Kafka Fundamentals

About

- ~10 lat w branży
- ~4 lata kontaktu z Kafką
- używana komercyjnie łącznie od 3 lat
- Scala od 4 lat
- Confluent Developer exam
- I like to break stuff

A Wy... ?

- Imię
- coś o sobie
- doświadczenie z Kafką?

Organizacja ćwiczeń

- Na przemian slajdy i ćwiczenia praktyczne
- Zachęcam do udziału i rozwiązywania przykładów
- Przerwy po sekcjach
- Gotowe rozwiązania w repo gdyby coś nie działało

Organizacja ćwiczeń

Tutaj zaczynamy

0. Intro

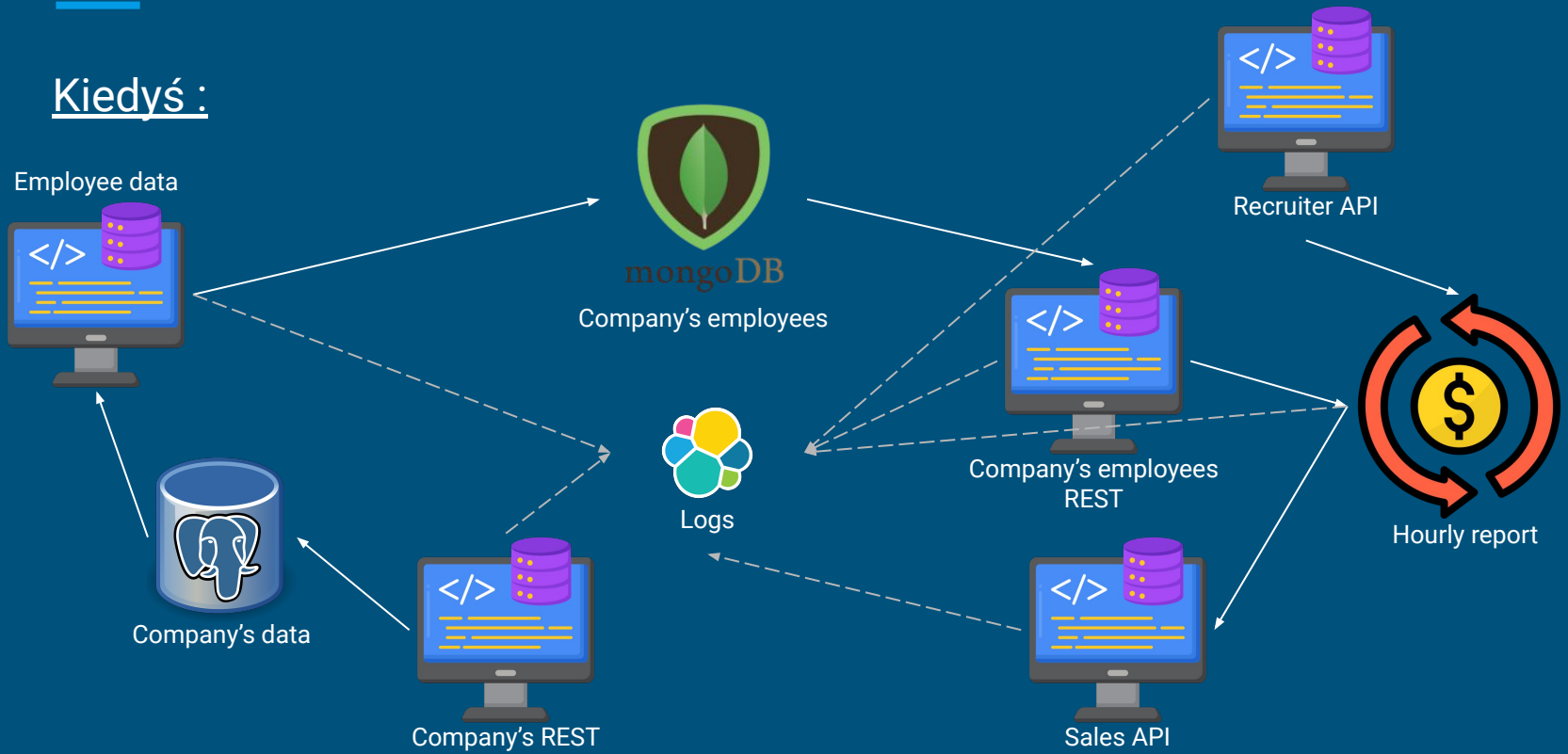
0. Intro

1. Apache Kafka - Jak powstała, jaki problem rozwiązuje?
2. Dlaczego warto uczyć się Kafki?
3. **Ex 0**: Stawianie środowiska, zapoznanie z domeną
4. Podstawowe komponenty: Broker, Topic, Producer, Consumer
5. **Ex 1**: Piszemy Producera
6. **Ex 2**: Piszemy Consumera
7. Topic cz.1
8. Przerwa!
9. Replication & Sharding
10. Kafka vs. Others

Kafka - Jak powstała, jaki problem rozwiązuje?

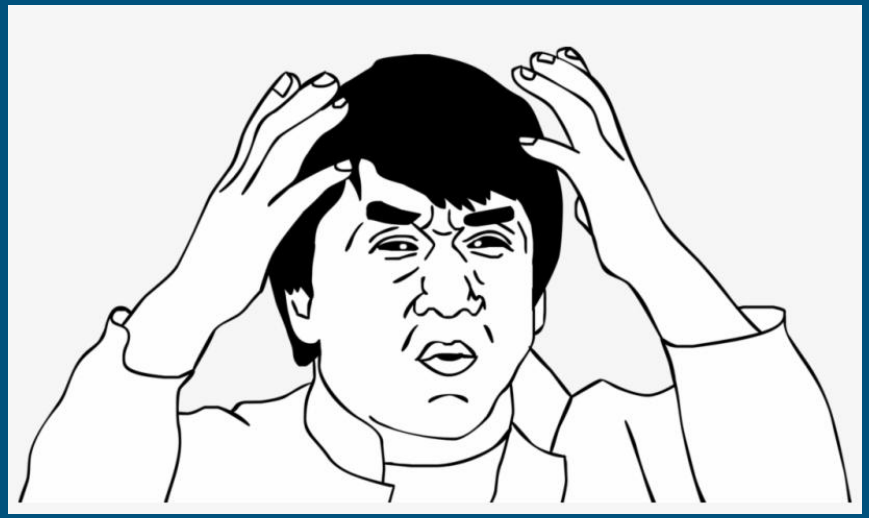
Kafka - Jak powstała, jaki problem rozwiązuje?

Kiedyś :



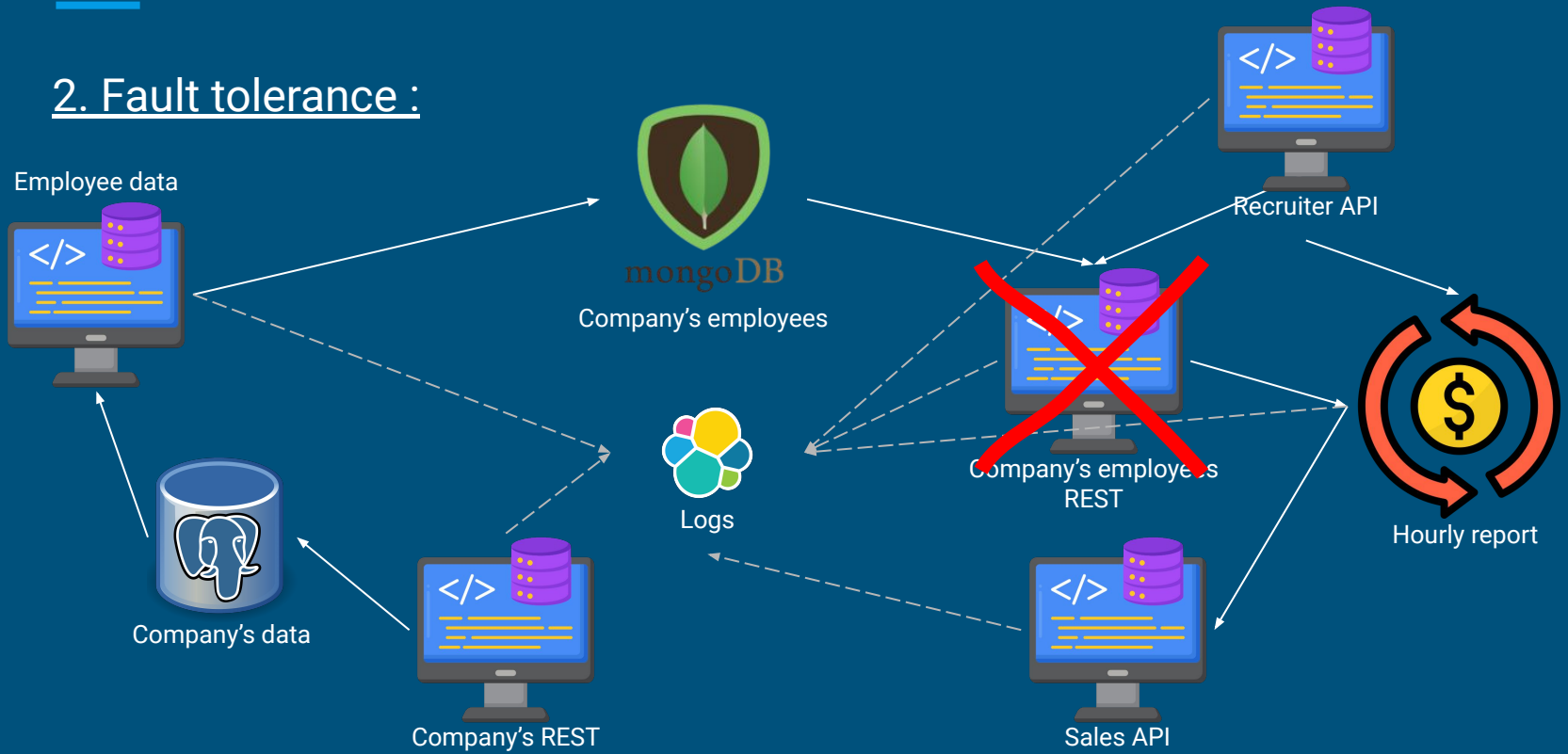
Kafka - Jak powstała, jaki problem rozwiązuje?

1 WTF :



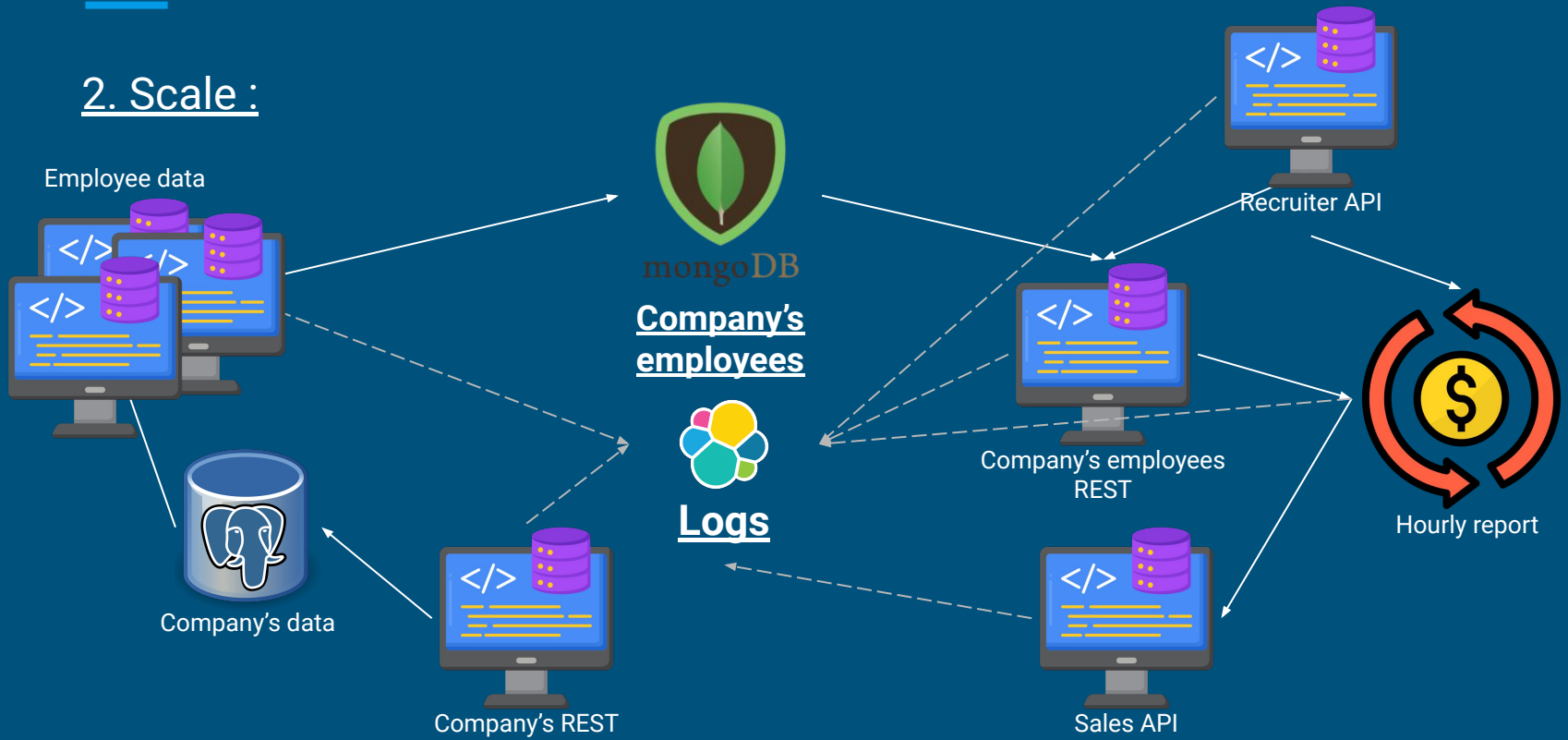
Kafka - Jak powstała, jaki problem rozwiązuje?

2. Fault tolerance :



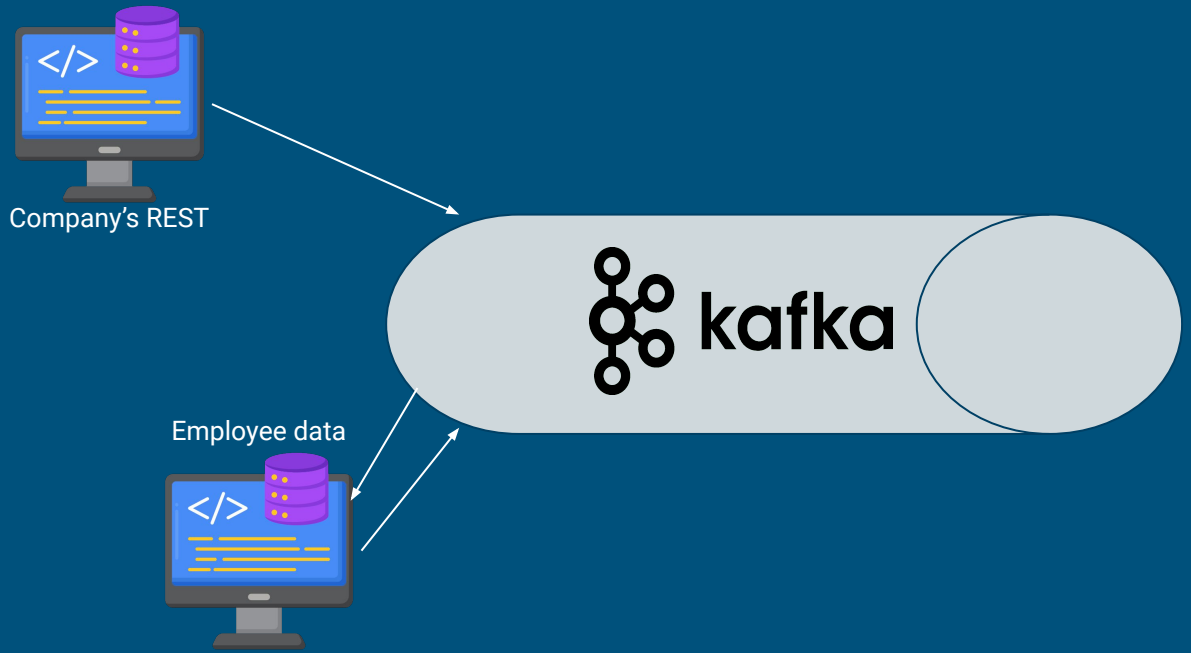
Kafka - Jak powstała, jaki problem rozwiązuje?

2. Scale :



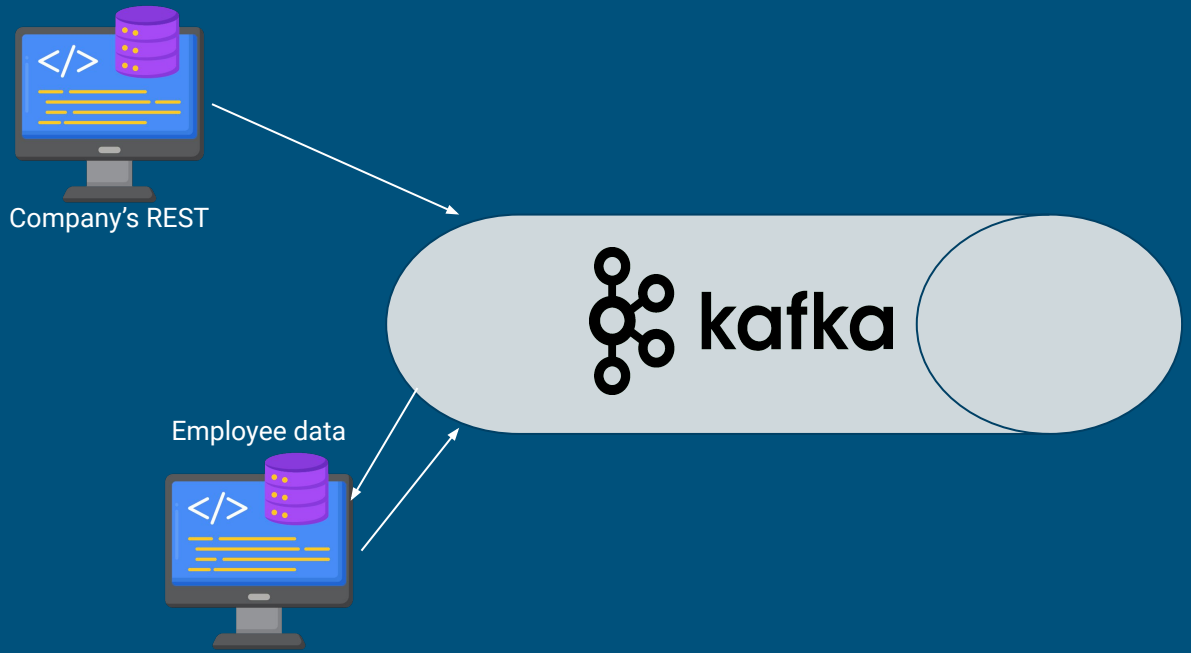
Kafka - Jak powstała, jaki problem rozwiązuje?

Dziś :



Kafka - Jak powstała, jaki problem rozwiązuje?

Dziś :



Fault tolerance

Scale

Dlaczego warto uczyć się Kafki?

Dlaczego warto uczyć się Kafki?

1. data analytics
    a. ML
    b. Bolt
2. Microservices
    a. czyli to o czym rozmawialiśmy
3. Low Latency Streaming
    a. Netflix, Uber,
4. IOT

