Event-Driven Programming

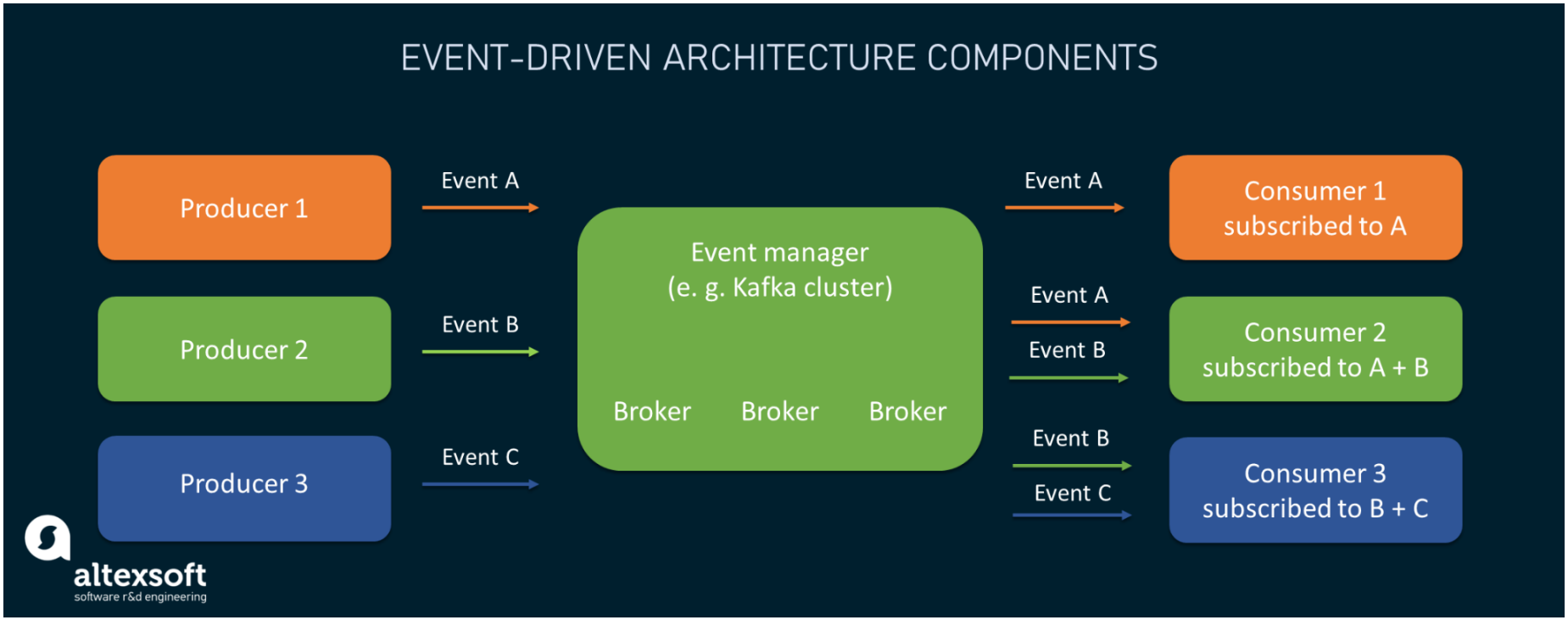
What is an Event?

REVATURE

- Event
  - An event is a change in state, or an update, like an item being placed in a shopping cart on an e-commerce website.
- Event Broker/Manager/Bus
  - Middleware that mediates the communication of event messages between producers and consumers using the various message exchange patterns

