Introduction to Kafka Dead Letter Queue

Implementation in Golang

About Me

Patrick Maurits Sangian

@patricksangian
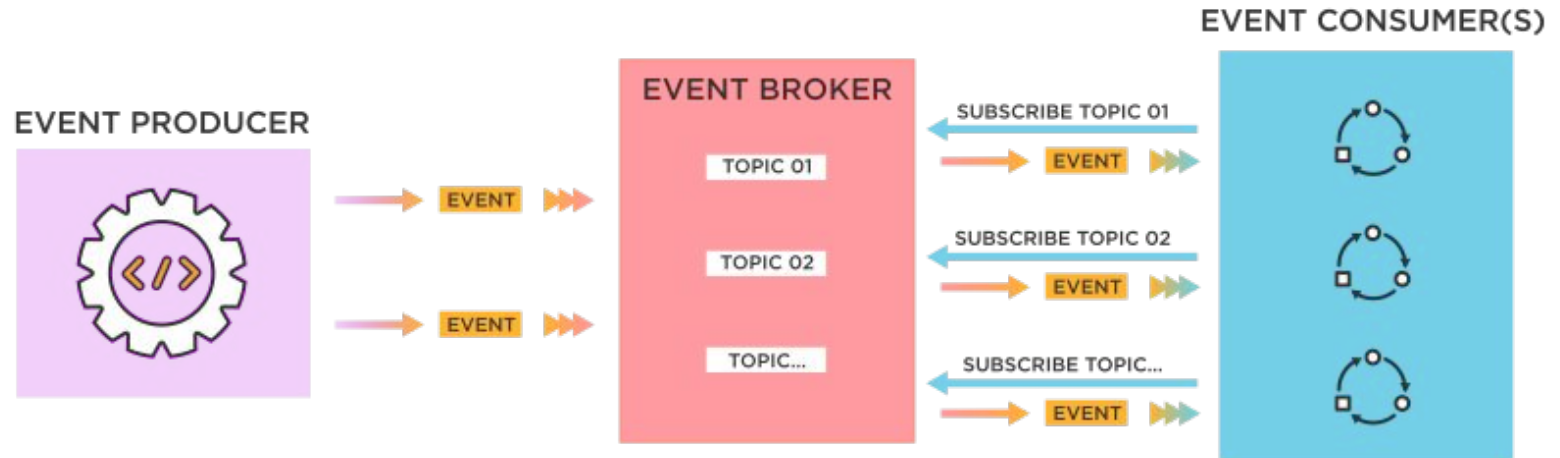
Software Engineer @ Telkom

Goals

- Understand about Event Driven Architecture
- Know what the DLQ is
- Know when to implement DLQ
- Have an exposure to Basic Monitoring and Alerting
- Understand how to build a simple demo in Golang

Event Driven Architecture

- Publisher / Producer
- Channel / Queue / Topic
- Subscriber / Consumer
- Event Bus / Message Broker
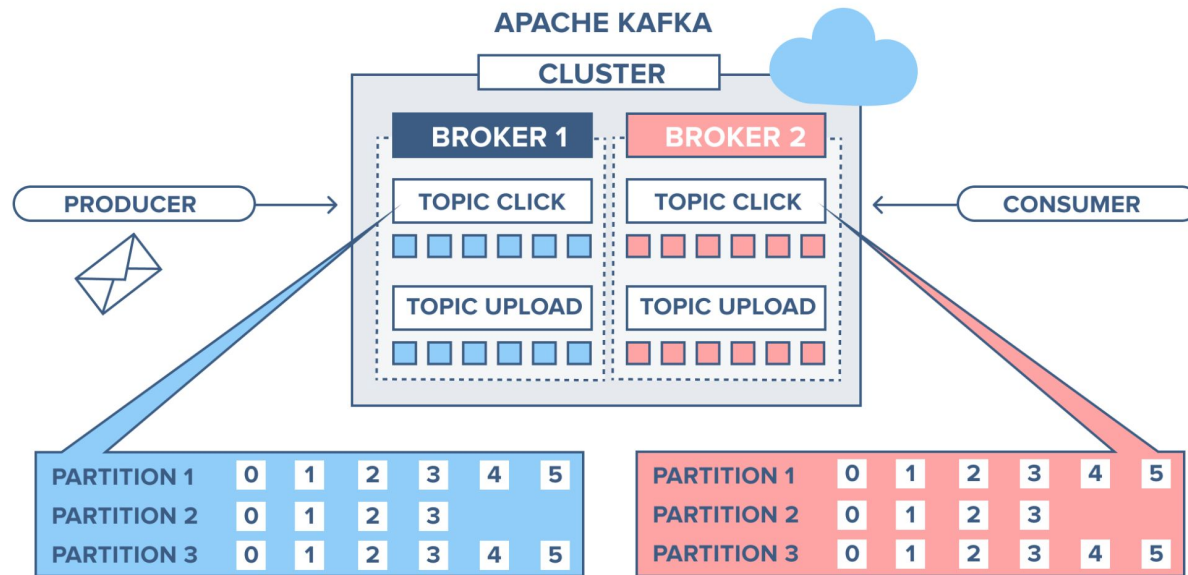
Event Driven Architecture

