AutoIndexほか
HUEにおけるKafka
活用事例

(株) ワークスアプリケーションズ
HUE&ATE Dept. 浅野　航平

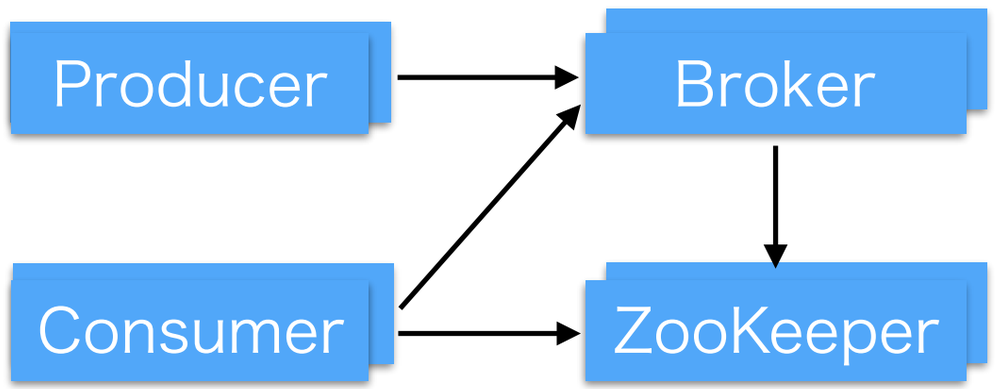
1. Kafkaについて

Kakfa概要

- OSSのメッセージキュー実装の一つ。
  独自仕様・独自プロトコル (on TCP)
- Fast/Scalable/Durable/Distributed
- pub-sub/queuing両方をサポート

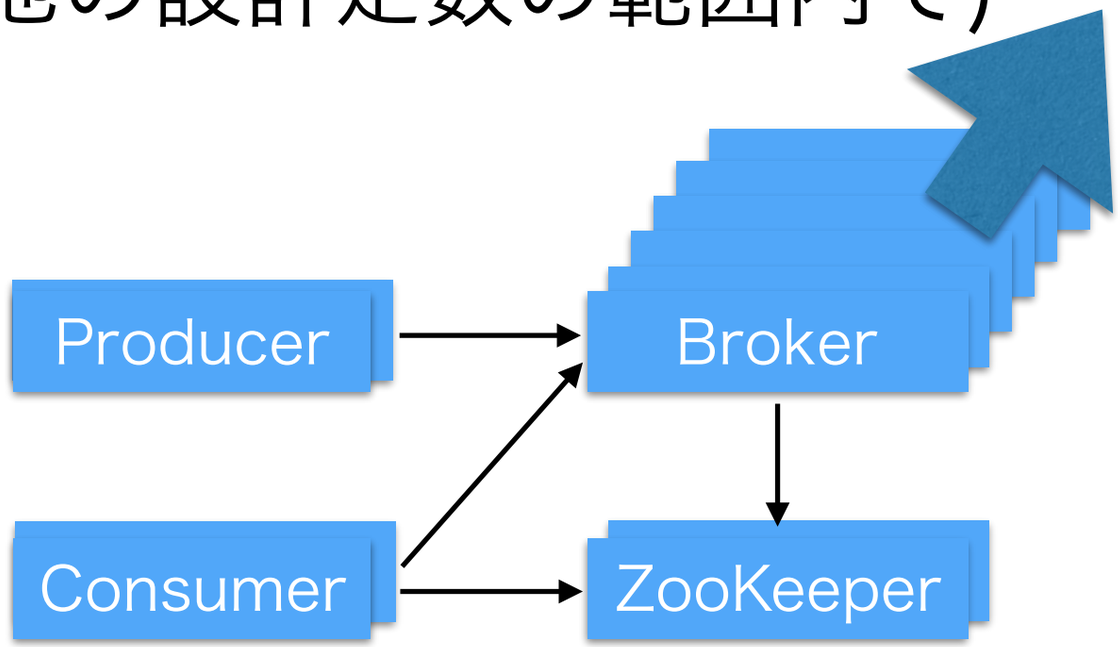
ノード構成

- クライアント (APサーバー、バッチ計算ノード; Producer/Consumer)
- Brokerノード (この部分をKafkaクラスタと考える
- ZooKeeperノード



Scalability

- メッセージ量が増大していっても、
  Brokerの数を増やせば
  (パーティション数など他の設計定数の範囲内で)
  処理能力を増やせる。



メッセージの保証

<u>一般に</u>

- At-Least-Once:
  必ずどれかで処理されるが、
  重複がありうる
- At-Most-Once:
  重複はないがロスするかもしれない
- Exactly-Once:
  重複もロスもしない

メッセージの保証

Kafkaではconumer処理の書き方の協調が必要:

