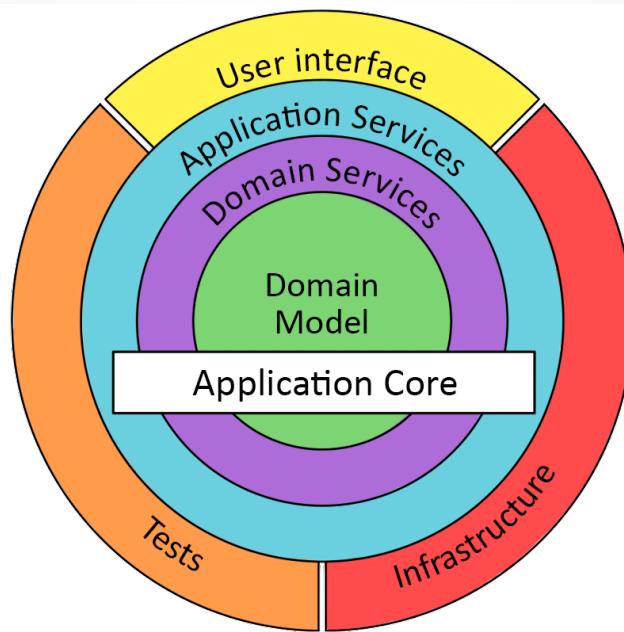
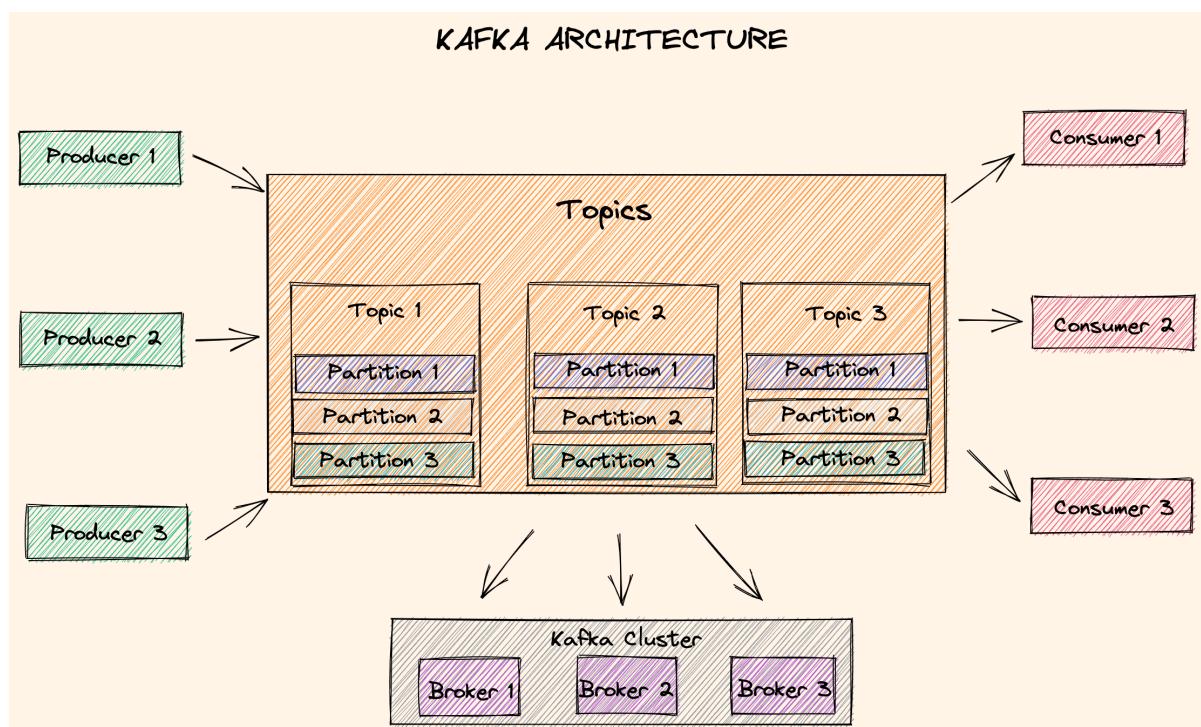
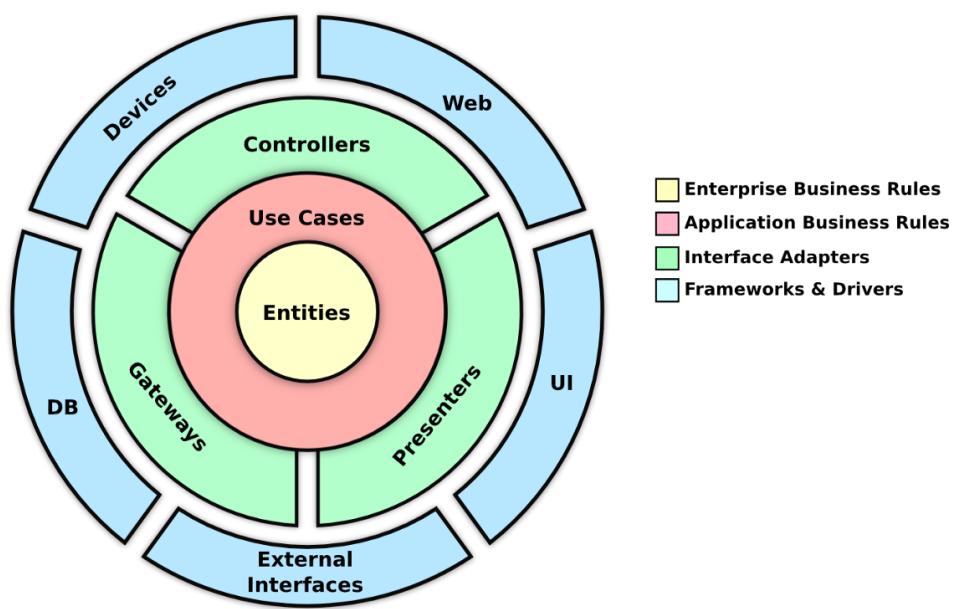


