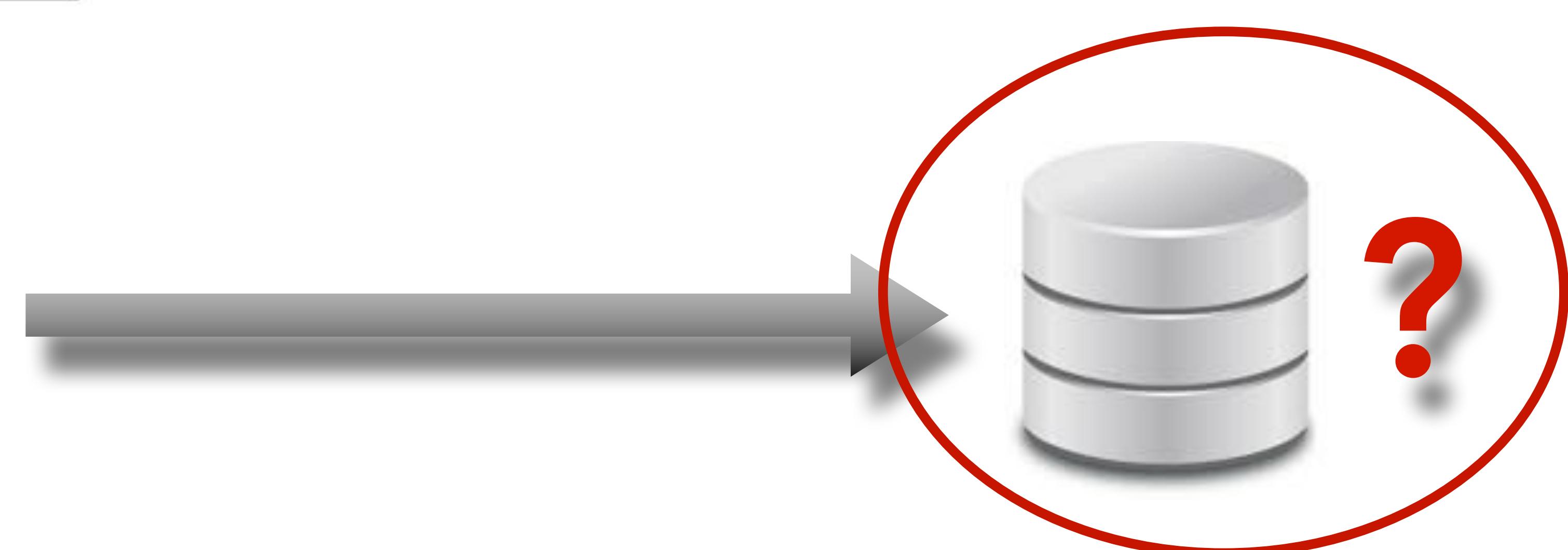
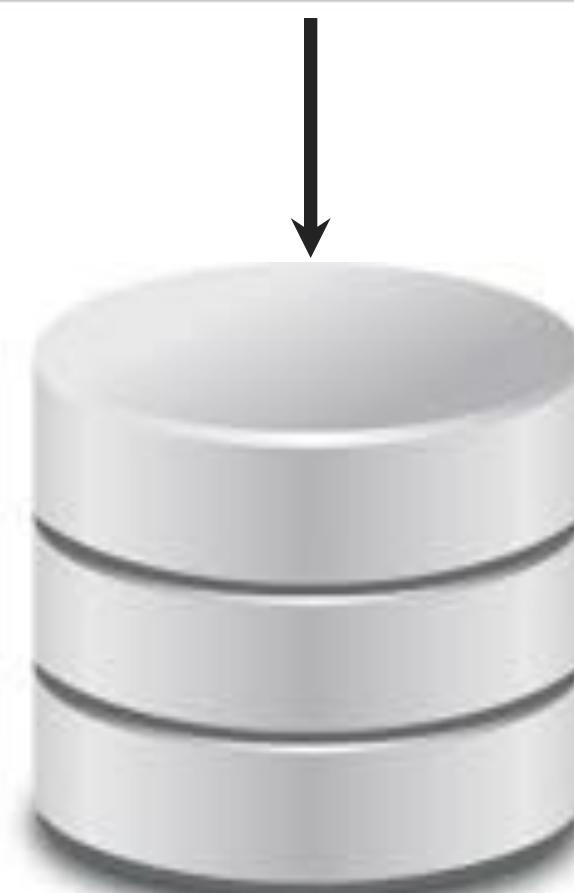
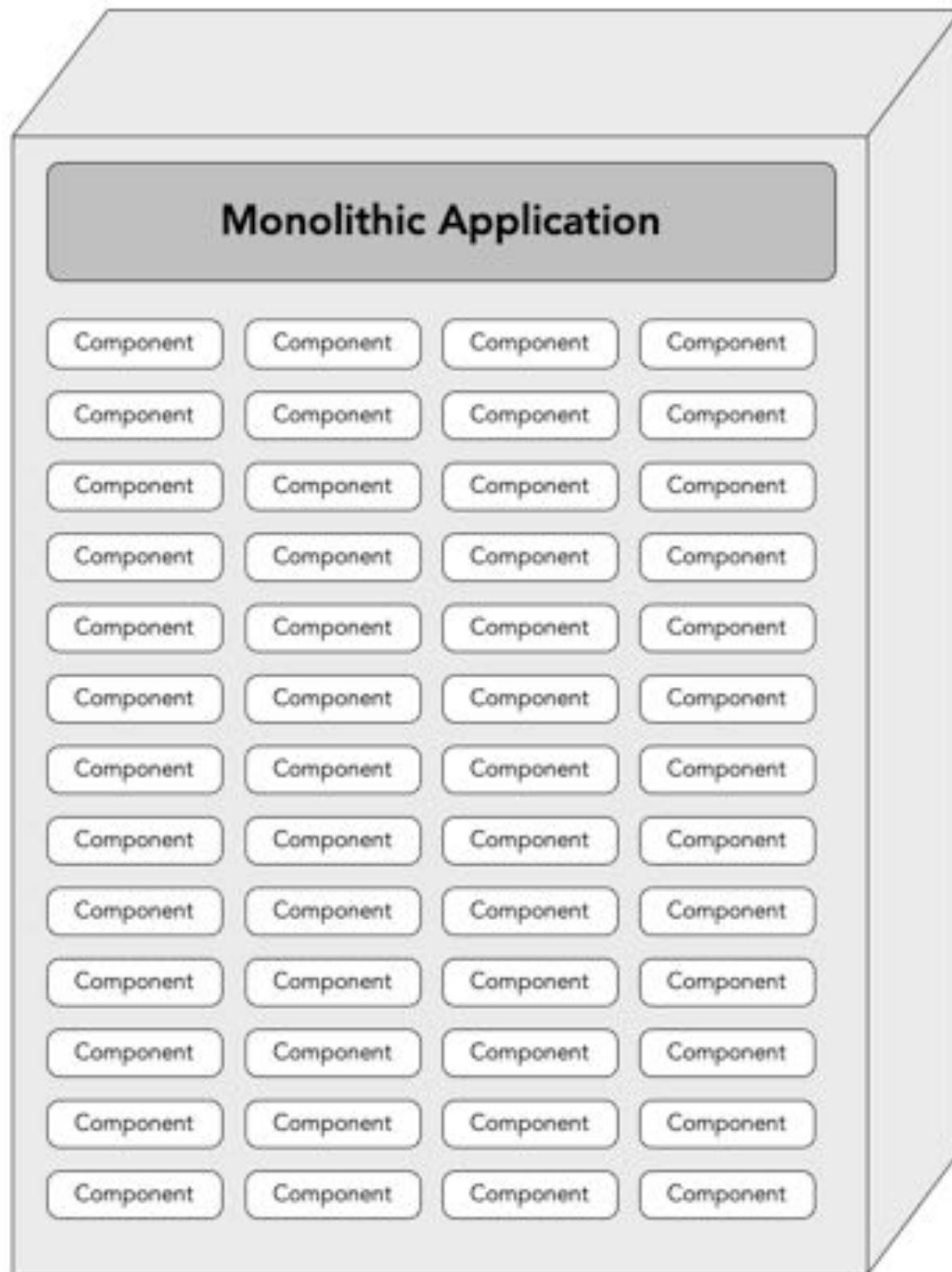
Independent Consultant

Hands-on Software Architect / Published Author

Founder, DeveloperToArchitect.com

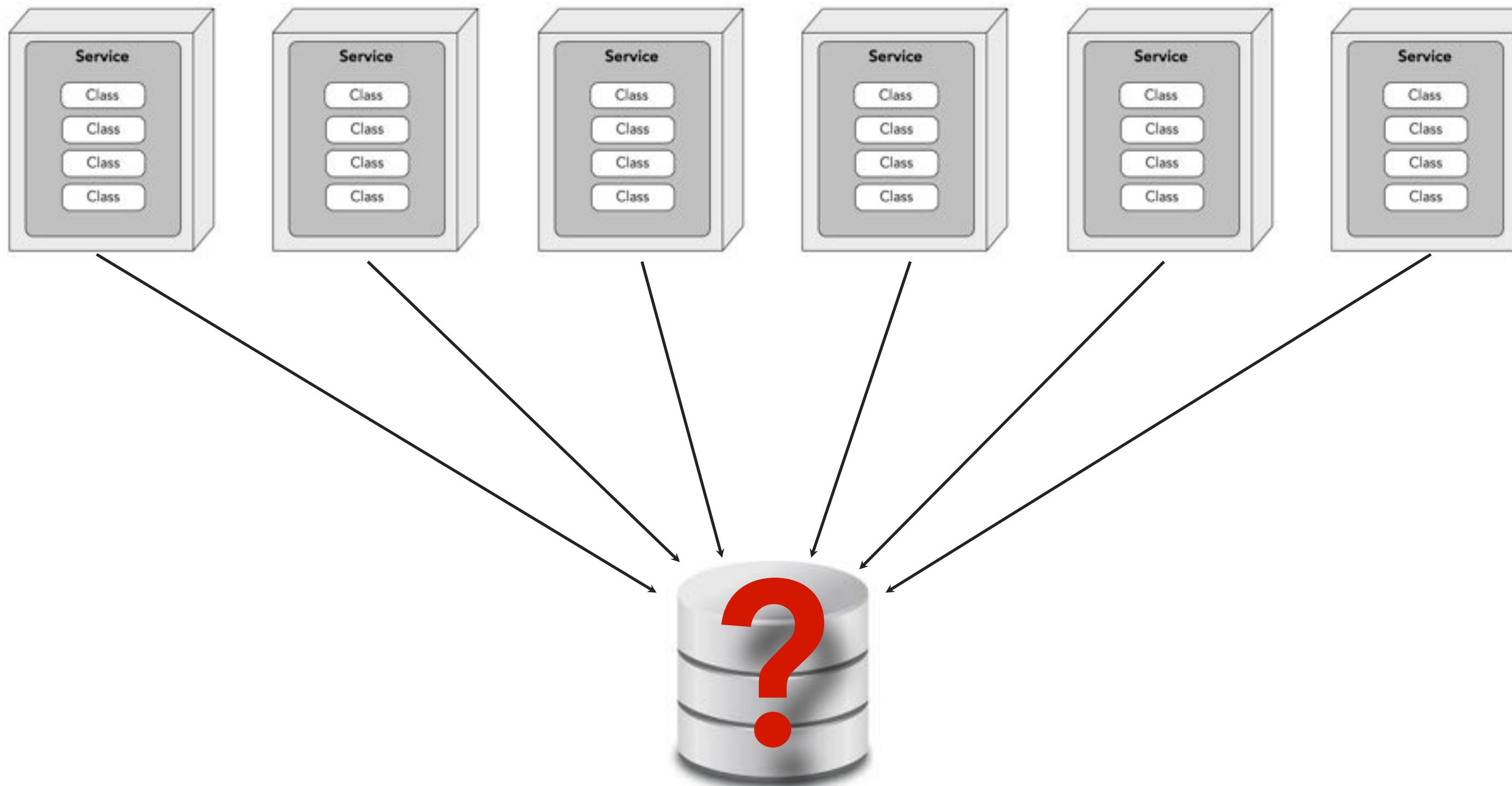
<https://www.linkedin.com/in/markrichards3>

@markrichardssa



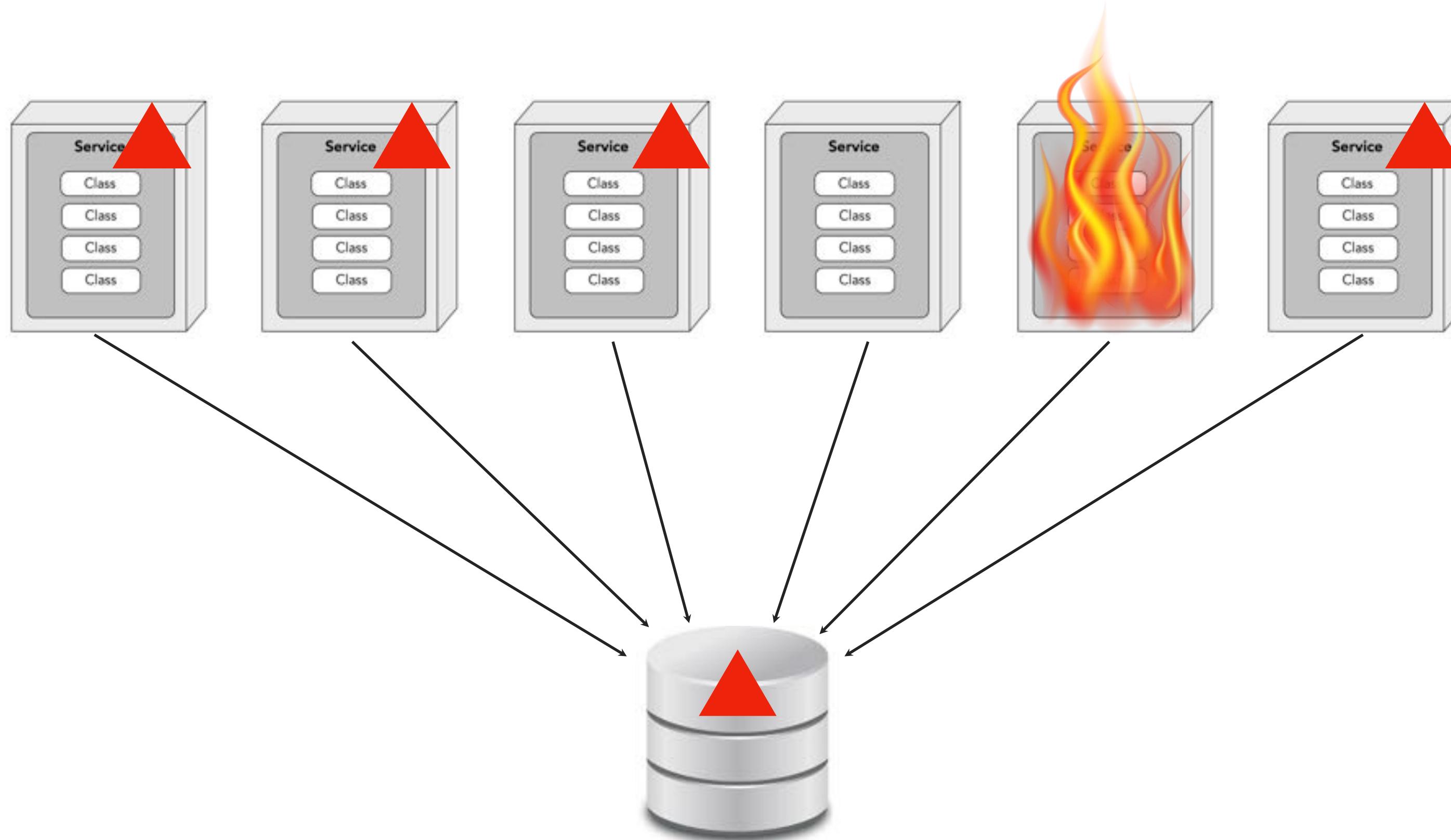
decomposing data

“when should I consider breaking apart my data?”