- At-Least-Once:
  メッセージを受け取って、
  処理が終わってからコミット
- At-Most-Once:
  メッセージを受け取りを
  コミットしてから処理
- Exactly-Once:
  メッセージの処理とコミットを同時に

2. AutoIndex

新製品HUEのMW基盤

- Cassandraをメイン永続層に採用し、
  スケーラビリティと、
  マルチデータセンタ要件、
  耐障害性を確保。
- Elasticsearchで全文検索能力の付与。
- 全てAWS上 (他クラウドもサポート予定)

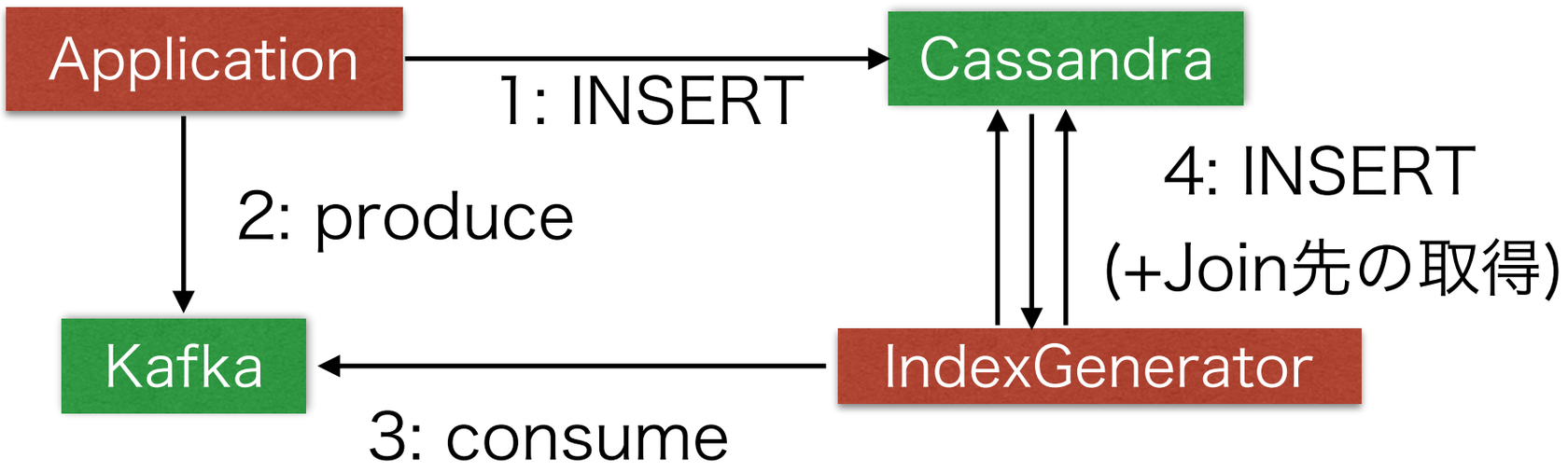
正規化するか否か?

- RDBであればクエリ時にジョインができるのだが、Cassandraにはジョインがない。
- Elasticsearchに入れるデータはそもそも非正規化しないと有意義なクエリができない。

