



# Spring Security

---

Beginner to Guru

Multi-tenancy Security



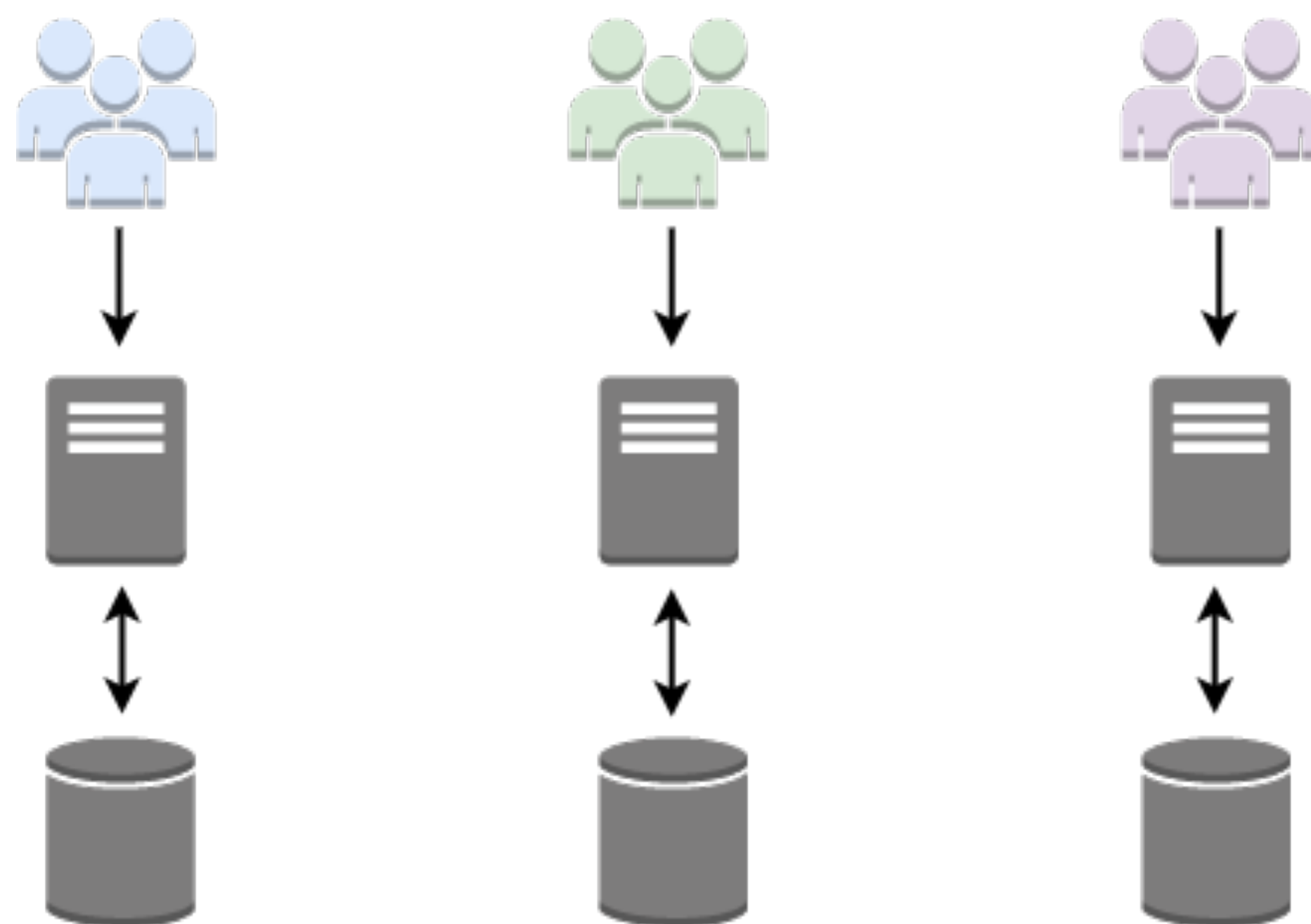
## Multi-tenancy Security

- Multi-tenant software architectures allows multiple users to share a single instance of a software application
- Tenants are users, can be individuals or groups
  - Individuals - example Gmail
  - Groups - example GitHub Organizations
- Benefit of multi-tenancy is efficiency
- Multi-tenancy comes in many different forms