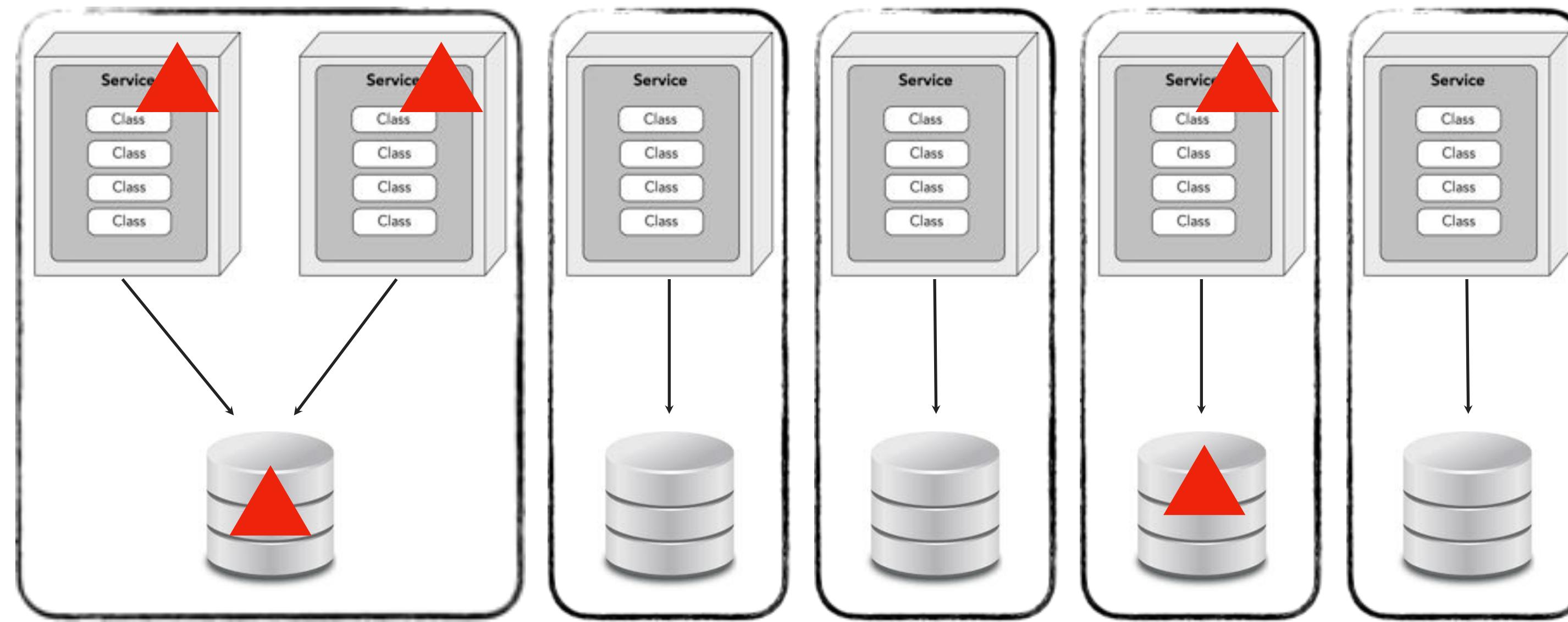
decomposing data

change control



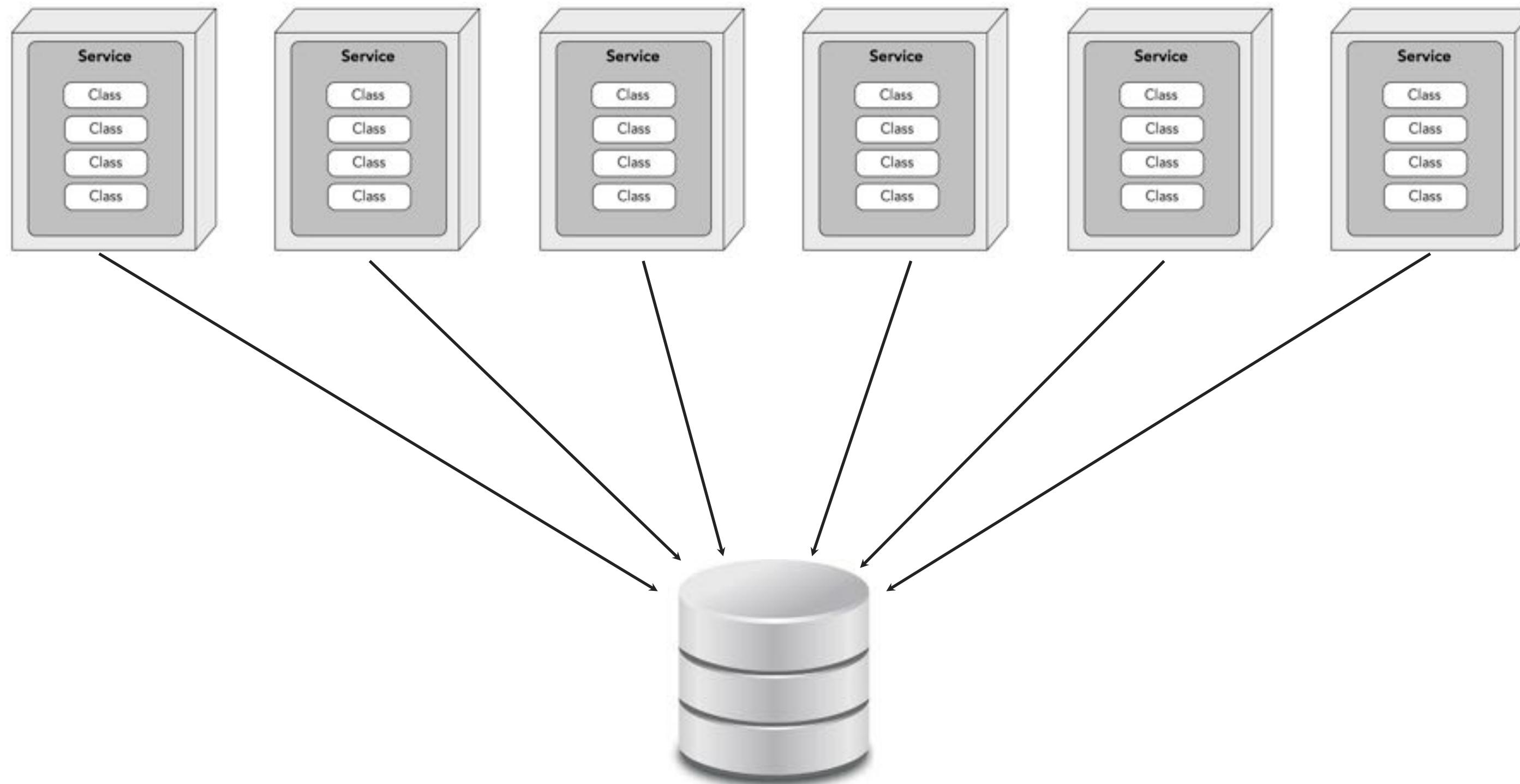
decomposing data

change control



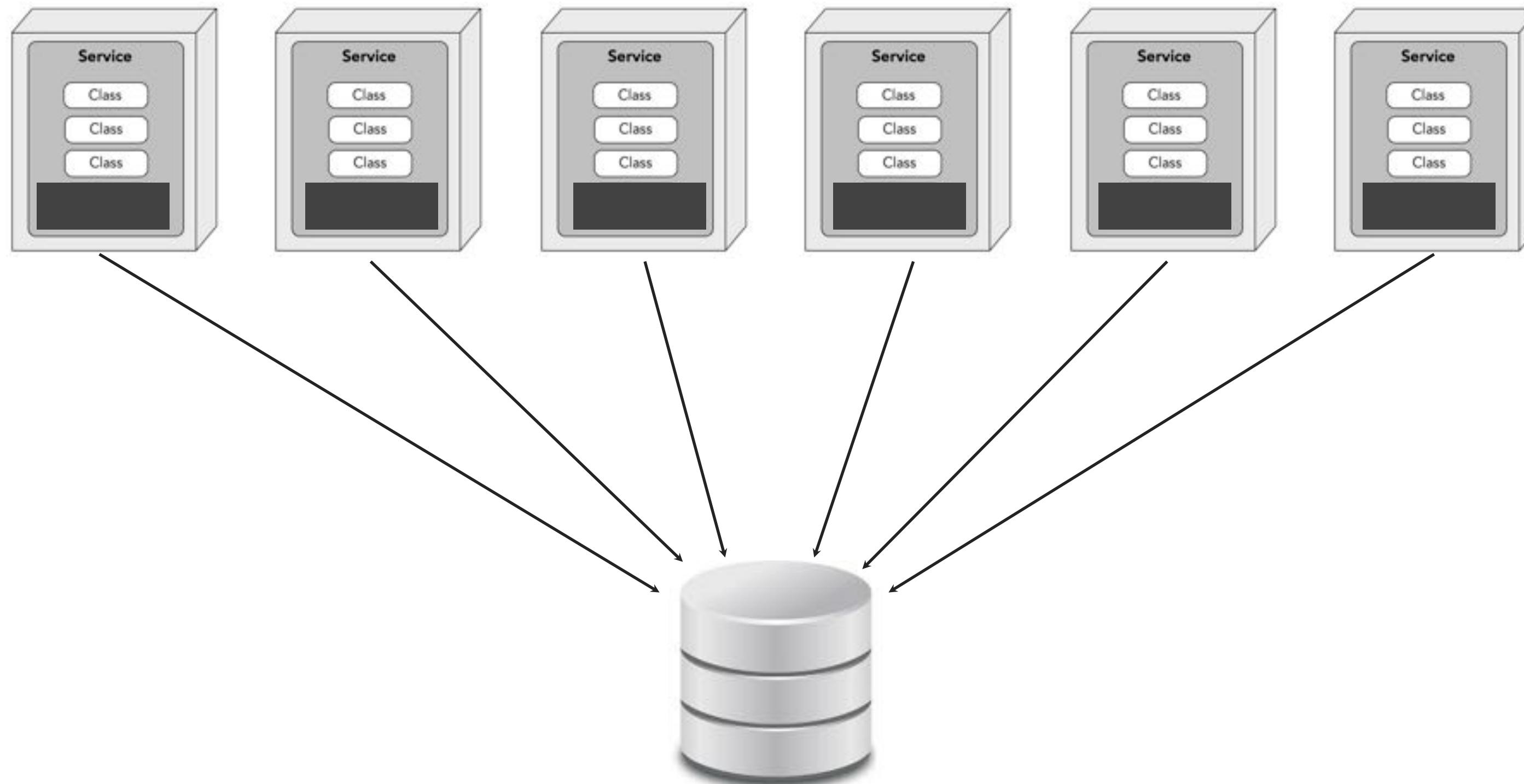
decomposing data

database connections



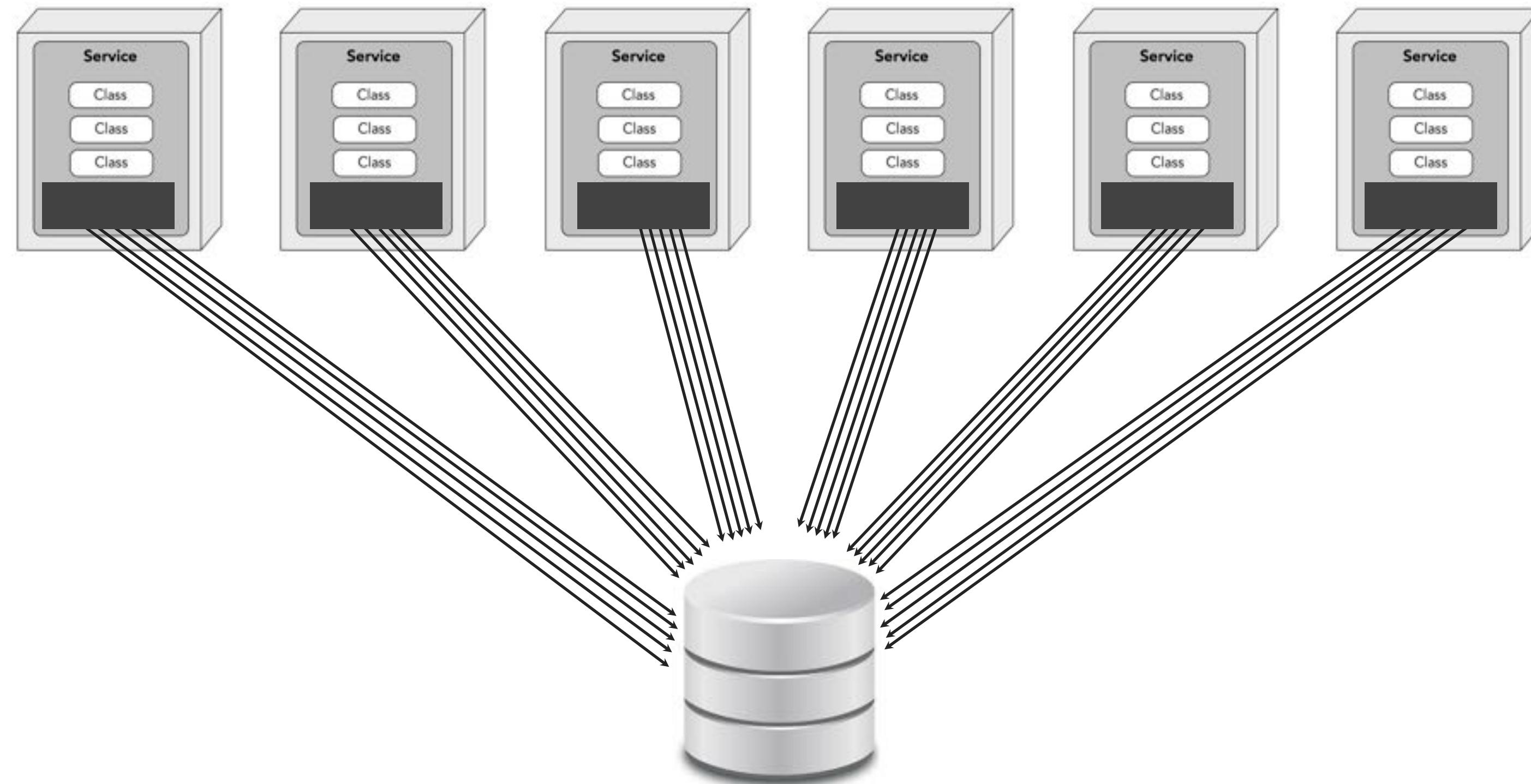
decomposing data

database connections



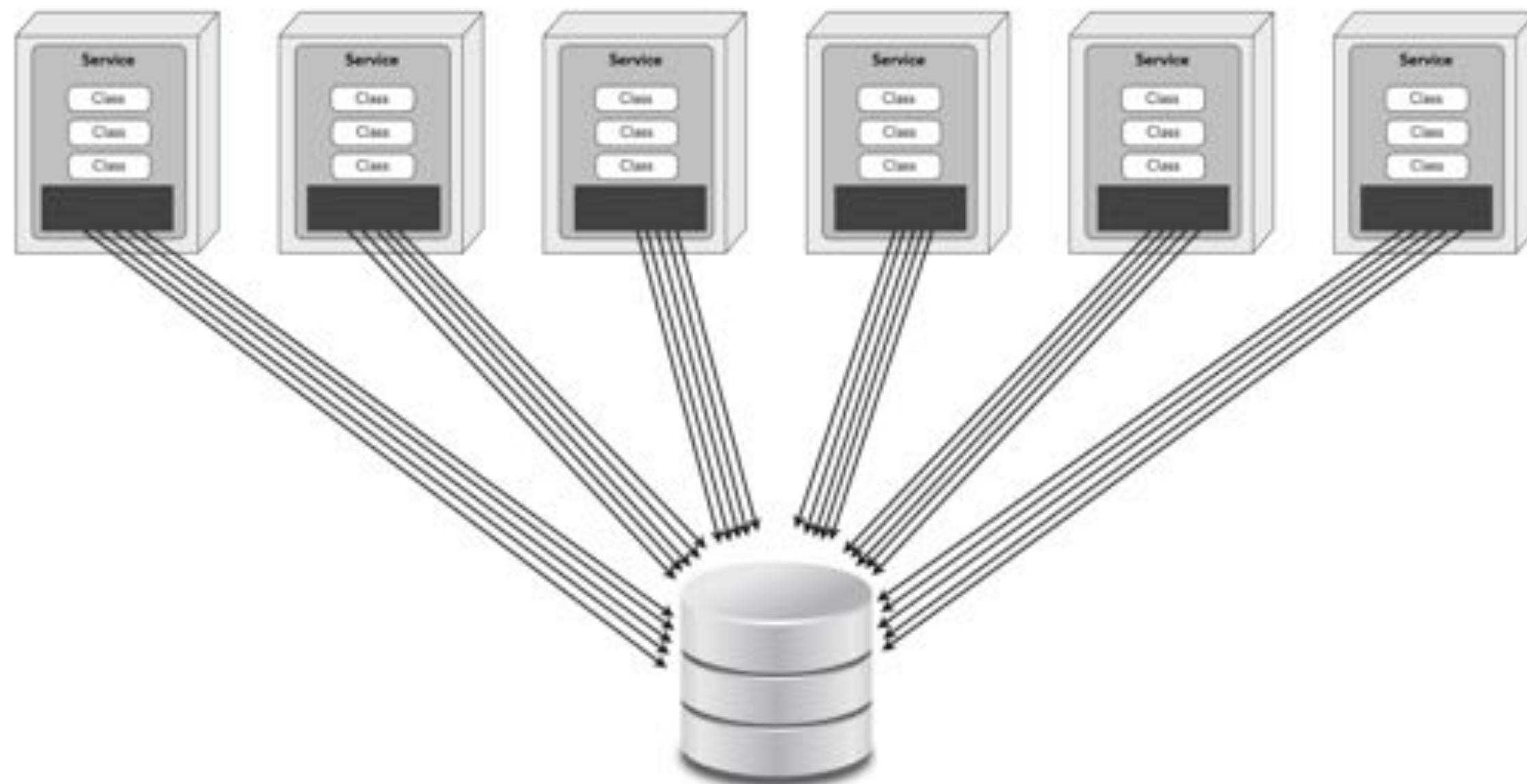
decomposing data

database connections



decomposing data

database connections



monolithic application: 200 connections

distributed services: 50