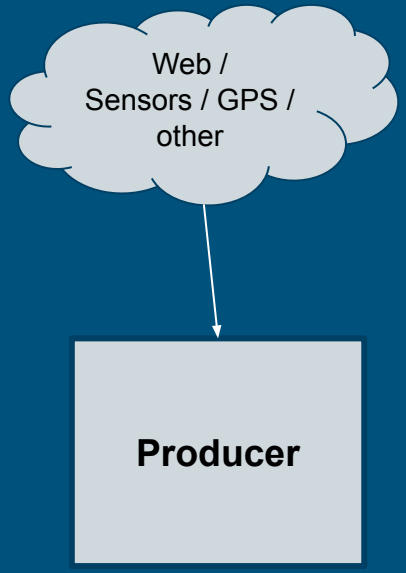
Ex 0

Stawianie środowiska, zapoznanie z domeną

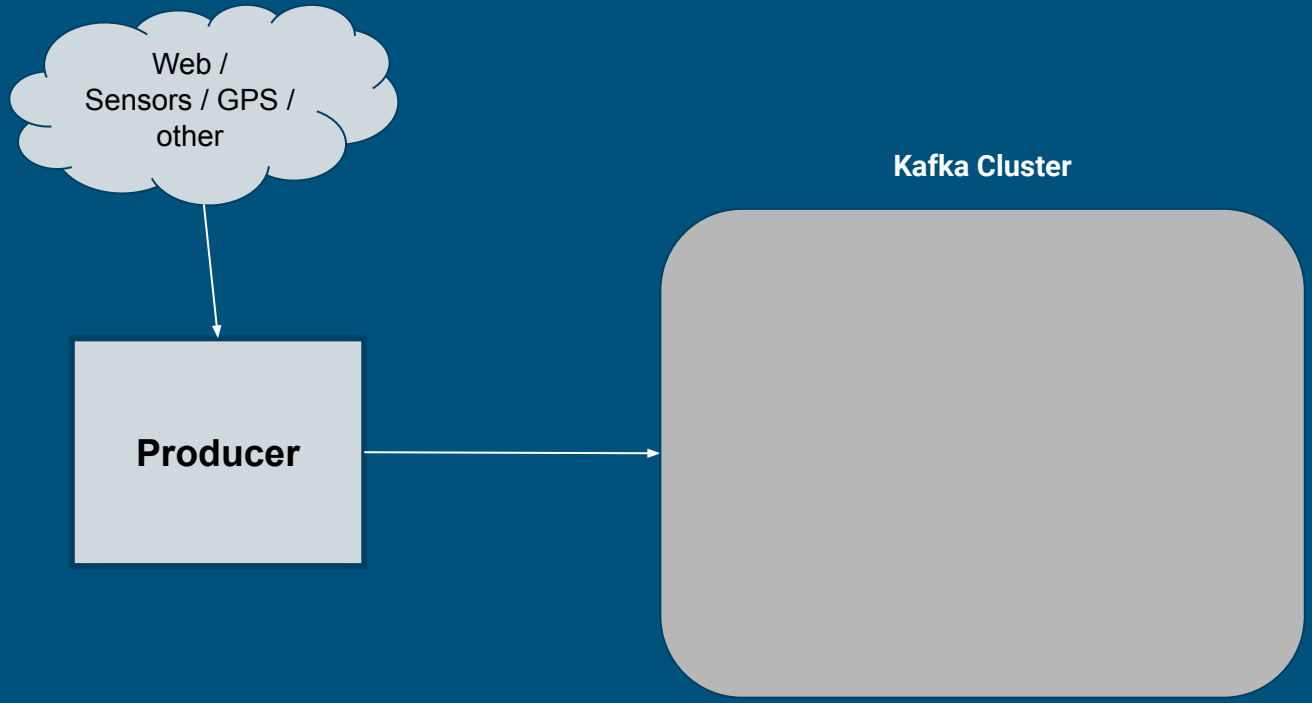
Podstawowe komponenty

Podstawowe komponenty

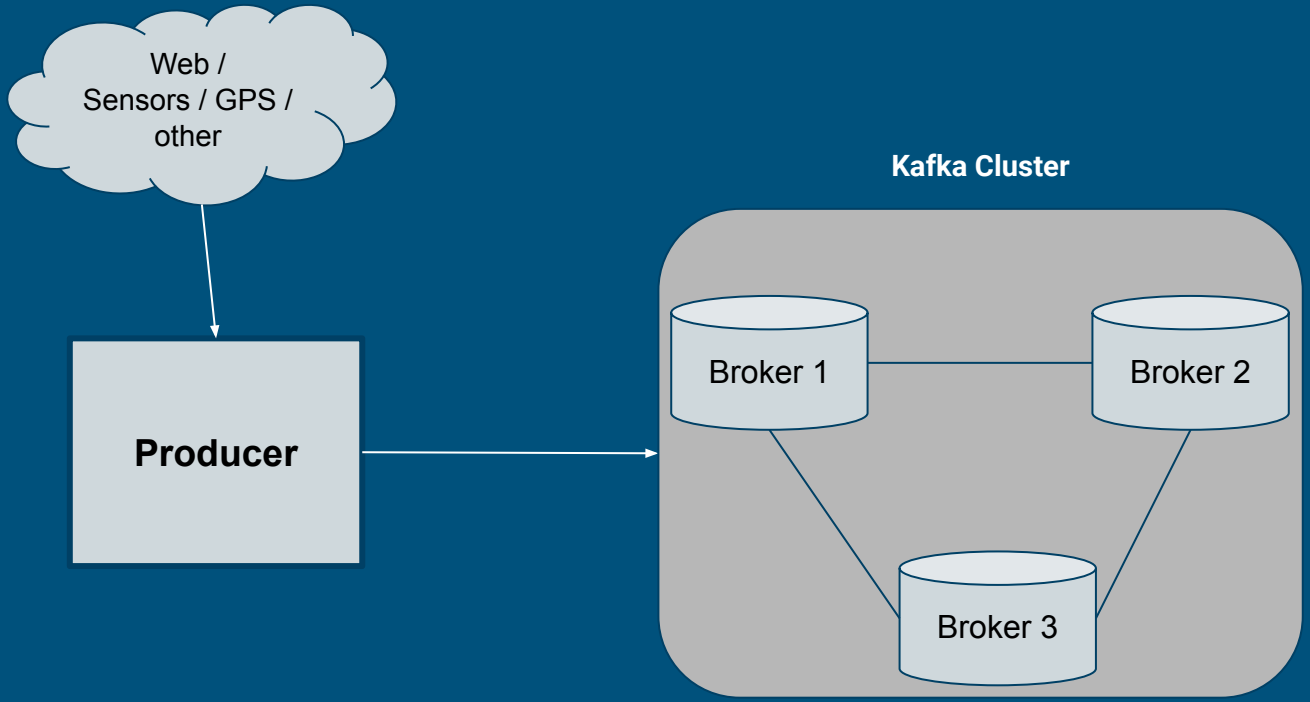
Podstawowe komponenty



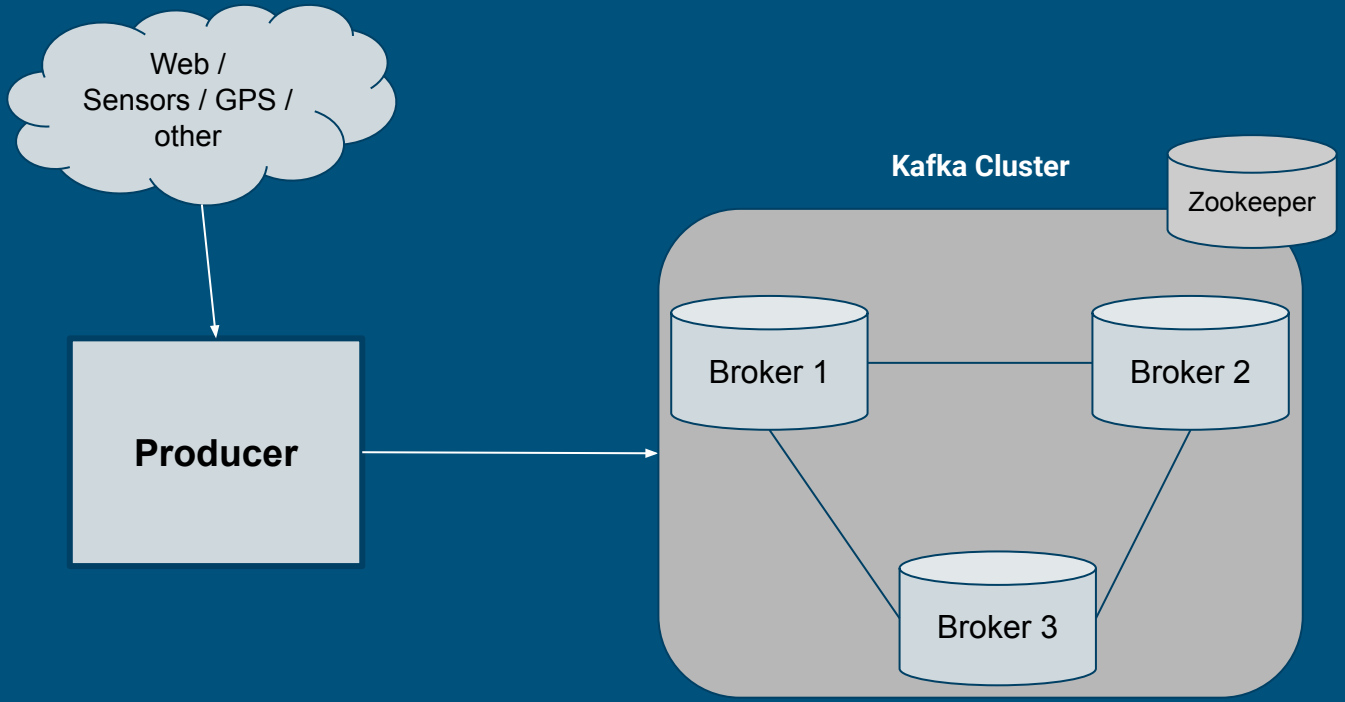
Podstawowe komponenty



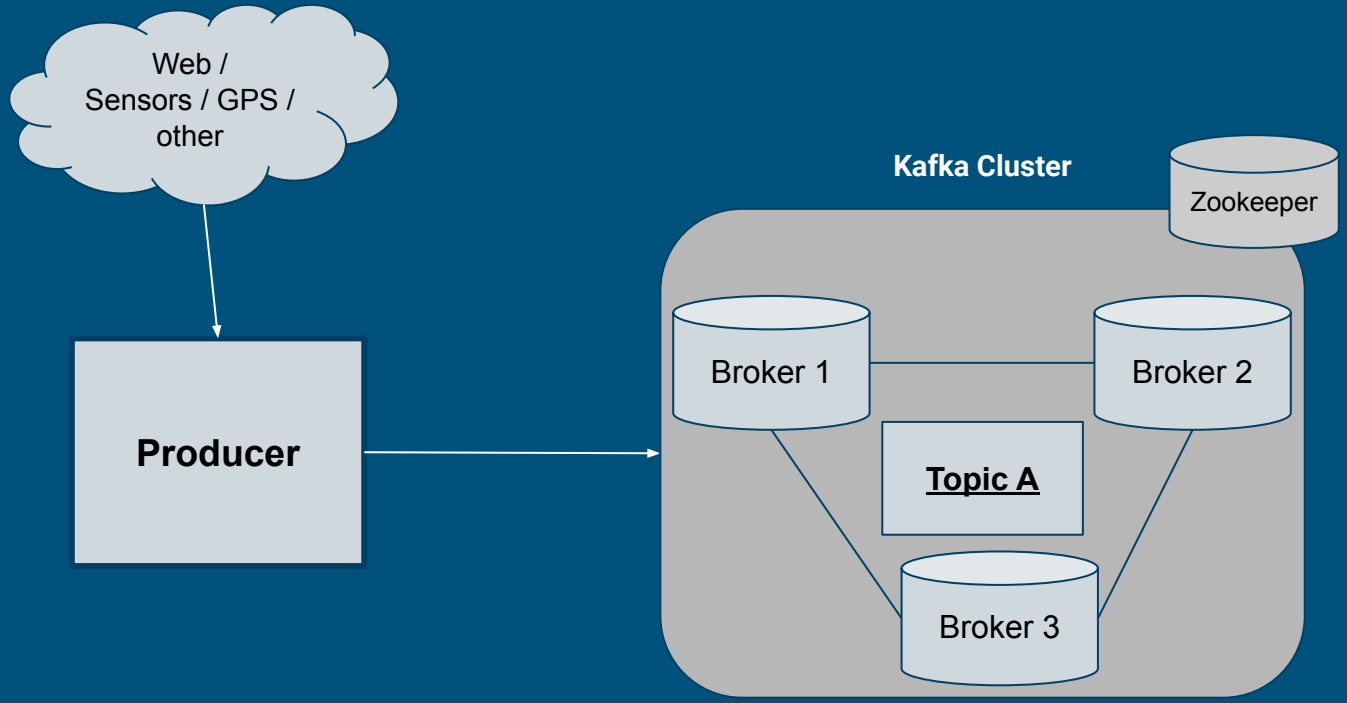
Podstawowe komponenty



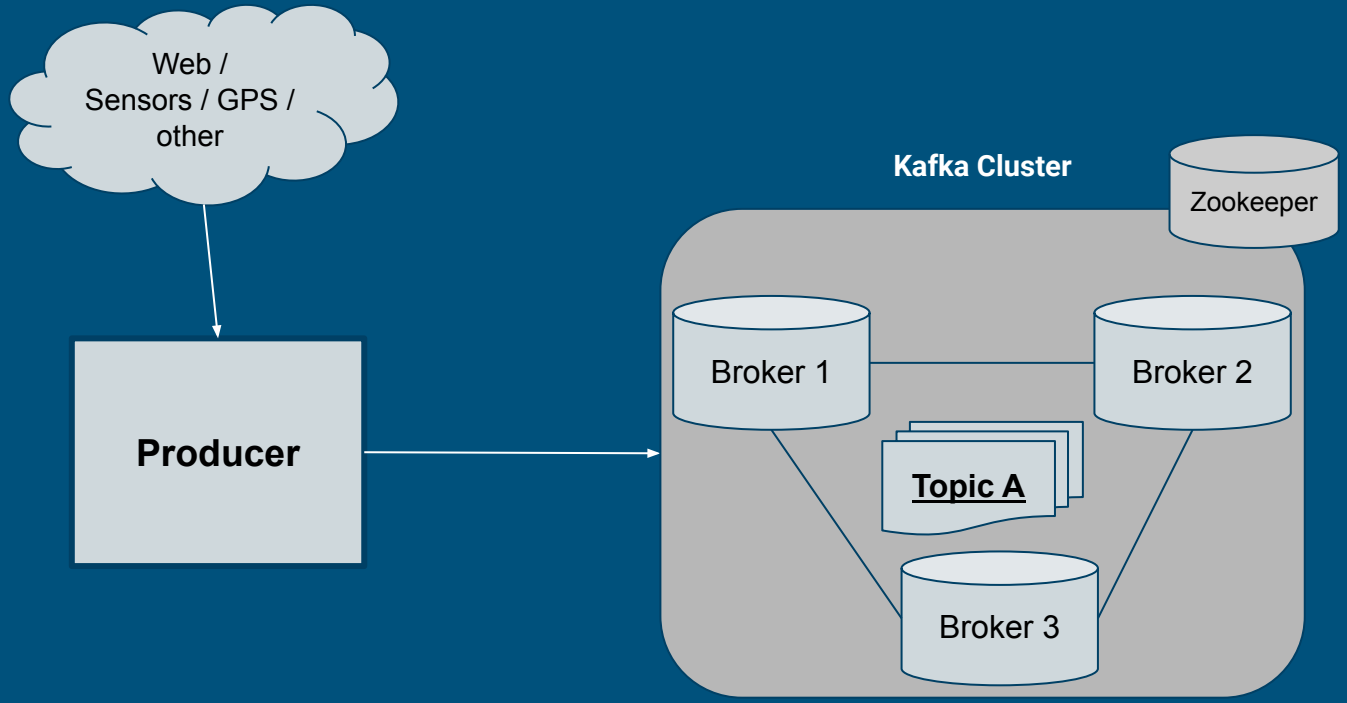
Podstawowe komponenty



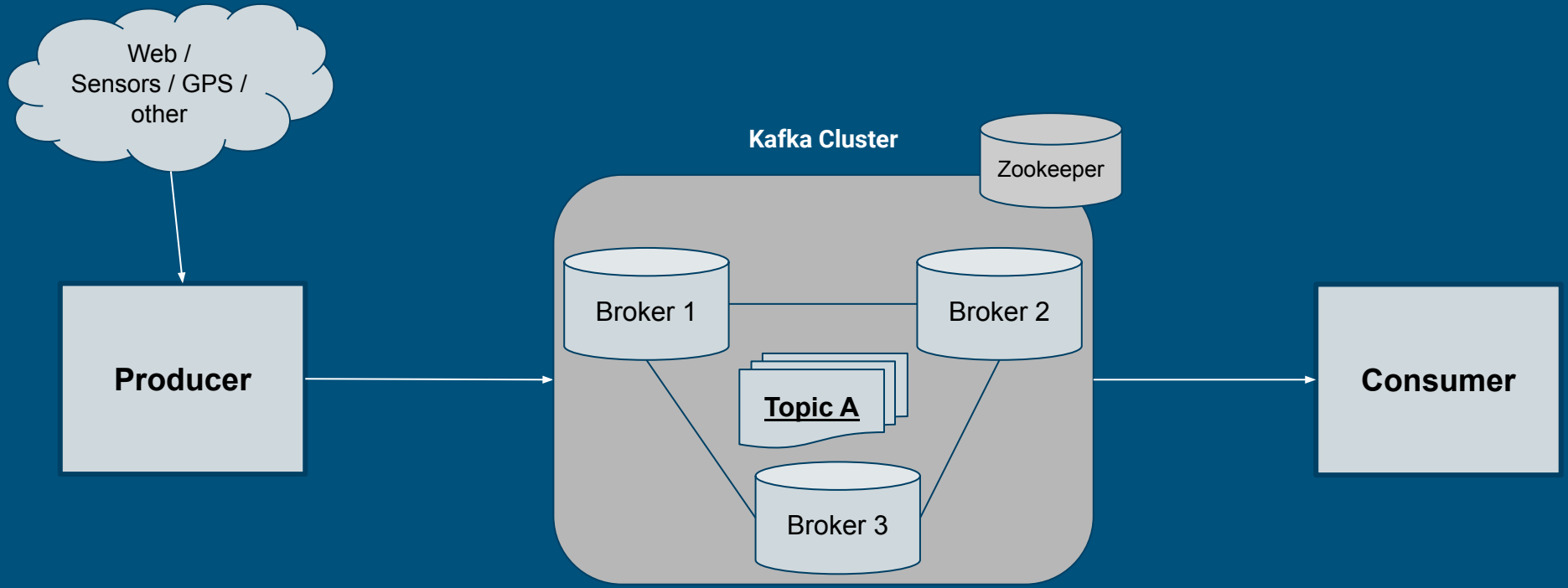
Podstawowe komponenty



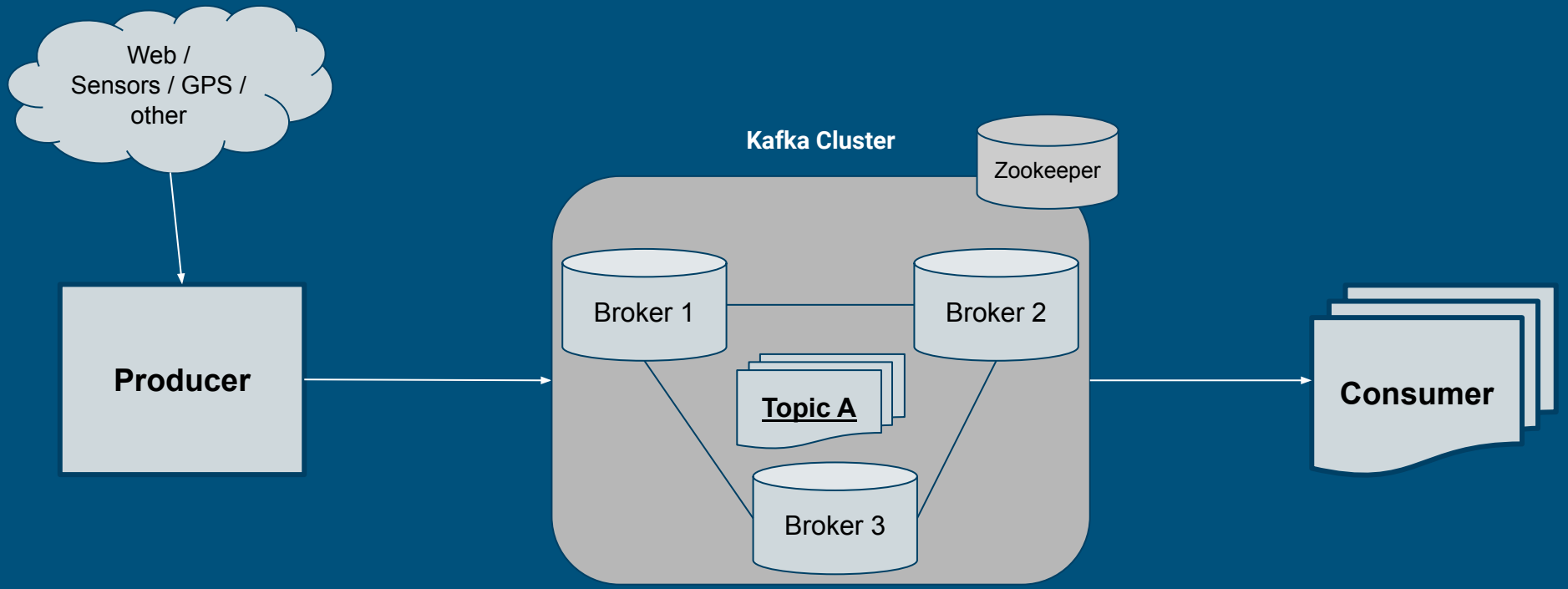
Podstawowe komponenty



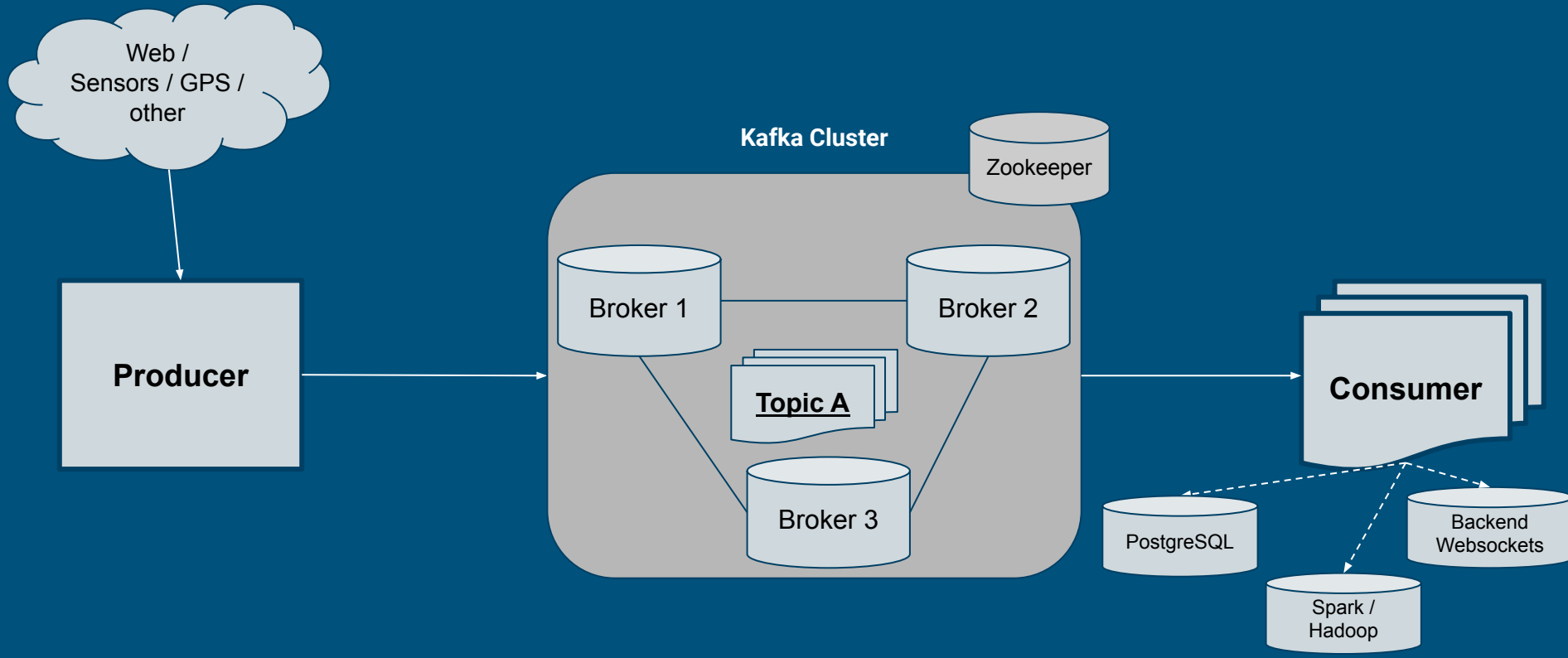
Podstawowe komponenty



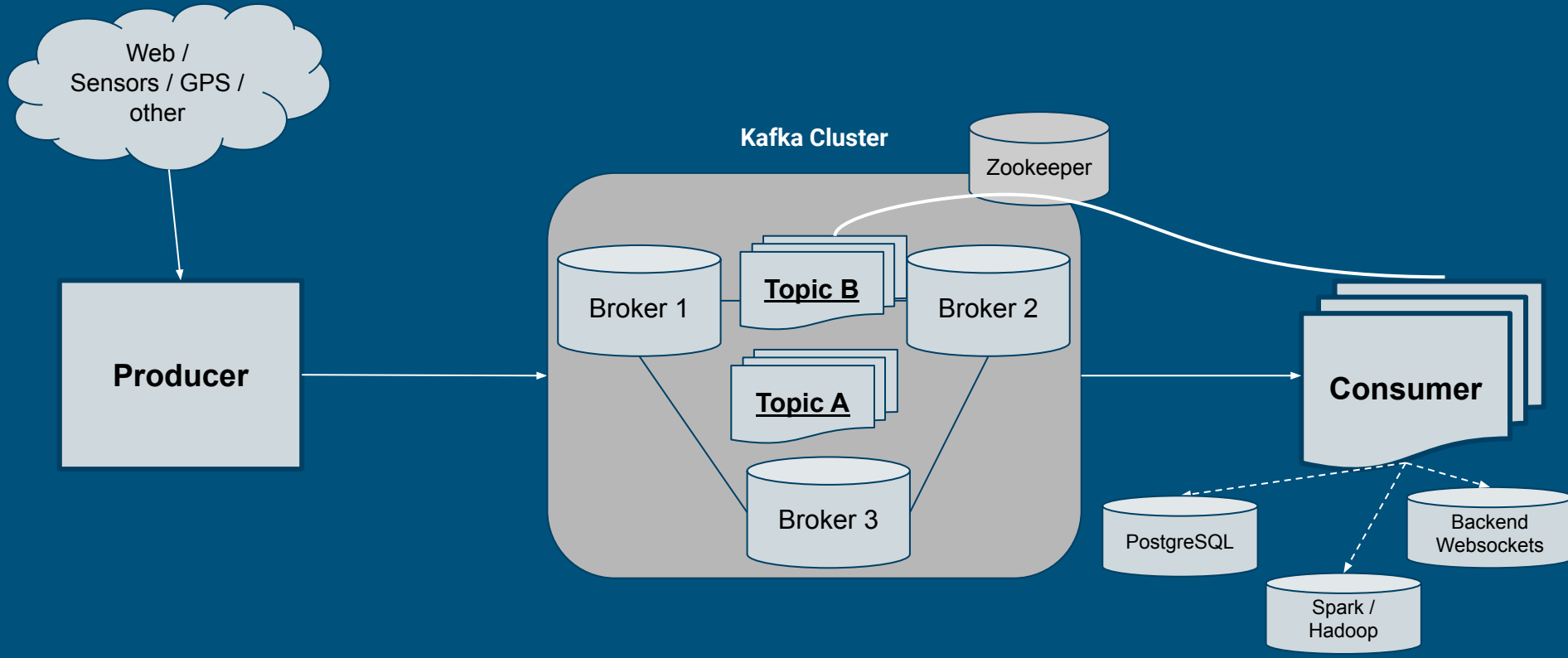
Podstawowe komponenty



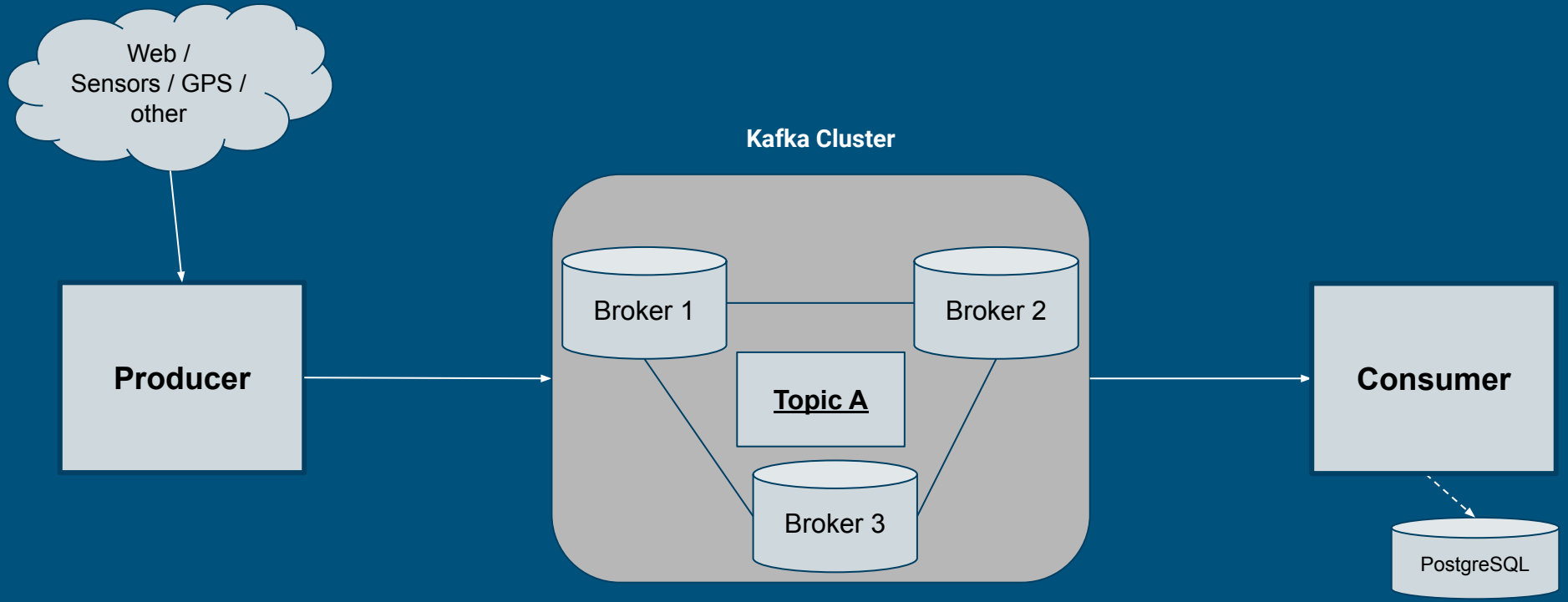
Podstawowe komponenty



Podstawowe komponenty



Podstawowe komponenty



Ex 1

Piszemy prostego Producera

Ex 2

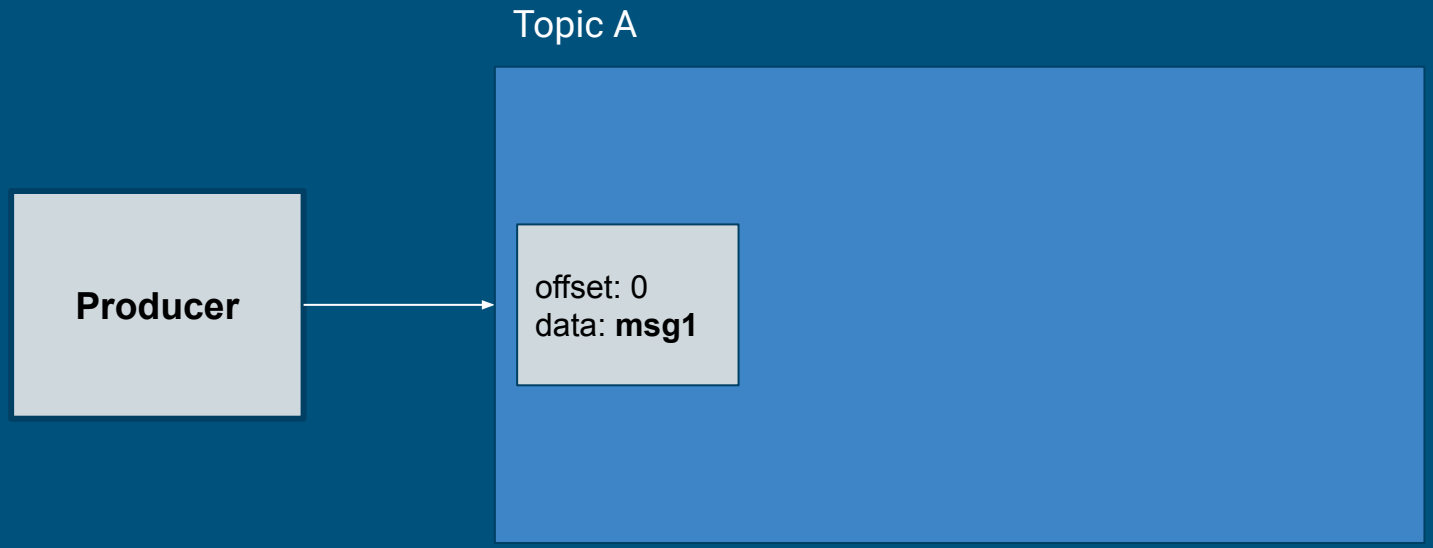
CLI: Jak przeglądać dane na topicu? i inne podstawowe operacje