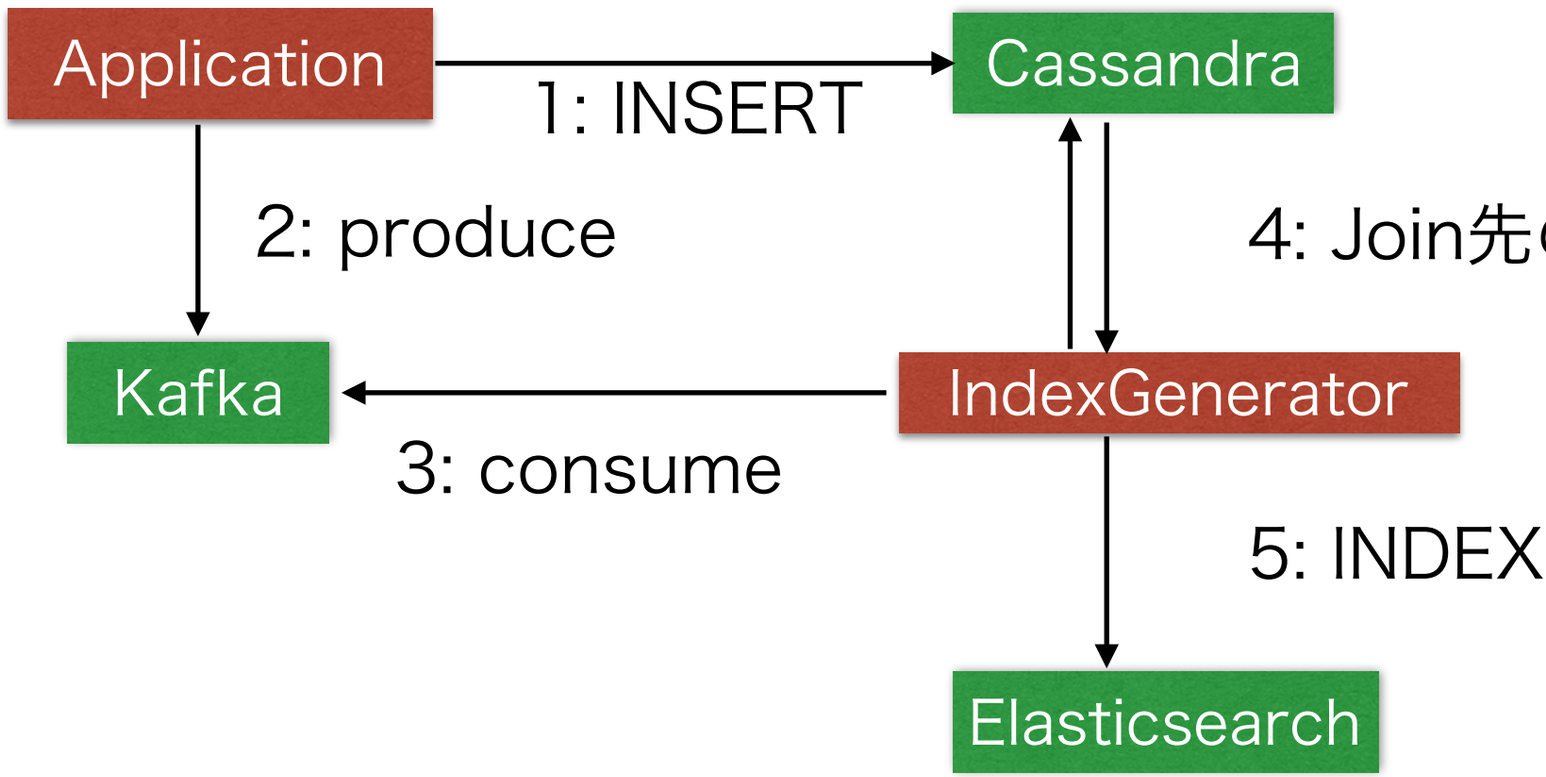
- 非正規化データを基本とすると、更新時に複数の書き込みが発生してしまい、性能維持と整合性確保が困難に。

AutoIndex

- Cassandraには正規化データのみ登録
- INSERTの際にメッセージをproduce
- 常時走るconsumerでメッセージを検知して非正規化データを常に更新



バリエーション


Application
1: INSERT
Cassandra
2: produce
4: Join先の取得
Kafka
IndexGenerator
3: consume
5: INDEX
Elasticsearch

Kafkaが向く理由

- 高いスループット要求。
- Cassandraの操作が基本的に冪等なので、At-Least-Onceモードが使える。
- Cassandra側のスケールに合わせてKafkaもスケールできる。

Kafkaで困る部分

- クラスタがデータセンタを跨げない
  - クラスタ内の障害検知に
    ハートビートをつかっていて、
    パケット遅延にシビア
- クラスタ内通信が平文・認証レス (0.9.0で解消)
- 順序の保証が必要なければ
  他プロトコルでクラスタ間を中継をして解決可能