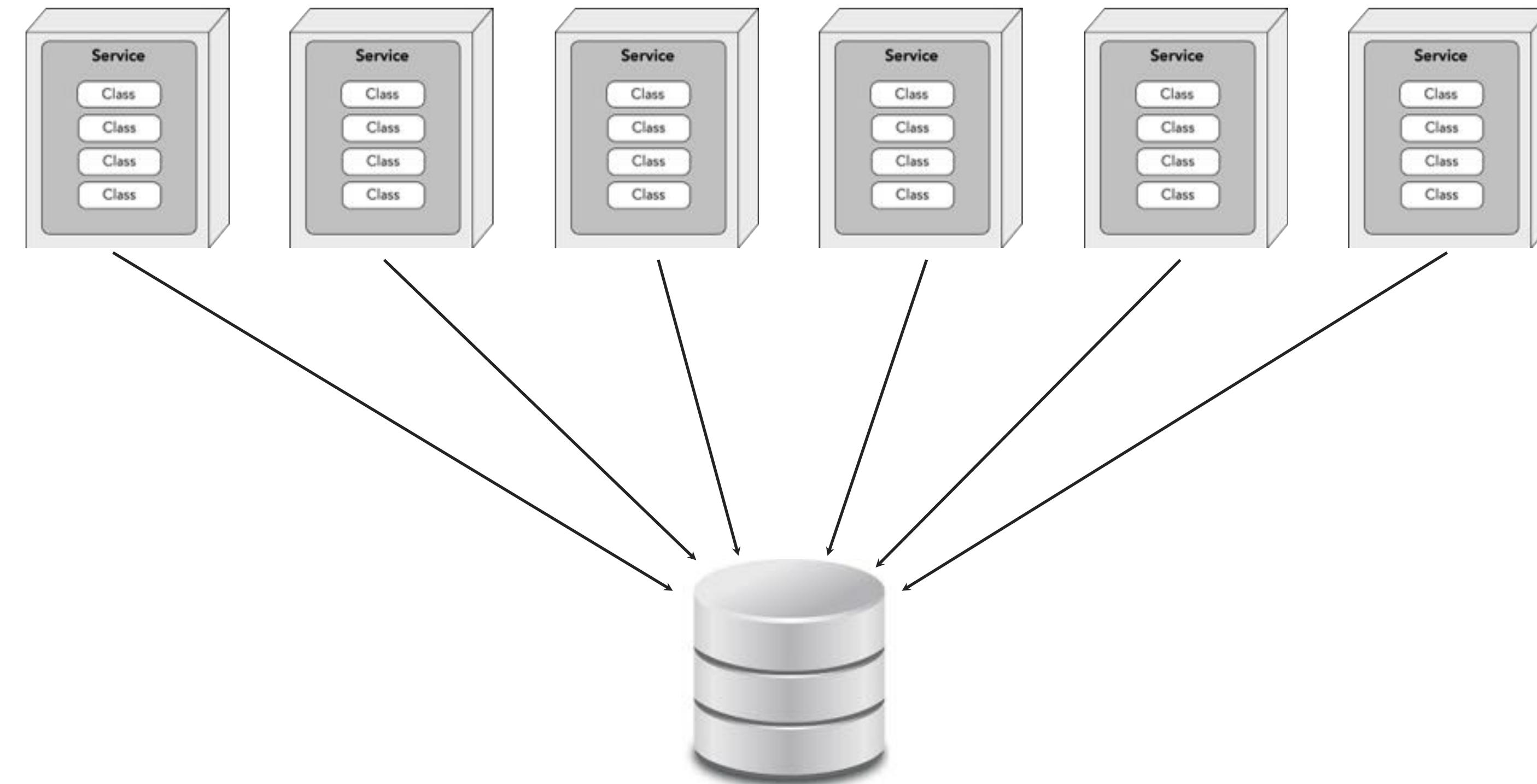
connections per service: 10

service instances (min): 2

distributed connections: 1000 connections

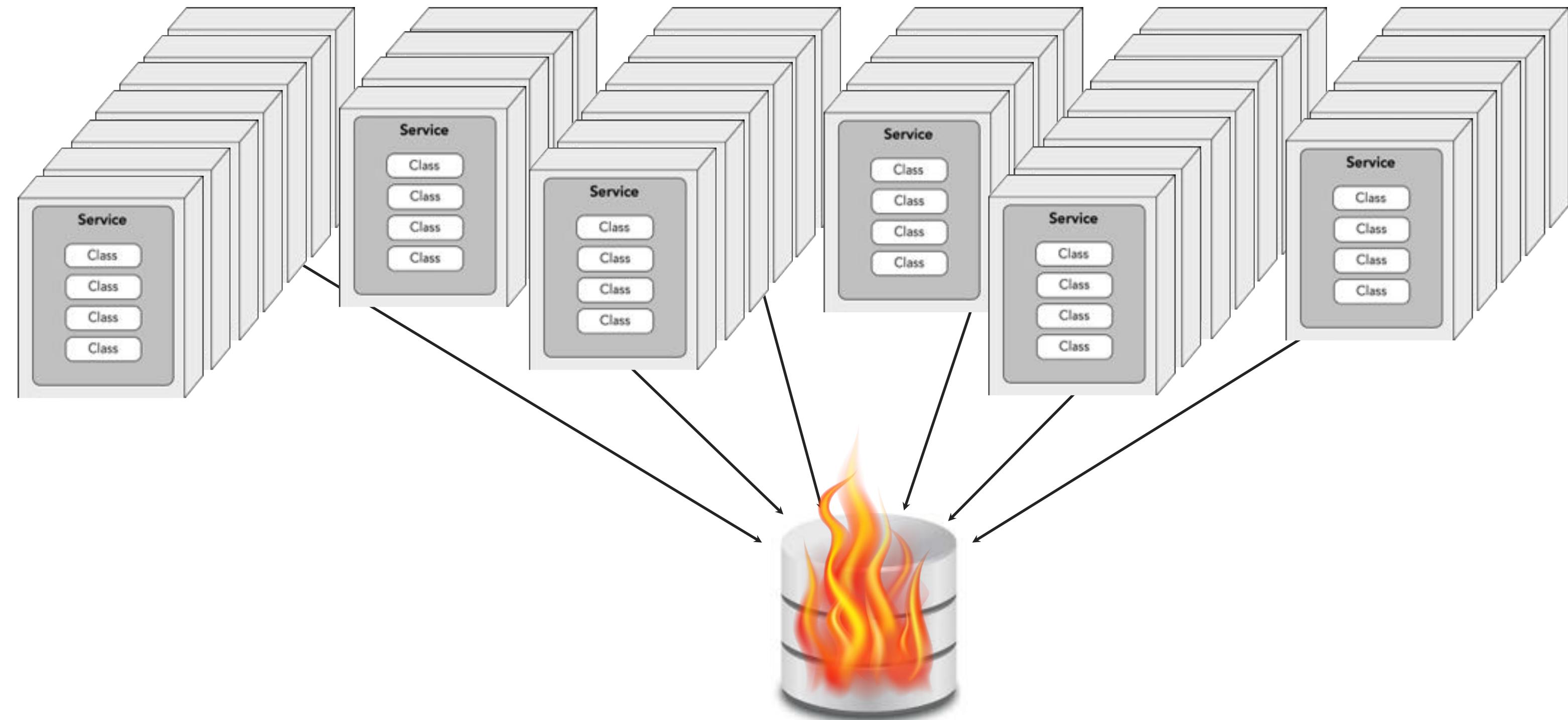
decomposing data

database scalability



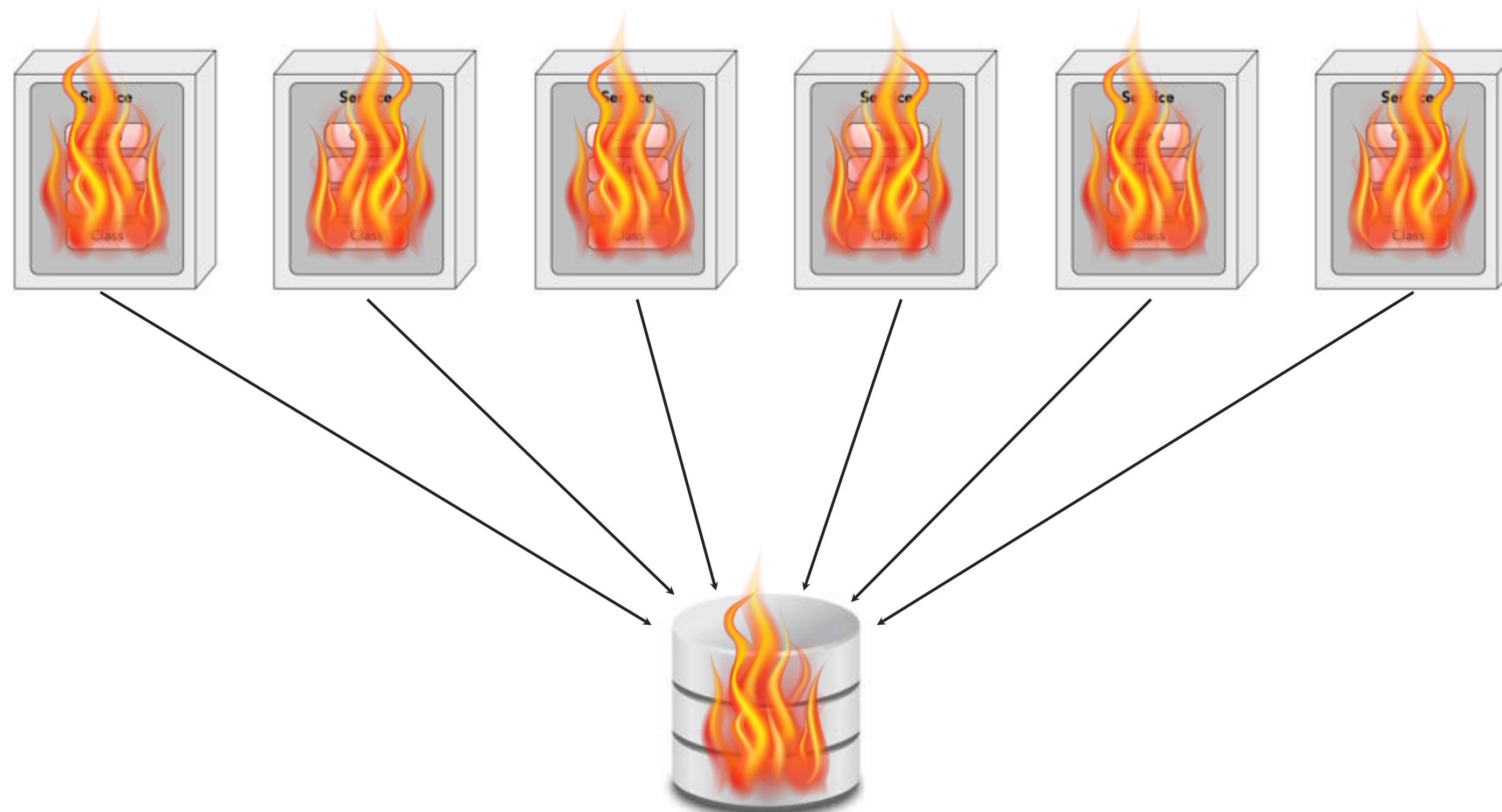
decomposing data

database scalability

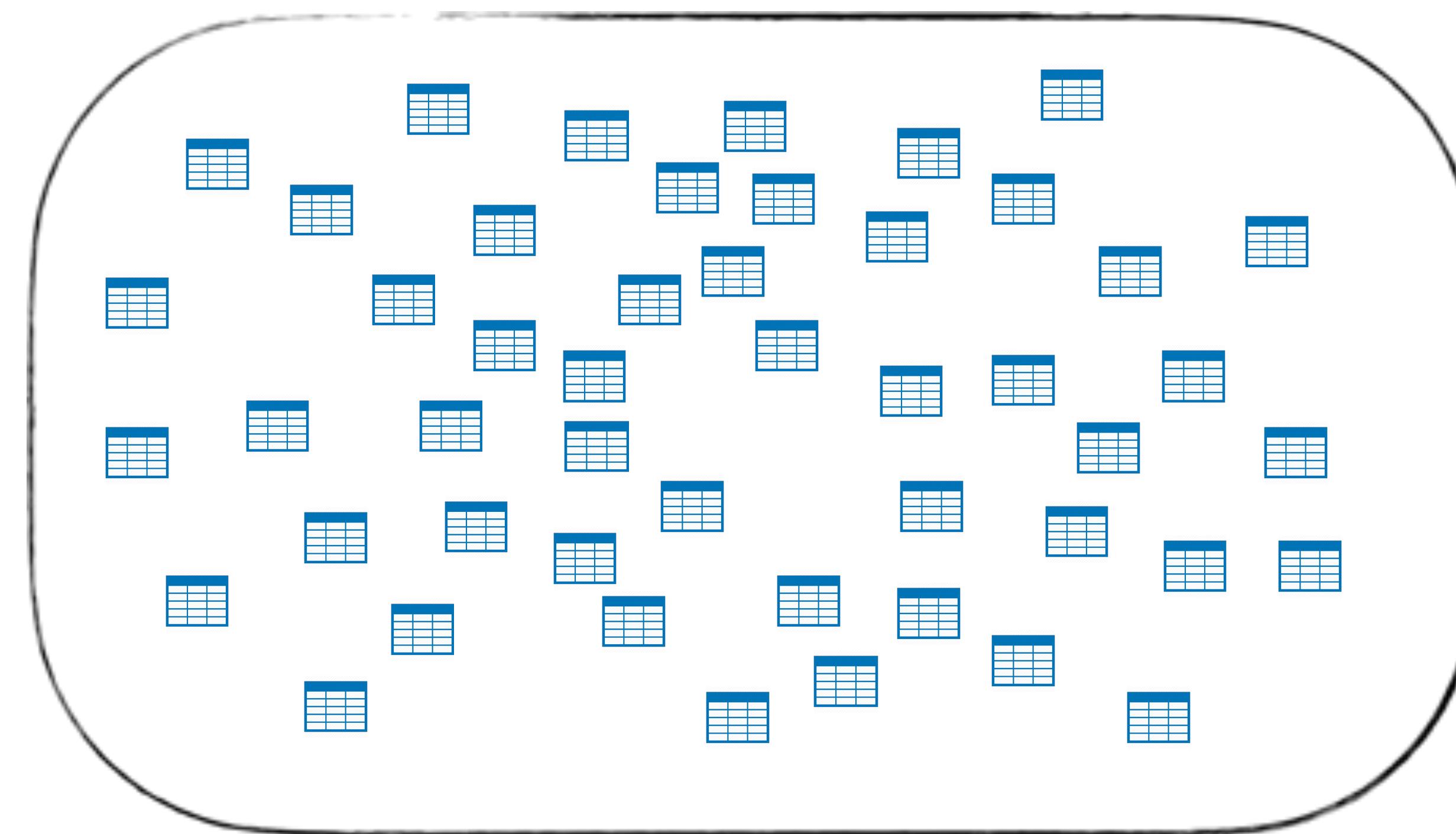


decomposing data

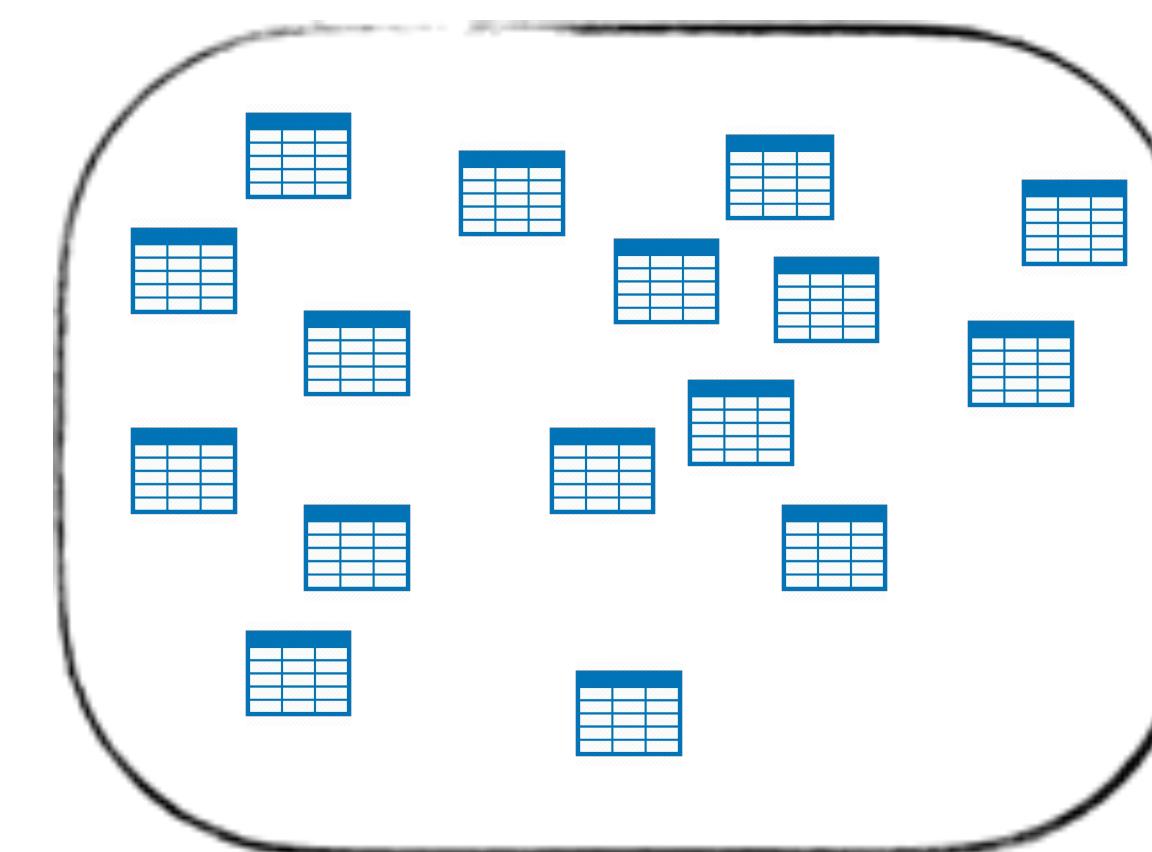
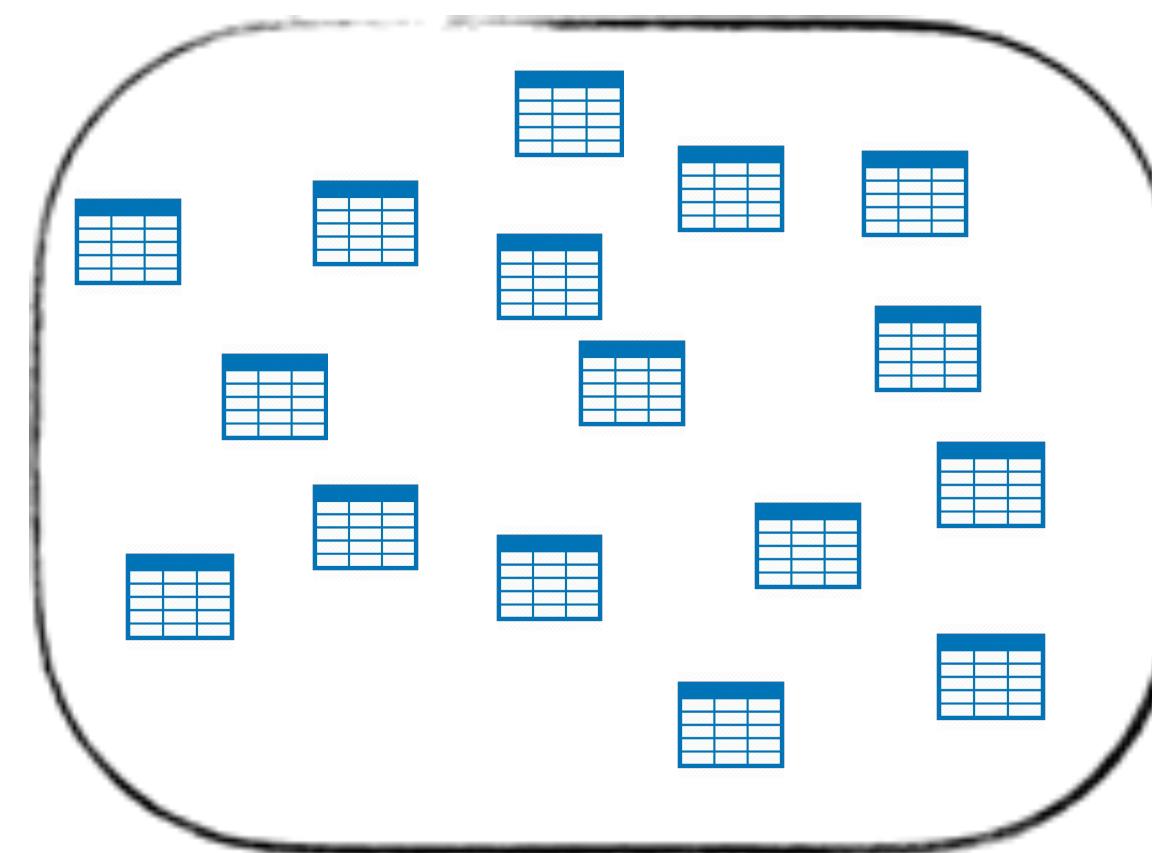
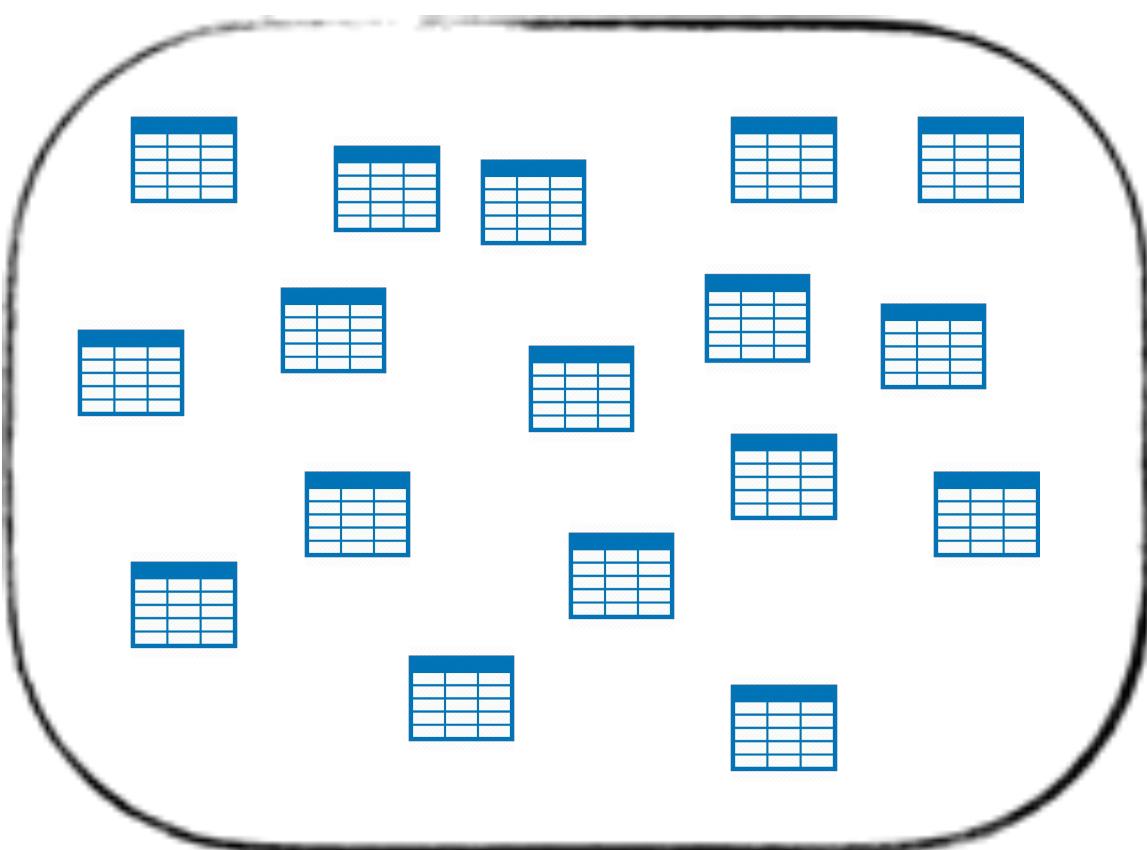
fault tolerance