Ex 3

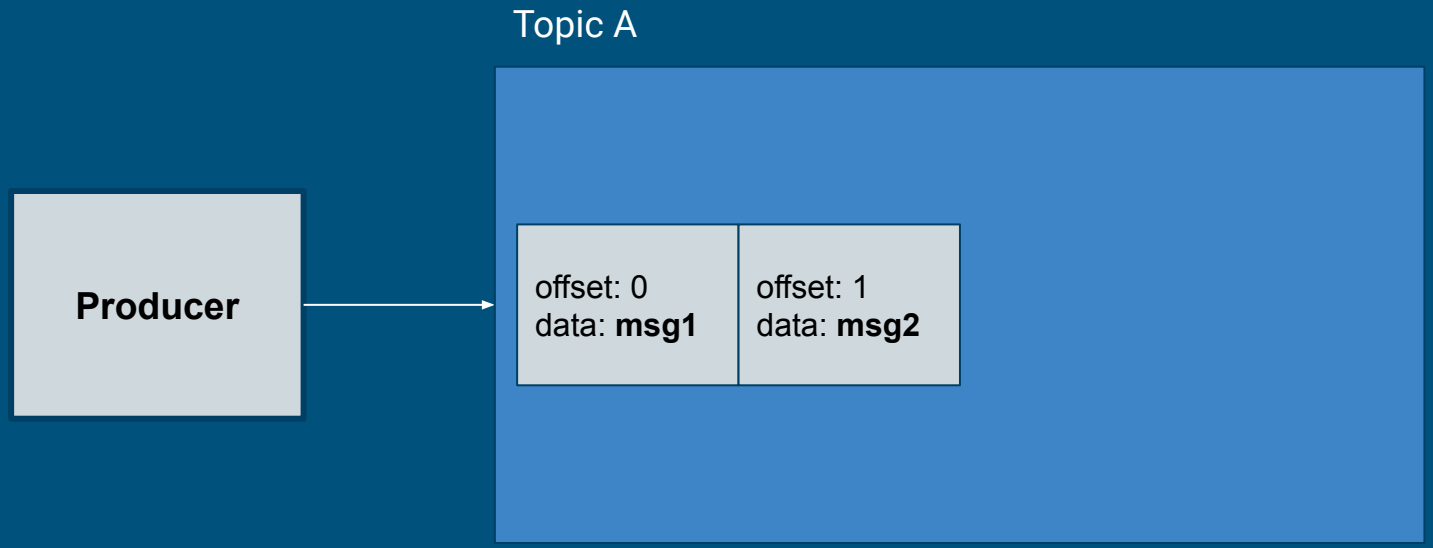
Piszemy prostego Consumera

Topic cz.1

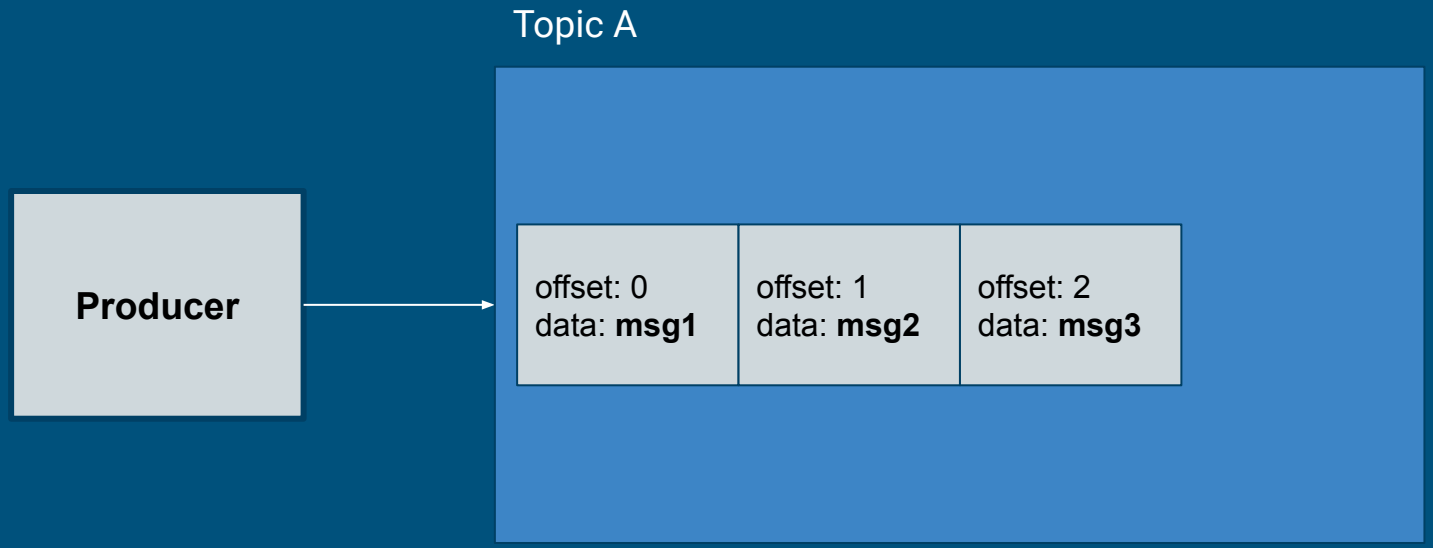
Topic cz. 1



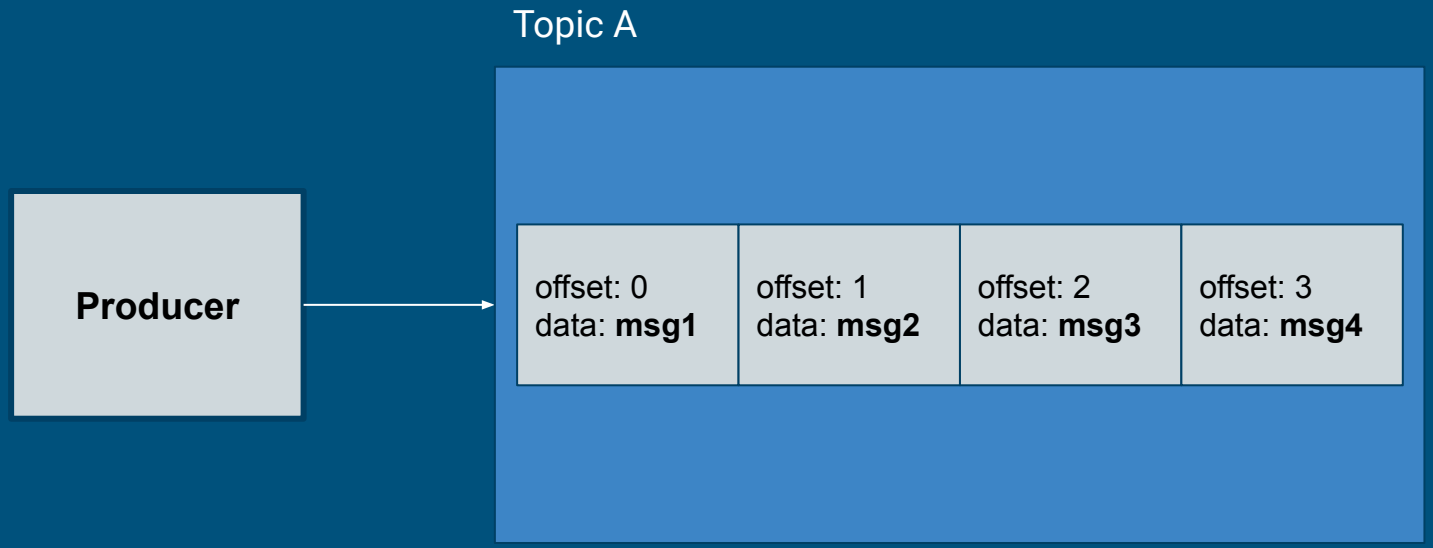
Topic cz. 1



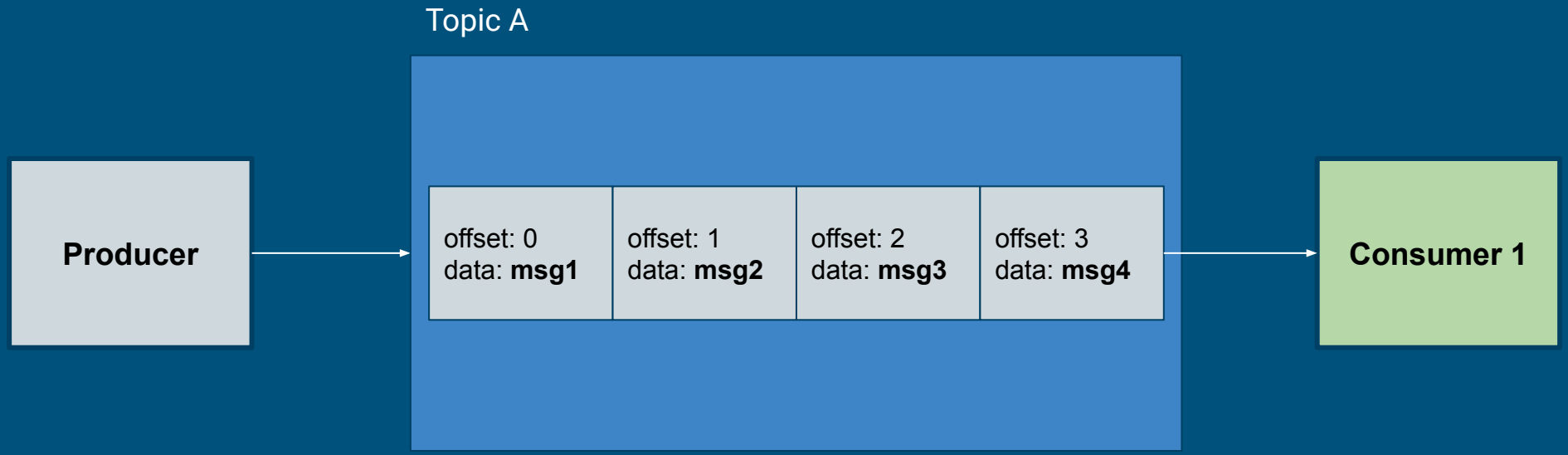
Topic cz. 1


Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3

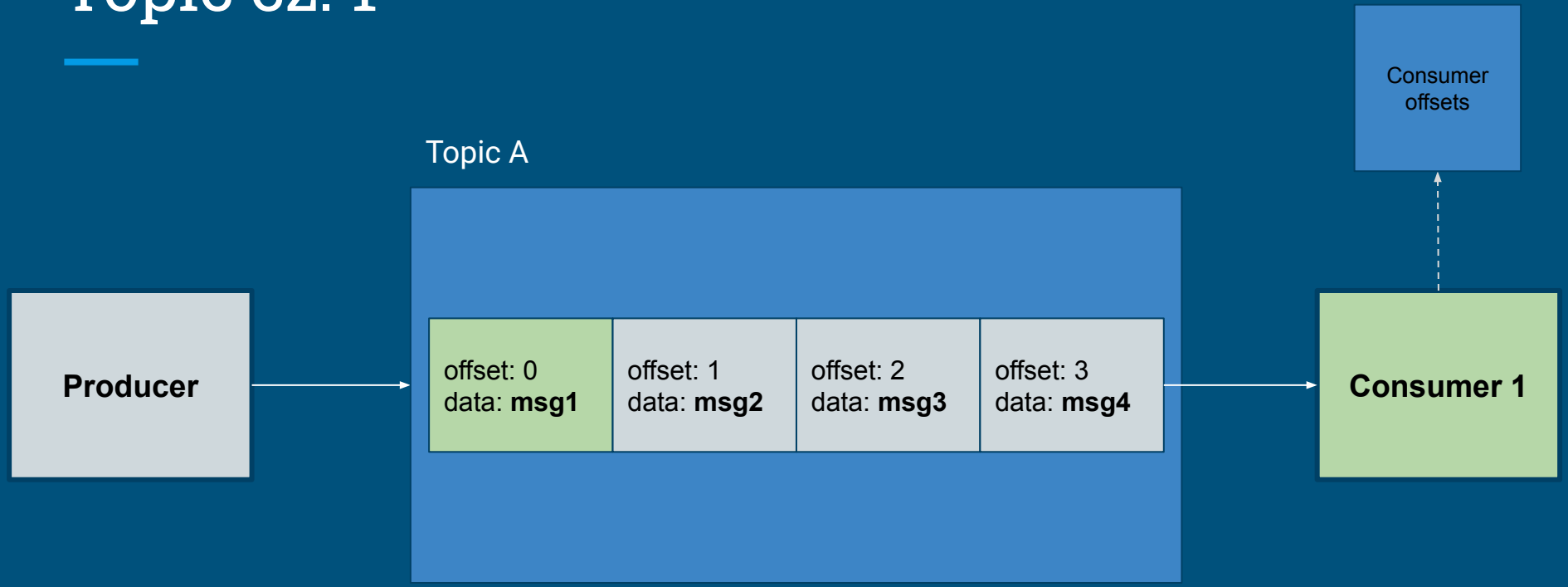
Topic cz. 1


Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4

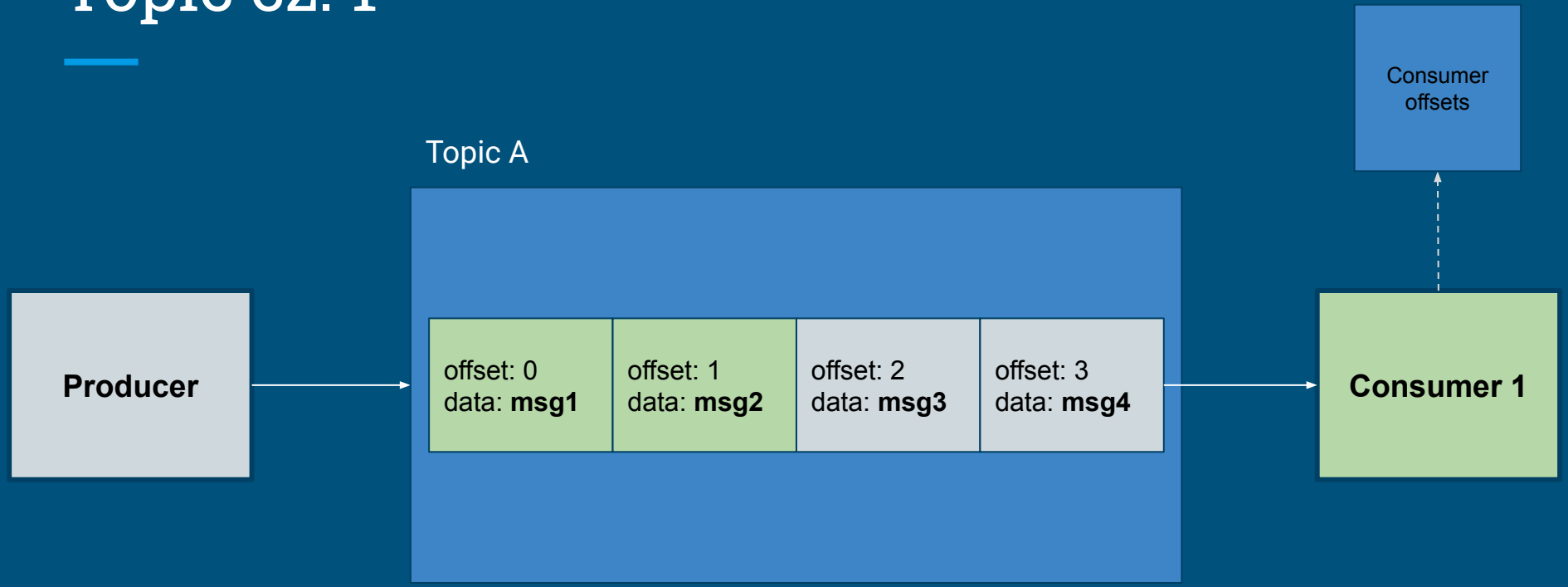
Topic cz. 1


Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

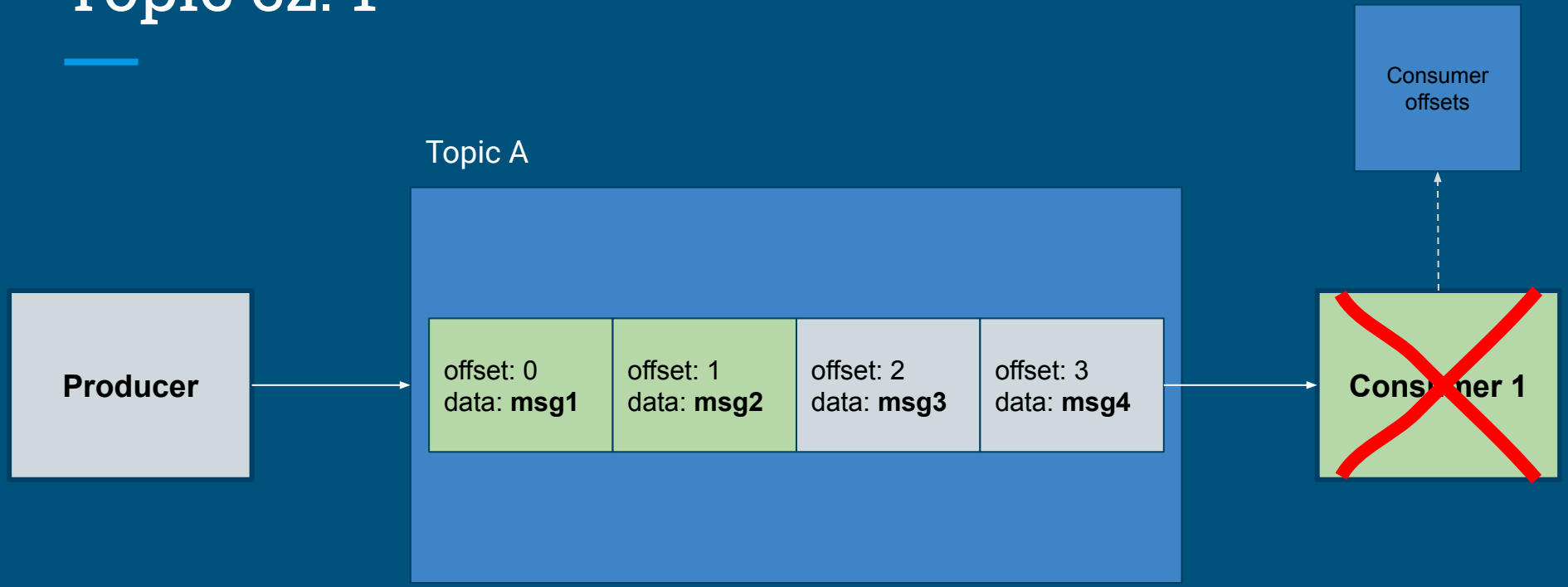
Topic cz. 1


Consumer offsets
Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

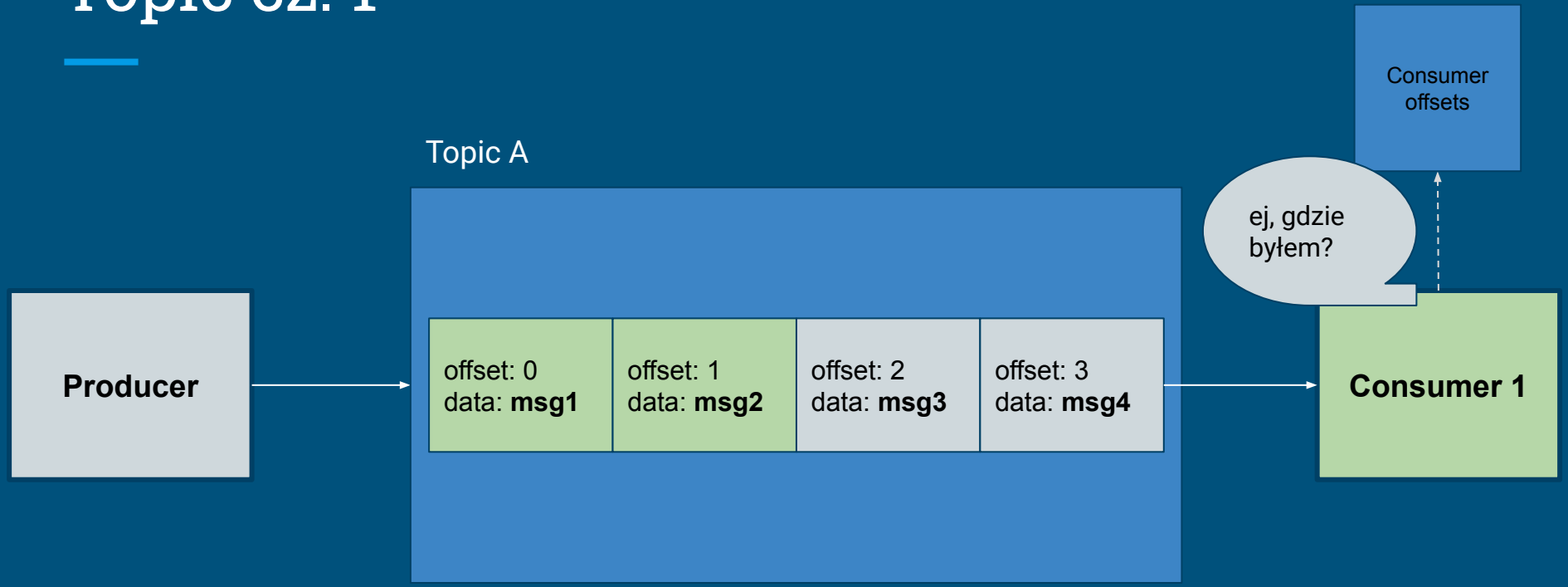
Topic cz. 1


Consumer offsets
Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

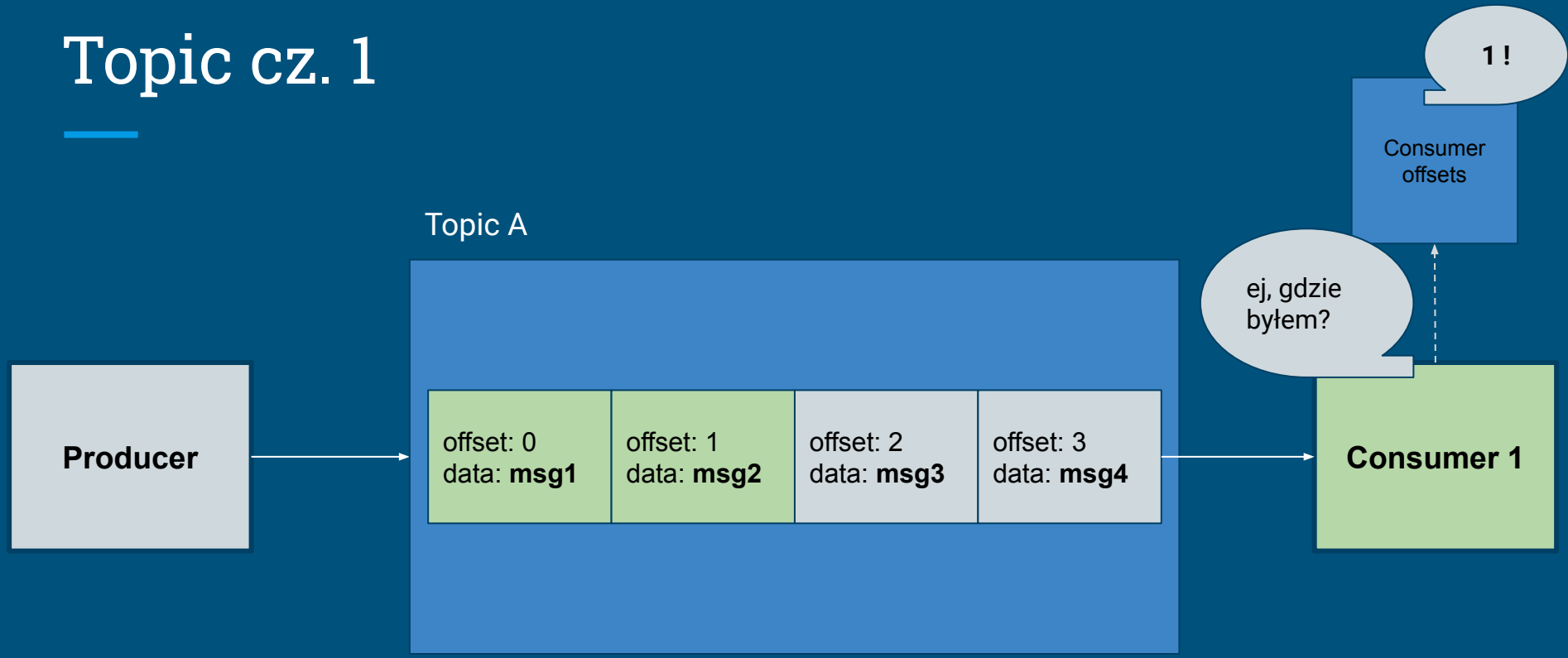
Topic cz. 1


Consumer offsets
Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

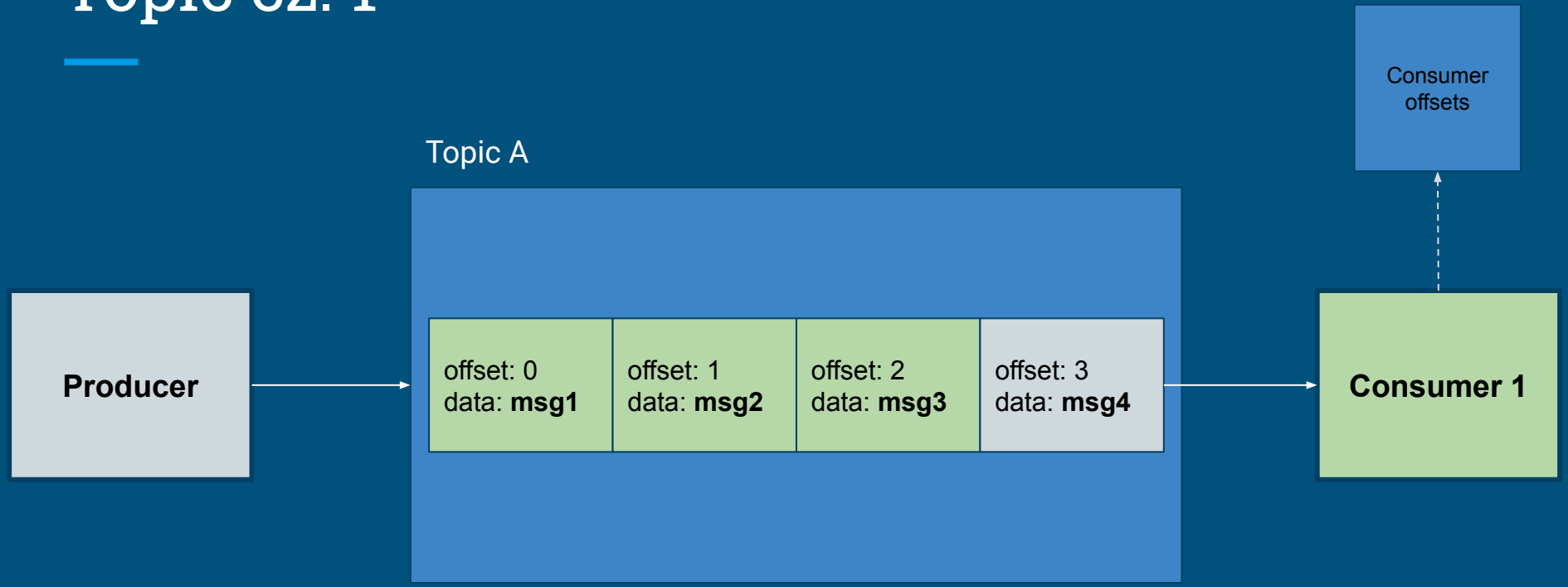
Topic cz. 1


Consumer offsets
Topic A
ej, gdzie byłem?
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

Topic cz. 1

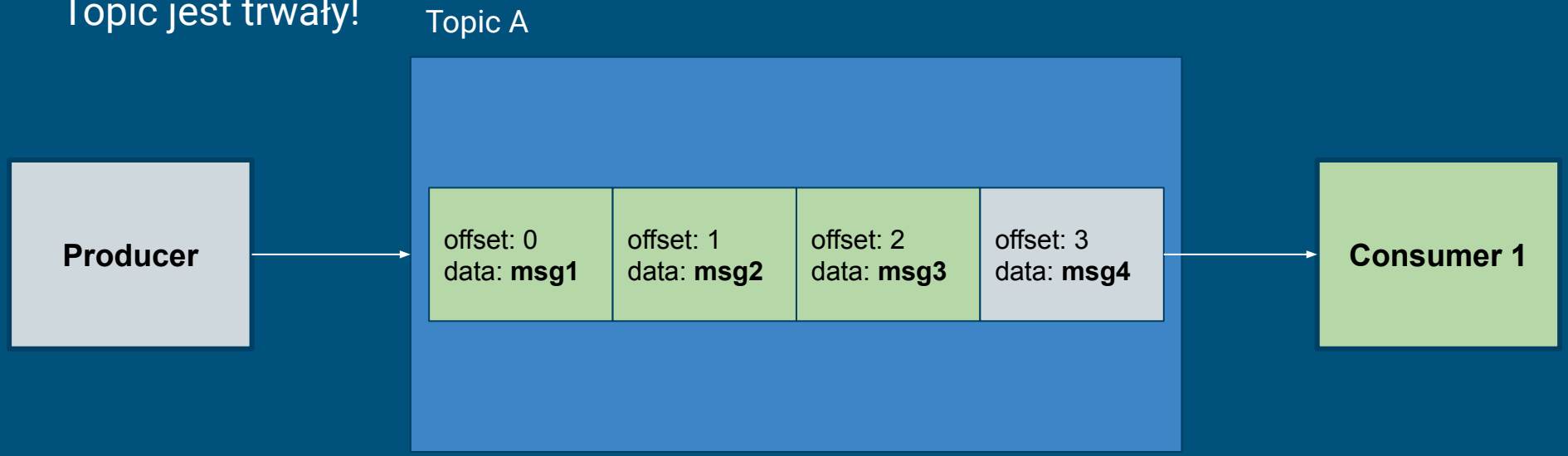

1 !
Consumer offsets
Topic A
ej, gdzie byłem?
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

Topic cz. 1


Consumer offsets
Topic A
Producer
offset: 0
data: msg1
offset: 1
data: msg2
offset: 2
data: msg3
offset: 3
data: msg4
Consumer 1

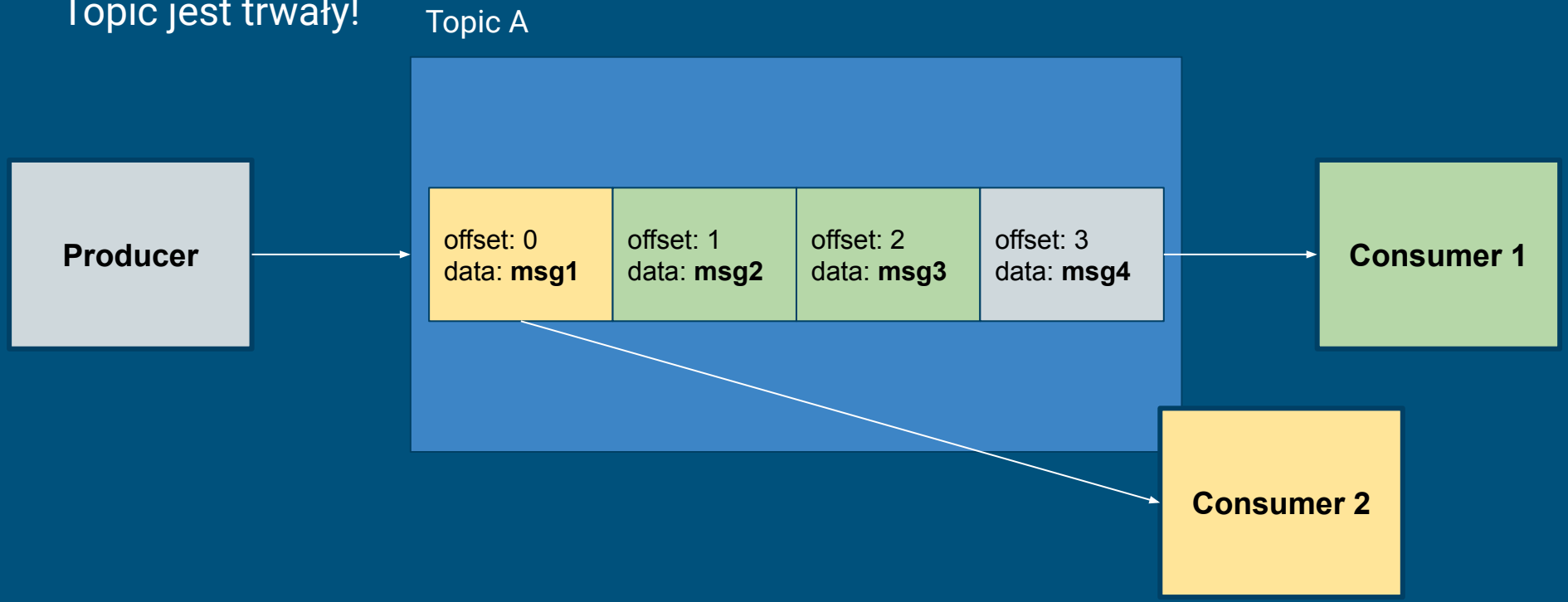
Topic cz. 1

Topic jest trwały!



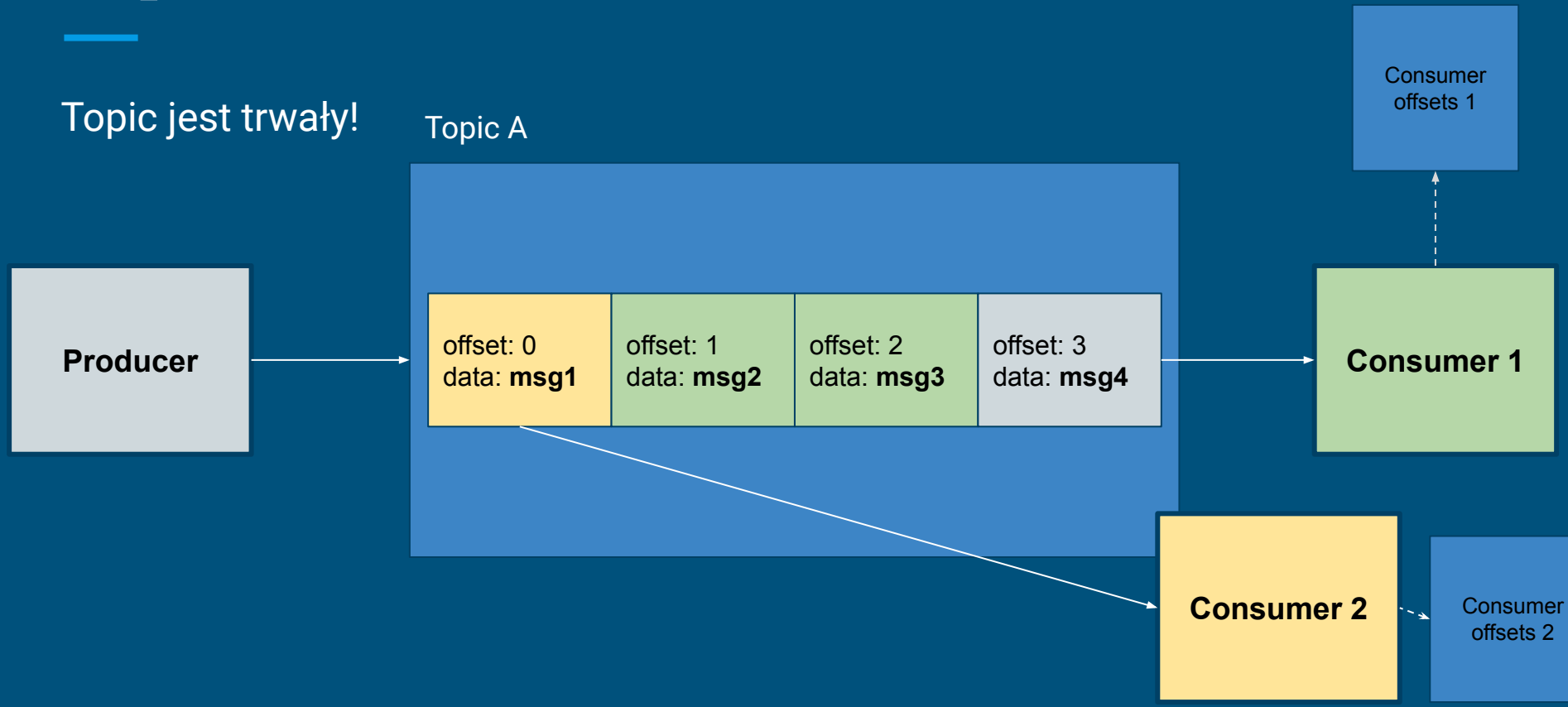
Topic cz. 1

Topic jest trwały!



Topic cz. 1

Topic jest trwały!



Topic cz. 1

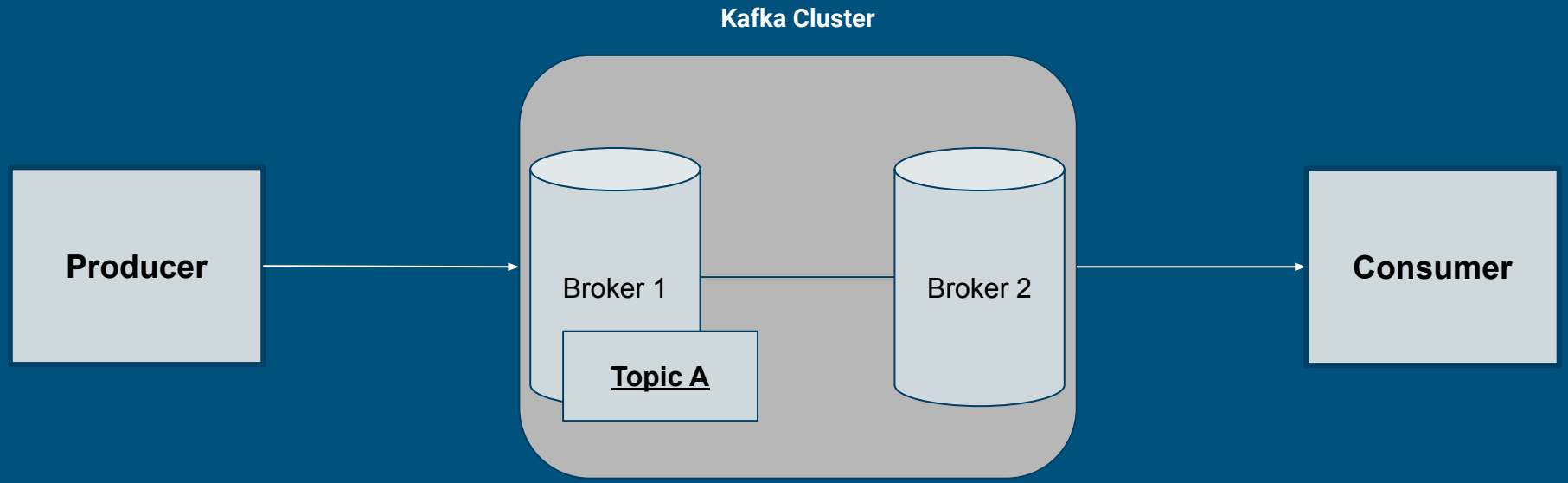
- retention.policy == 7 dni
- compacted topic

Większość rzeczy w Kafce ma swoje defaulty...