Kafka

- Message Broker
- Streaming Platform
- Components
  - Cluster
  - Broker
  - Topic
  - Partition
  - Offset
- Equivalent
  - NSQ, NATS, Google PubSub, RabbitMQ, Amazon SQS, etc
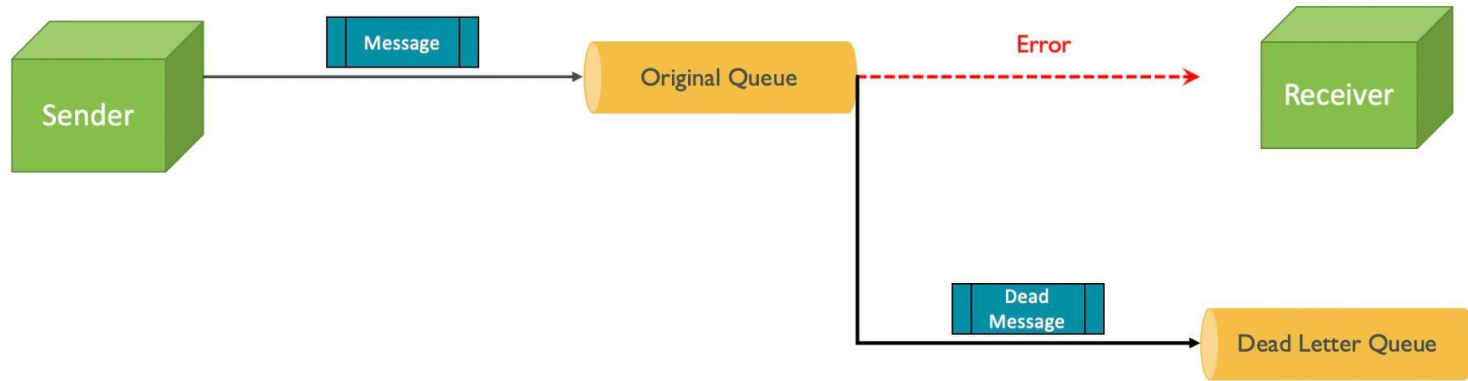
Kafka

Dead Letter Queue

- Queue that store all the unprocessed messages after received by the consumer
- Why the message did not processed by the consumer ?
    - Invalid Content Type
    - Eligibility of data
    - Due to server error such as database down, unreachable service, etc