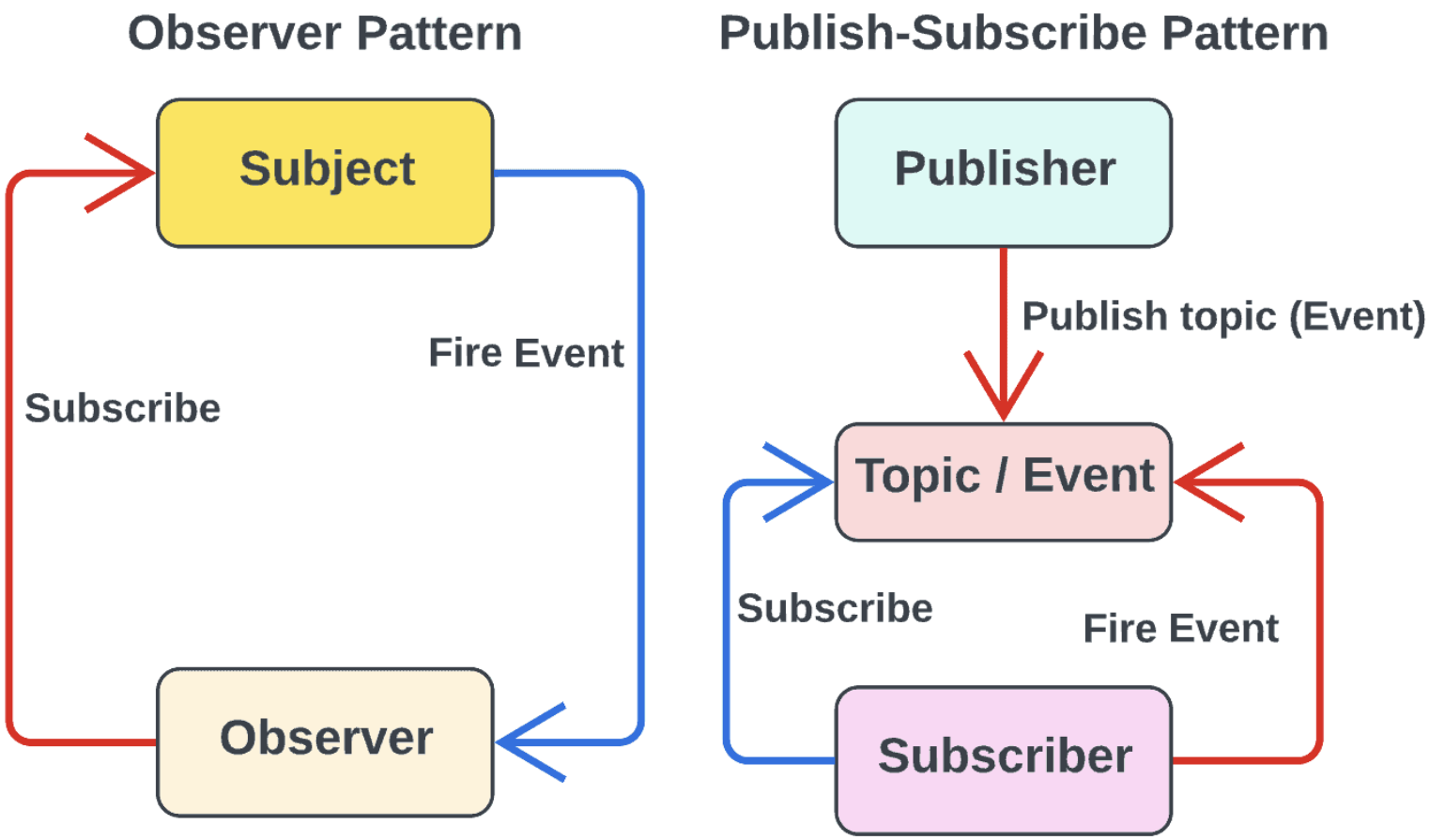
What is Event-Driven Architecture?