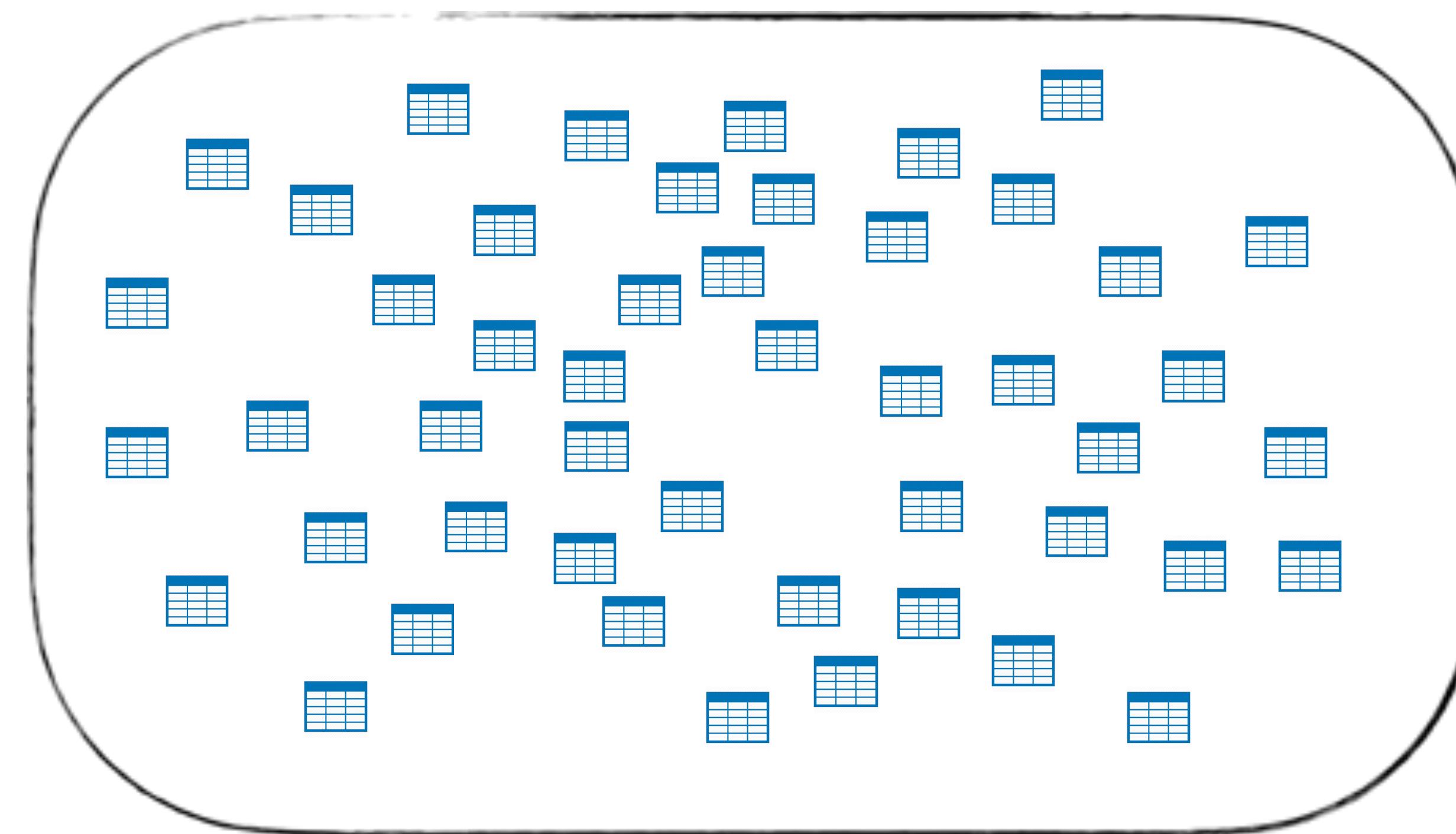
decomposing data



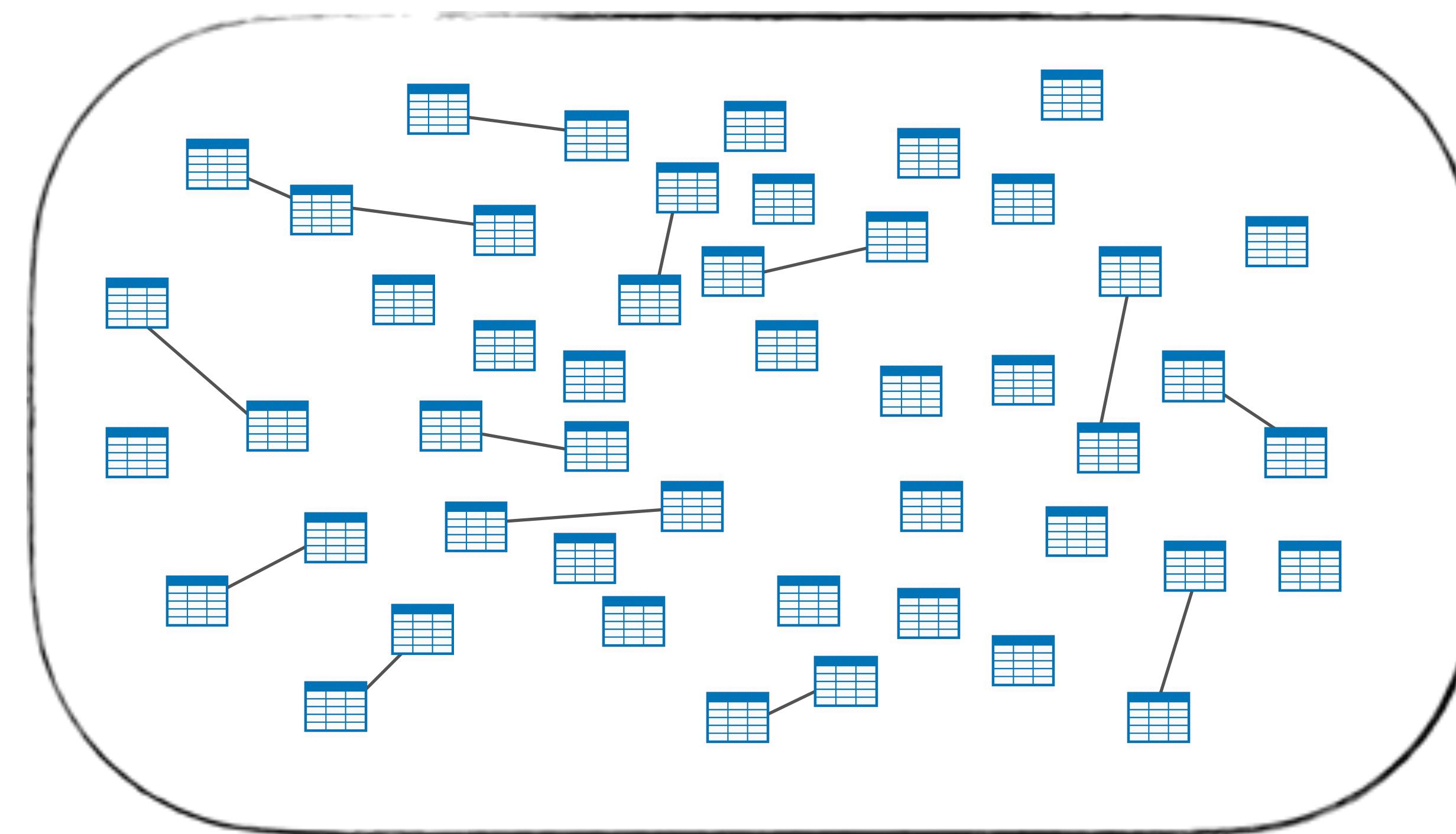
decomposing data



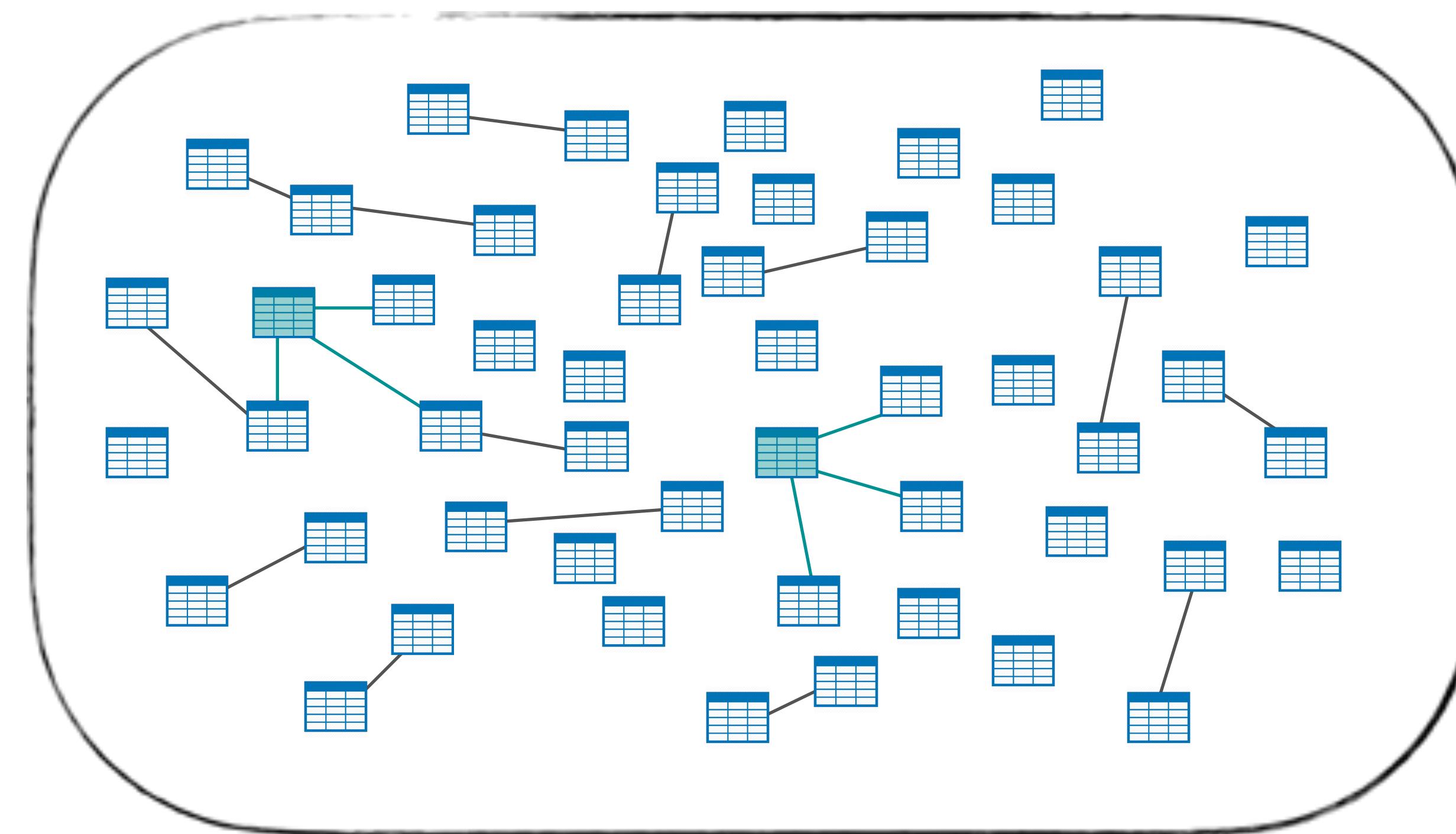
decomposing data



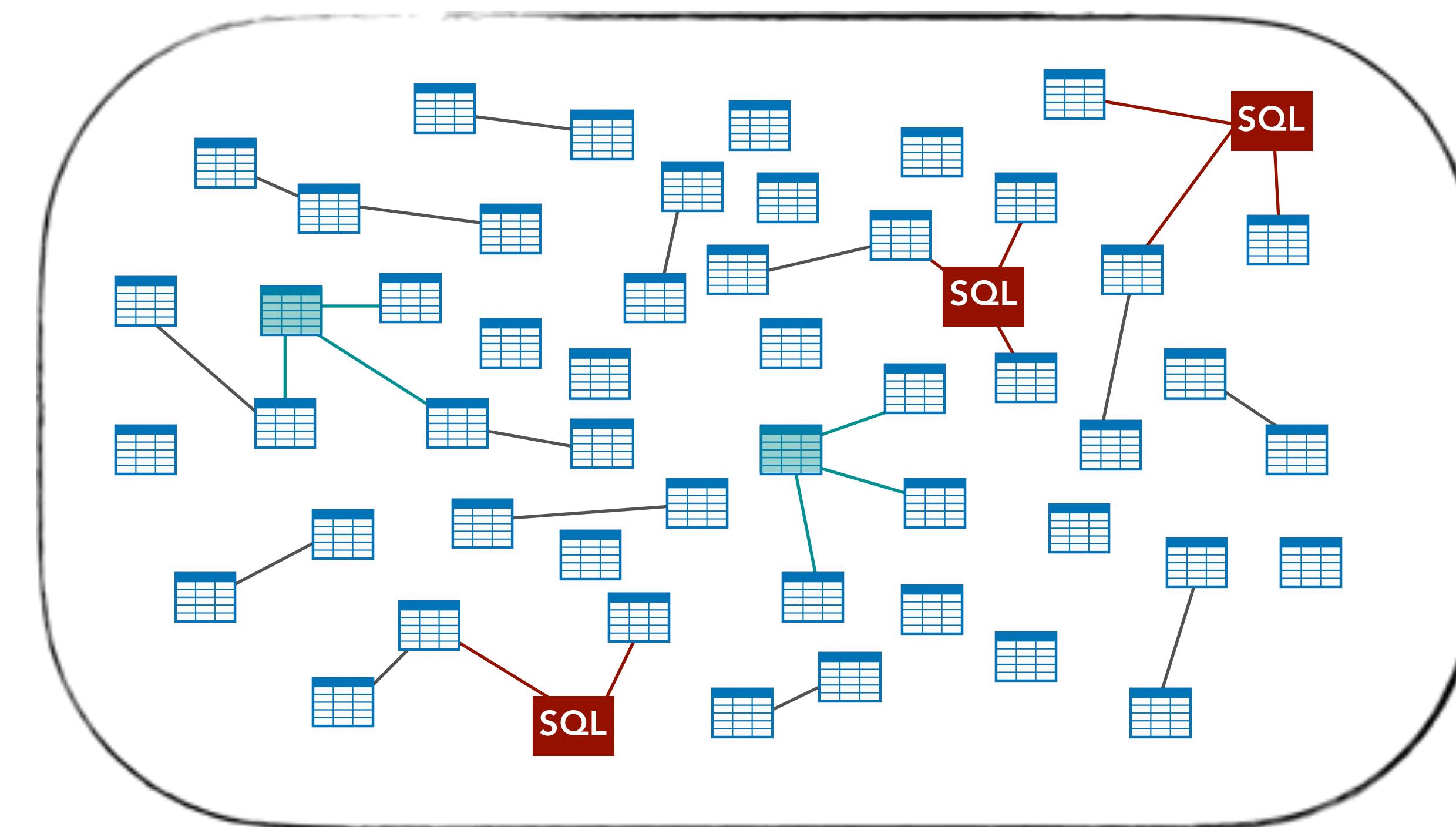
decomposing data



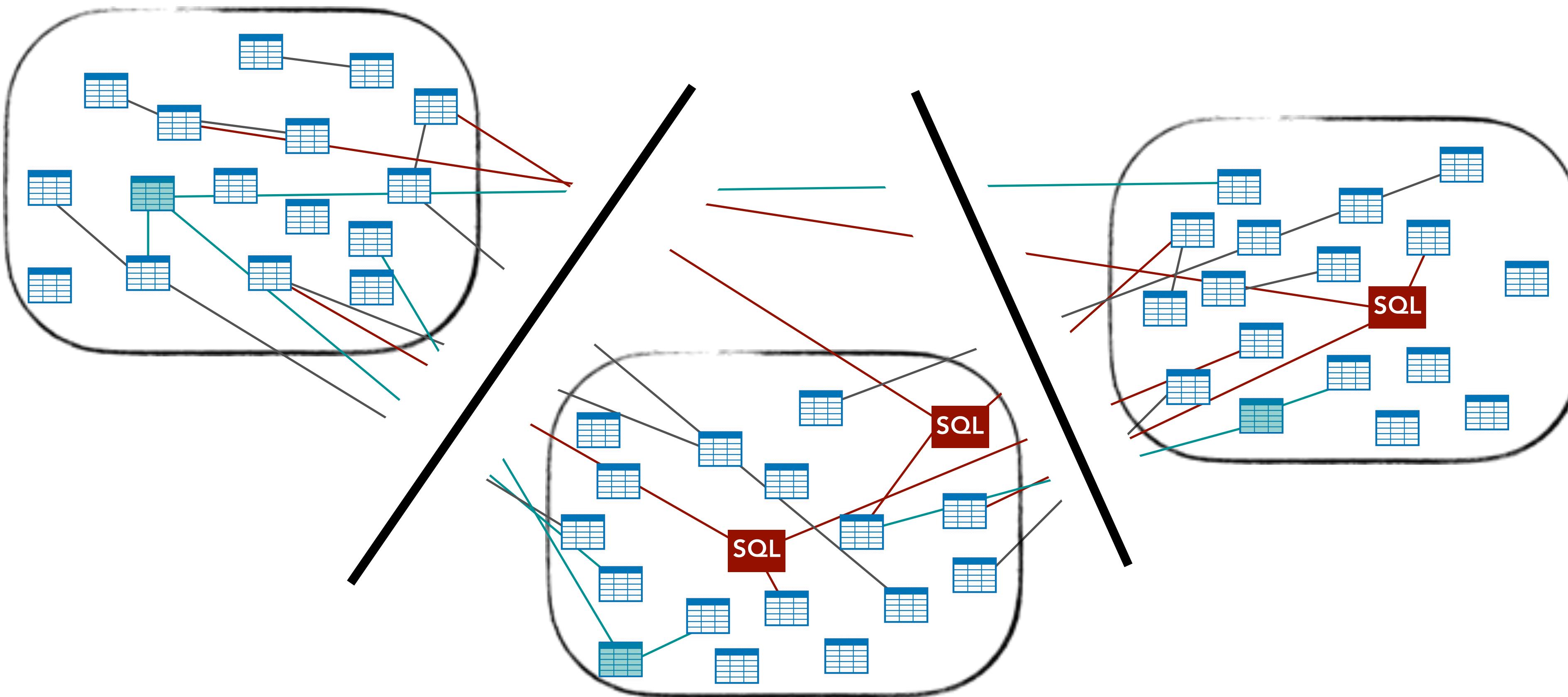
decomposing data



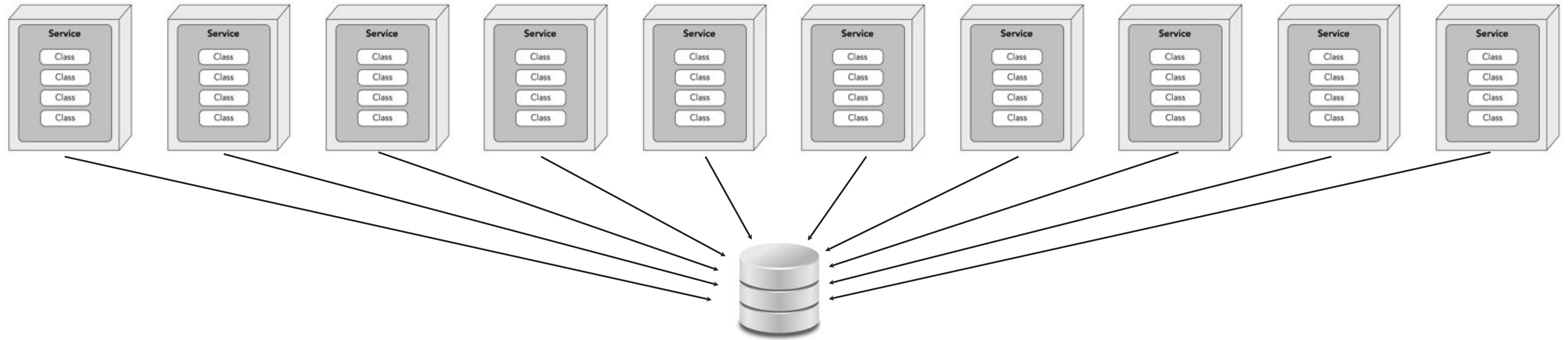
decomposing data



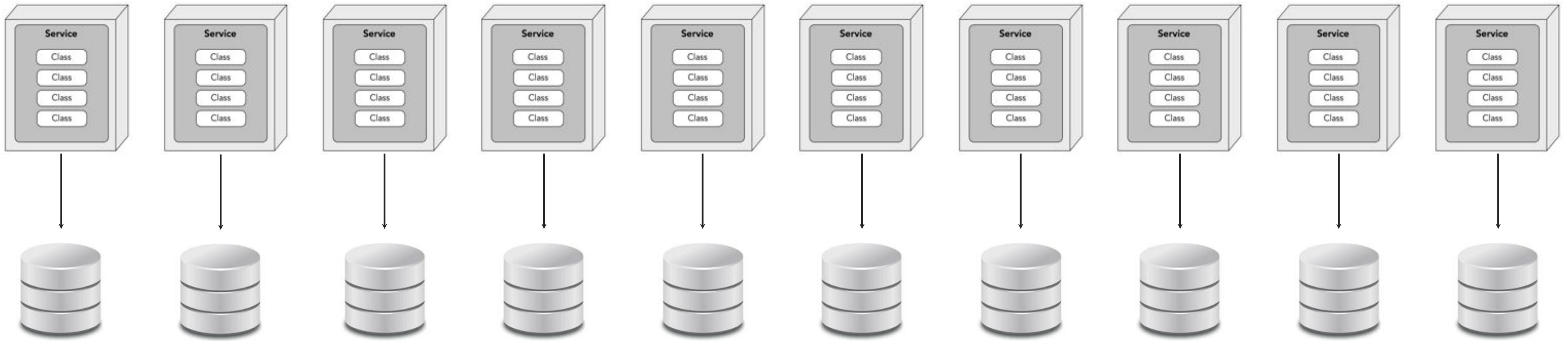
decomposing data



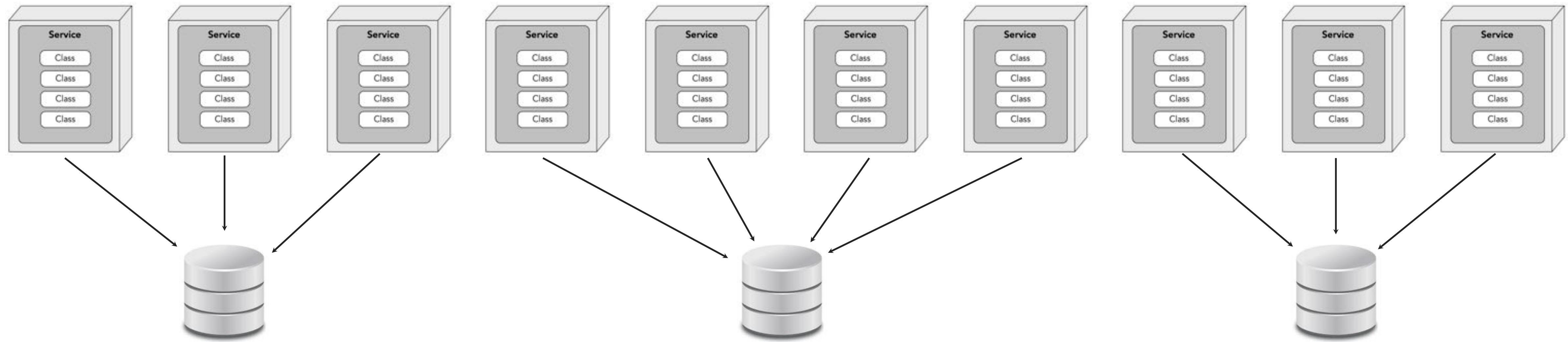
monolithic database pattern



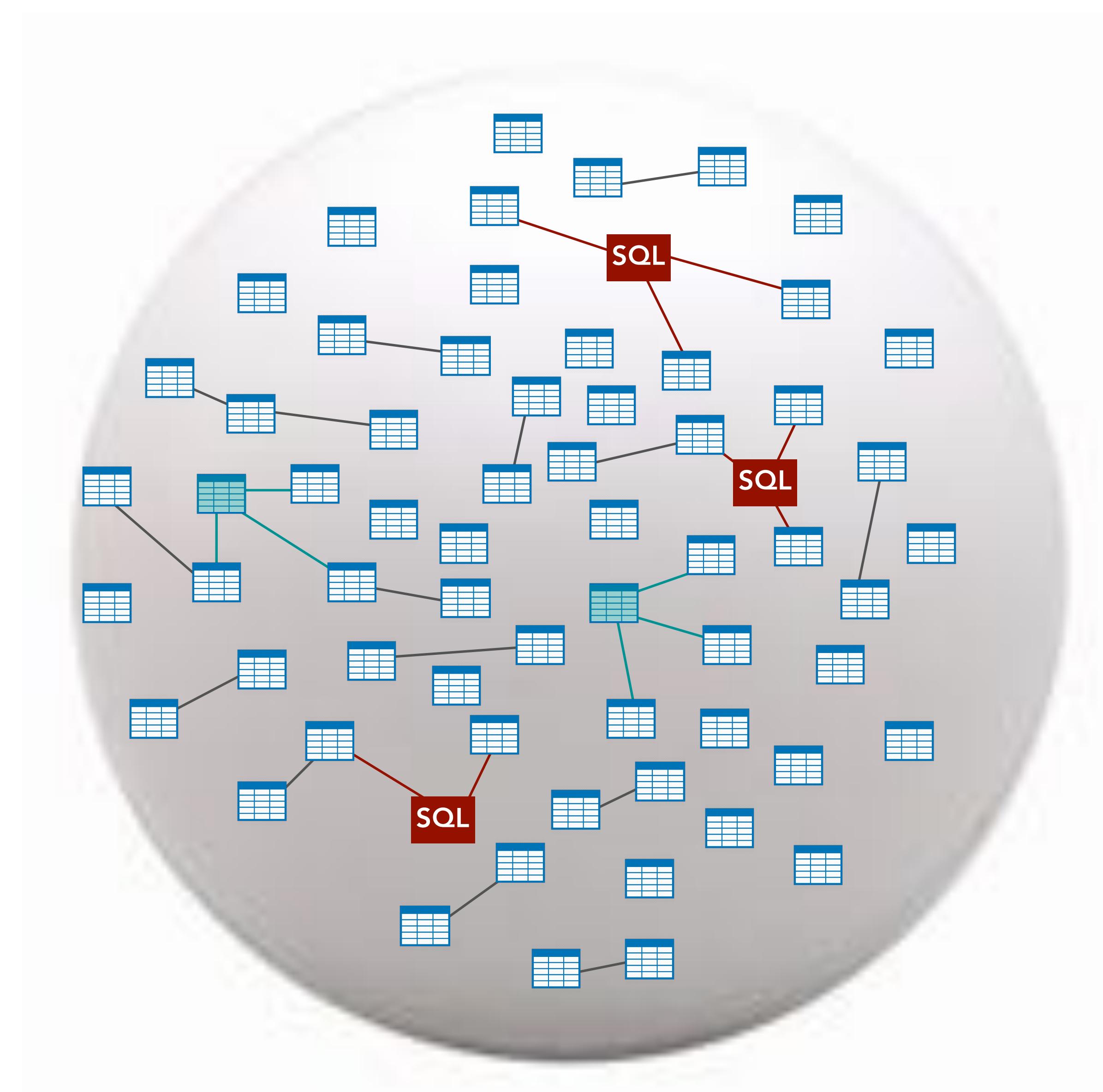
database-per-service pattern



data domain pattern



data domain pattern



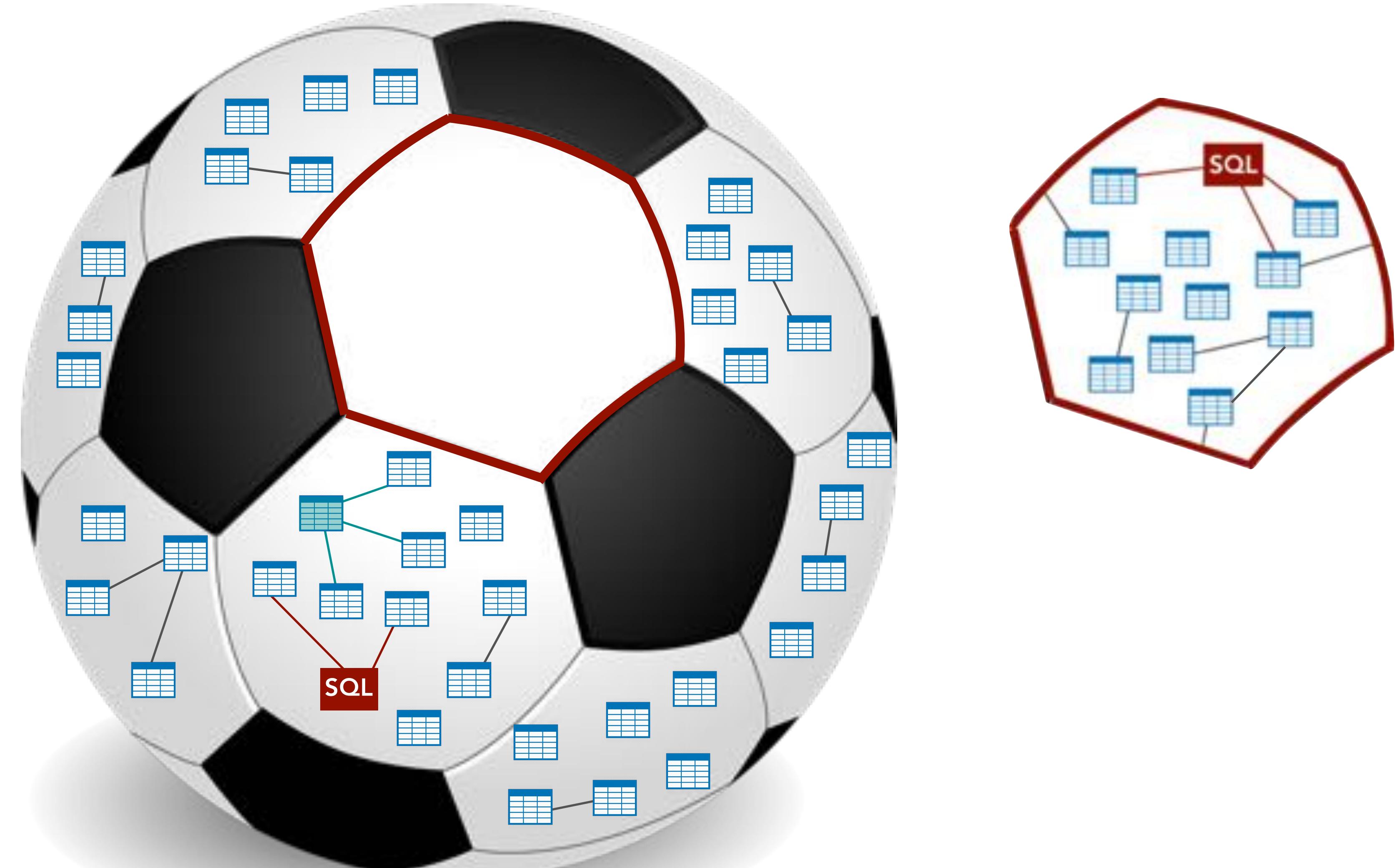
data domain pattern



