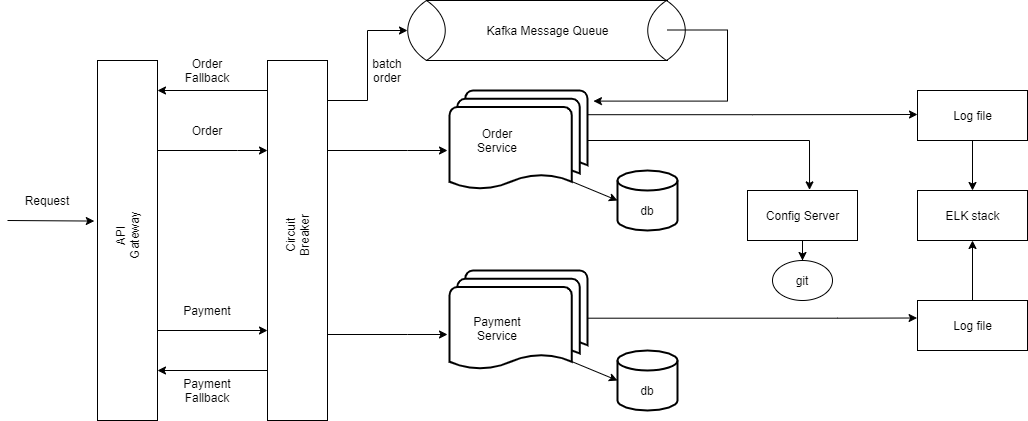
Goutham Krishna Reddy Sagam

[gsagam@buffalo.edu](mailto:gsagam@buffalo.edu)

+1-319-671-1796

# EGEN ECOMMERCE ORDER-SERVICE ARCHITECTURE

**System Architecture.**

****

1. Ecommerce application contains many requirements from customer displaying products, purchasing products, reviews, payment service. We can make all this services loosely coupled and make microservices out of it which is more beneficial and also offers effective solutions for scalability.
2. Every Id in the order table will be unique because of using universally unique identifier. Also, order-id is generated by encrypting customer’s unique email ID and current time stamp using MD5 algorithm.
3. Distributed architectures have many dependencies. If the application is not isolated from dependency failures or network failures, the application itself is at risk of being taken down. Netflix’s Hystrix or a Circuit Breaker helps by providing protection and control over latency and failures from dependencies, most commonly those accessed over network. It helps stop cascading failures and allows you to fail fast and rapidly recover.
4. Performance of bulk order is improved by sending the request event to a distributed message queue and running them instead of calling them synchronously. Also, We can use weighted load balancing techniques to prioritize client order requests increasing overall performance of the system.
5. In a microservices architecture, a request might call many other services and we should know the flow of the request if any error occurs in the middle of the system. So, for distributed tracing we can use Zipkin and Sleuth to monitor the requests. Also, for centralized logs we can use ELK stack. It makes a developer life easy to filter and figure out logs.
6. The idea of splitting a payment is debatable. However, We can give customer a choice using purchasing gift cards which can be used to split payment.