まとめ

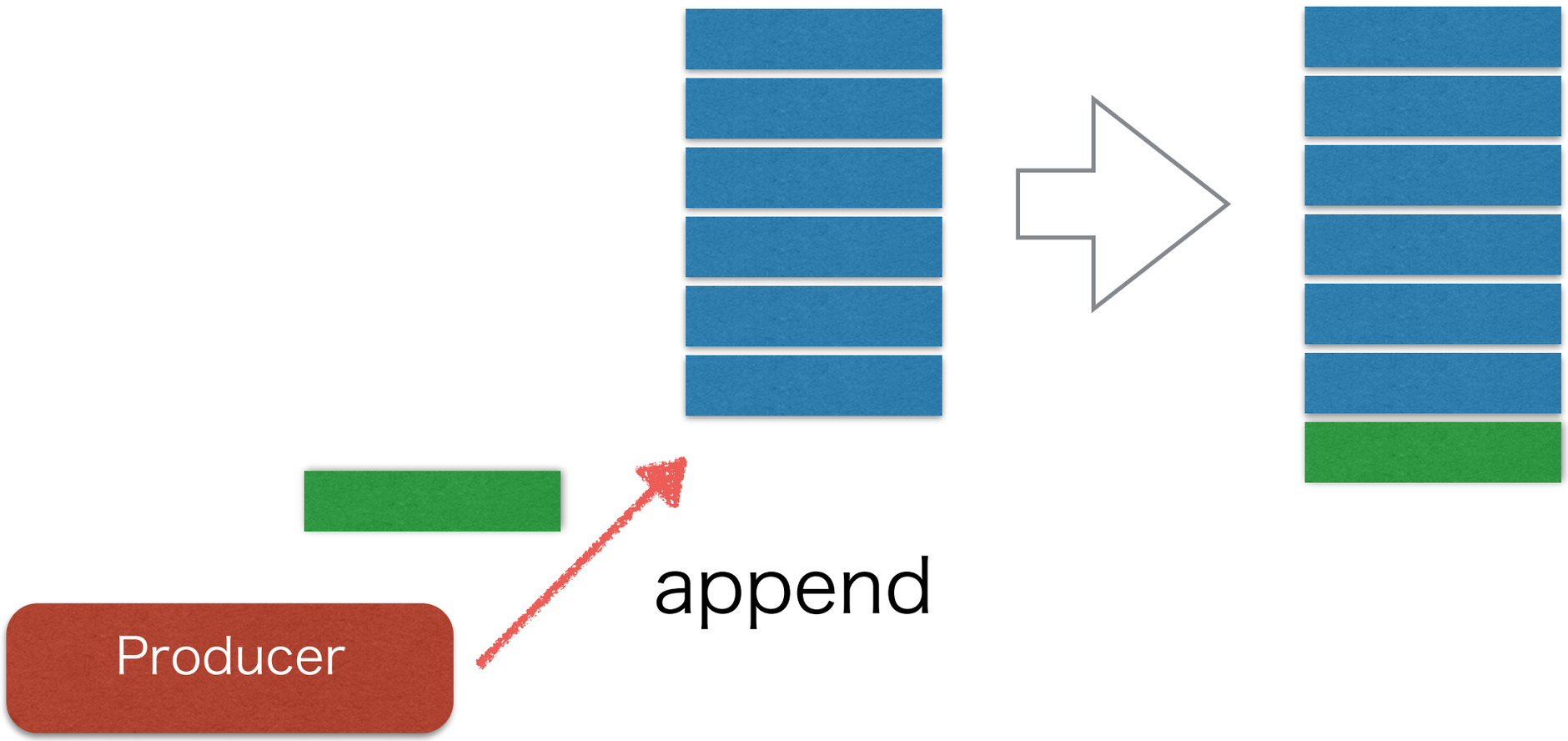
- At-least-once保証で十分な応用であれば scalabilityもあり、クラウド上でも使いやすい。