data domain pattern



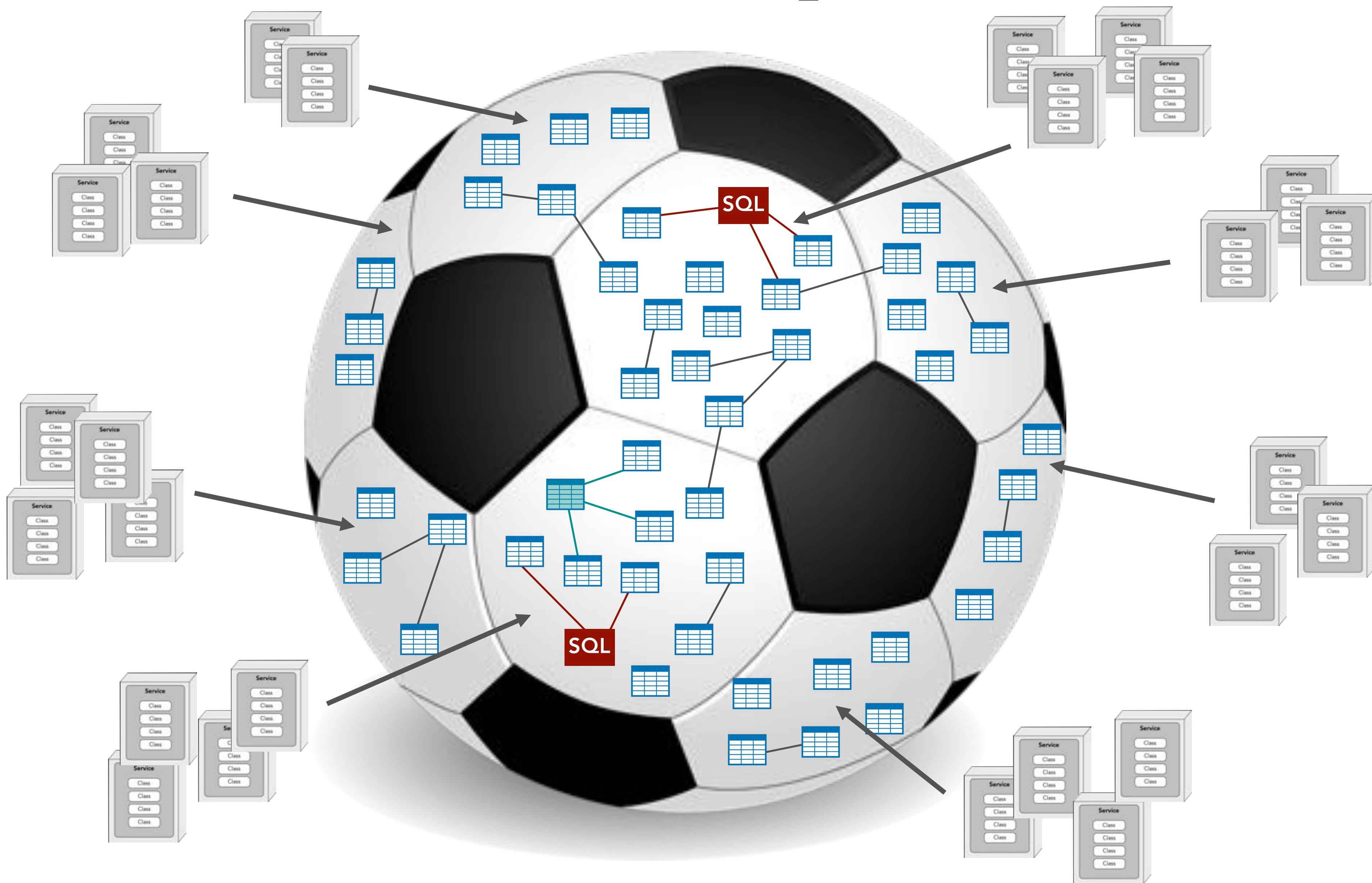
data domain pattern



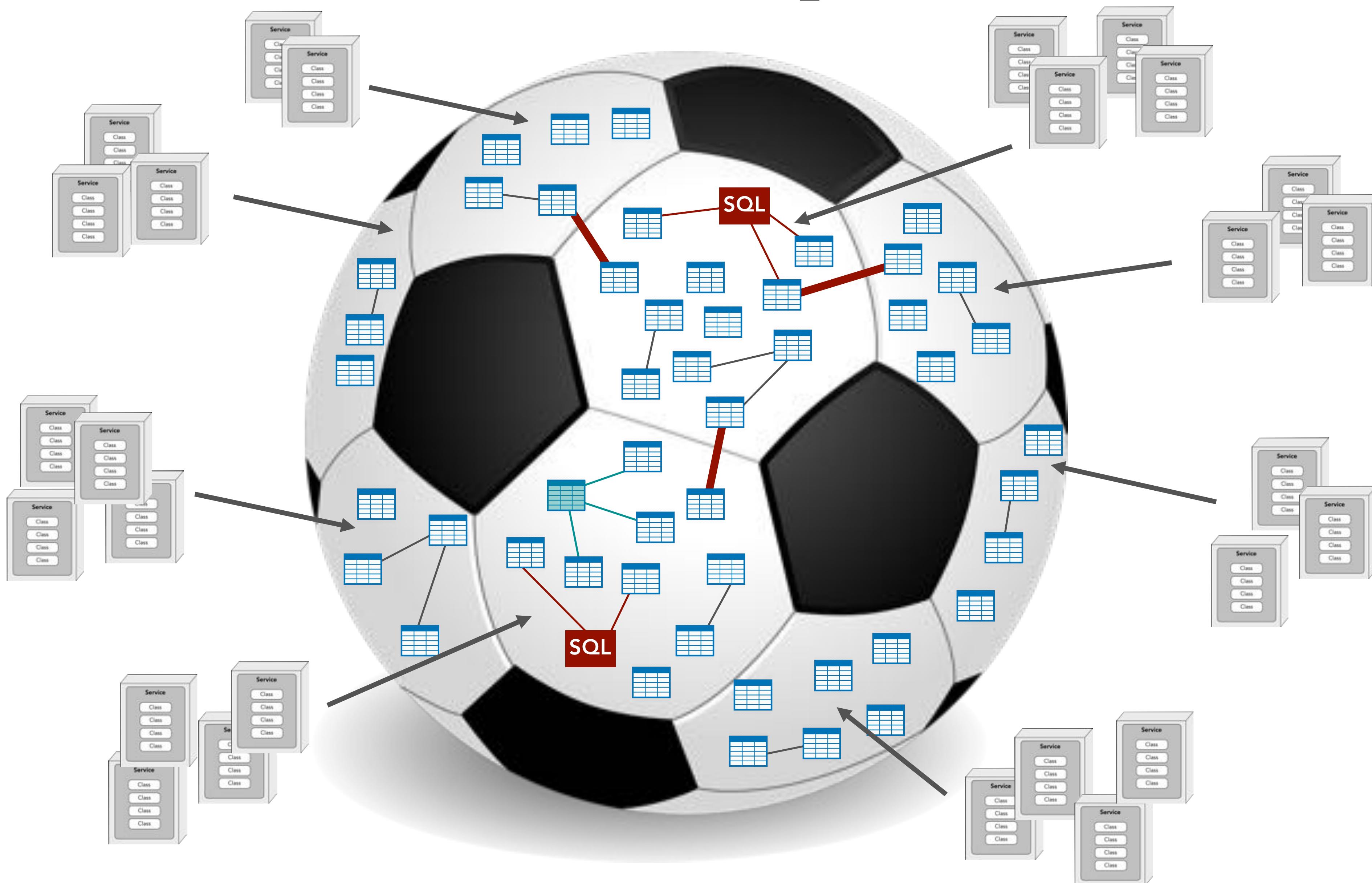
data domain pattern



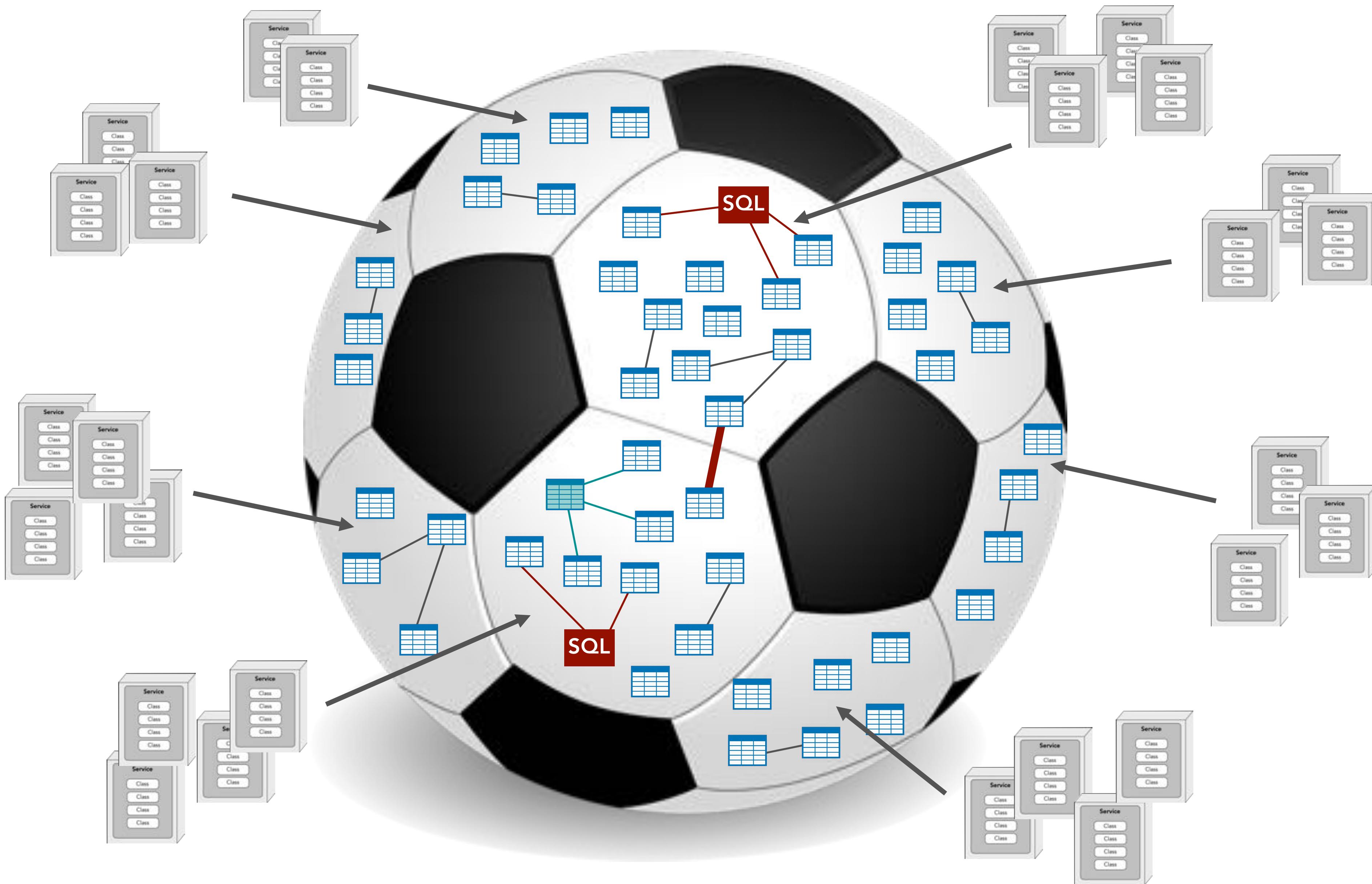
data domain pattern



data domain pattern



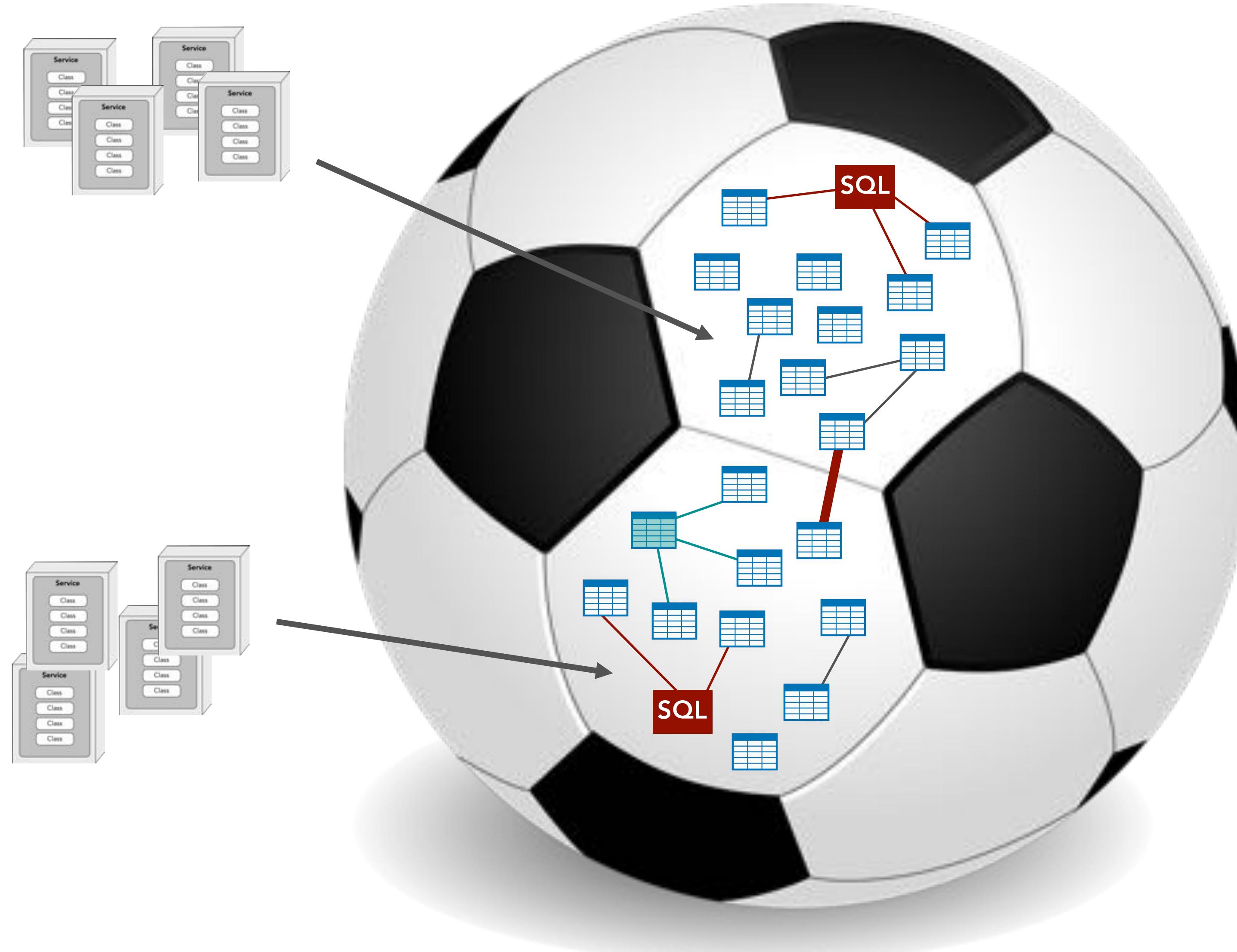
data domain pattern



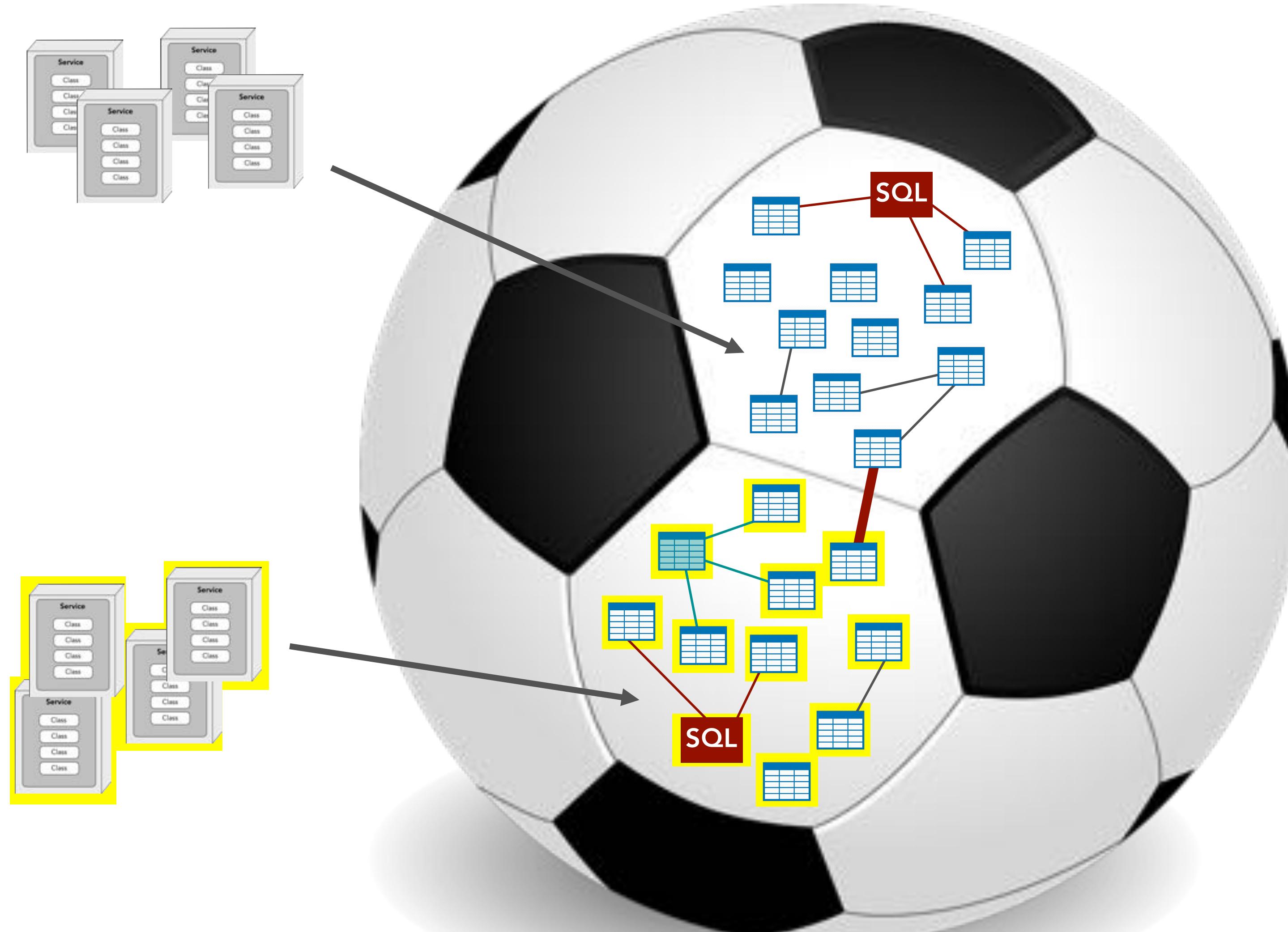
data domain pattern



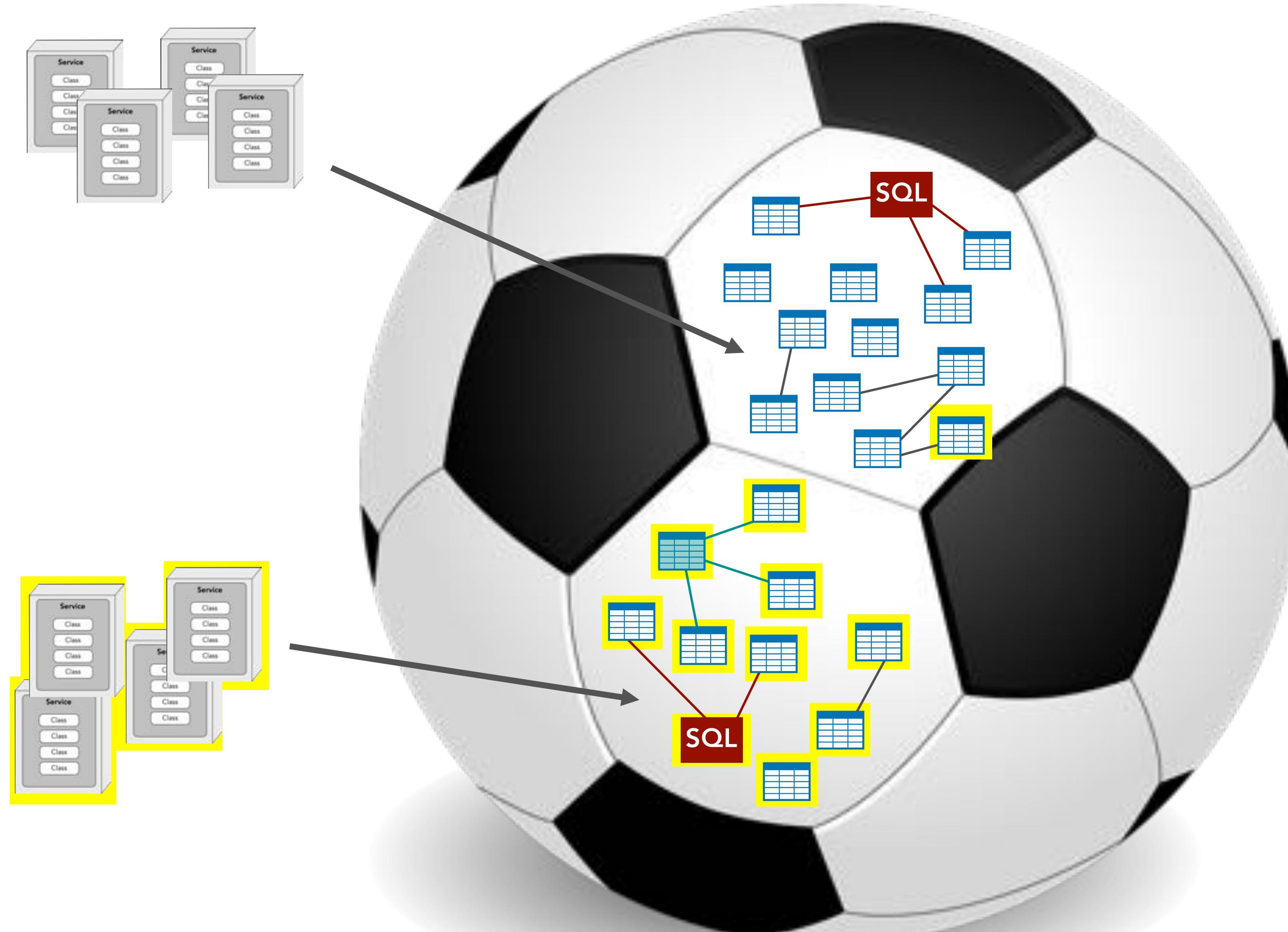
data domain pattern



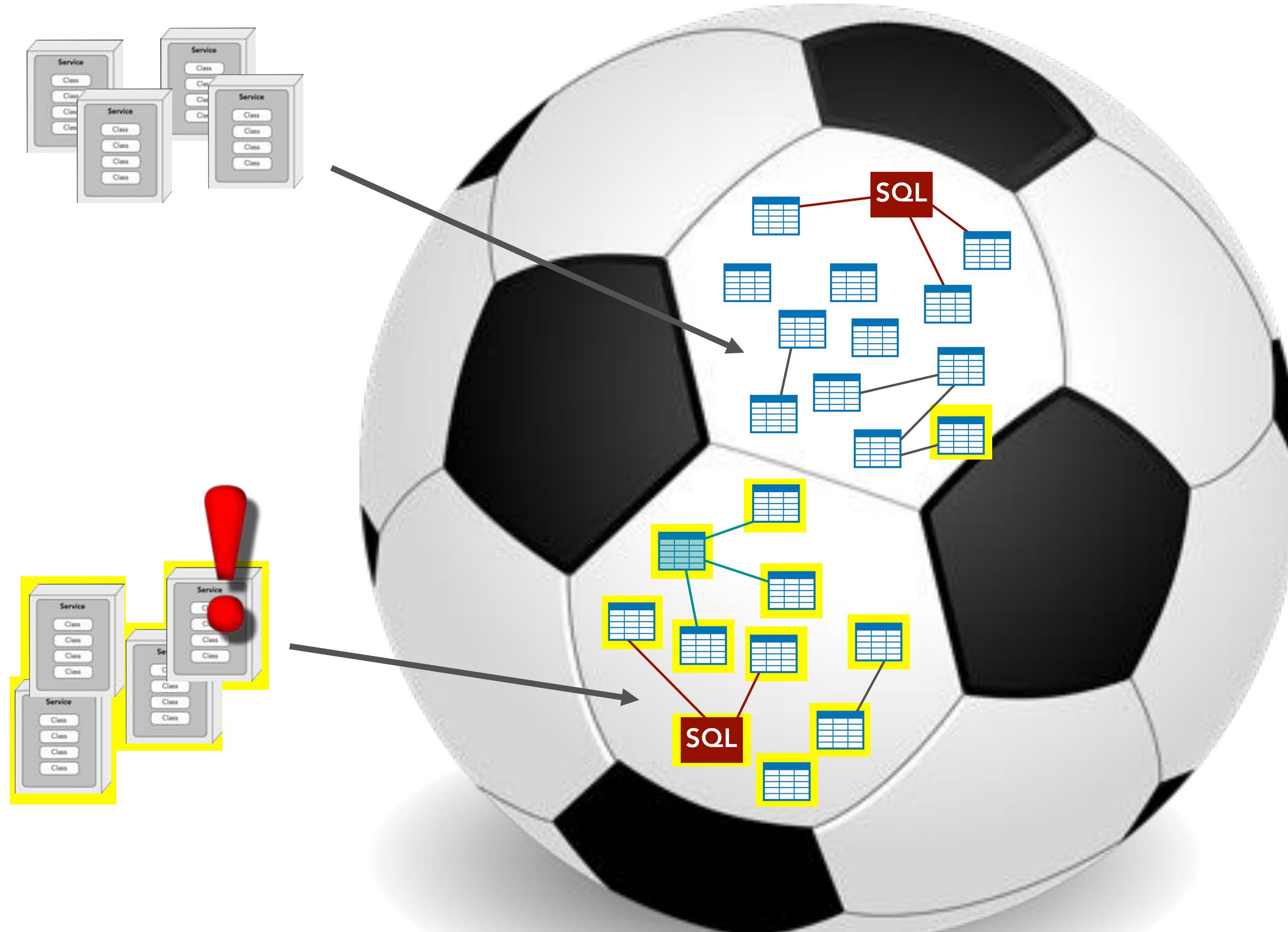
data domain pattern



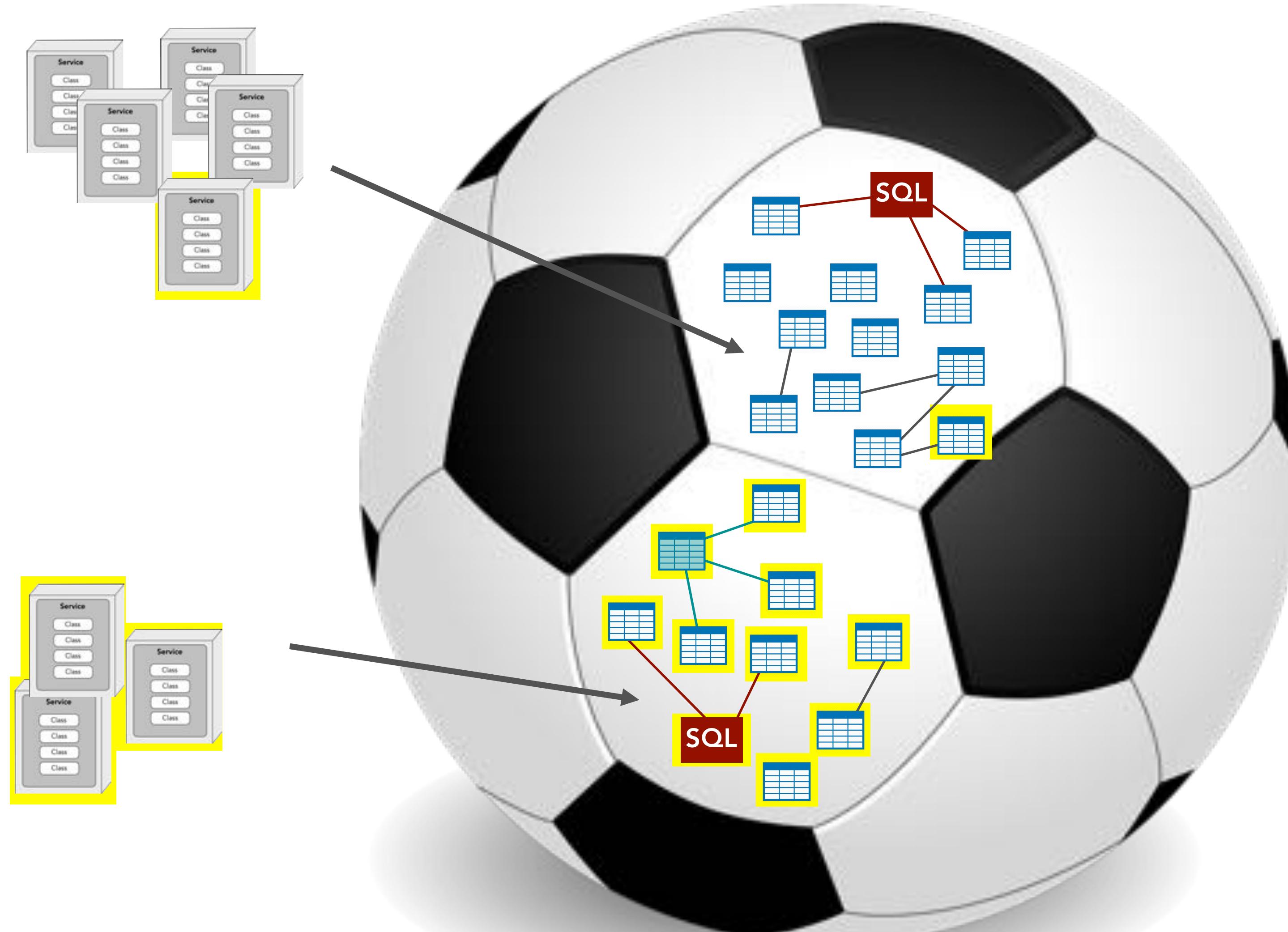
data domain pattern



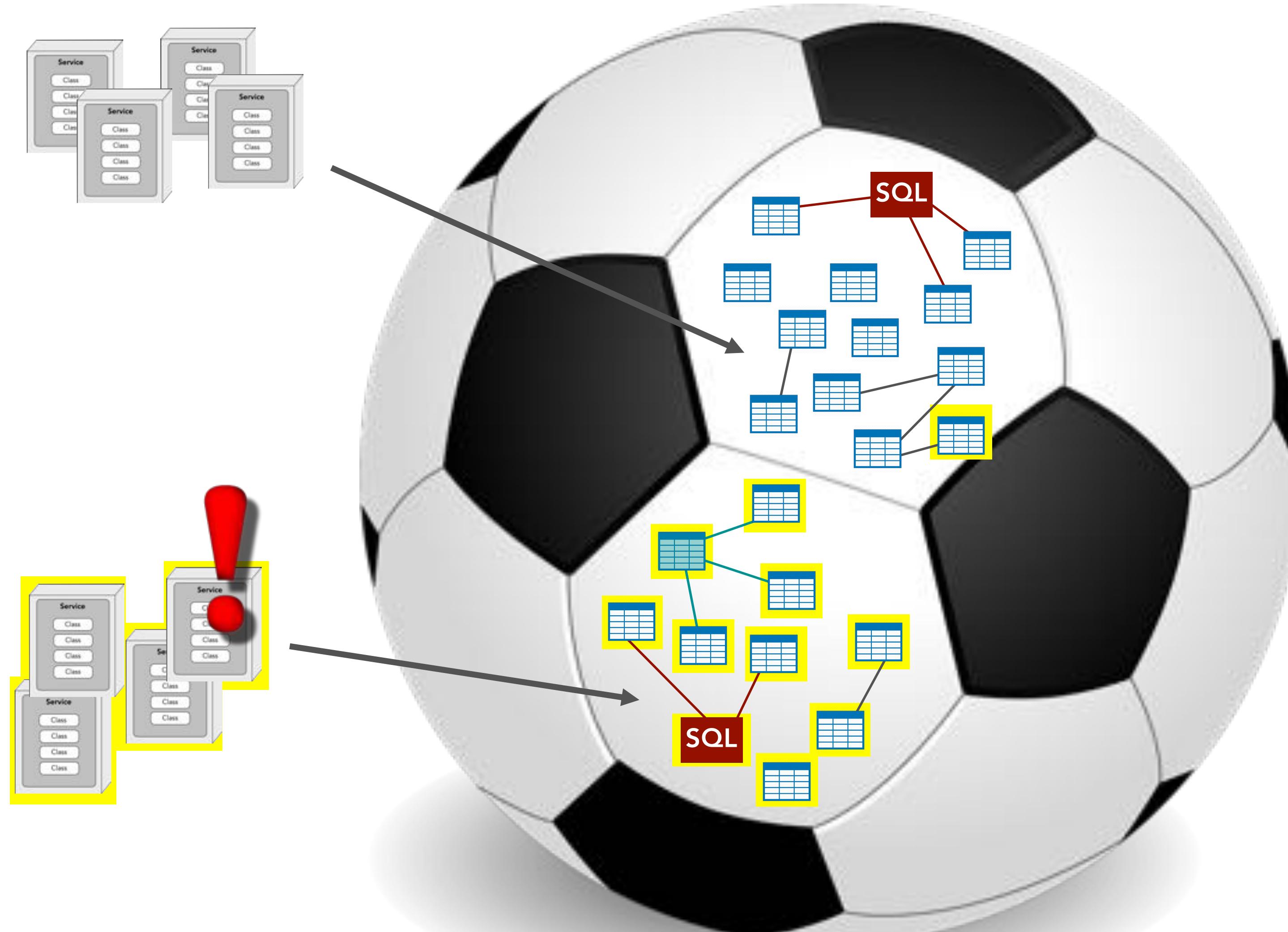
data domain pattern



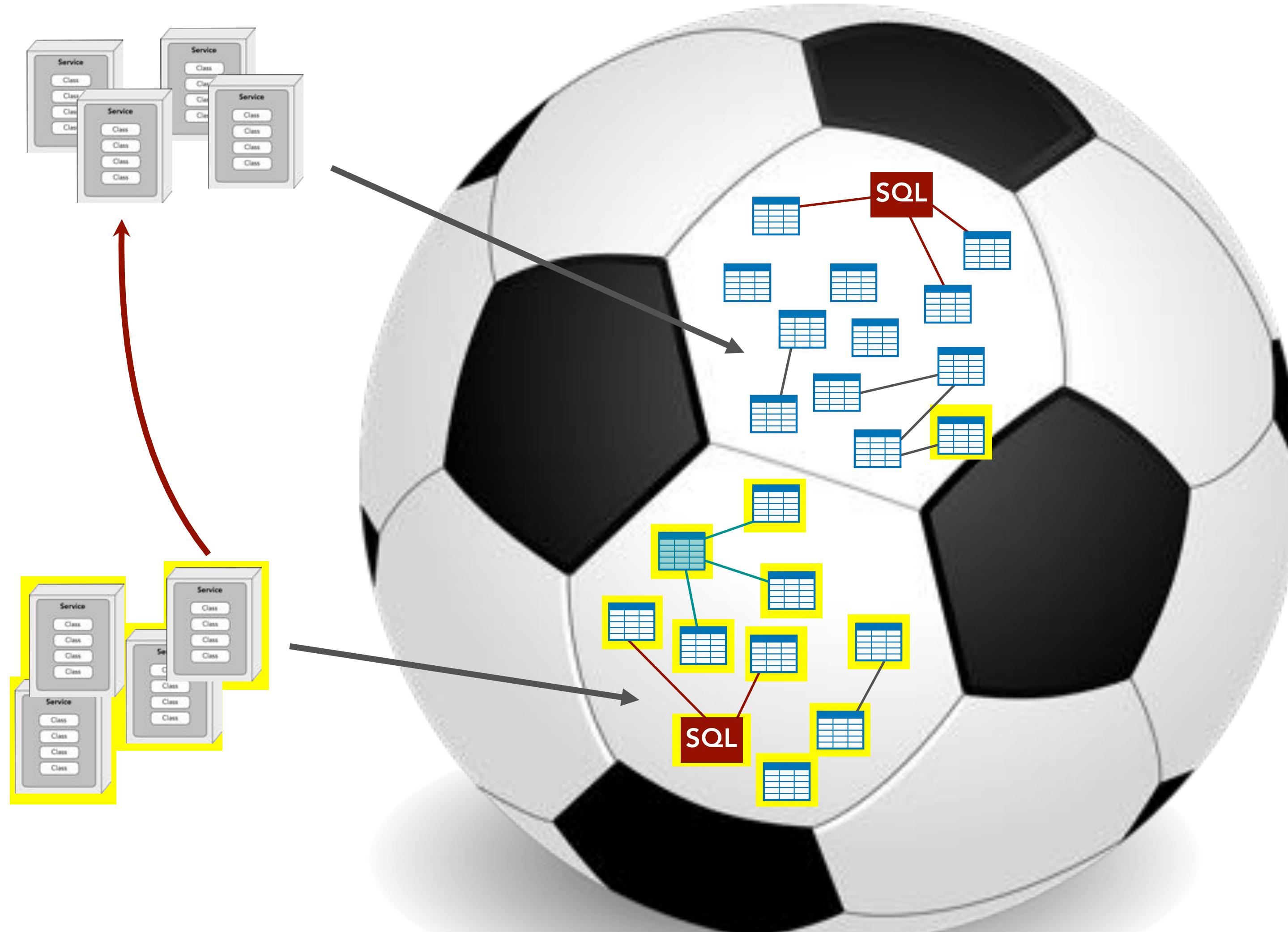
data domain pattern