REVATURE



Pub/Sub and Observer Patterns

REVATURE



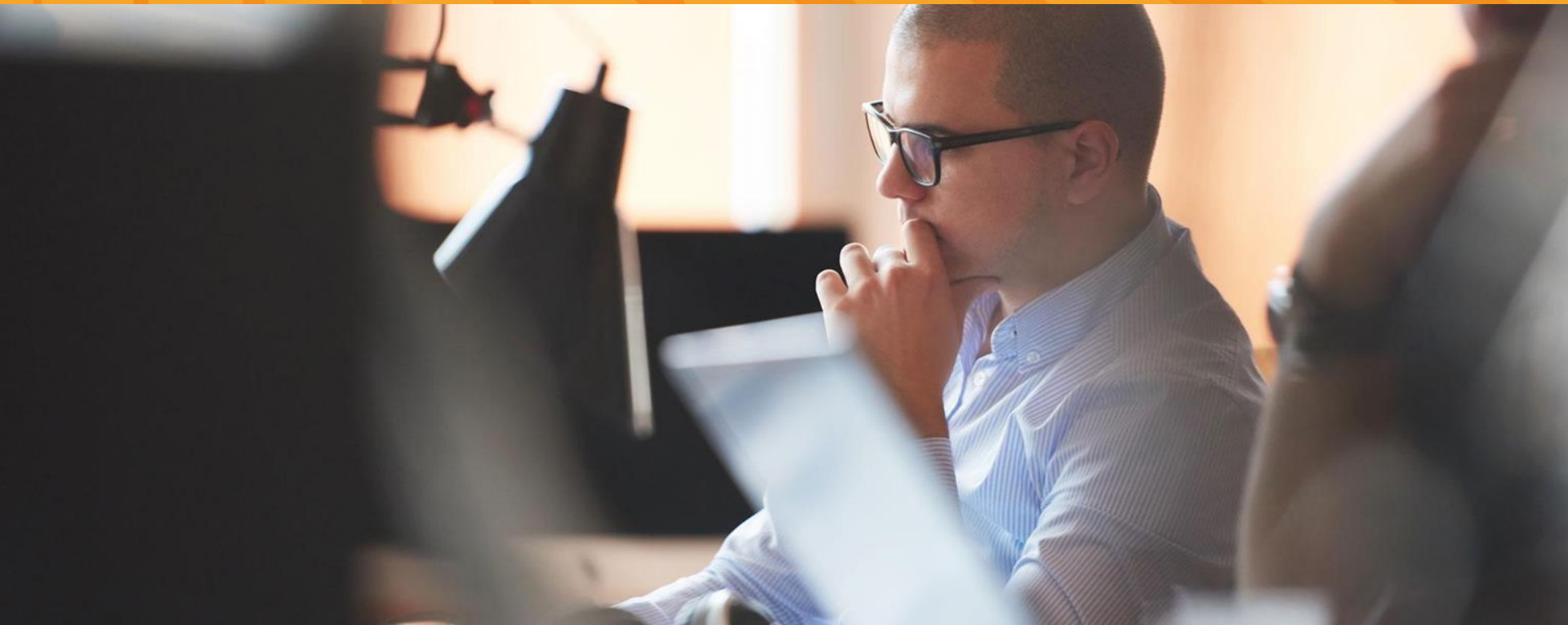
Kafka

REVATURE

- Kafka
  - Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.