Dead Letter Queue
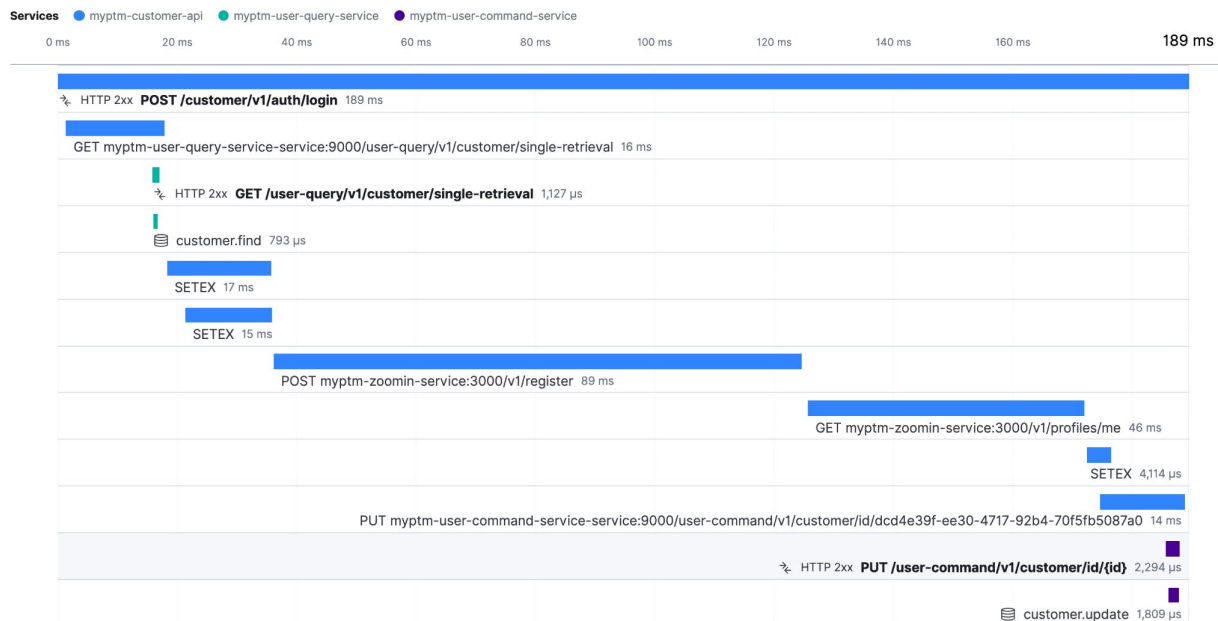
Don't Need DLQ If …

- There's no event driven architecture
- You implement Orchestration Pattern
- There's no guarantee the data / message will be delivered

Monitoring & Alerting

- Monitoring
  - Prometheus
  - Grafana
  - ELK Stack (Elasticsearch, Logstash, Kibana, APM)
  - New Relic
- Alerting
  - PagerDuty
  - Sentry.io
  - Slack/Discord/Telegram

Monitoring & Alerting

Services ● myptm-customer-api ● myptm-user-query-service ● myptm-user-command-service

0 ms	20 ms	40 ms	60 ms	80 ms	100 ms	120 ms	140 ms	160 ms	**189 ms**

⤳ HTTP 2xx **POST /customer/v1/auth/login** 189 ms

GET myptm-user-query-service-service:9000/user-query/v1/customer/single-retrieval 16 ms

⤳ HTTP 2xx **GET /user-query/v1/customer/single-retrieval** 1,127 µs

🗄 customer.find 793 µs

SETEX 17 ms

SETEX 15 ms

POST myptm-zoomin-service:3000/v1/register 89 ms

GET myptm-zoomin-service:3000/v1/profiles/me 46 ms

SETEX 4,114 µs

PUT myptm-user-command-service-service:9000/user-command/v1/customer/id/dcd4e39f-ee30-4717-92b4-70f5fb5087a0 14 ms

⤳ HTTP 2xx **PUT /user-command/v1/customer/id/{id}** 2,294 µs

🗄 customer.update 1,809 µs

Monitoring & Alerting

Errors

Group ID	Error message and culprit	Occurrences	Latest occurrence ↓
5760e	{"Err":null,"ErrMessage":"","Data":null,"PartialMeta":null,"Status (*successDeliveryOrderEventHandler).HandleMessage	11	11 days ago
f7626	Point is under 1 (*deliveryOrderCommand).AddPoint	11	11 days ago
4ee85	KafkaError: Code: Local: All broker connections are down | Message (*KafkaConsumerAdapter).logErr	1	20 days ago
68b1f	{"Err":{},"ErrMessage":"Error 1366: Incorrect string value: '\\xF0 (*loggedInCustomerEventHandler).HandleMessage	1	11 days ago
88323	{"Err":{},"ErrMessage":"Error 1366: Incorrect string value: '\\xF0 (*customerCommand).InsertOne	1	11 days ago
8d99c	Application is running on port 9000 main	1	9 days ago
cc28a	Error 1366: Incorrect string value: '\xF0\x9F\x8C\xB1pa...' for co (*customerMysqlCommand).Insert.func1	1	11 days ago

Rows per page: 25 ∨

🐮 : By default DLQ Processing isn't available in Kafka.

🐒 : What ? Then, why we go so far ?

🐮 : Wait bro. If it's not available, why we don't just build it ?

🙈 : Did you mean diving into the code ?

🐮 : Correct !

Codes

- Dead Letter Queue Processing Service
- Source Code: https://github.com/sangianpatrick/go-kafka-dlq-demo

Example

Kafka Cluster

Produce/Consume

Worker

DLQ Processing Service

Produce/Consume

Produce/Consume

webhook

DLQ Handler

DLQ Handler

Order Service

Payment Service

Payment Gateway / Bank