- メッセージ順序保証の喪失を許容できれば、マルチデータセンタでも利用可能。

- AutoIndexのように特性のマッチする問題で使うことが大切。

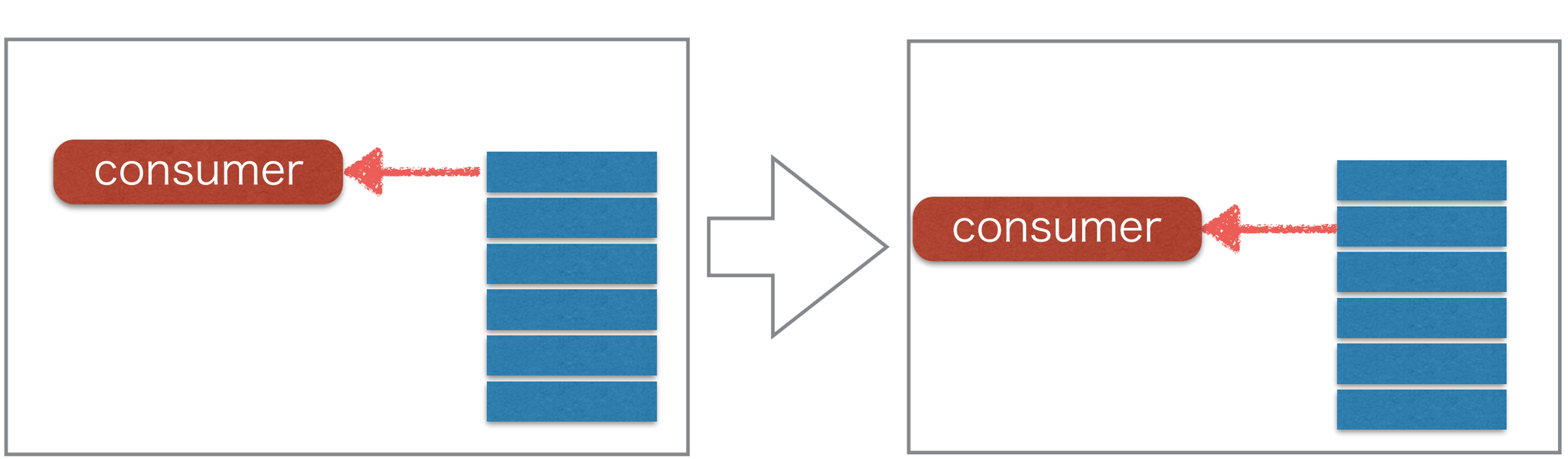
Appendix

メッセージのproduce



メッセージのconsume

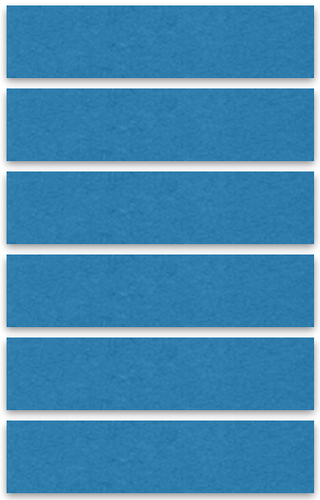
読み終わると、offsetが+1されるだけ。
メッセージ自身は消えない。