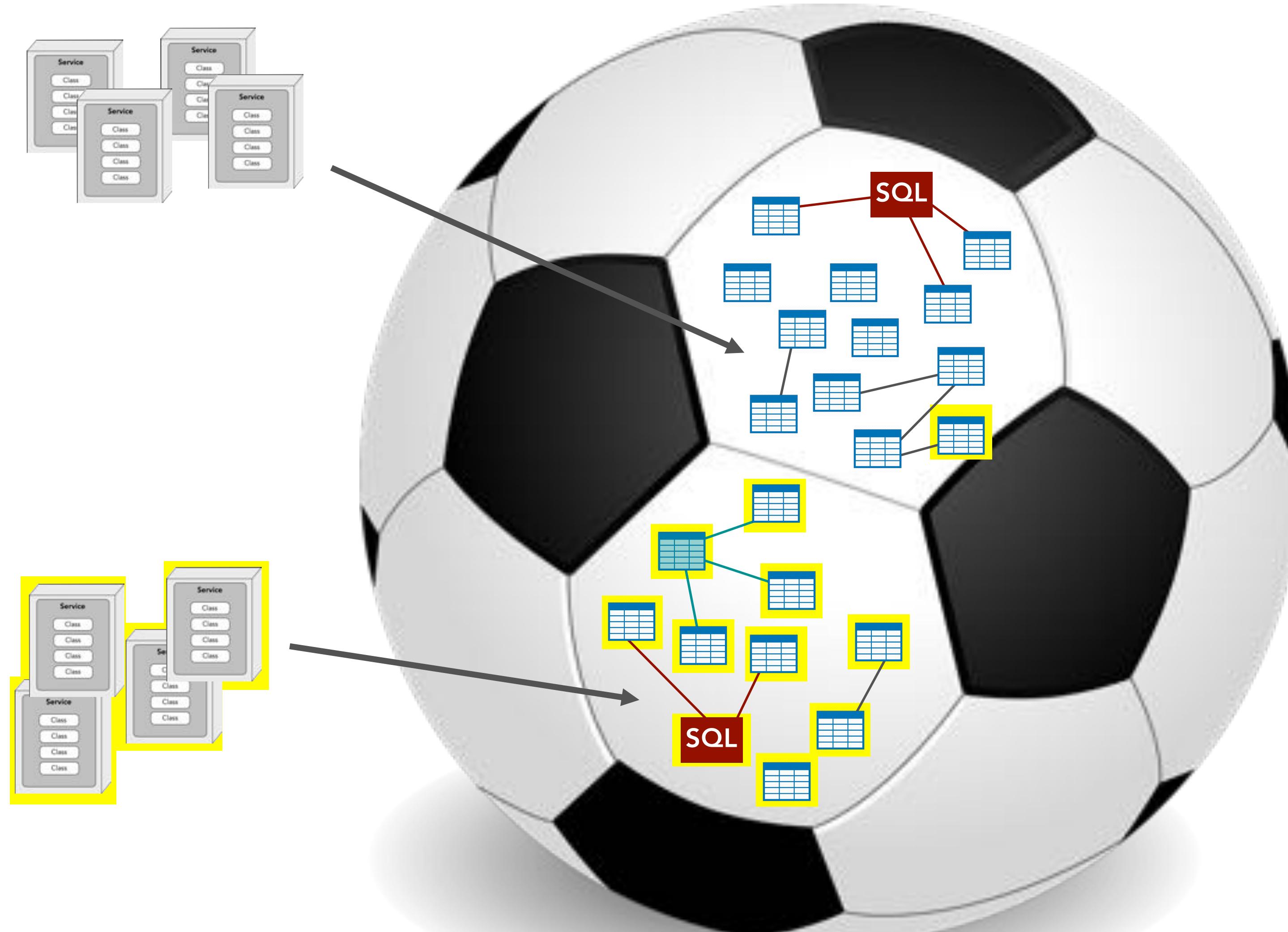
data domain pattern



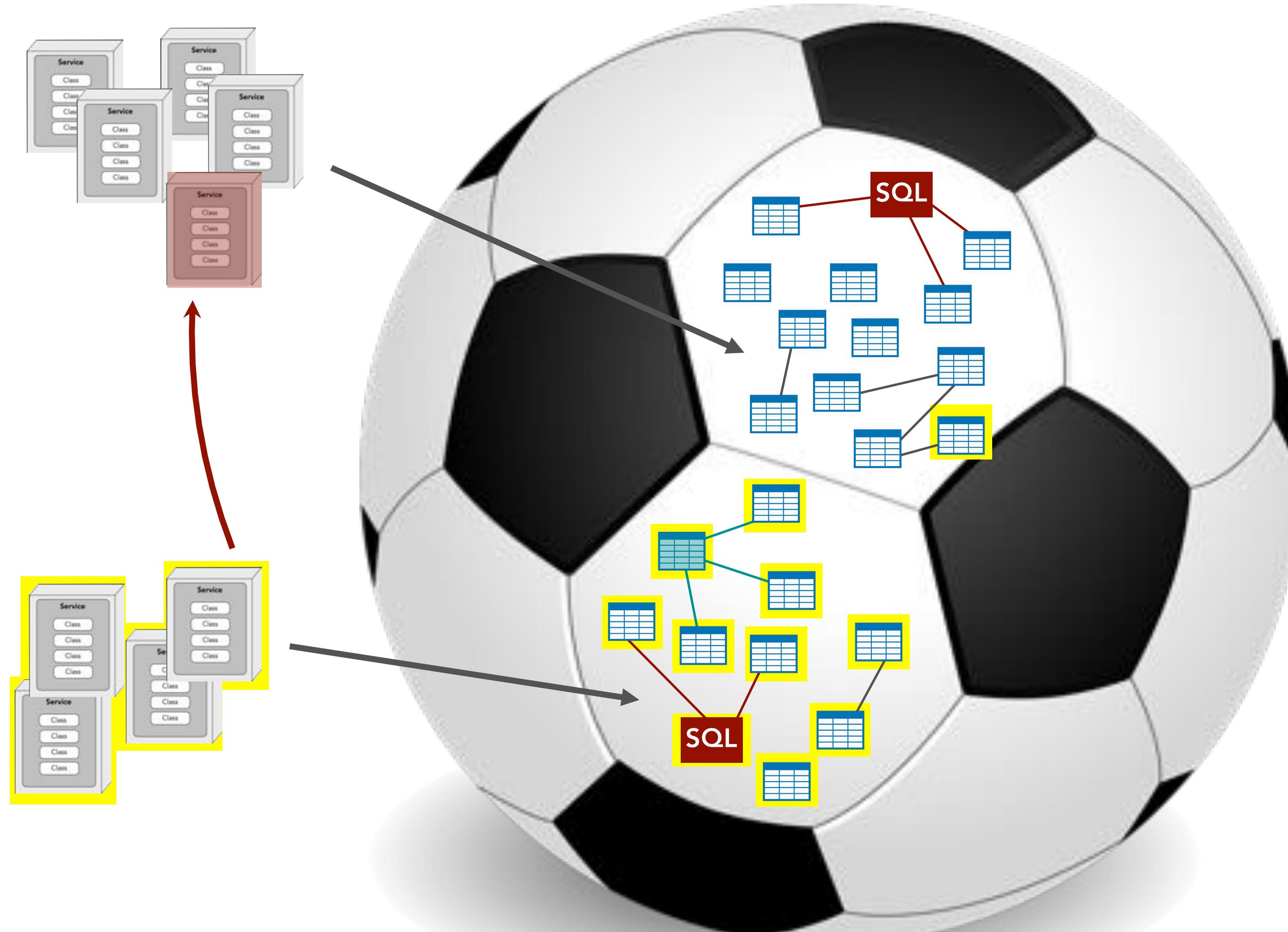
data domain pattern



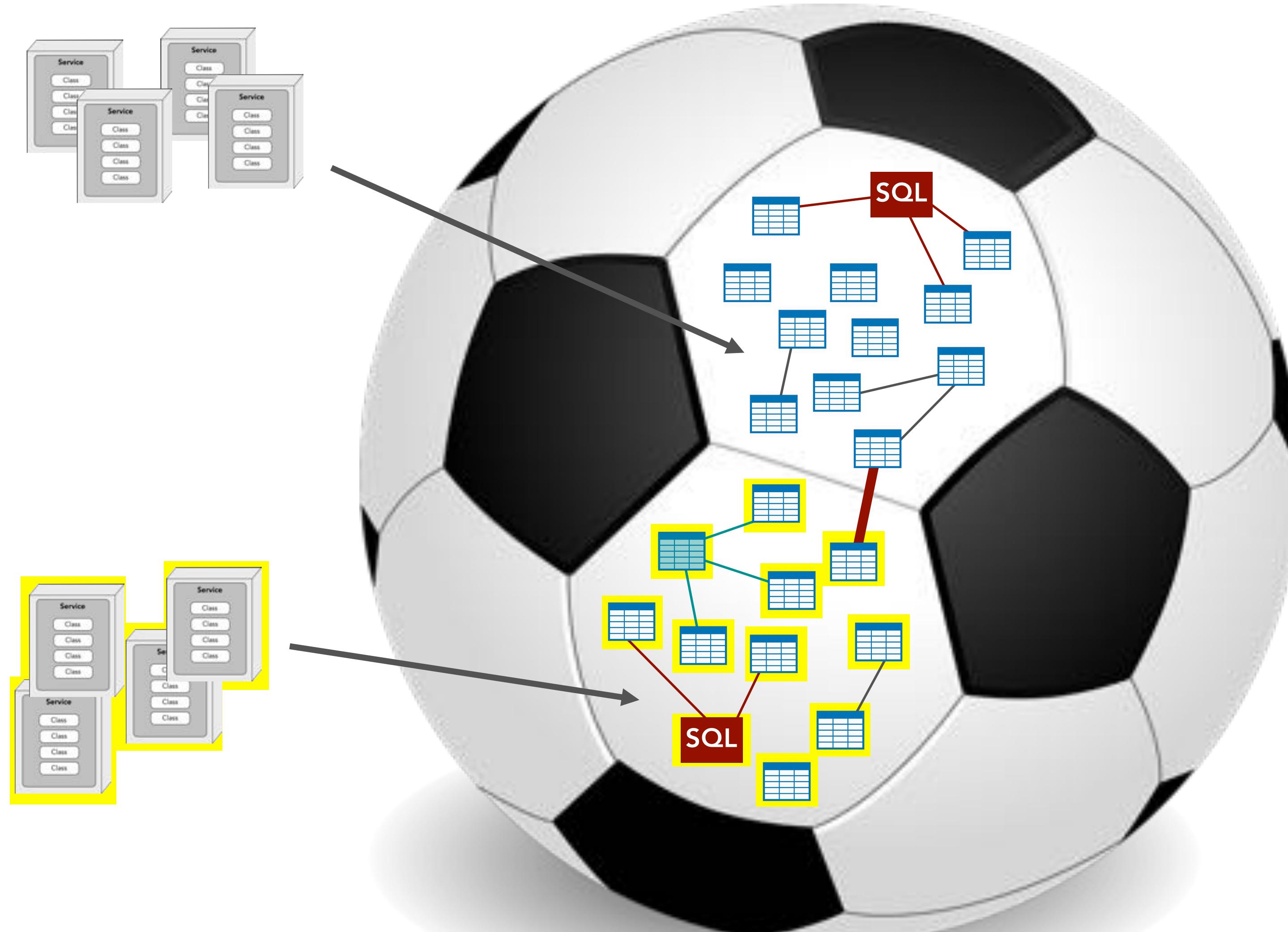
data domain pattern



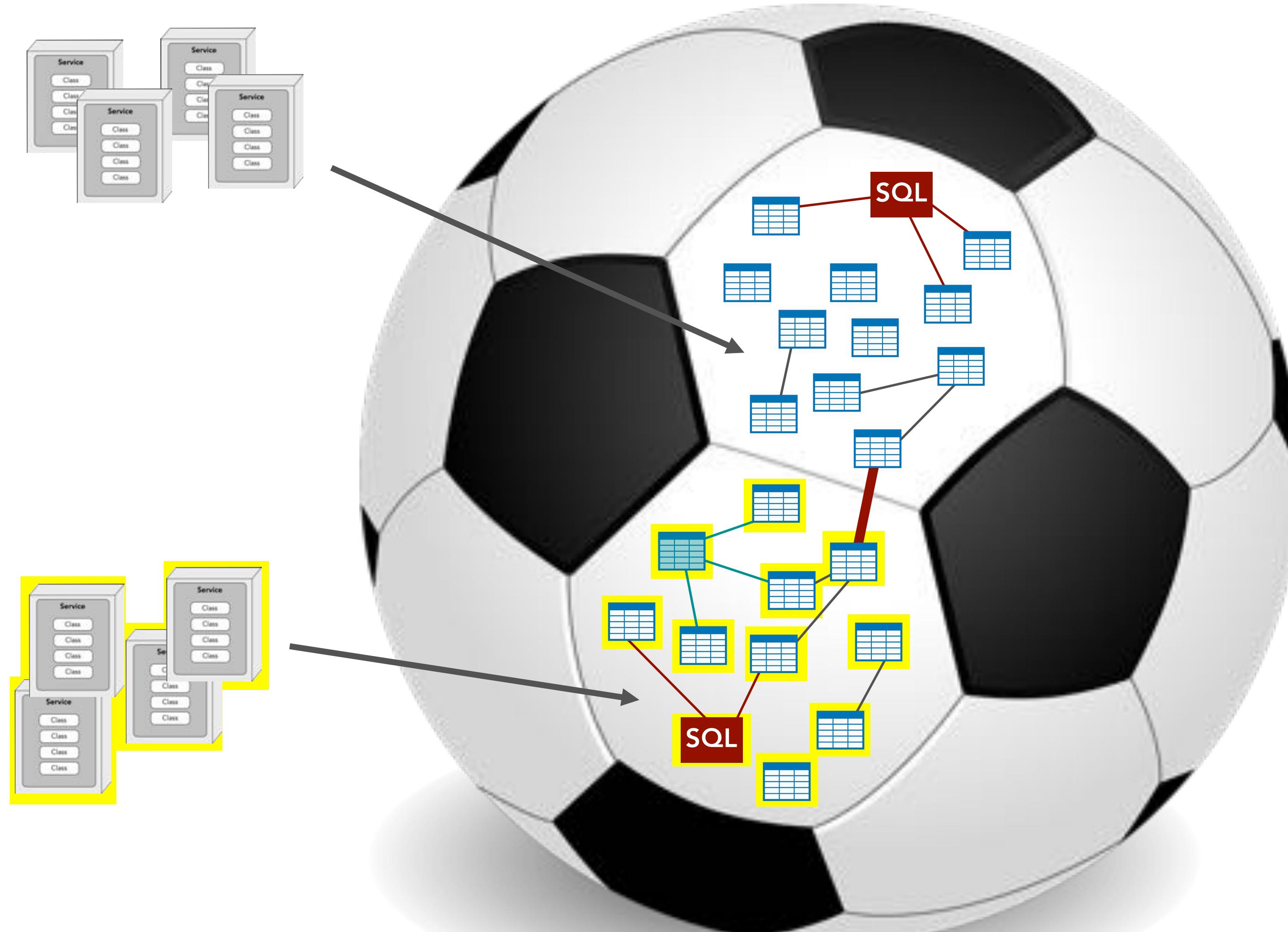
data domain pattern



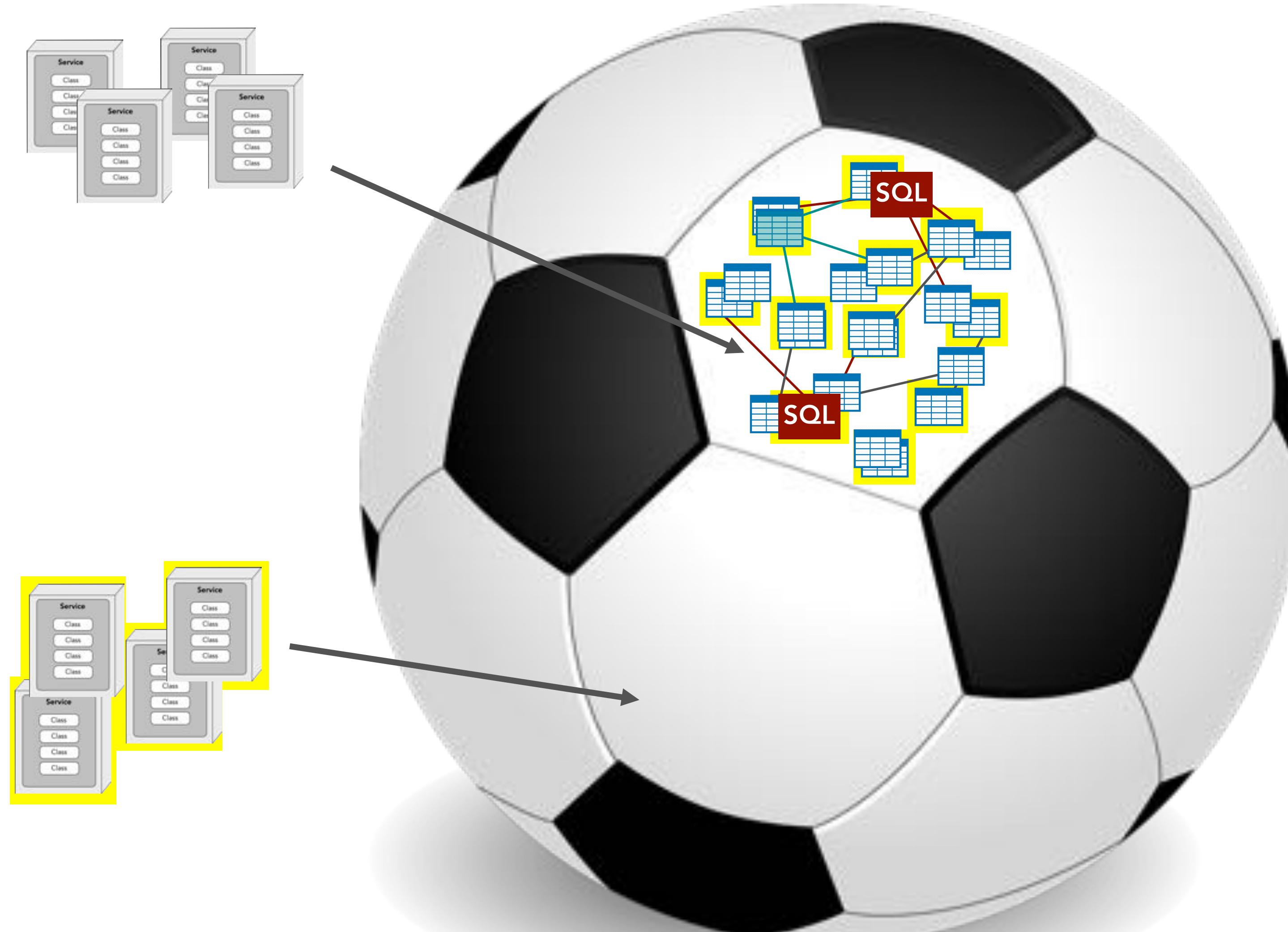
data domain pattern



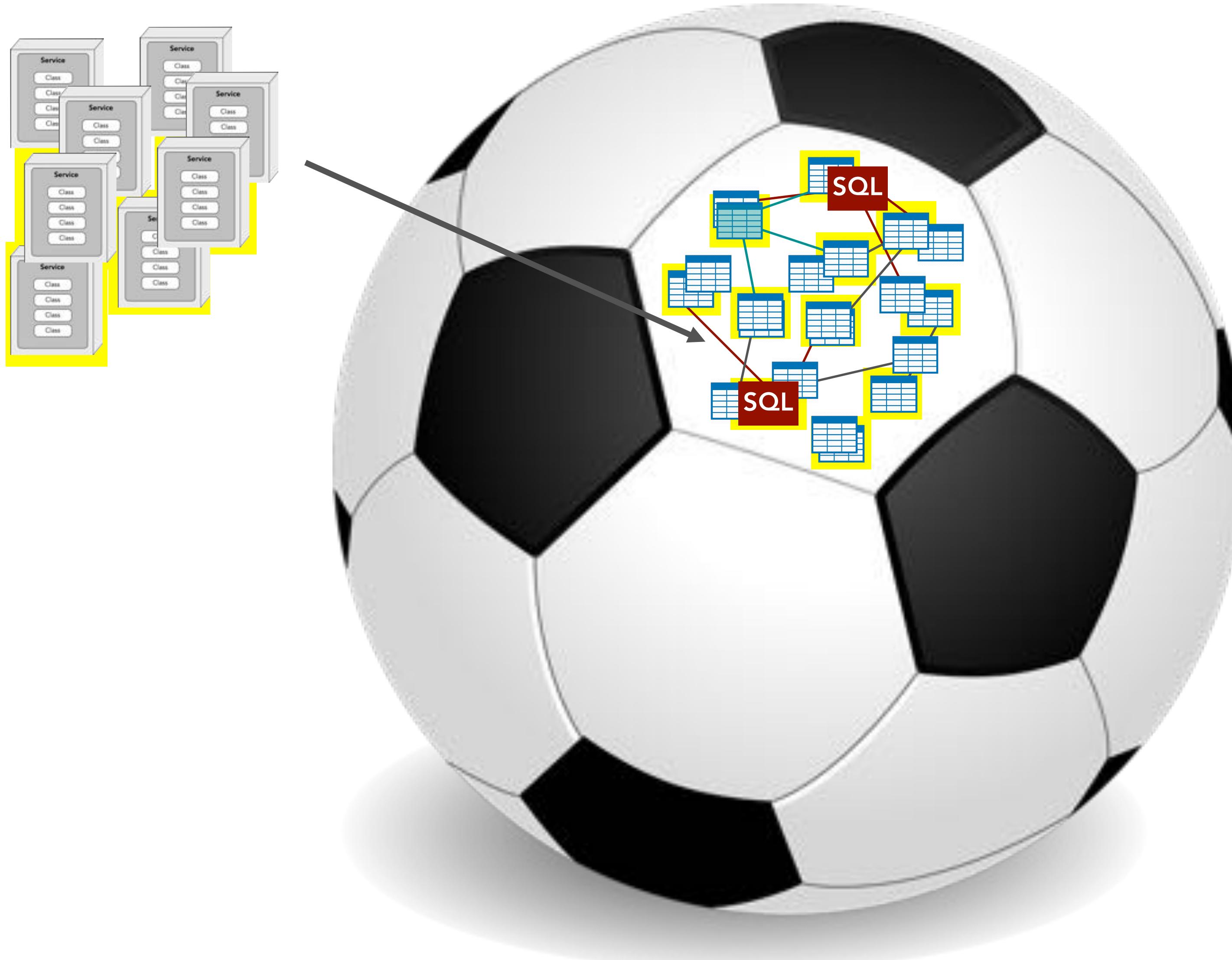
data domain pattern



data domain pattern



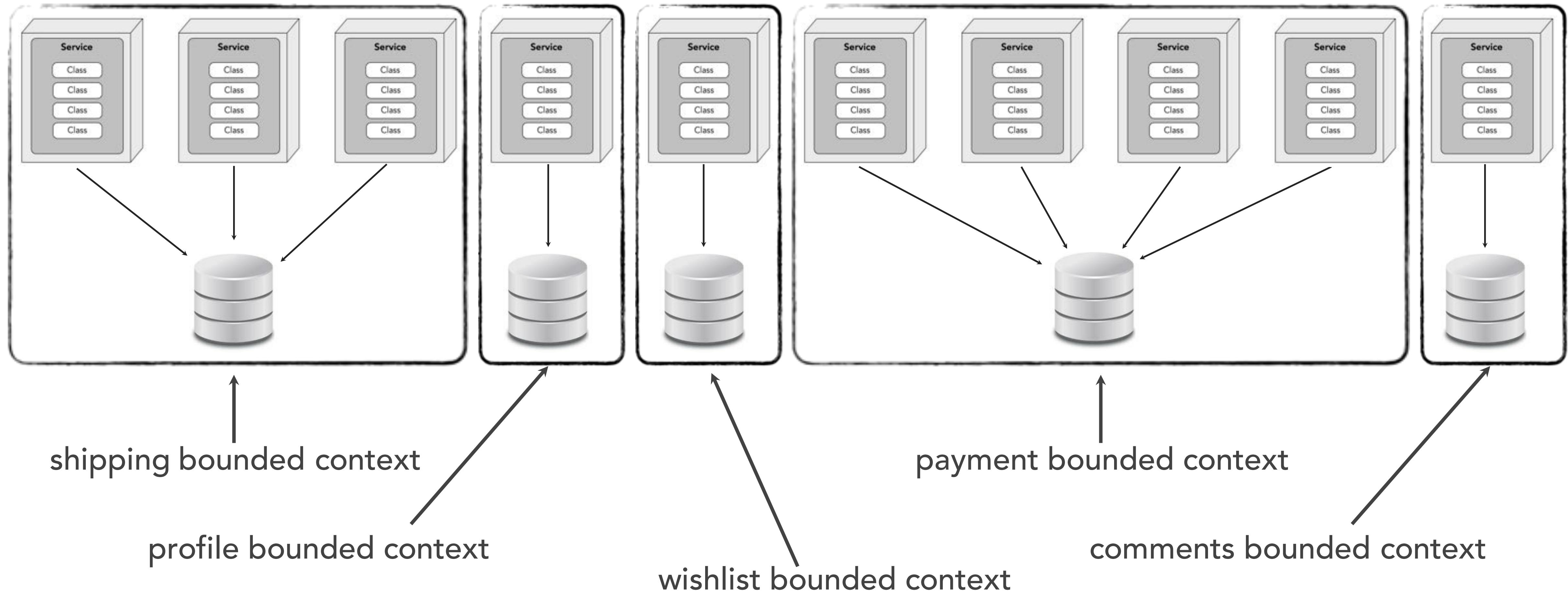
data domain pattern



data domain pattern



bounded context



bounded context

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
public @interface BoundedContext {
    public Context value();

    public enum Context {
        SHIPPING,
        PAYMENT,
        WISHLIST,
        COMMENTS,
        PROFILE
    }
}
```

bounded context

```
@ServiceEntrypoint  
@FunctionalService  
@BoundedContext(Context.PAYMENT)  
public class CreditCardServiceAPI {  
    ...  
}
```

bounded context

```
@ServiceEntrypoint  
@FunctionalService  
@BoundedContext(Context.PAYMENT)  
public class CreditCardServiceAPI {
```

...

```
}
```

```
@ServiceEntrypoint  
@FunctionalService  
@BoundedContext(Context.PAYMENT)  
public class GiftCardServiceAPI {
```

...

```
}
```

bounded context

```
@ServiceEntrypoint  
@FunctionalService  
→ @BoundedContext(Context.PAYMENT)  
public class CreditCardServiceAPI {
```

...