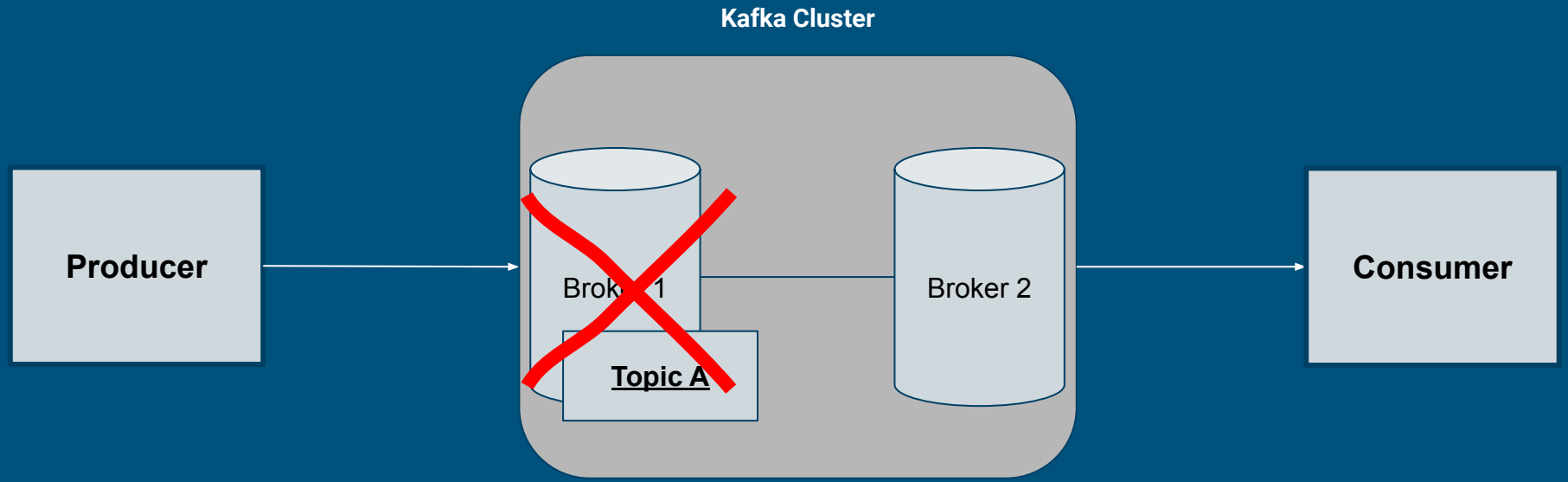
Replication factor == 1 !?

Wszystko źle! **Tracimy dane**

Replication factor == 1



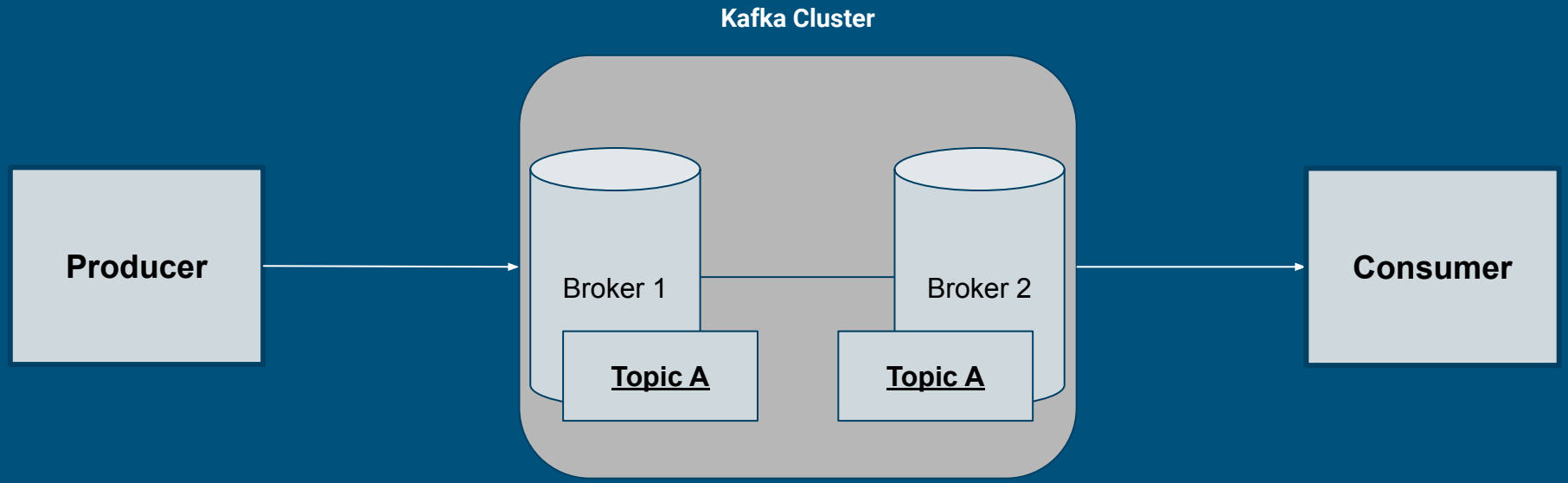
Replication factor == 1



Ex 4

Replication factor == 1 !?
Wszystko źle! Tracimy dane

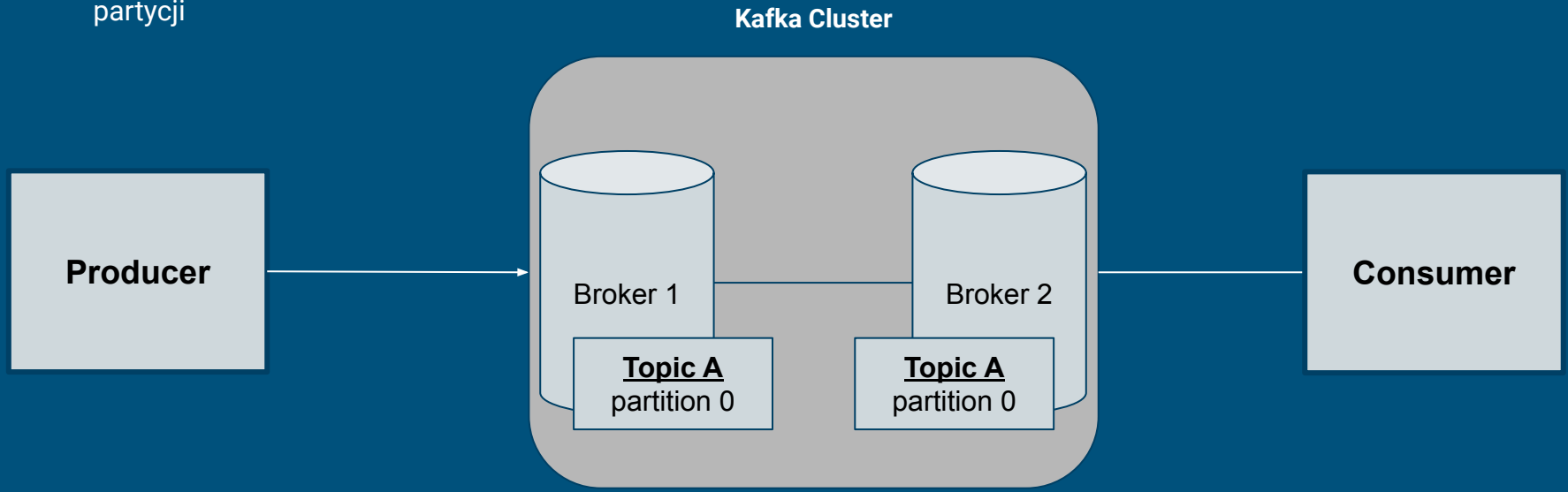
Replication factor == 2



Skalowanie, czyli partycje

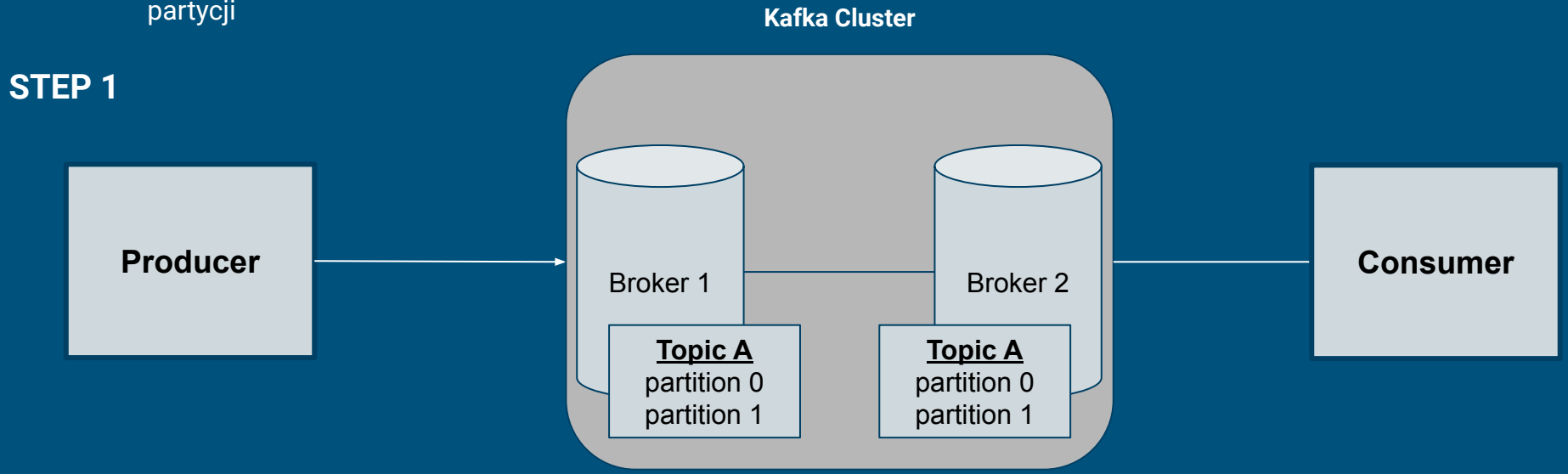
Replication factor == 2

- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji



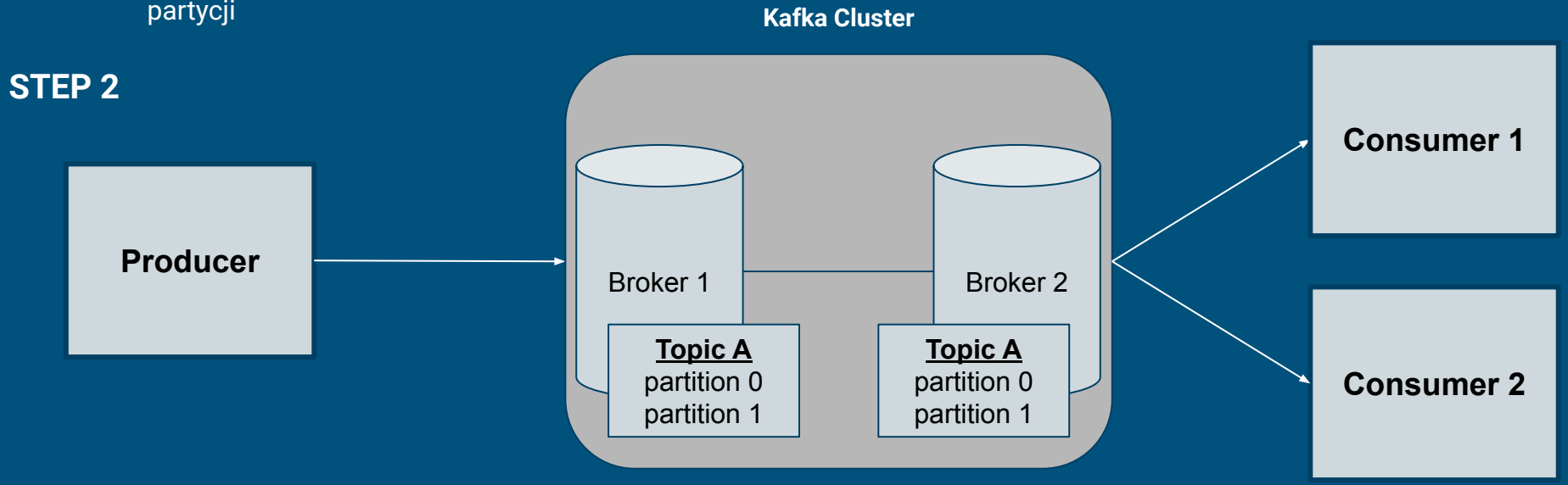
Replication factor == 2

- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji



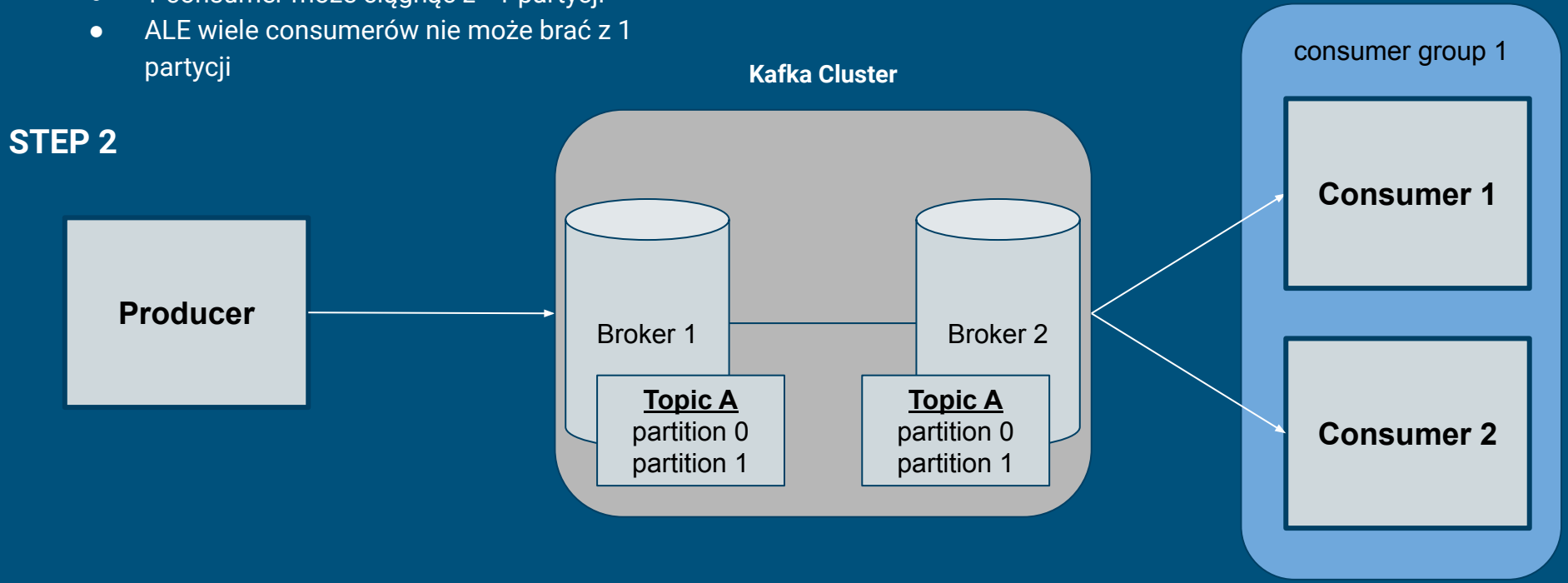
Replication factor == 2

- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji



Replication factor == 2

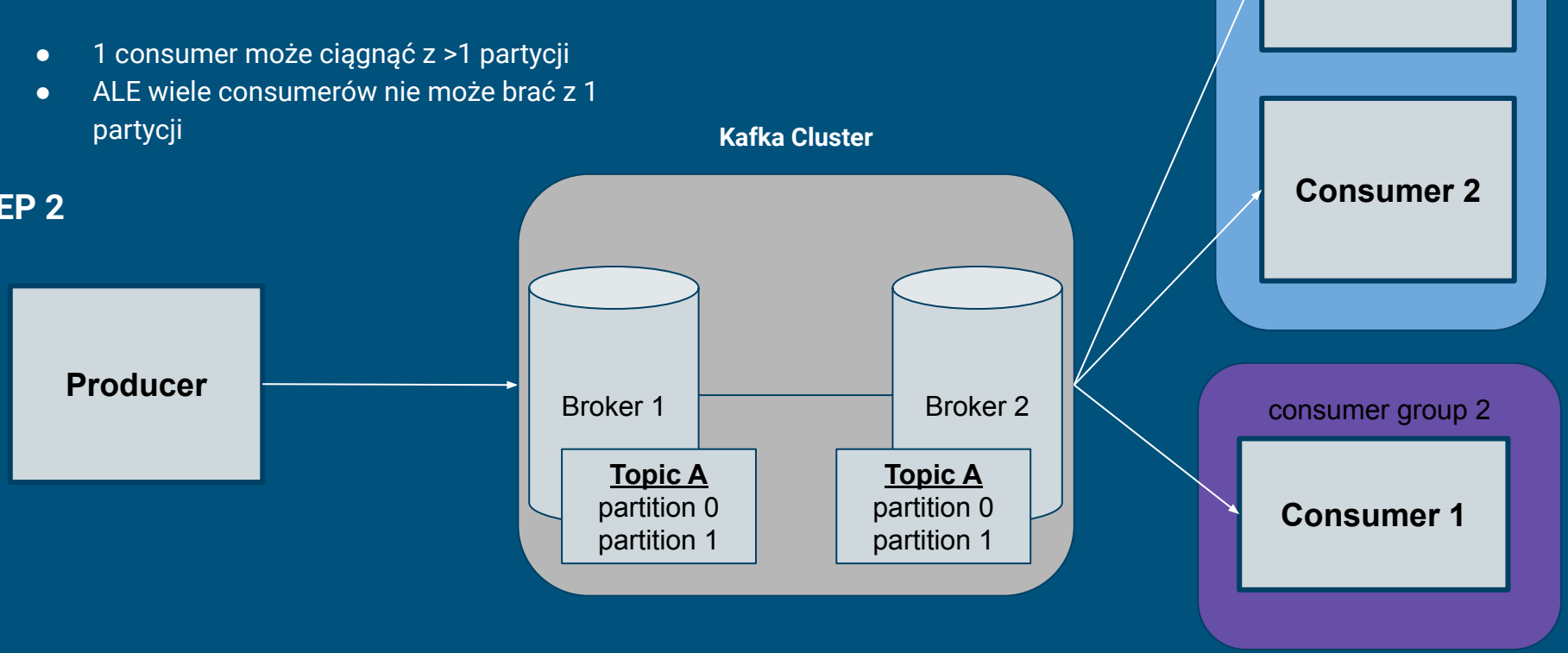
- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji



Replication factor == 2

- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji

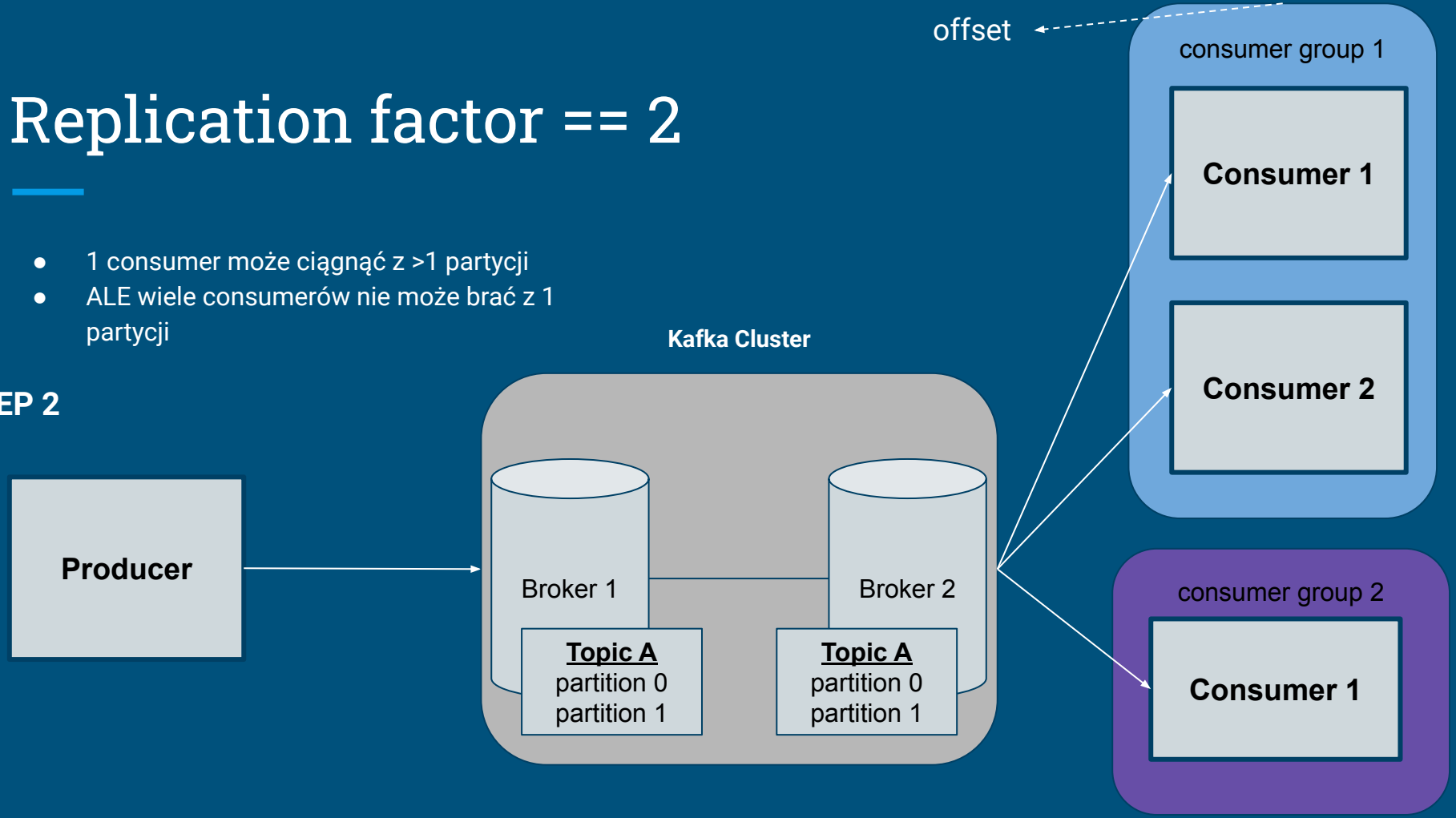
STEP 2



Replication factor == 2

- 1 consumer może ciągnąć z >1 partycji
- ALE wiele consumerów nie może brać z 1 partycji