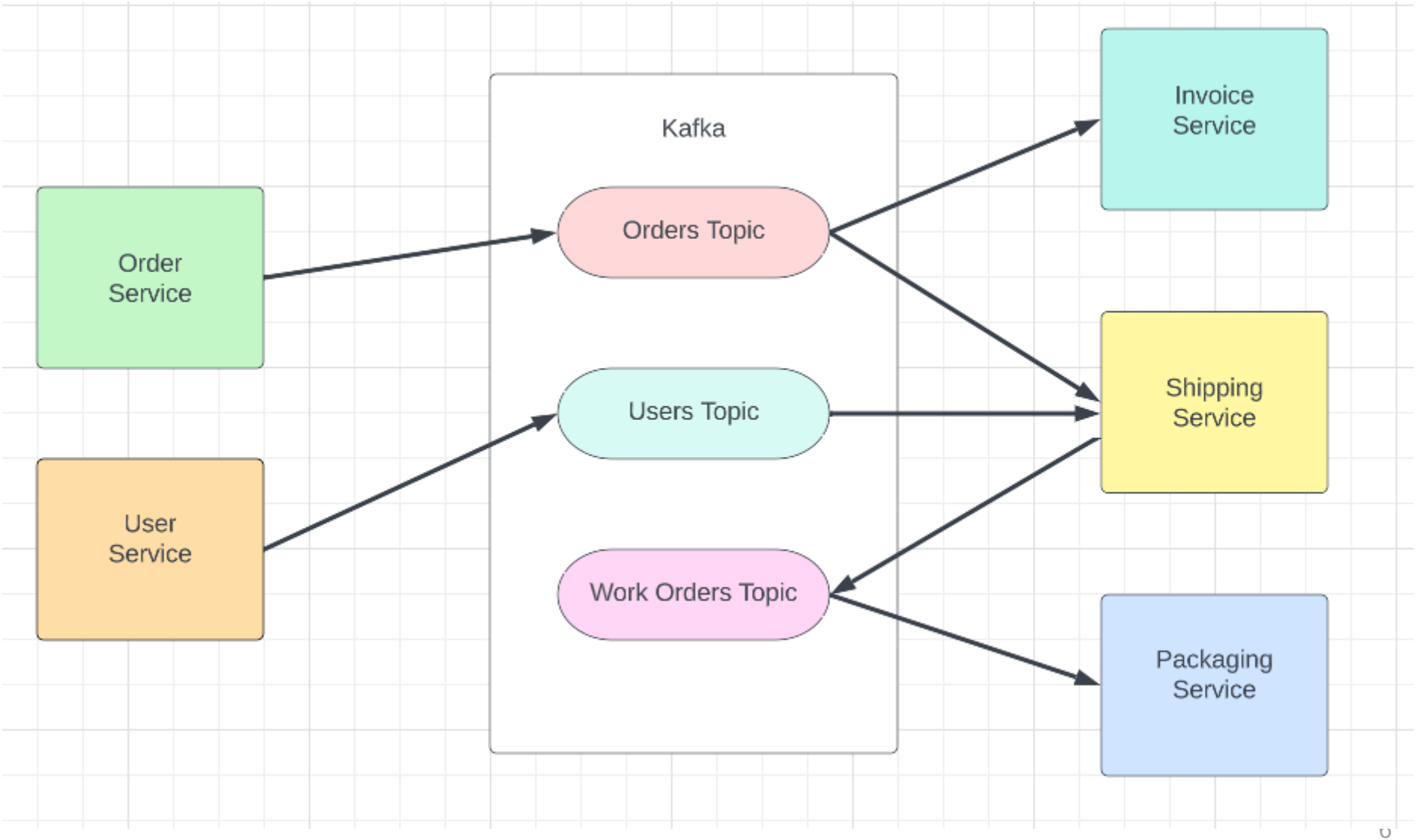
REVATURE

Basic Kafka Demo



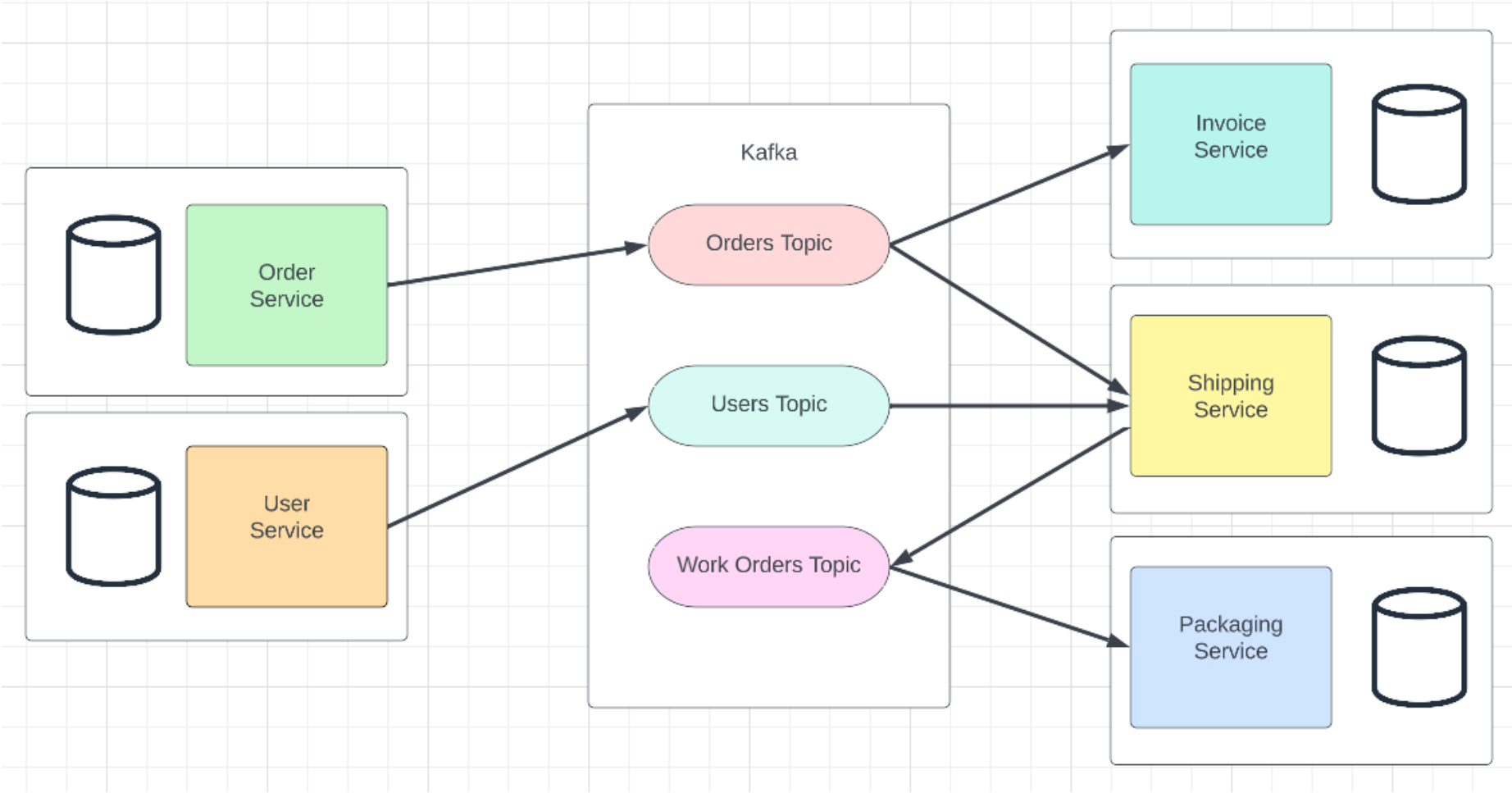
More Advanced Architecture

REVATURE



Microservices

REVATURE



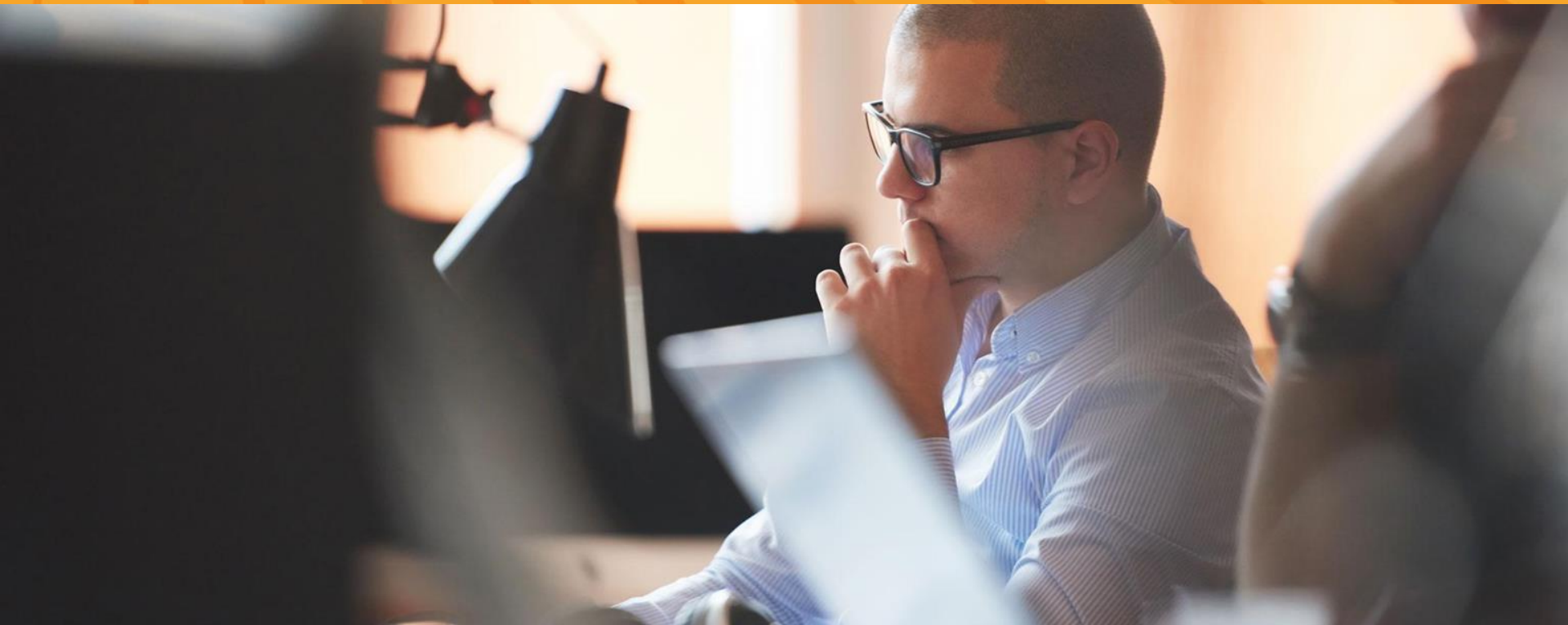
Advantages of Event Driven Architecture

REVATURE

- Agility
  - Everything happens as soon as possible and nothing is waiting on anything else.
- Scalability
  - You don’t have to consider what’s happening downstream, so you can add service instances to scale.
- Loosely-Coupled
  - To add another service, you can just have it subscribe to an event and have it generate new events of its own – there’s no impact on existing services.

REVATURE

Ecommerce Demo



Drawbacks of Event Driven Architecture

REVATURE

- Increased complexity
  - With too many events, producers and consumers associated with different business processes and workflows can be daunting to manage.
- Debugging and Troubleshooting Challenges
  - With the distributed and decoupled nature of event driven applications, it can be hard to trace an event from source to destination.
- Difficulties with Monitoring
  - Monitoring distributed, highly decoupled applications and systems can be trickier.