```
}
```

```
@ServiceEntrypoint
```

```
@FunctionalService
```

```
→ @BoundedContext(Context.PAYMENT)
```

```
public class GiftCardServiceAPI {
```

...

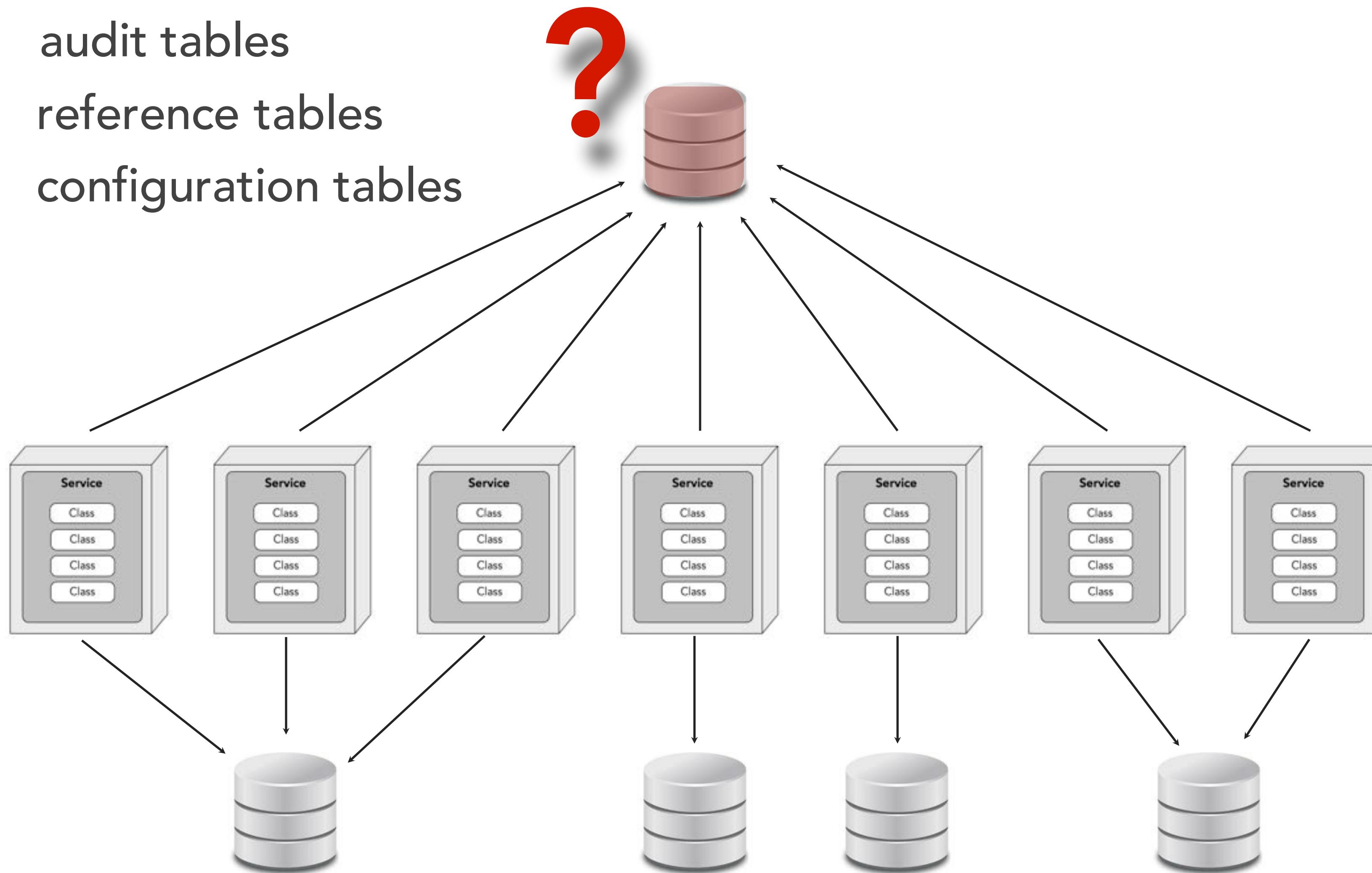
```
}
```

managing common data

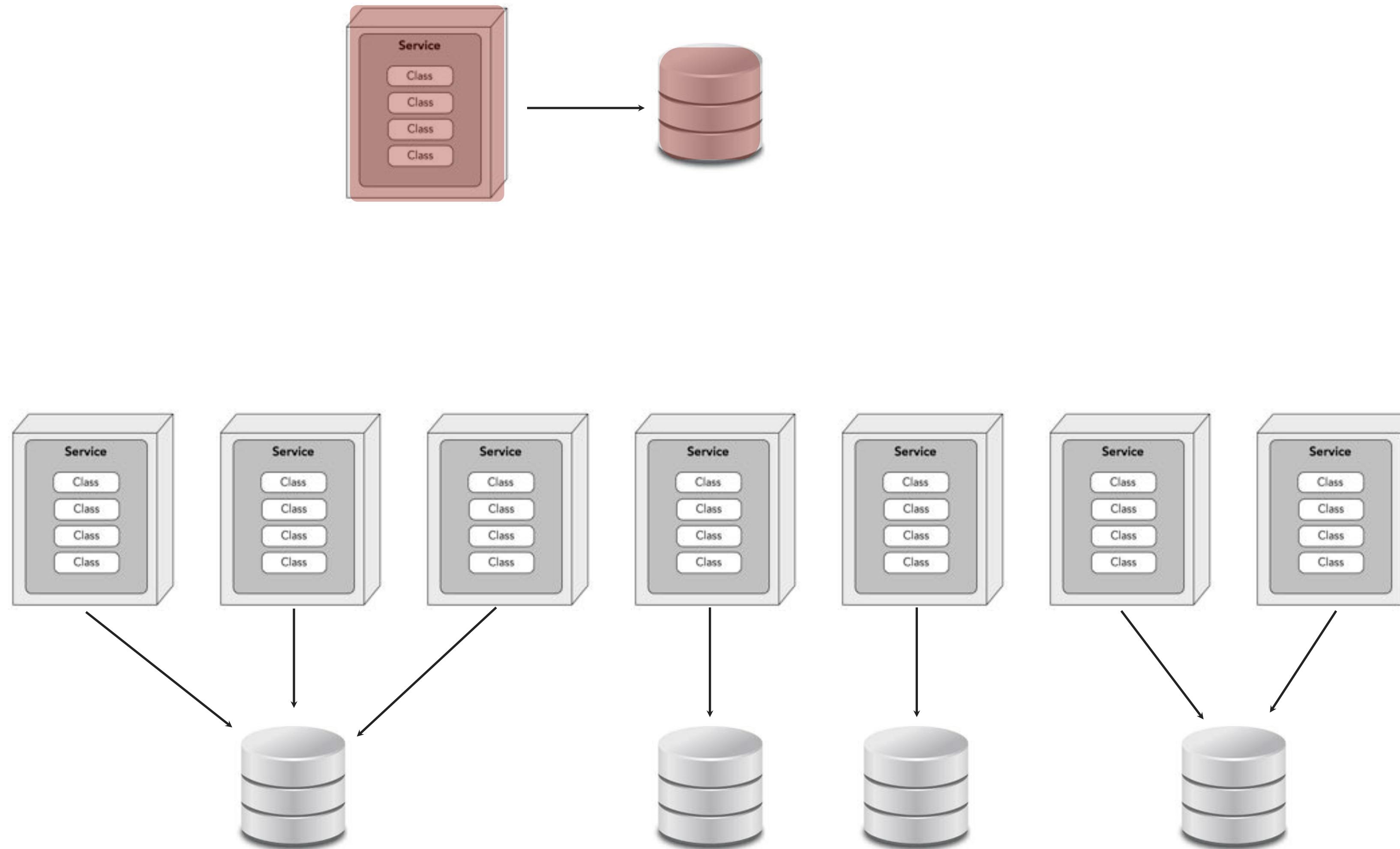
audit tables

reference tables

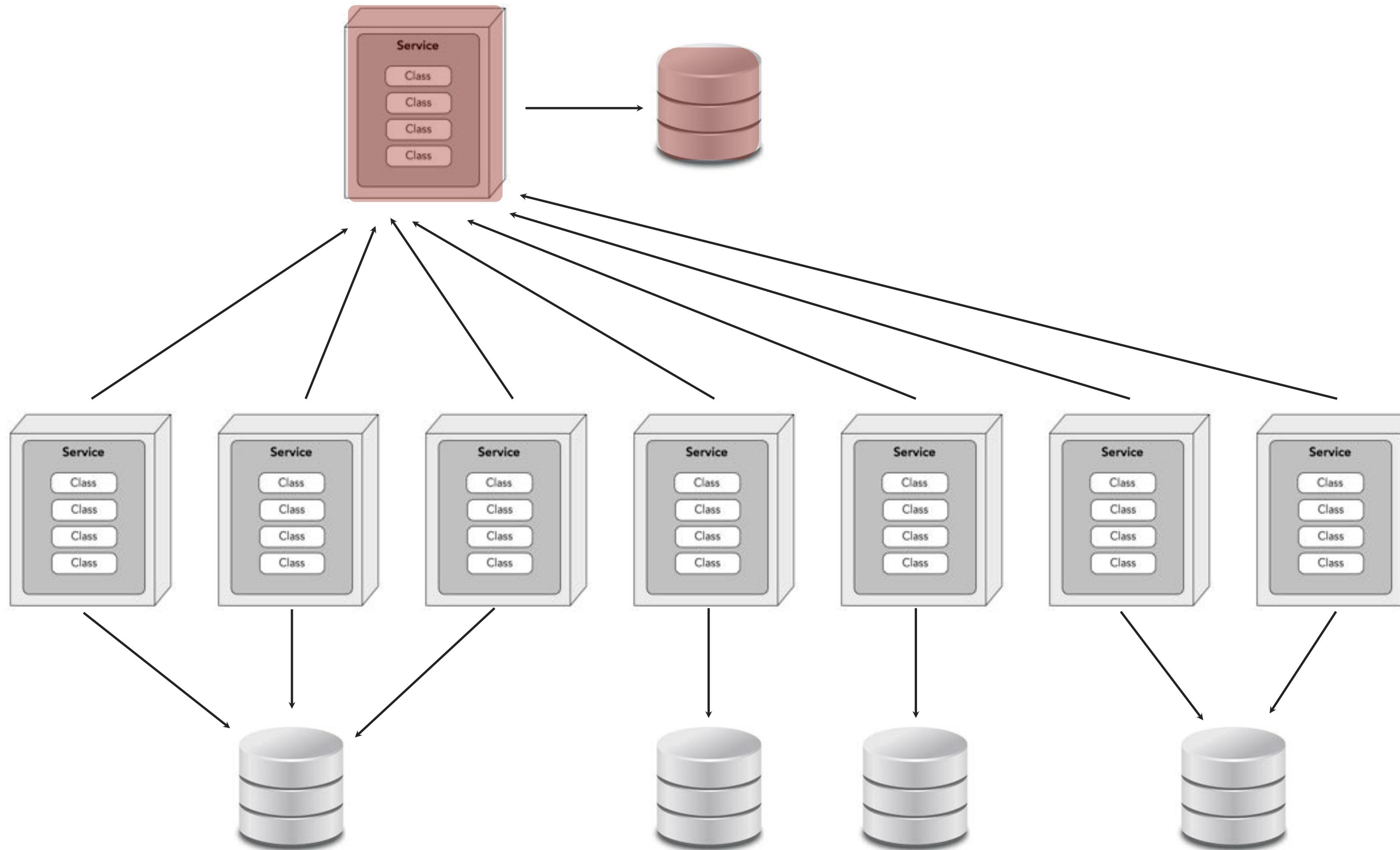
configuration tables



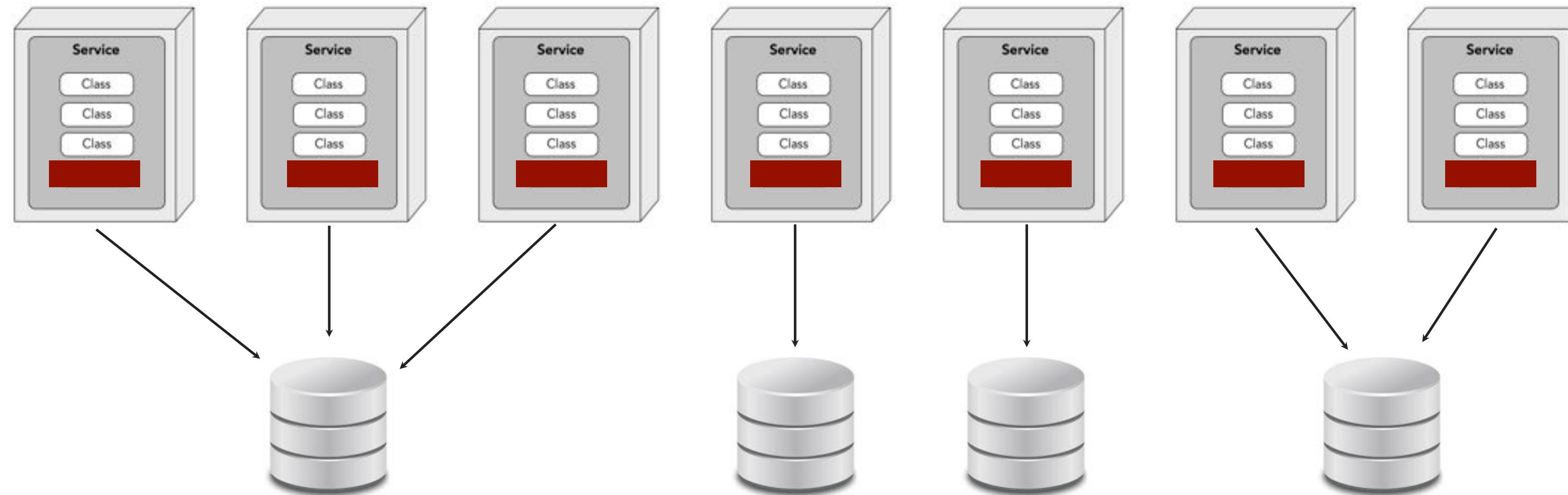
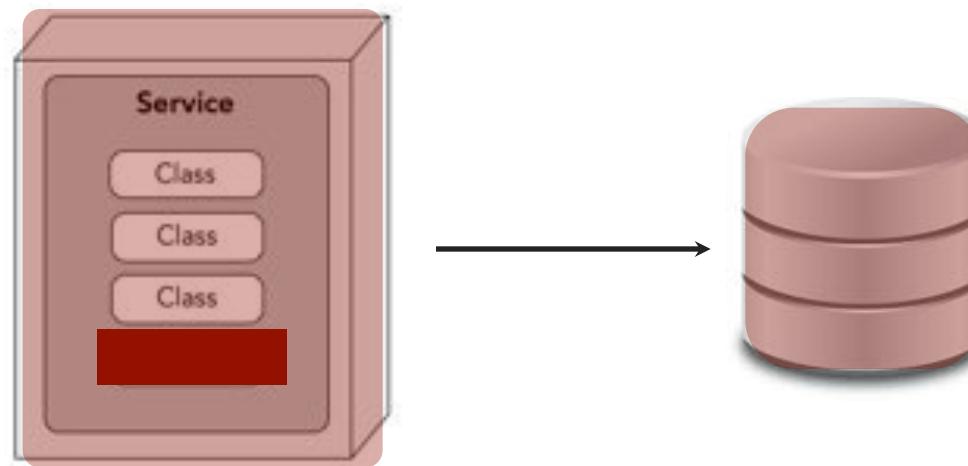
managing common data



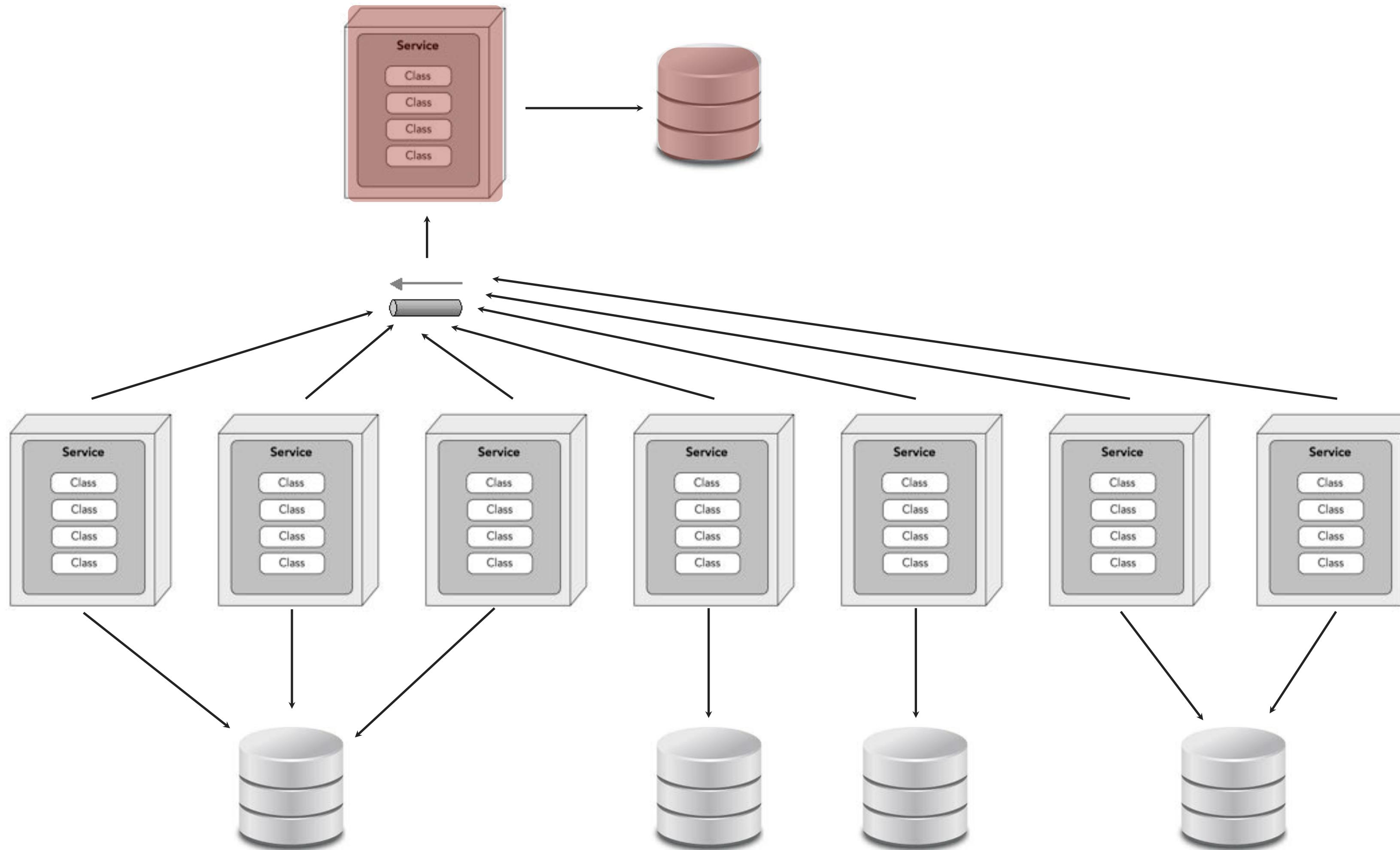
managing common data



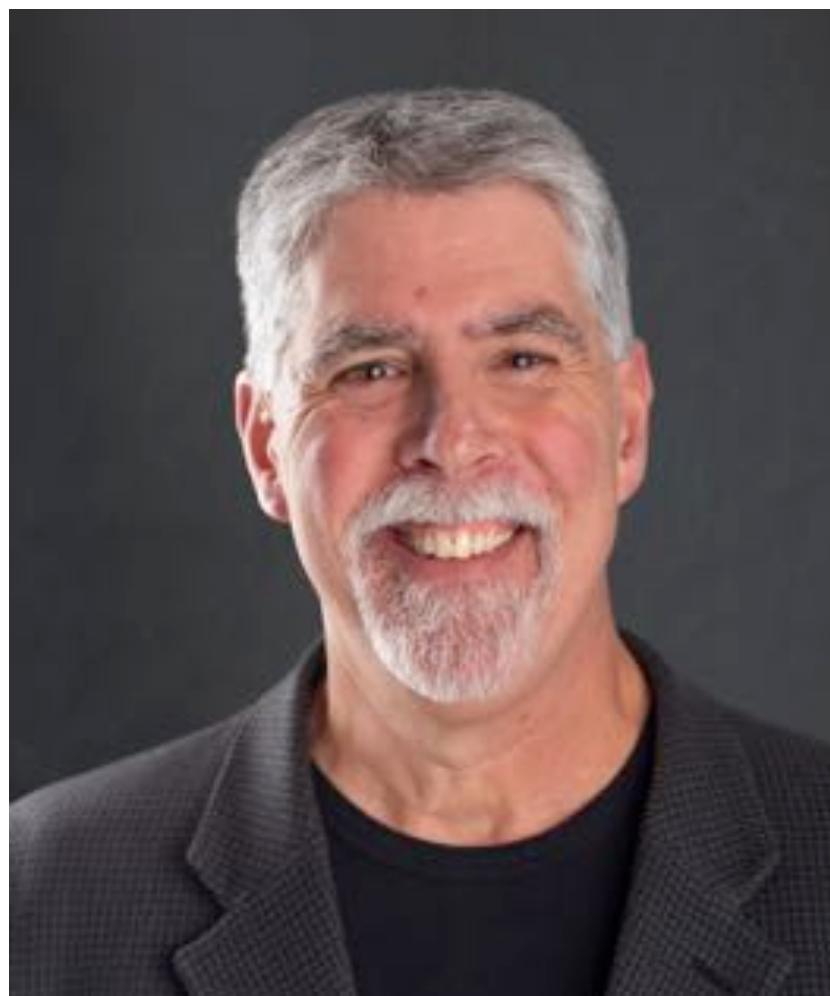
managing common data



managing common data



Decomposing Data Within Distributed Systems



Mark Richards

Independent Consultant

Hands-on Software Architect / Published Author

Founder, DeveloperToArchitect.com

<https://www.linkedin.com/in/markrichards3>

@markrichardssa