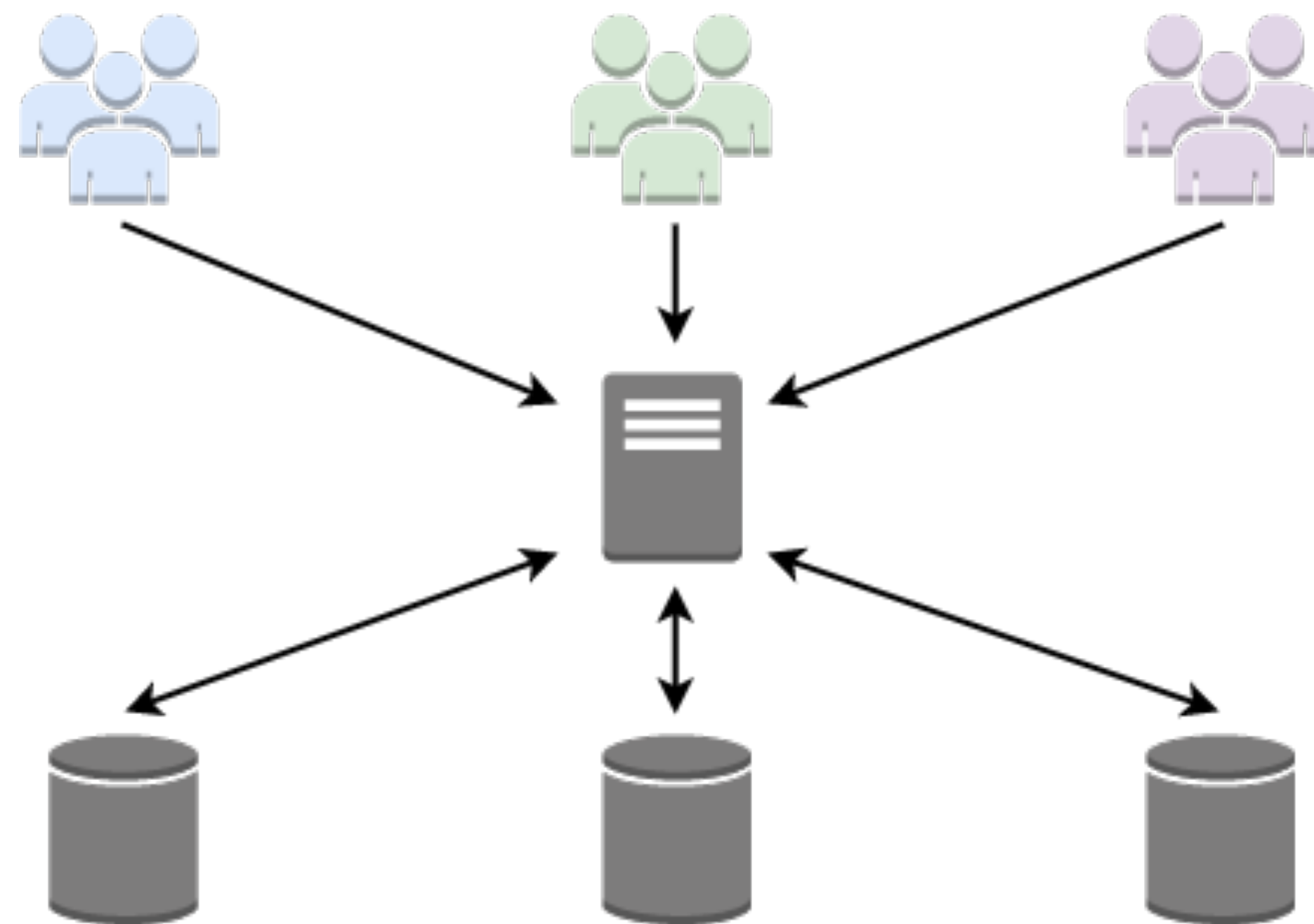


# Instance Replication





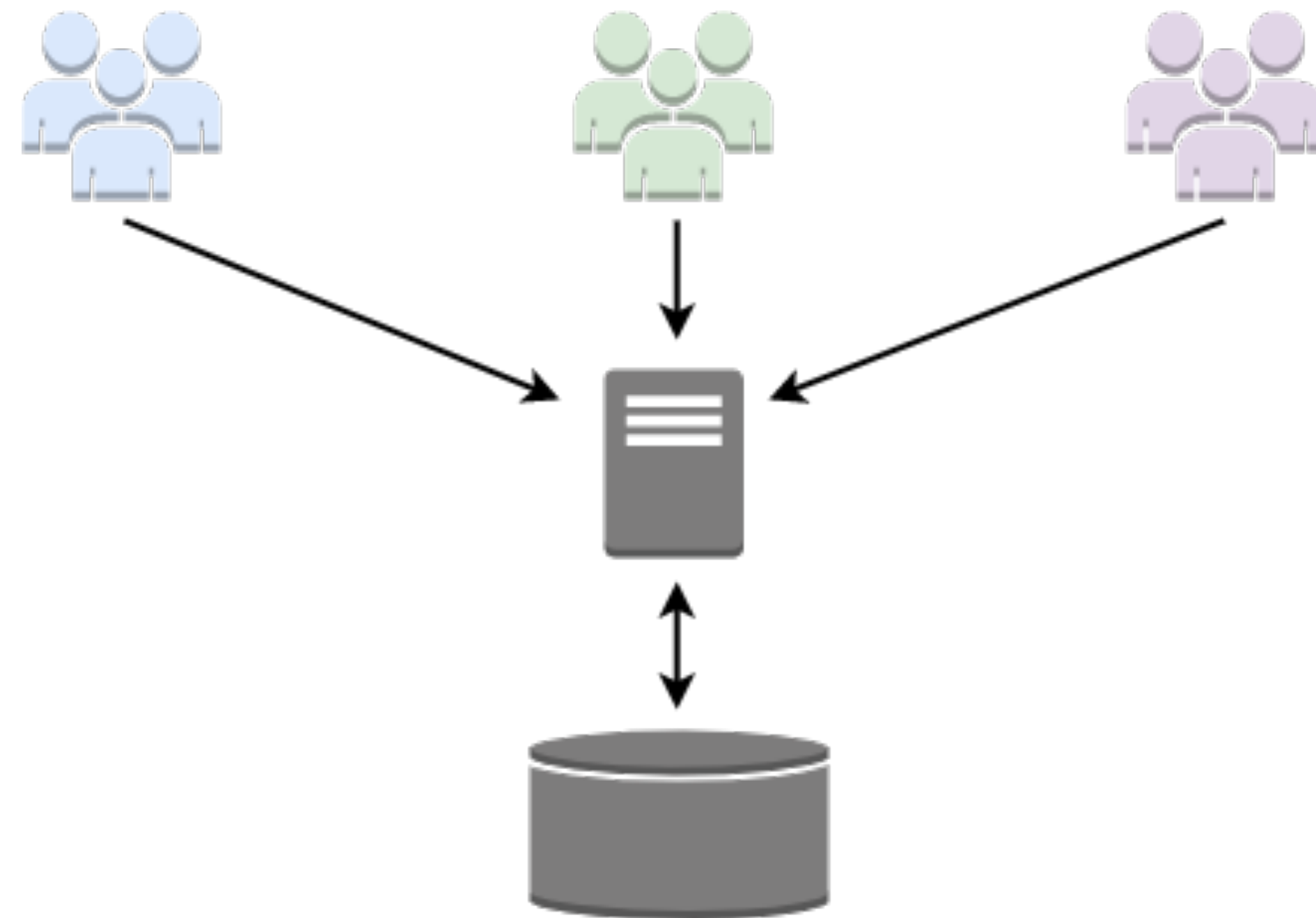
## Shared Application, Separate Databases



2020



## Shared Application, Shared Database





## Multi-tenancy Security

- Shared Databases can use a schema per tenant or shared tables with a tenant attribute
- When the application and/or database is shared application security needs to prevent unauthorized access
  - Customer A should not be able to read Customer B's data
  - Customer A should not be able to update or delete Customer B's data
  - A super user may need access to all customer data
  - Service accounts may need access to all customer data







## Multi-tenancy Security with Spring Security

- Use Case: Allow Customers to Place Orders, Read Orders, and Cancel Orders
- A customer can have one or more users
- A custom UserDetails Object is used to hold necessary attributes in Security Context
- SPeL is used to access attributes to make access decisions
- For our use case, we will add Customer to the UserDetails object
  - Rather than using the Spring Security User object, we will implement our own