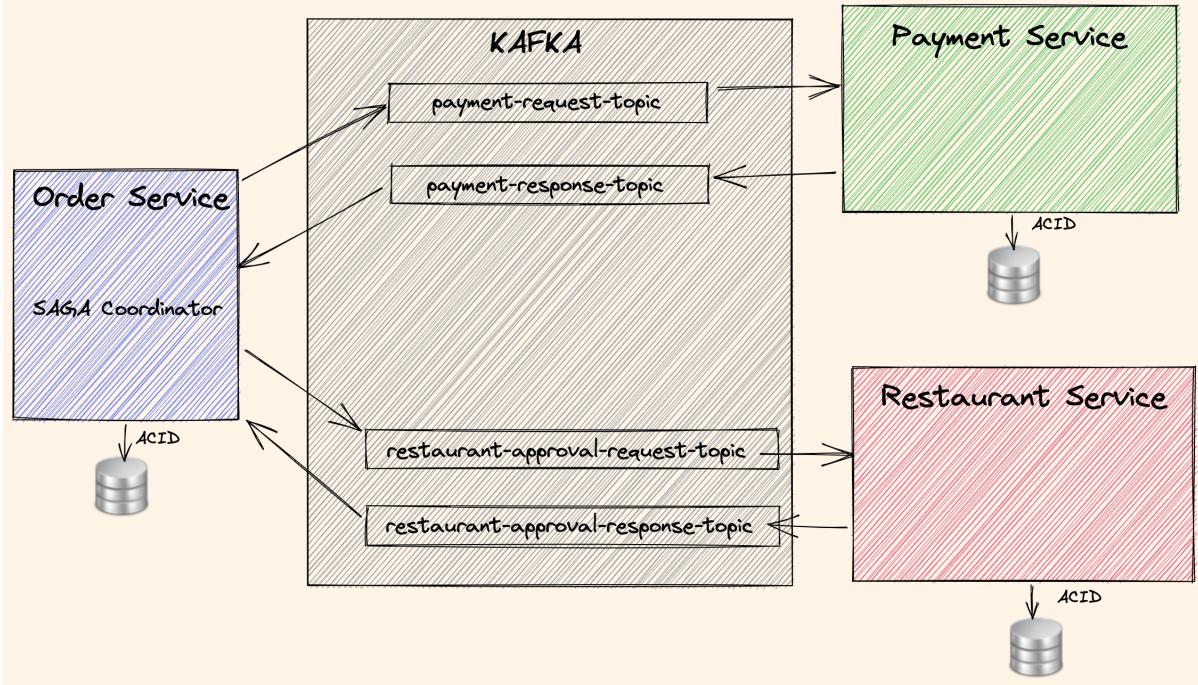
## Structure of the Course

- Microservices: Powered by Spring boot
  - \* Independent development and deployment by different teams
  - \* Easy to scale for a specific service
  - \* Better fault isolation
  - \* Enables to use different technology and languages for different services
- Clean & Hexagonal Architectures: Isolate the domain logic from outside dependencies.
- Domain Driven Design (DDD): Bounded context, Entities, Aggregates, Value Objects, Domain services, Application services, and Domain Events.
- Kafka: Event store for Event-driven services. Enable loosely coupled services that communicates through events.
- SAGA: Distributed long running transactions across services. Used for Long Lived Transactions (LLT). First invented in a publication on 1987 <https://www.cs.cornell.edu/~andru/cs711/2002fa/reading/sagas.pdf>





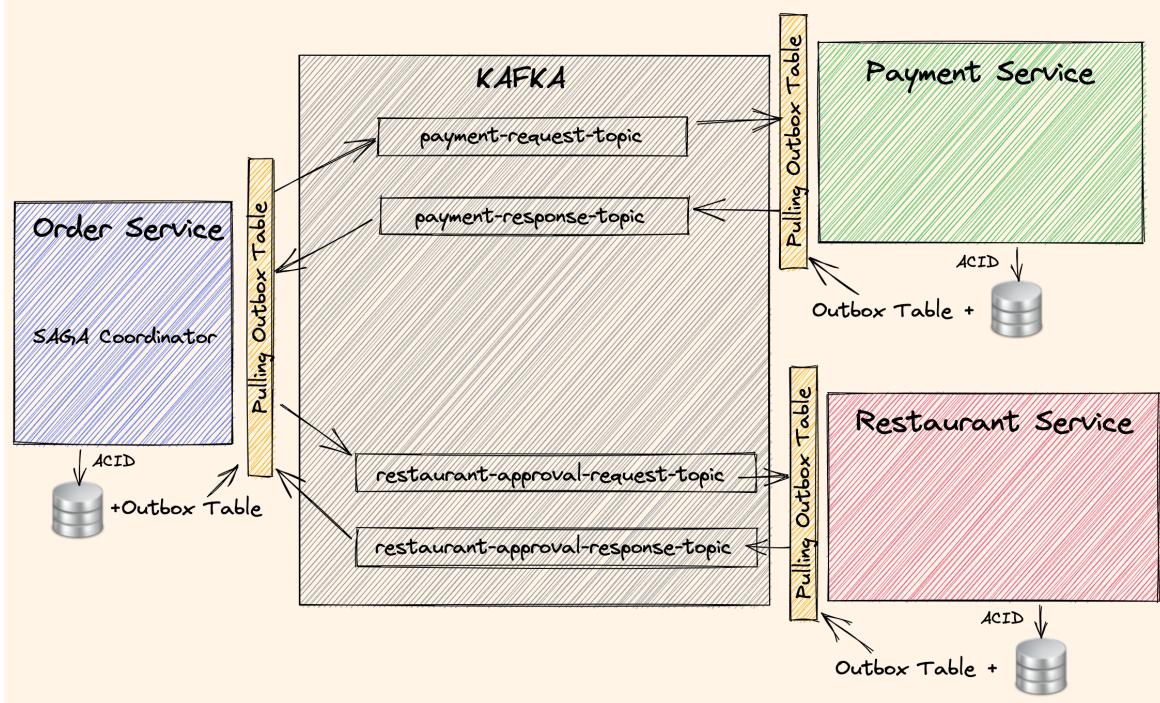
## SAGA PATTERN



## Structure of the Course

- Outbox: Help use of local ACID transactions to let consistent (eventual) distributed transactions. It will complete SAGA in a safe and consistent way.
- Command Query Responsibility Segregation (CQRS): Separate read and write operations. Better performance on read part using right technology for reading, and preventing conflicts with update commands. Scale each part separately. Leads to eventual consistency.
- Kubernetes (K8s) & Docker: Kubernetes is a container orchestration system that automates deployment, scaling and management of cloud native applications. It allows to run docker containers while reducing operational complexities.
- Google Cloud & Google Kubernetes Engine (GKE): Google's solution for Cloud computing. GKE is a managed, production-ready environment for running containerized applications.

## OUTBOX PATTERN



## CQRS PATTERN