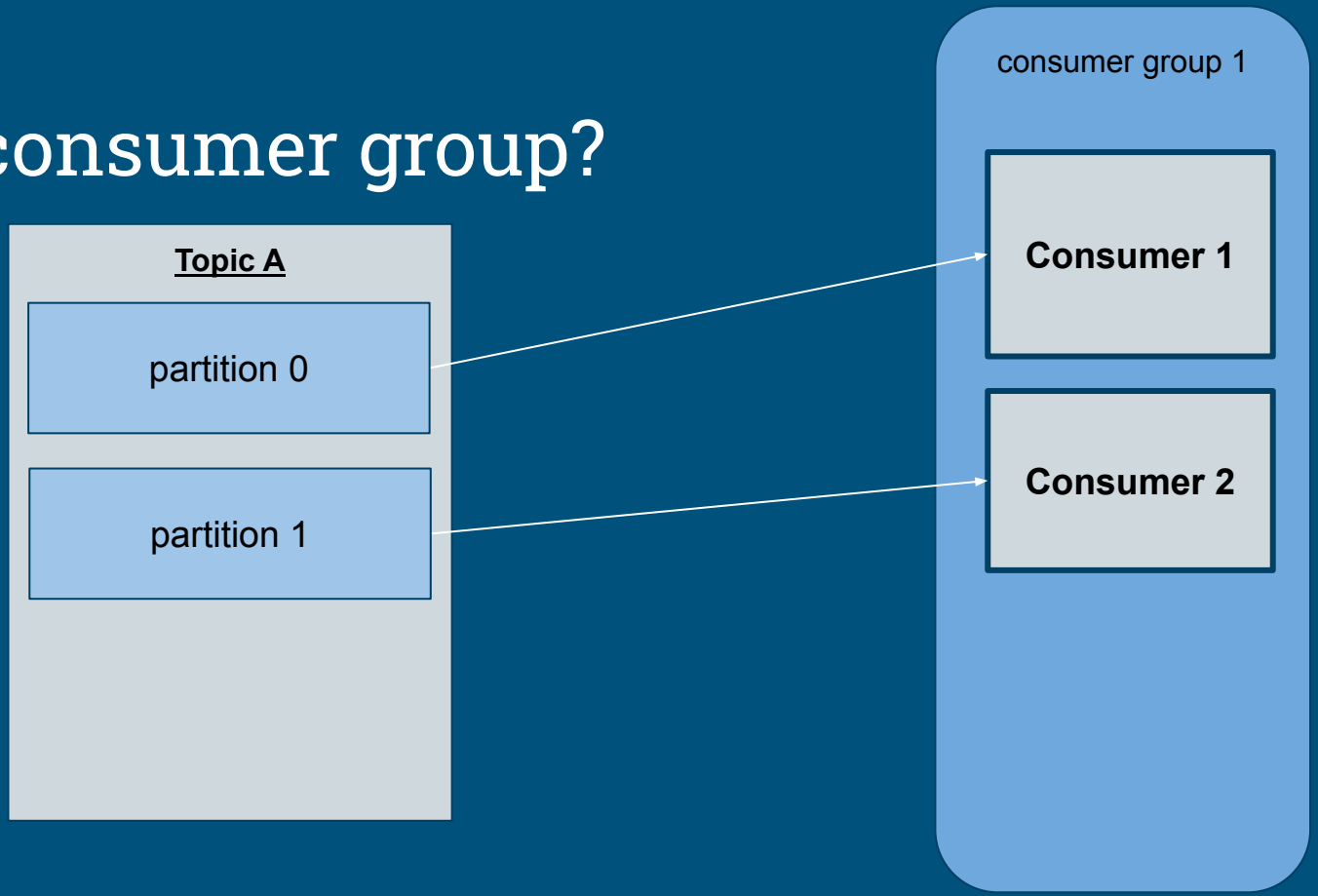
STEP 2



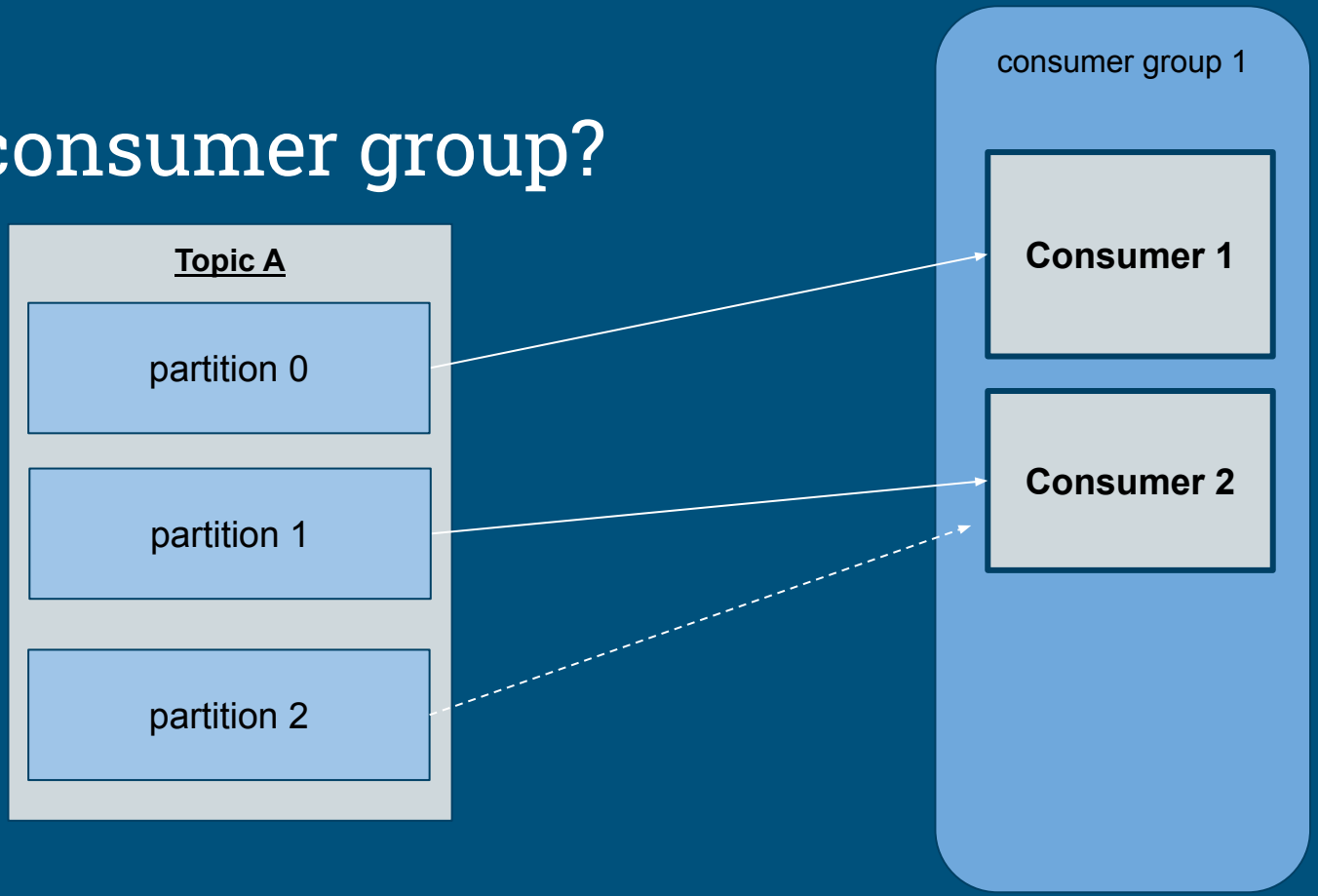
Ex 5

Skalujemy nasz topic

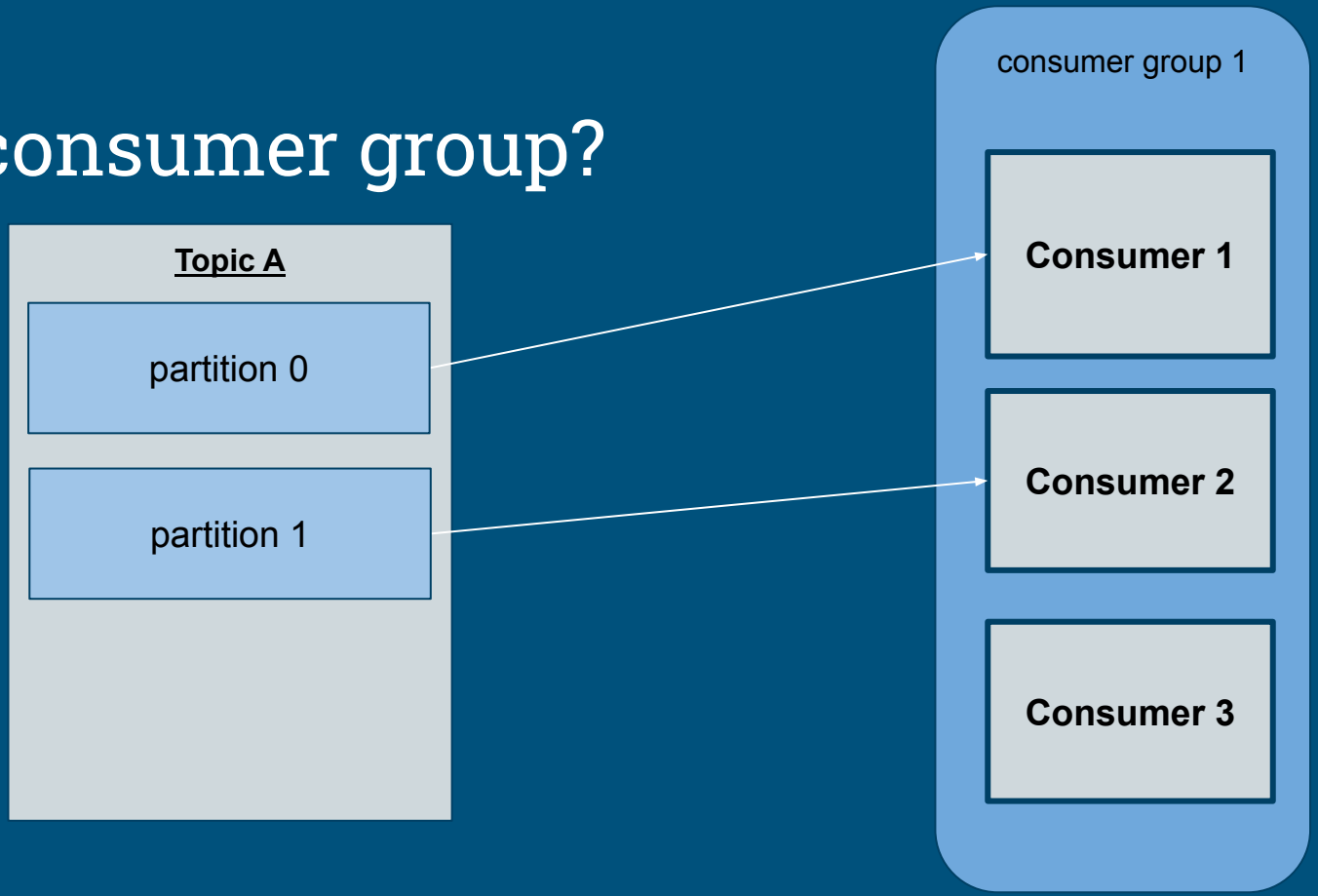
Jak działa consumer group?



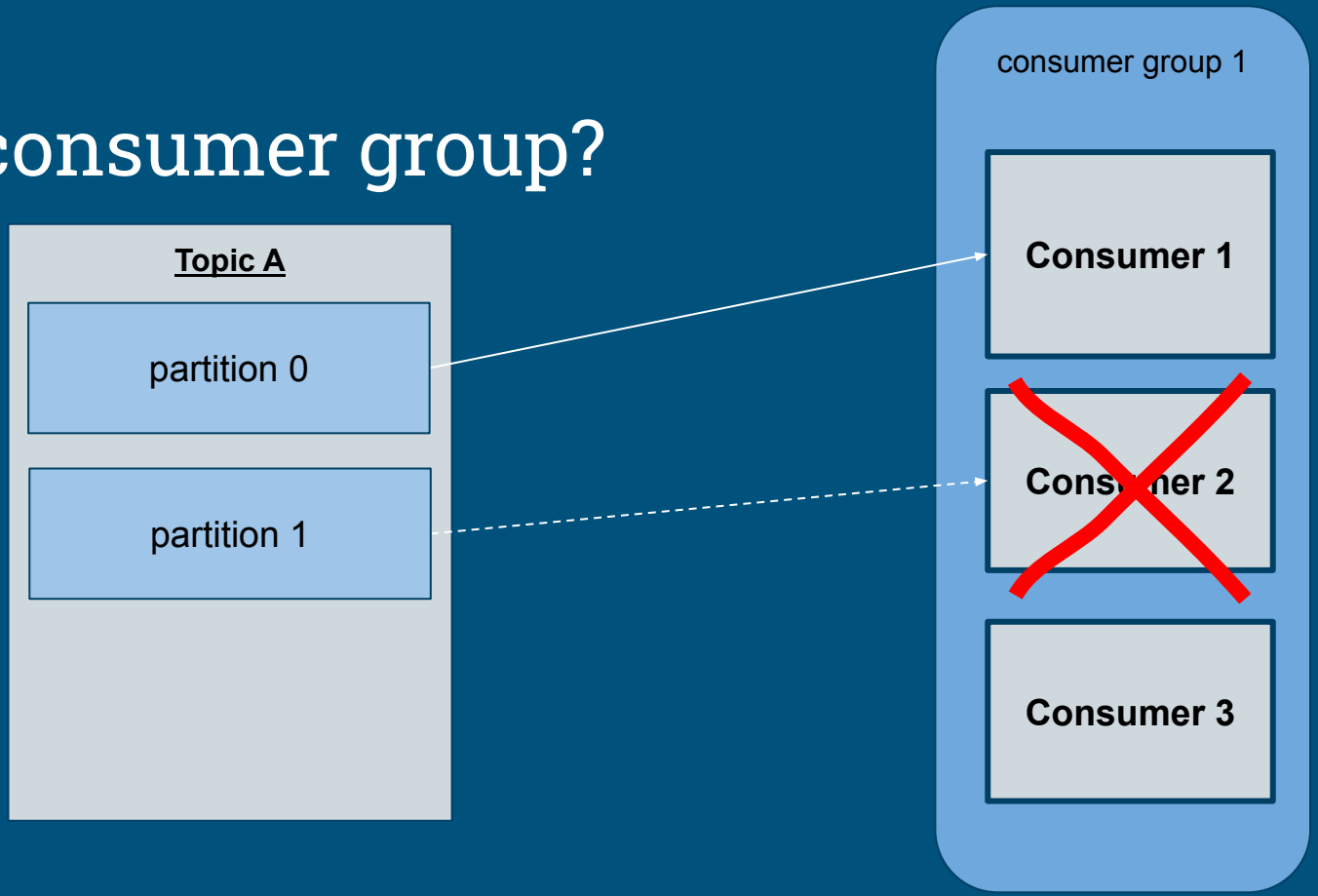
Jak działa consumer group?



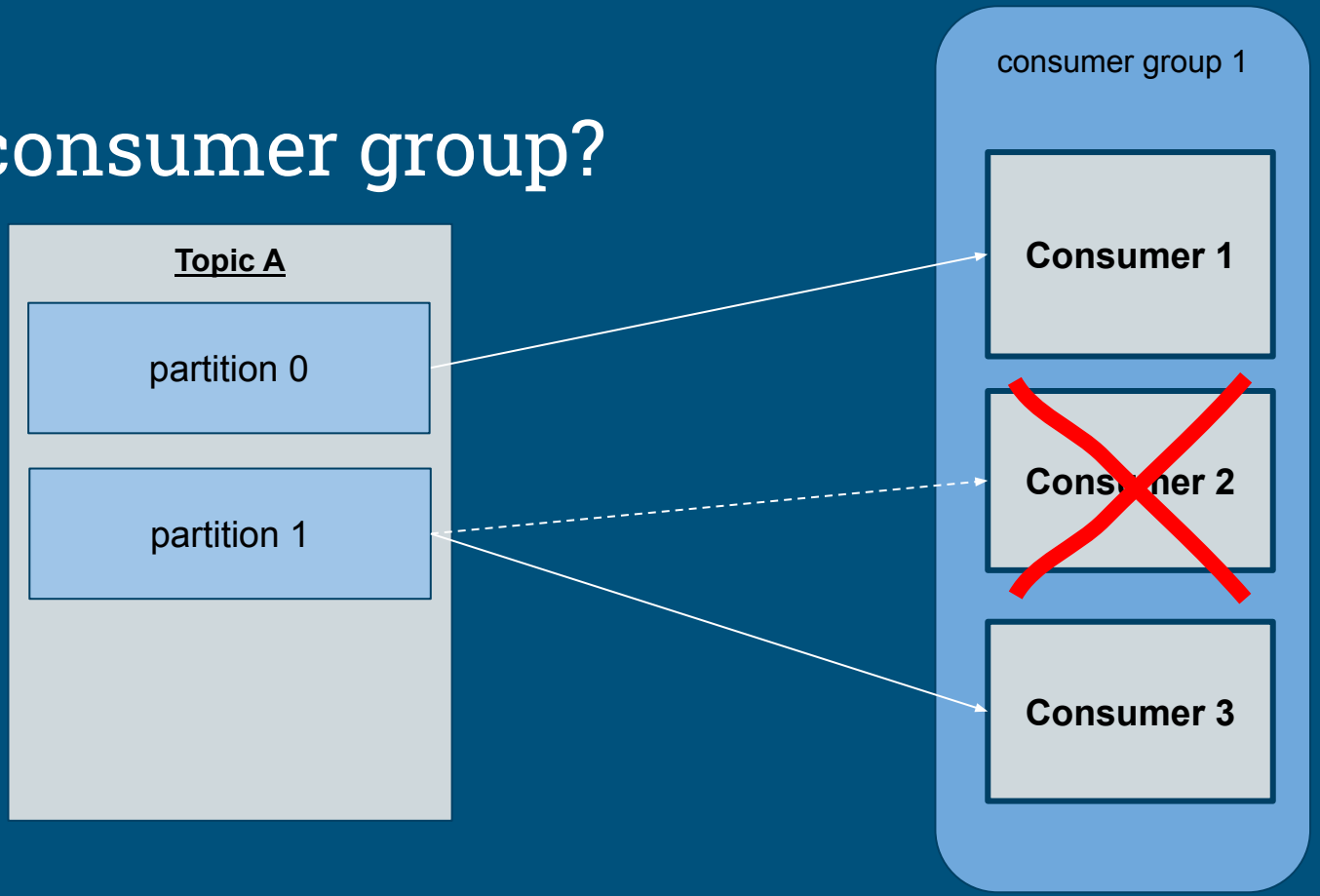
Jak działa consumer group?



Jak działa consumer group?



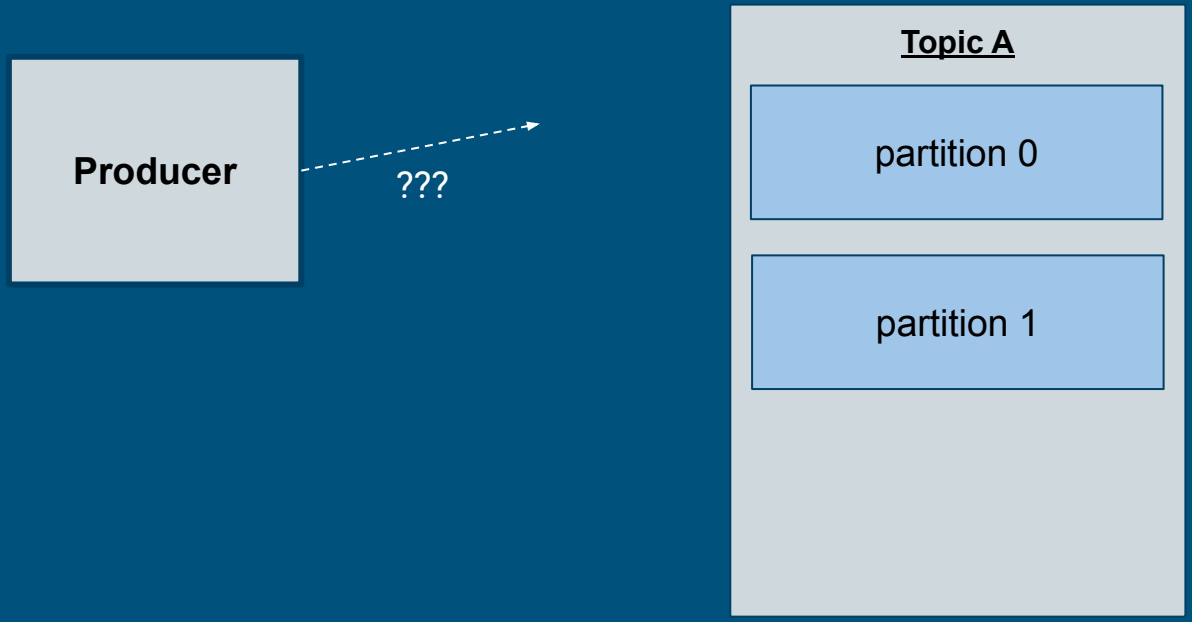
Jak działa consumer group?



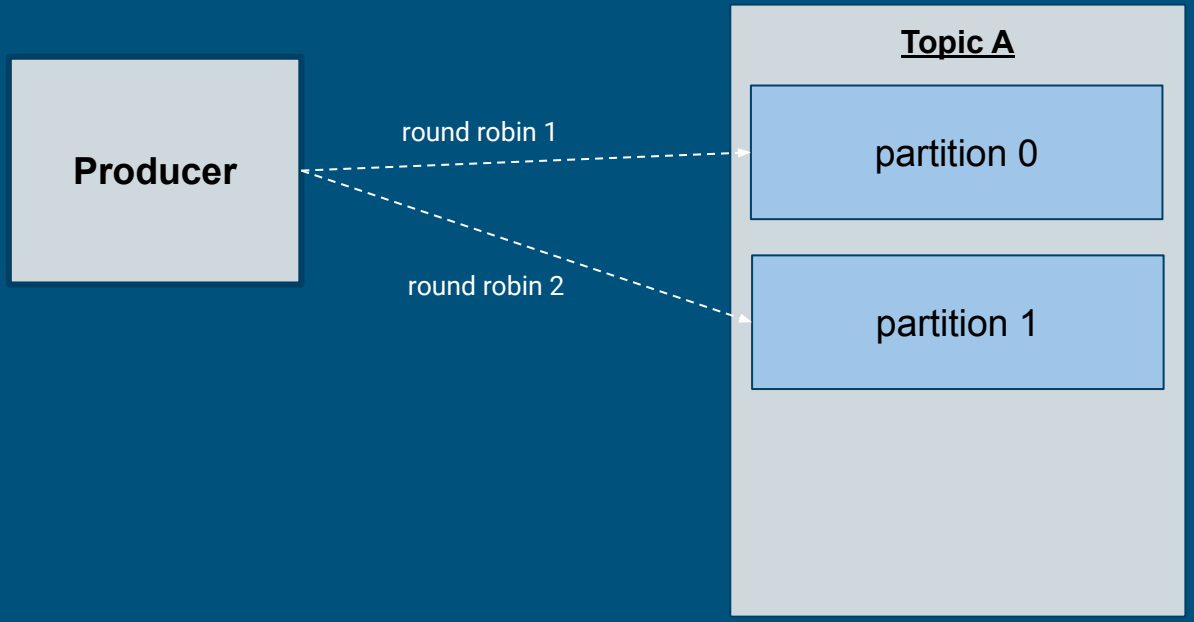
Ex 6

Naprawiamy Ordering

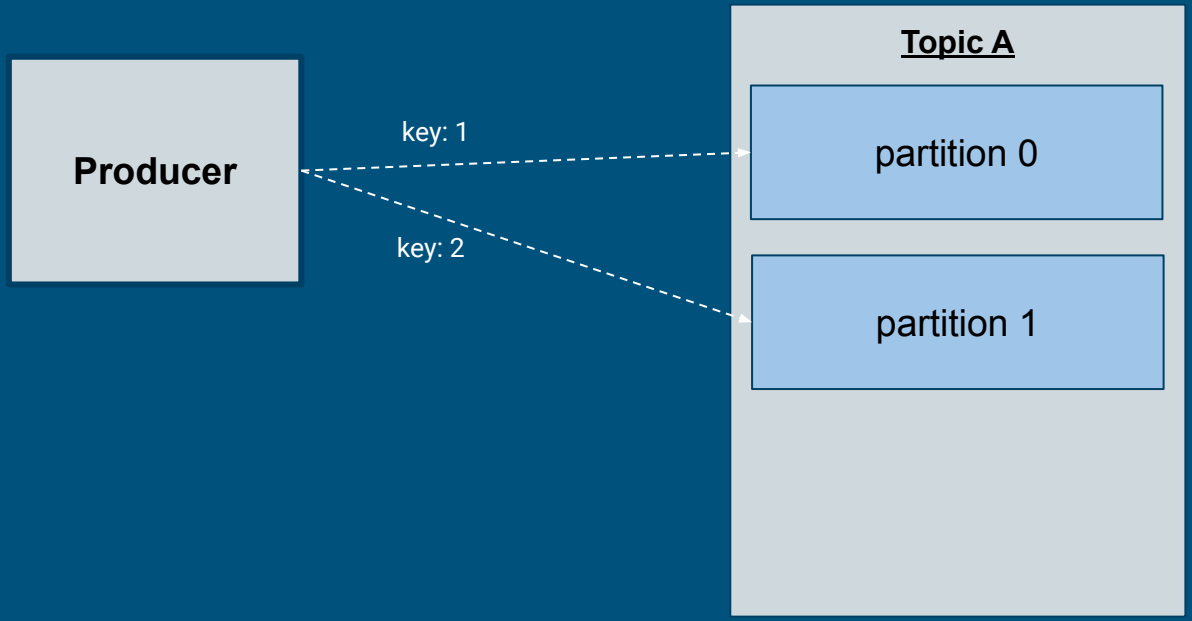
Jak Producer dodaje record?