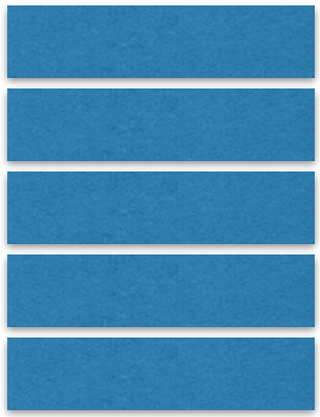
topic

複数のキューのインスタンスがあって
それぞれに名前が付いている。

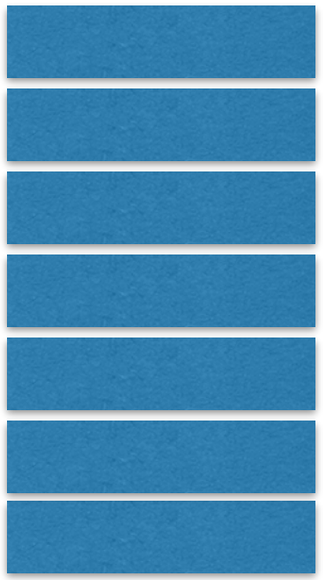
topic1



topic2

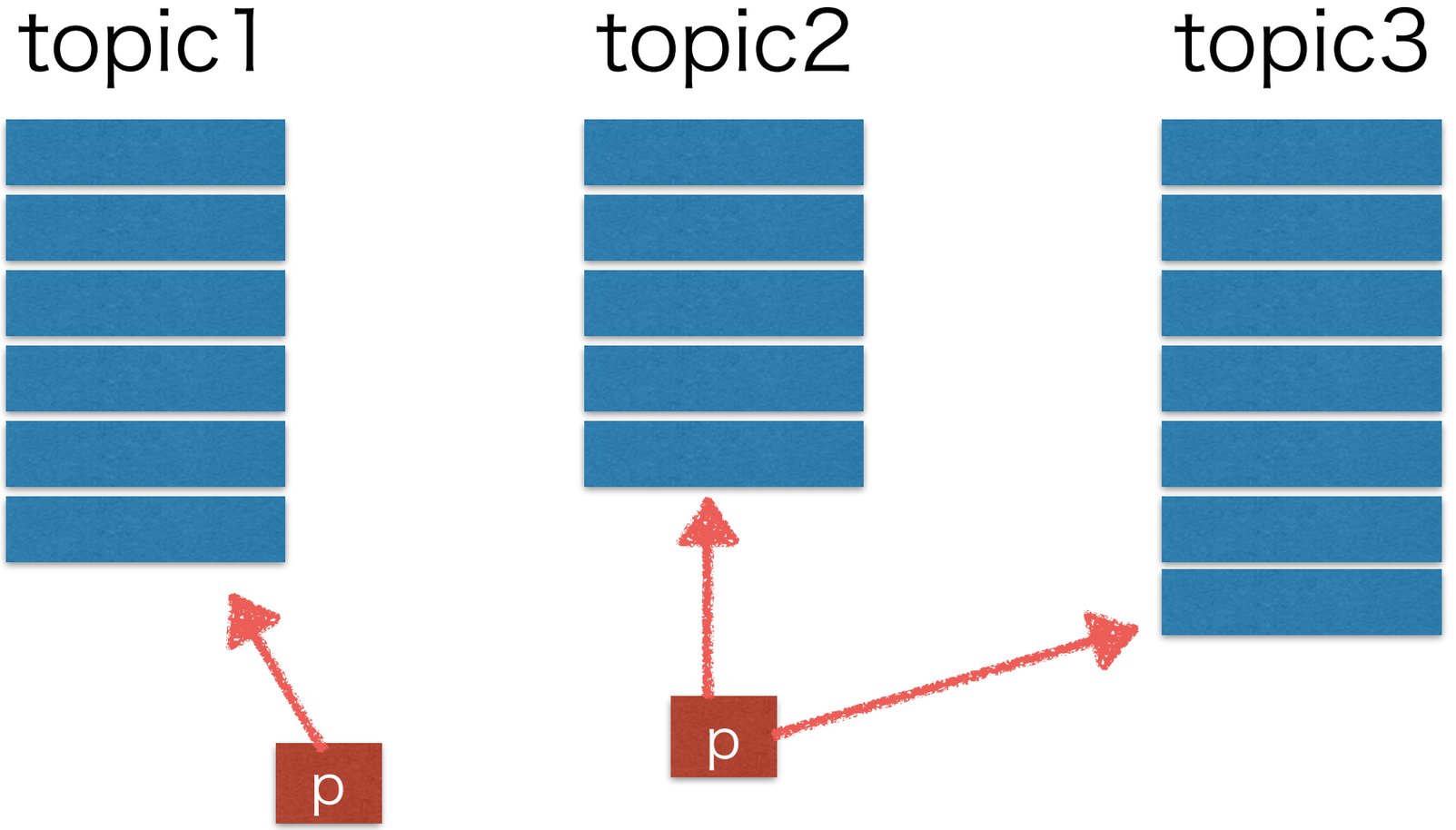


topic3



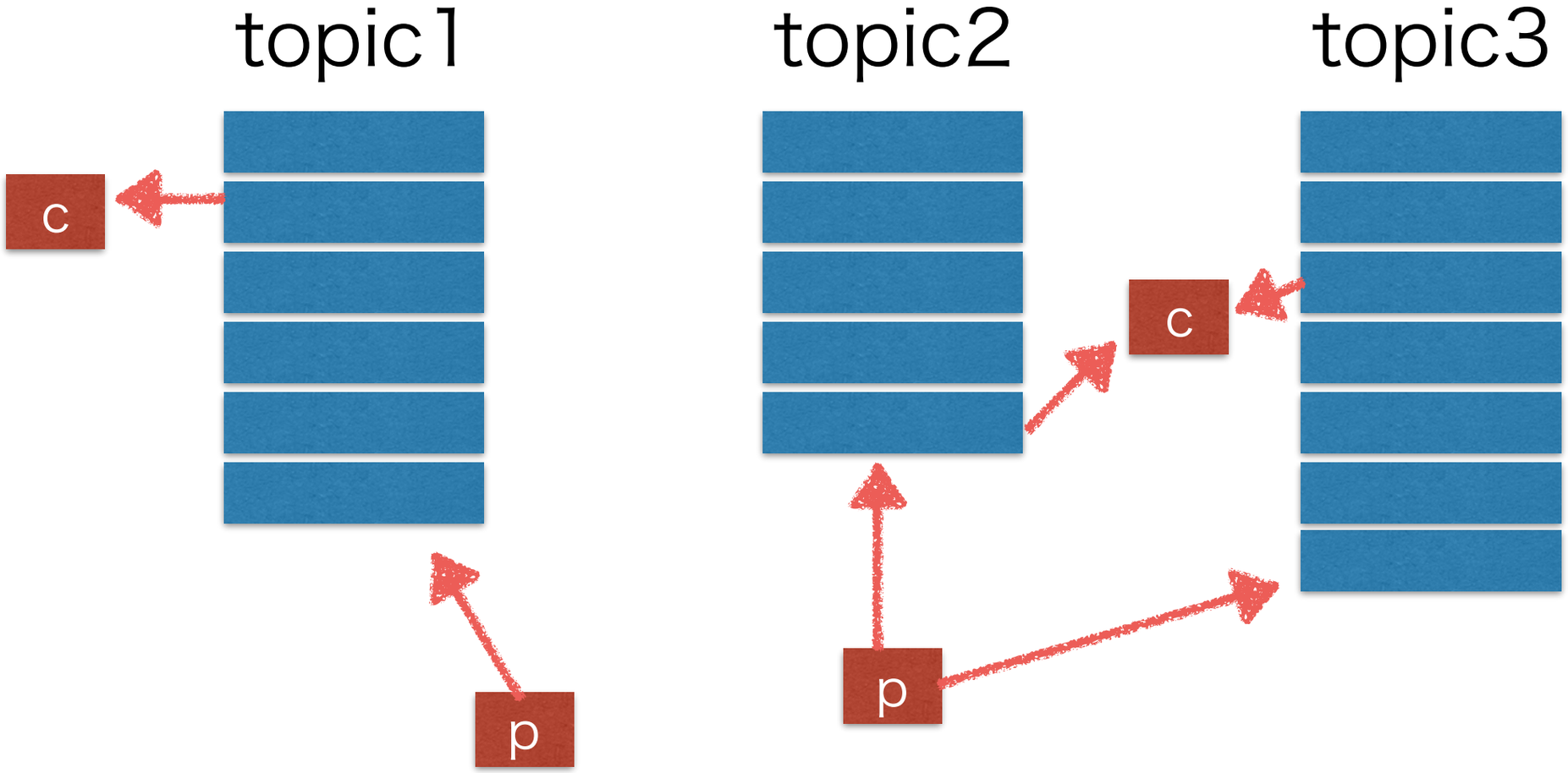
topic

複数のキューのインスタンスがあって
それぞれに名前が付いている。



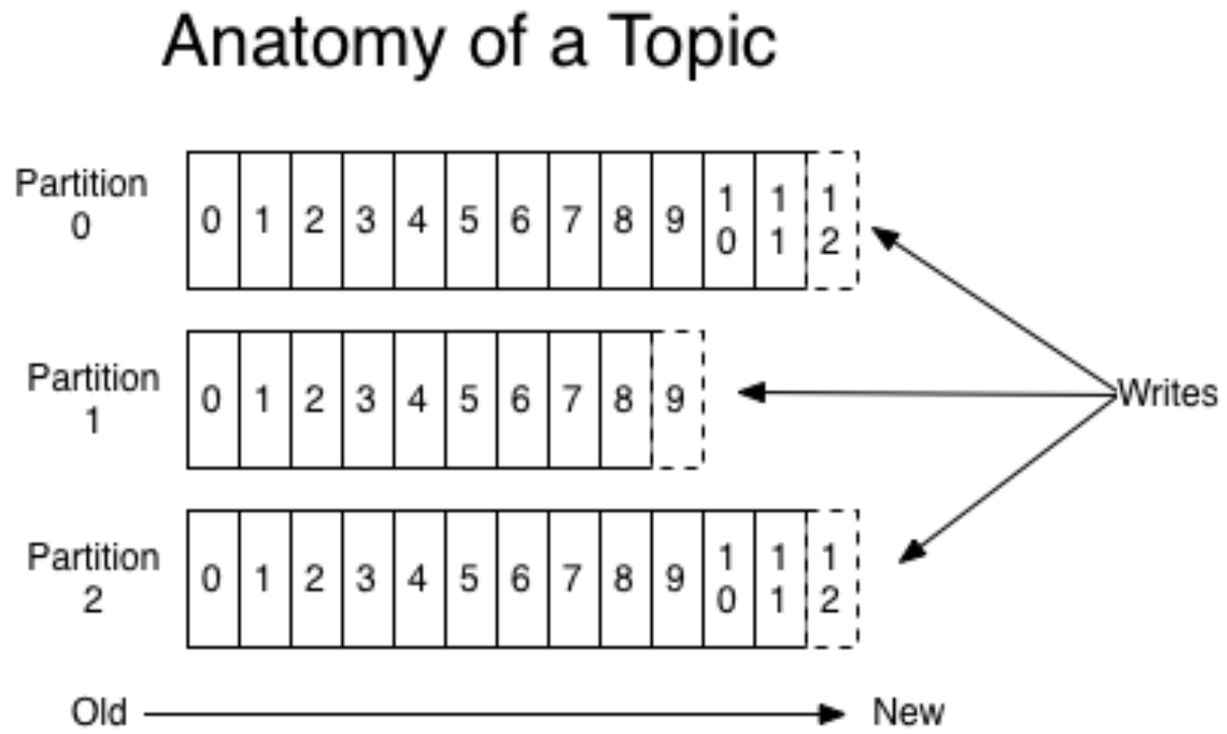
topic

複数のキューのインスタンスがあって
それぞれに名前が付いている。



partition

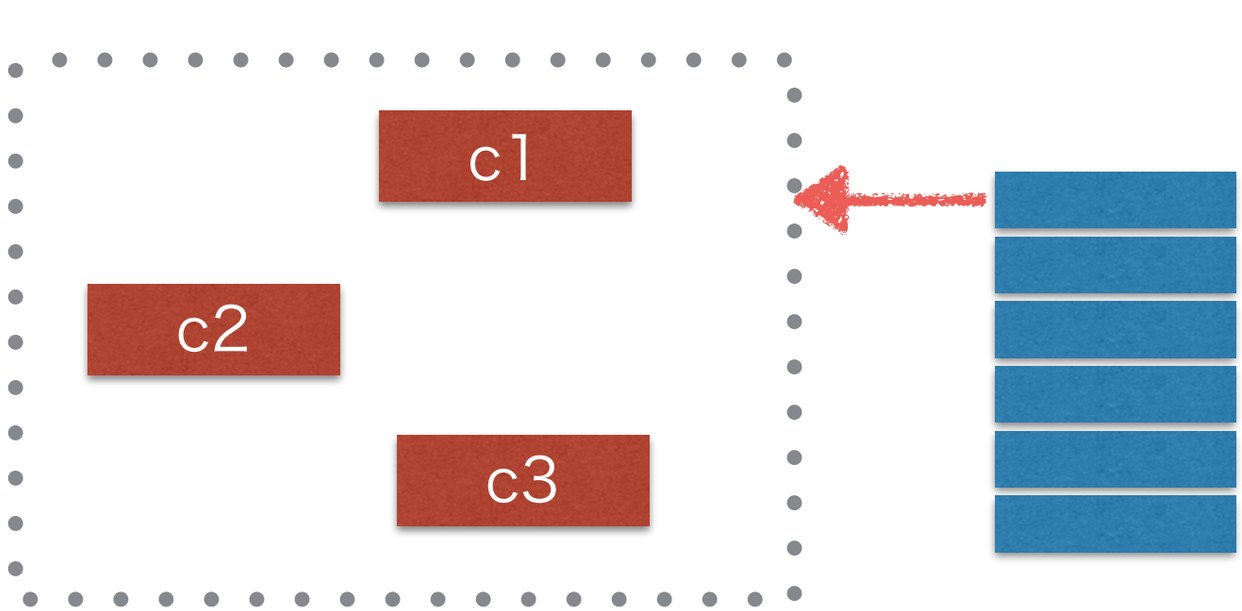
topicを更に分割した単位に自動的に振り分け。
同一partition内でしか順序が付かない。