Jak Producer dodaje record?



Jak Producer dodaje record?



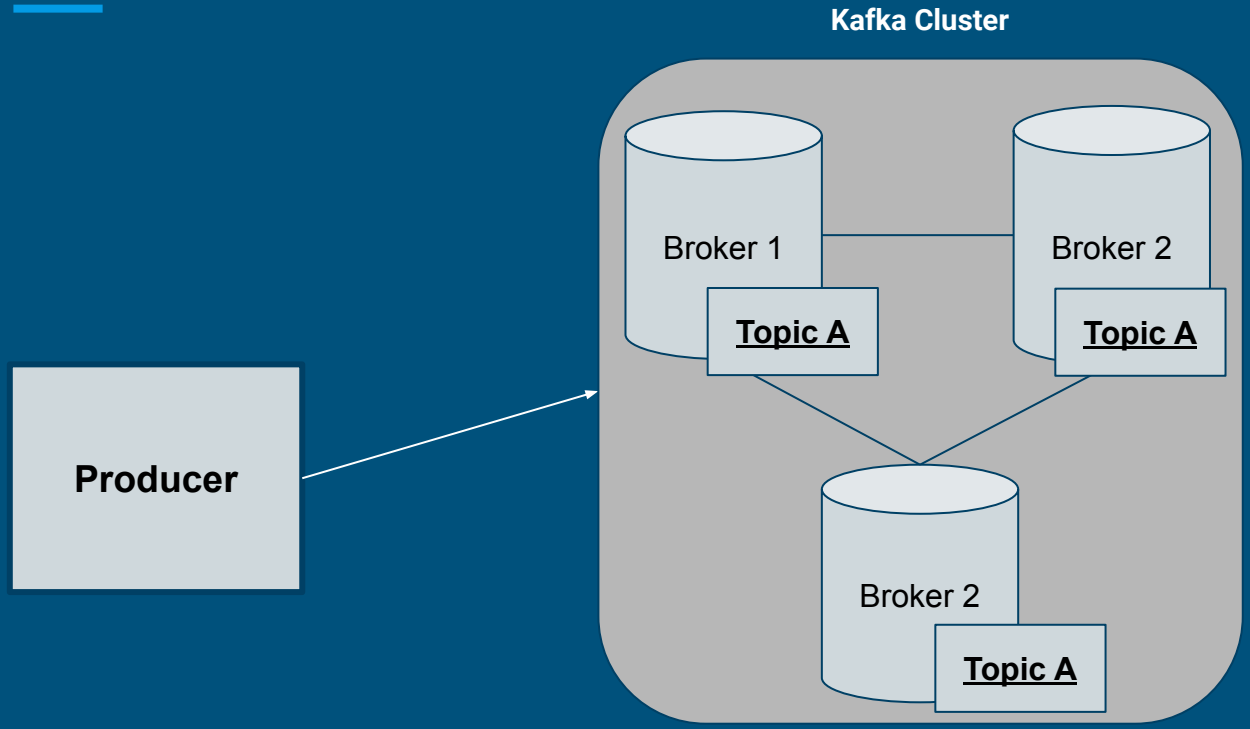
Przerwa
do 13.15

Będzie depresyjnie...

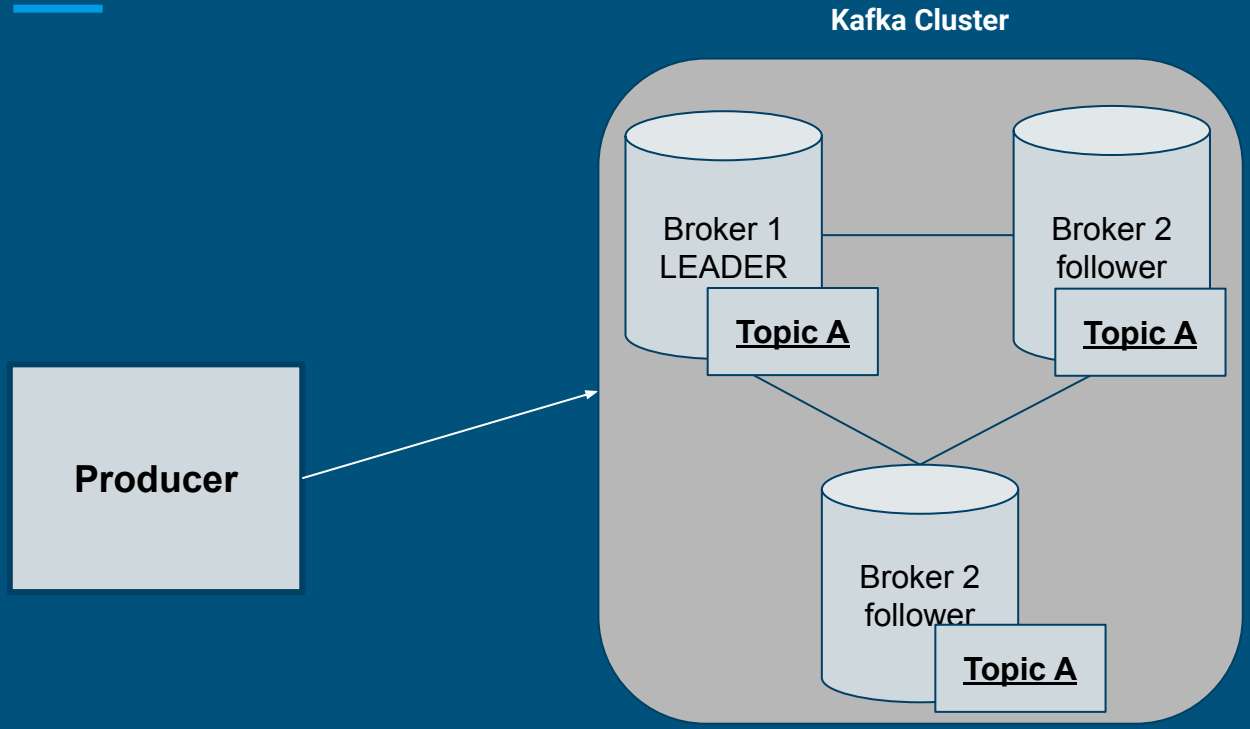
Dzień 1

Dzień 2

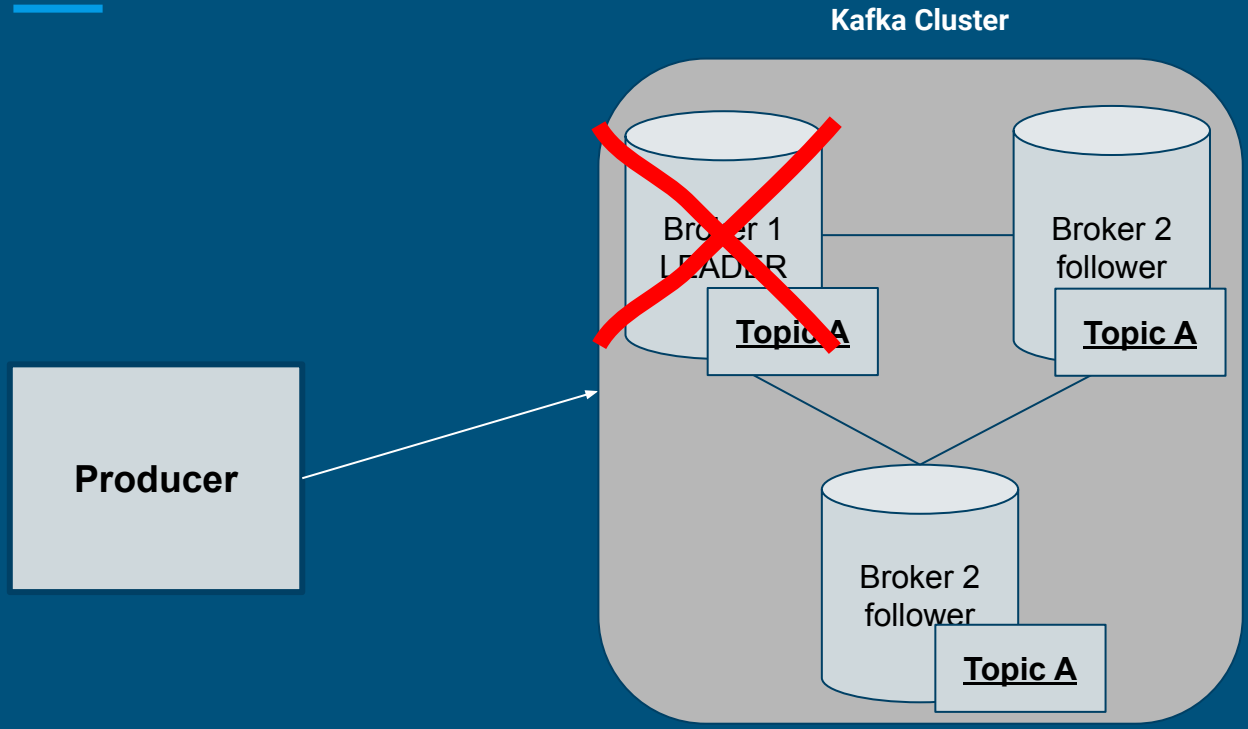
ACK - o co tutaj chodzi?



ACK - o co tutaj chodzi?



ACK - o co tutaj chodzi?



ACK

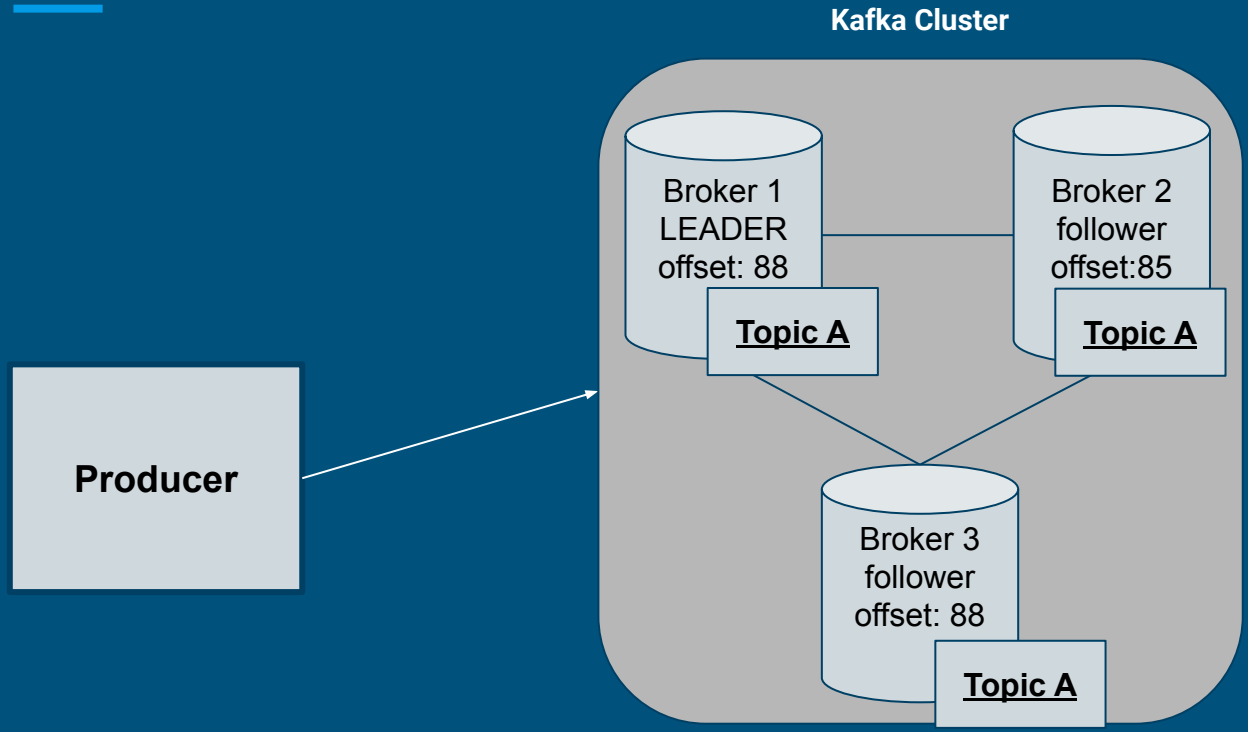
- -1 ⇒ zero powierdzeń. Wysyłamy wiadomość i nie czekamy na żadną zwrotkę.
  - ryzykowne, ale i turbo szybkie
- 1 ⇒ default. Czekamy na zwrotkę tylko od leadera
- all ⇒ czekamy na zwrotkę od wszystkich replik.

Ex 7

Offsety

Co to ISR?

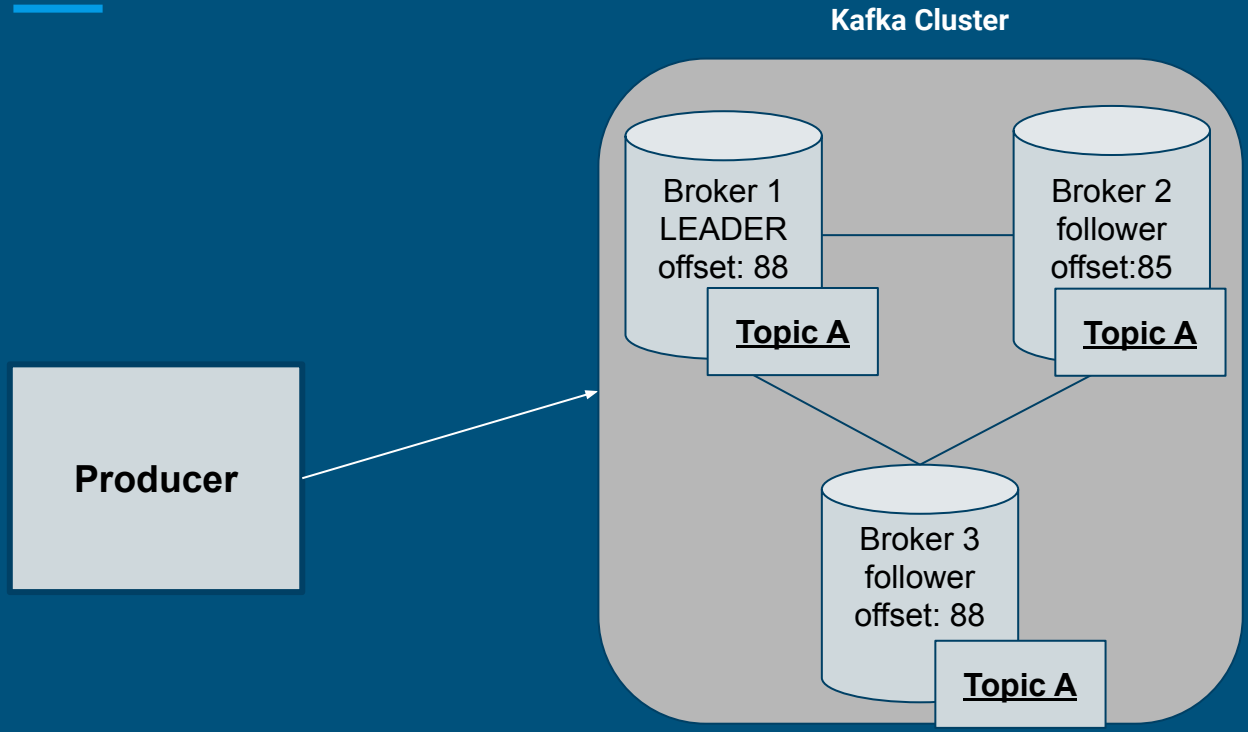
Co to ISR?


Kafka Cluster
Broker 1
LEADER
offset: 88
Topic A
Broker 2
follower
offset:85
Topic A
Broker 3
follower
offset: 88
Topic A
Producer

ISR - po co to komu?

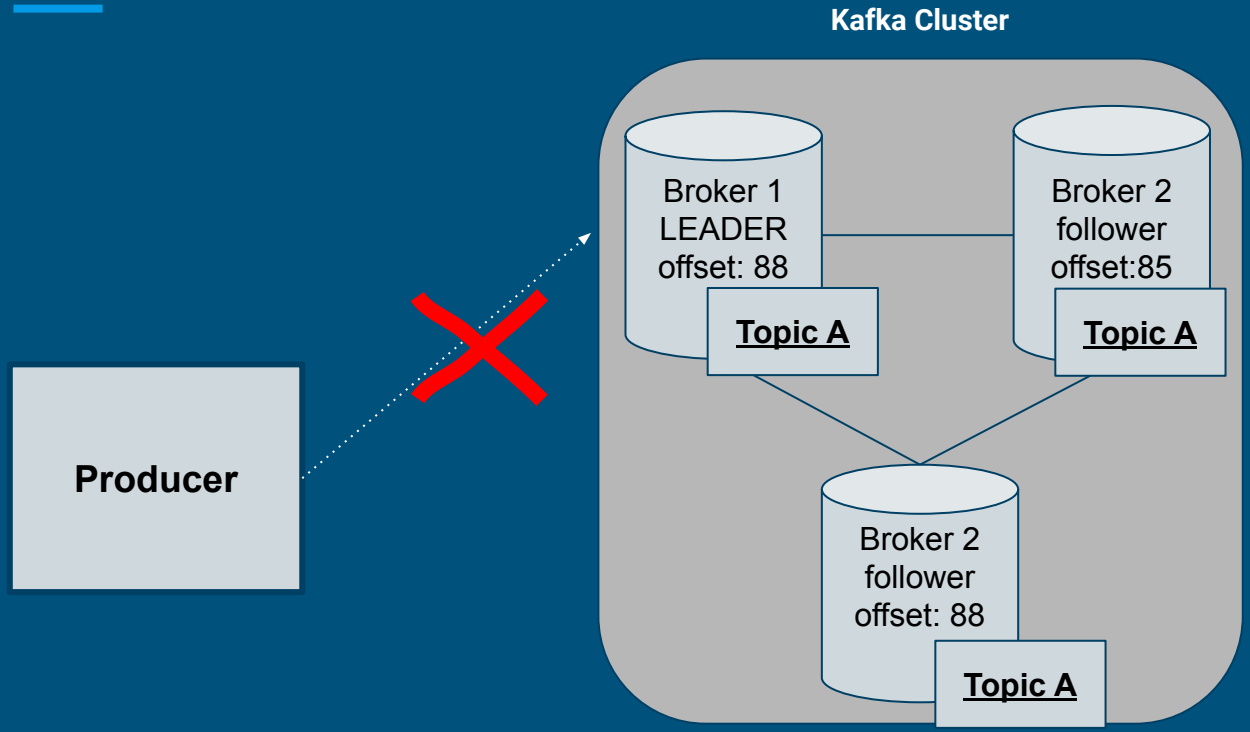
- ważne pojęcie diagnostyczne - widzimy ile followersów jest do tyłu
- jest specjalna flaga - min.in.sync.replicas ⇒ ile replik musi być na bieżąco żeby w ogóle zwracać pozytywny ack

Co to ISR?

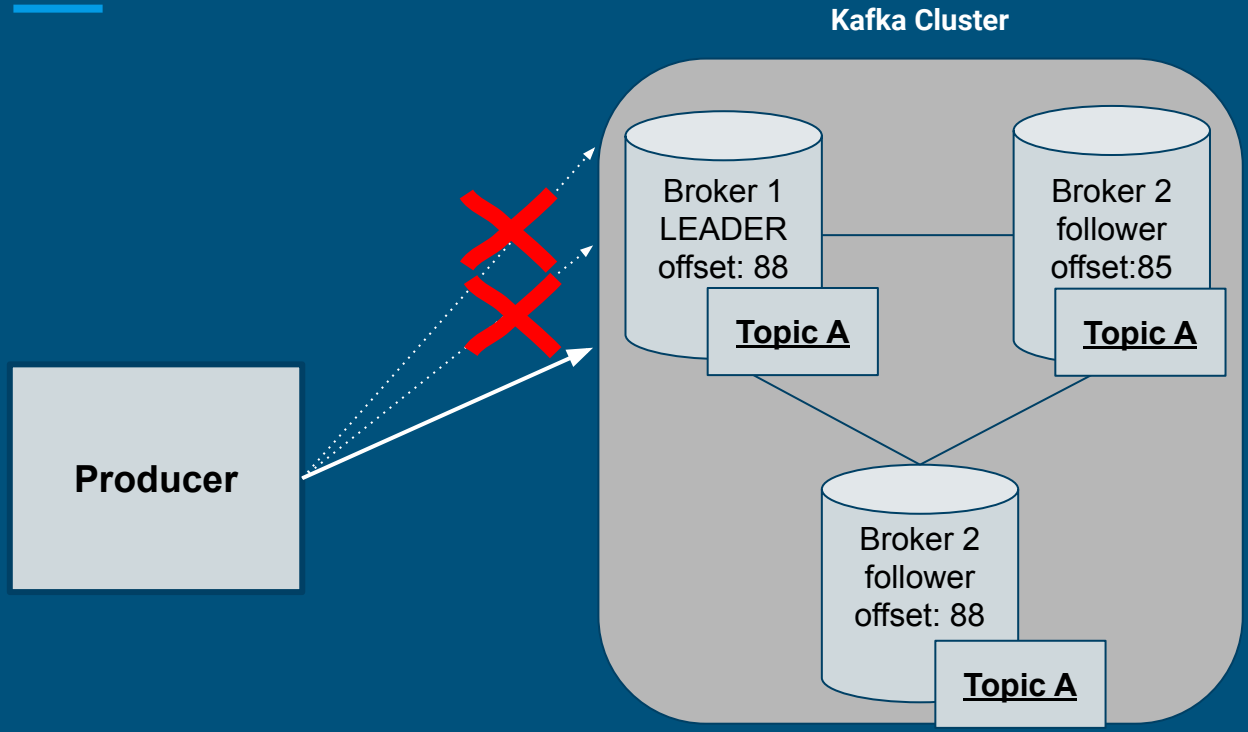

Kafka Cluster
Broker 1
LEADER
offset: 88
Topic A
Broker 2
follower
offset:85
Topic A
Broker 3
follower
offset: 88
Topic A
Producer

Co to Retry?

Co to Retry?


Kafka Cluster
Broker 1
LEADER
offset: 88
Topic A
Broker 2
follower
offset:85
Topic A
Broker 2
follower
offset: 88
Topic A
Producer

Co to Retry?