出典: http://kafka.apache.org/documentation.html

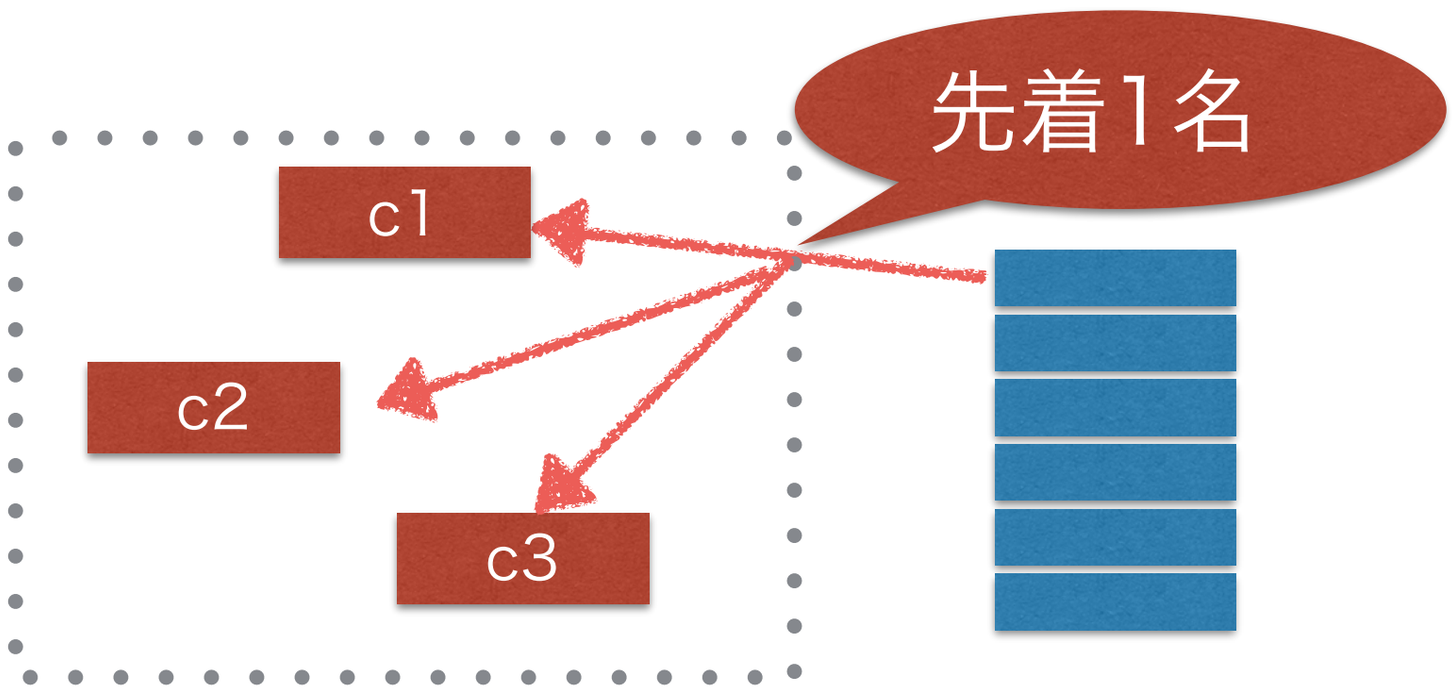
consumer group

同じoffsetを共有するconsumerの集団。
スレッド/プロセス/実行ノードなどは違うかもしれない。



queuing mode

全consumerが同じgroupの場合、
どれか一つがかならず受信する。



pub-sub mode

全consumerが異なるgroupの場合、
全consumerがかならず受信する。