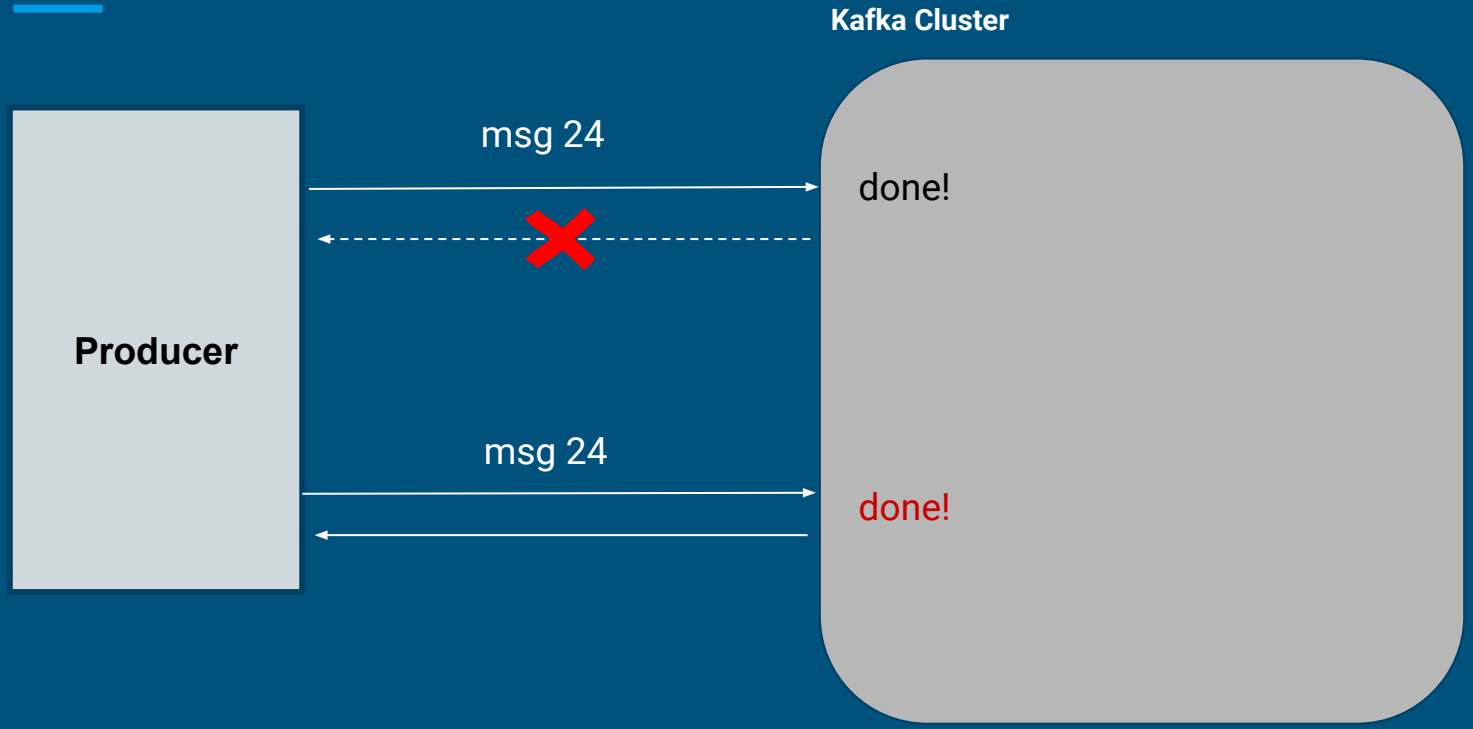

Kafka Cluster
Broker 1
LEADER
offset: 88
Topic A
Broker 2
follower
offset:85
Topic A
Broker 2
follower
offset: 88
Topic A
Producer

Ex 1

Poprawiamy flagi w naszym Producerze

- ACK
- ISR
- Retry

Problem Retry'ów



enable.idempotence

- https://www.cloudkarafka.com/blog/apache-kafka-idempotent-producer-avoiding-message-duplication.html

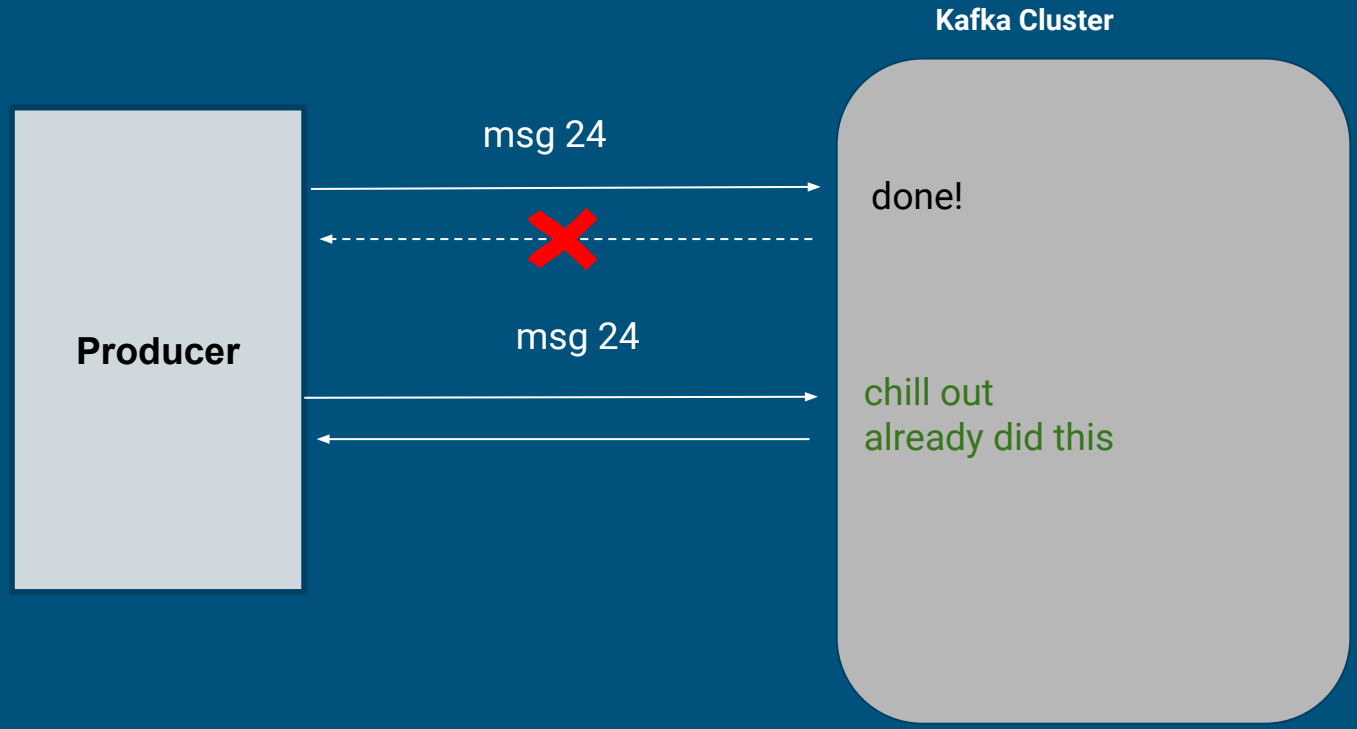
- 'enable.idempotence': True
- acks == all
- max.in.flight.requests.per.connection > 1 <= 5

Ex 2

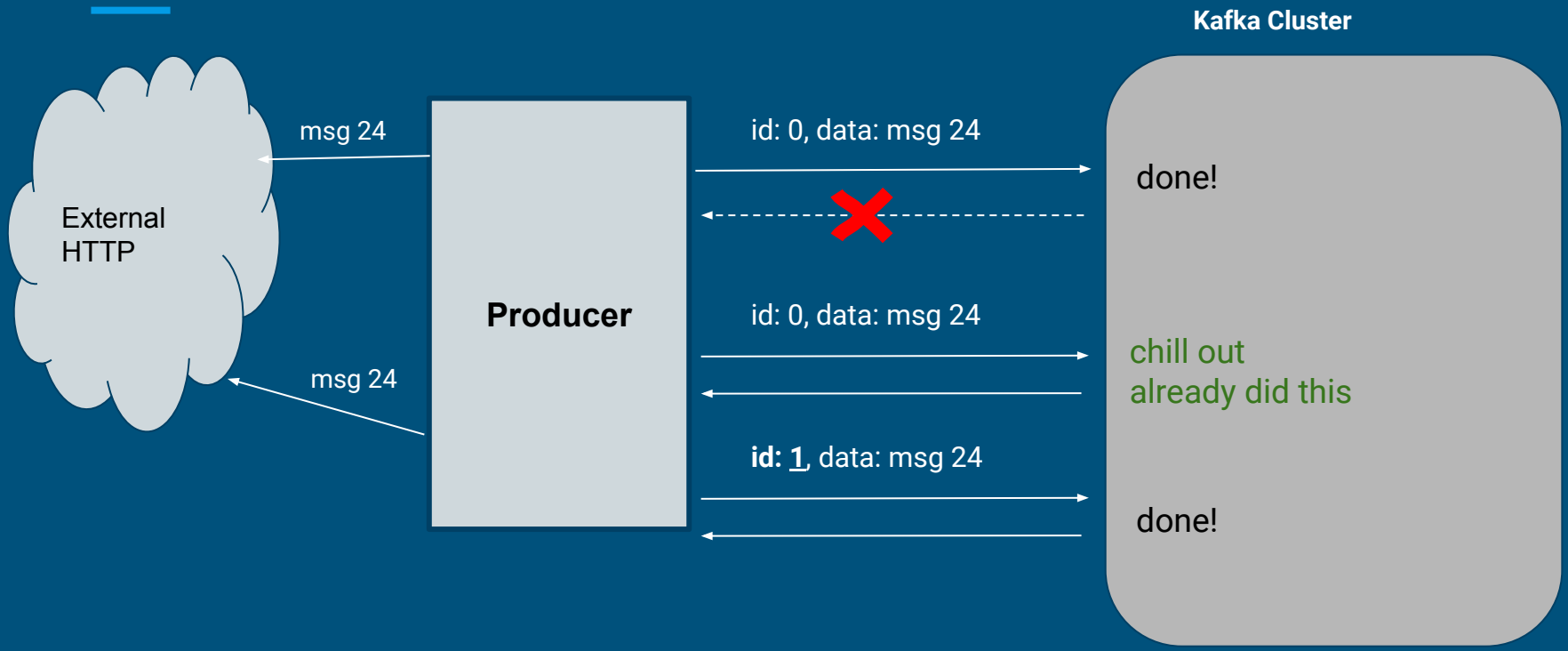
Poprawiamy flagi w naszym Producerze

- 'enable.idempotence': True
- acks == all
- max.in.flight.requests.per.connection <= 5

Problem z enable.idempotence

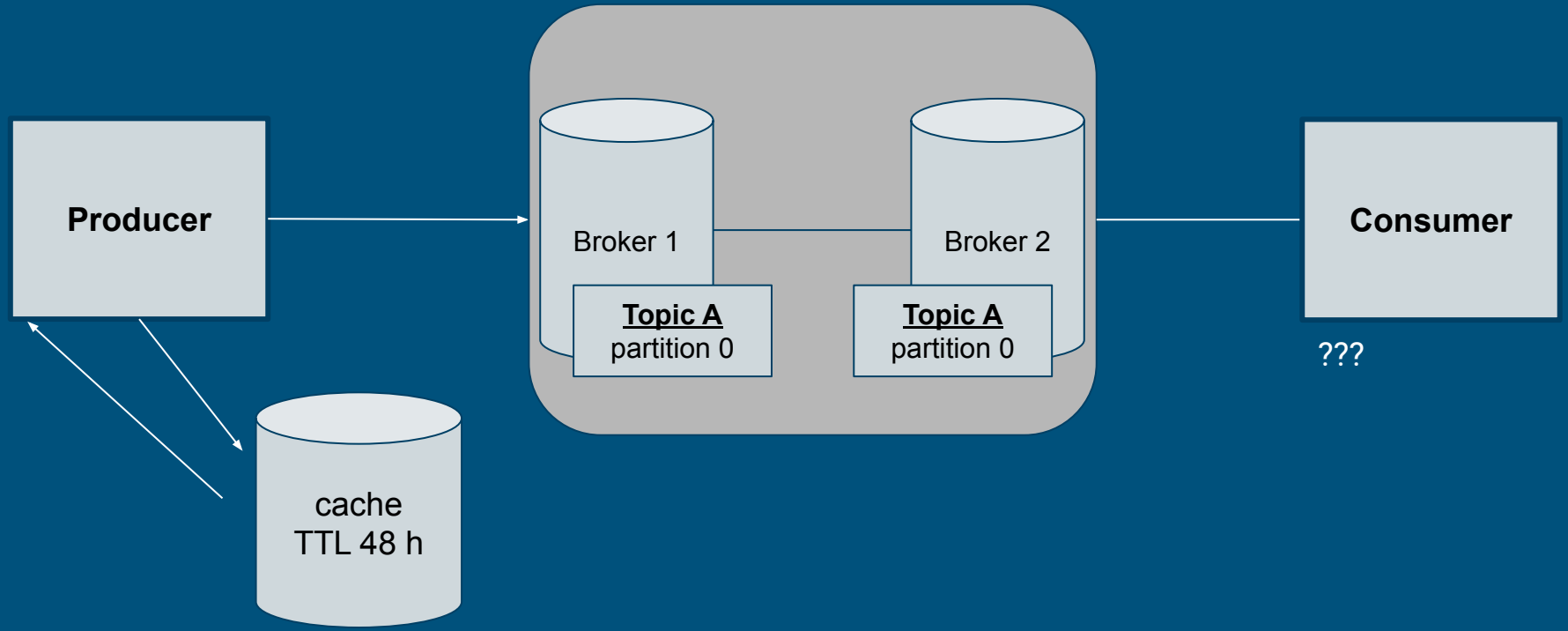


Problem z enable.idempotence

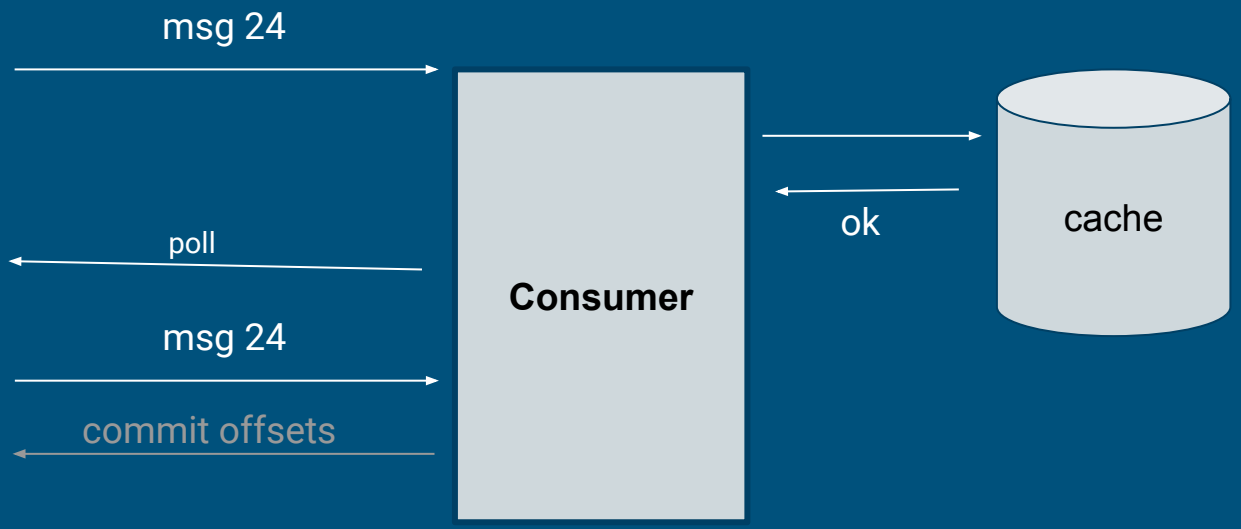


Duplicate handling - jest trudny i to wybór architektoniczny

Duplicate handling - jest trudny i to wybór architektoniczny

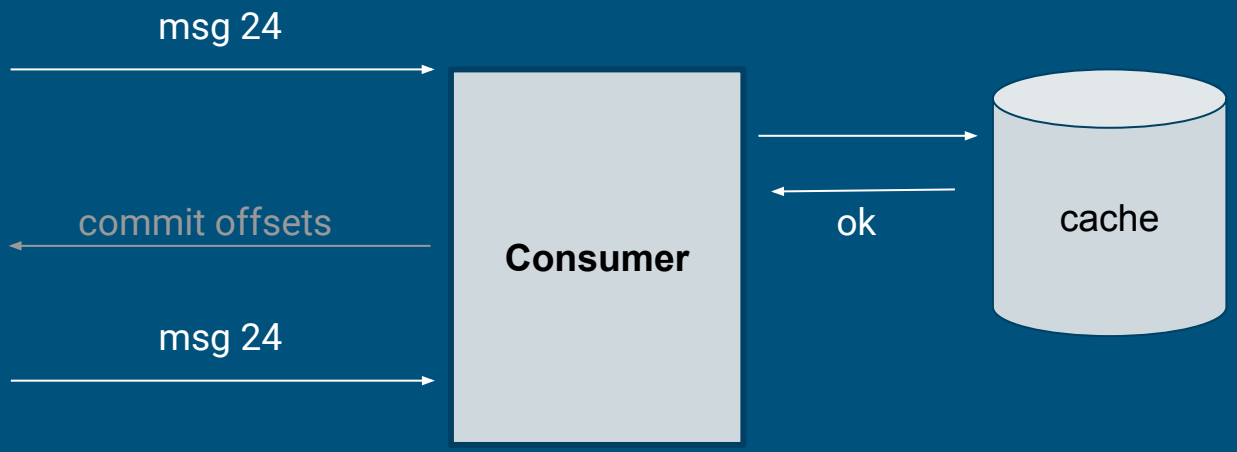


Duplicate handling - opcja po stronie consumera

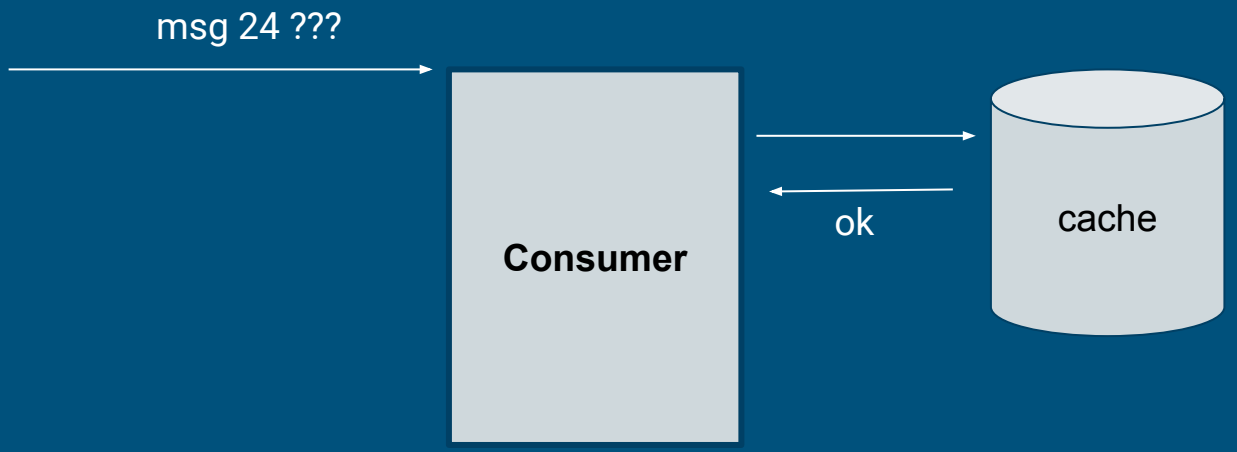


At Least One, At Most once, Exactly once delivery - różnice wady i zalety

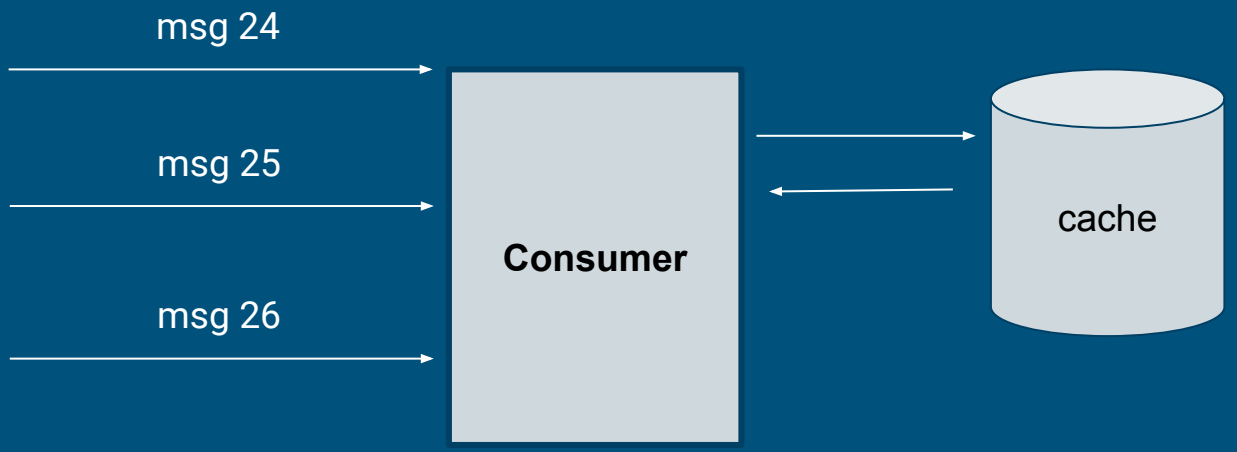
At Least One



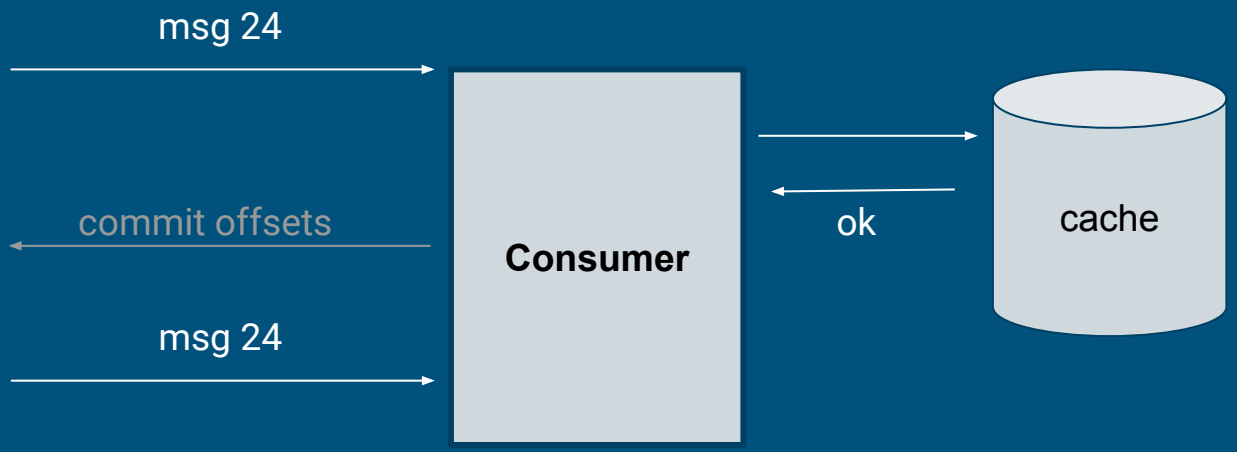
At Most once



Exactly once delivery (święty Gral)



Effectively once (processed)



Effectively once (processed)

Distributed cache ma swoją cenę - potencjalne awarie sieci, ale i jest potrzebny dla Consumer Groupy