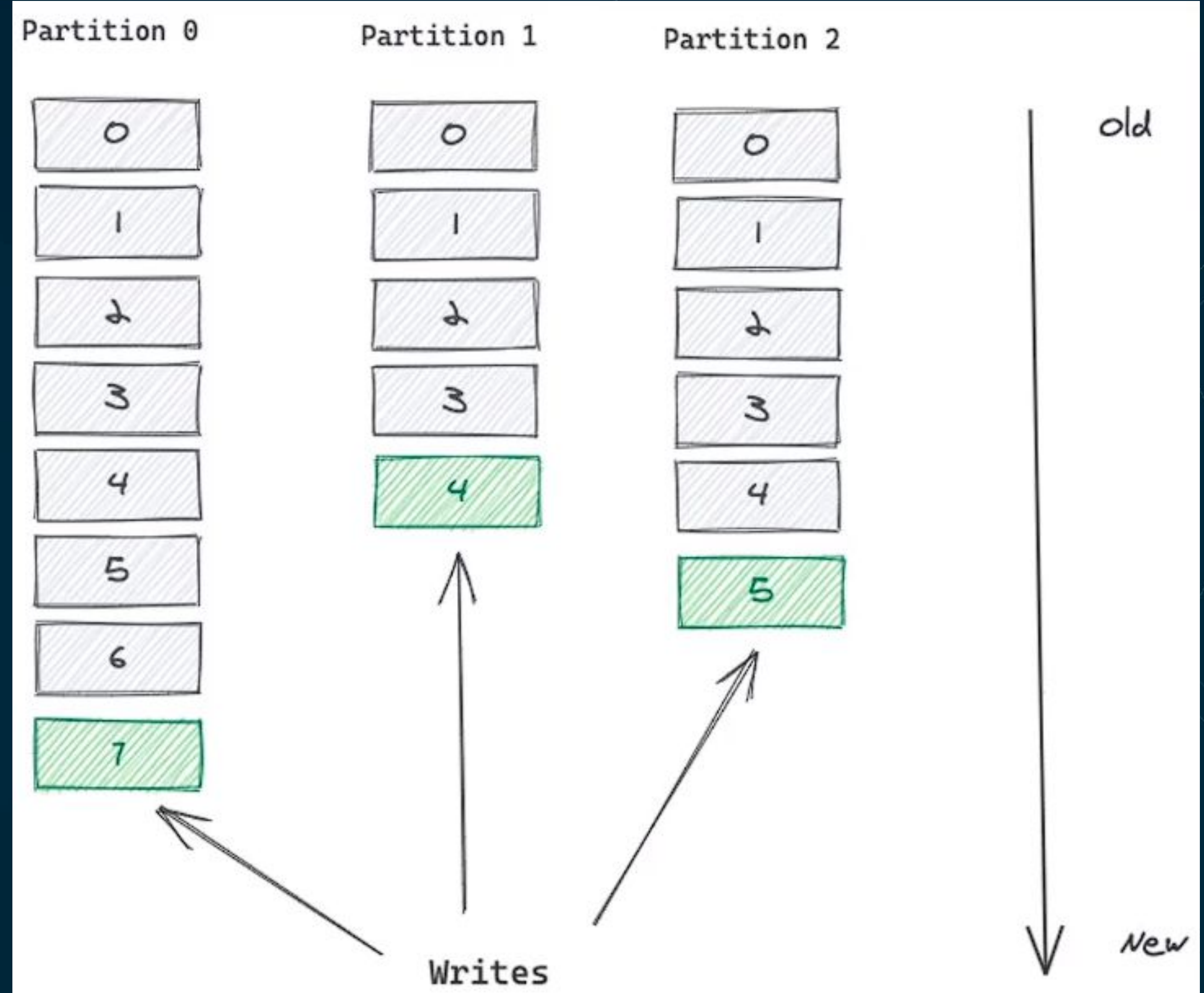


Produtores, Consumidores, Tópicos e suas principais linguagens e drivers

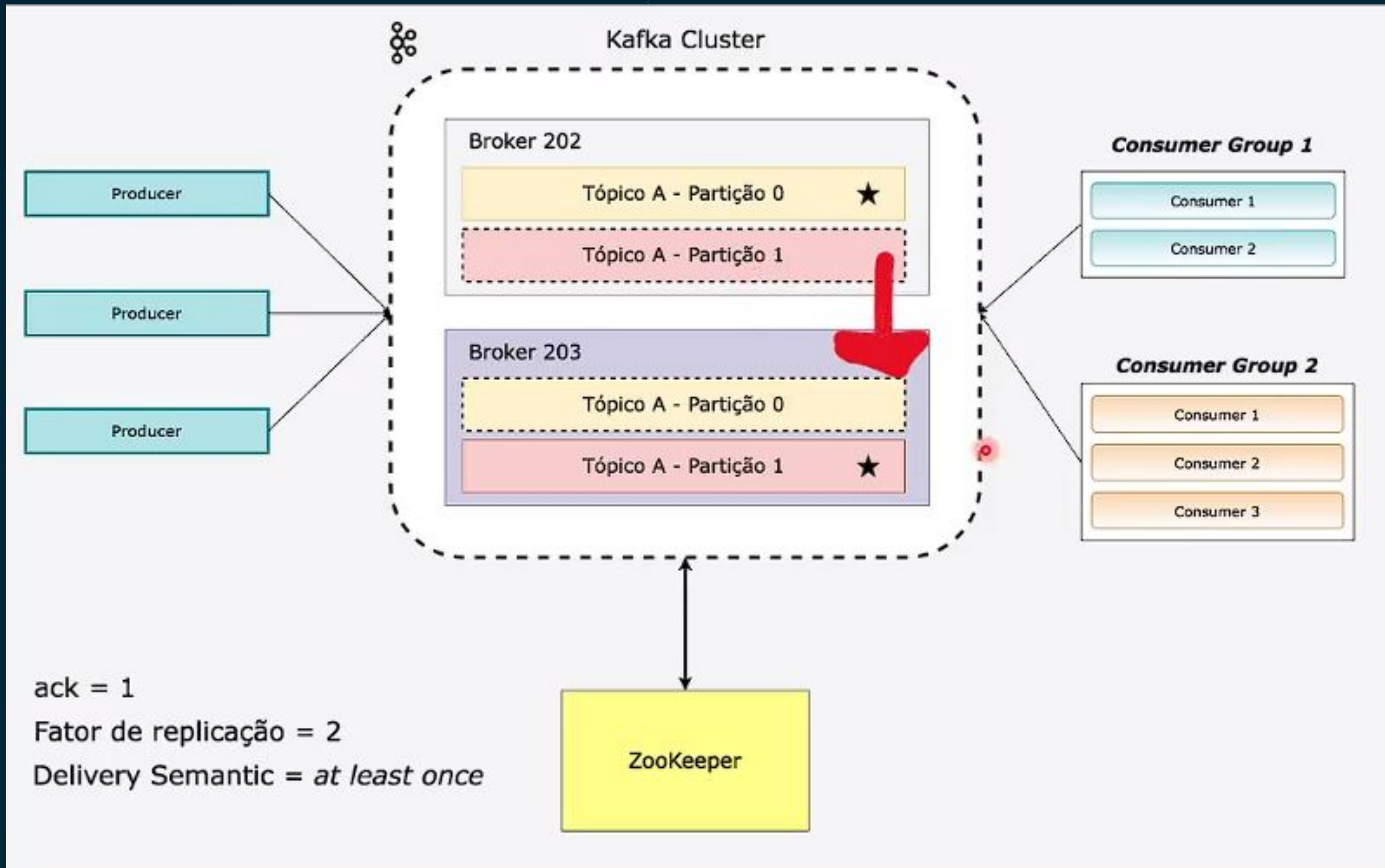
Tópicos e partições

- Tópicos são unidades lógicas, divididos fisicamente em partições.
- Um tópico pode ter milhares de partições.
- Partições garantem escalabilidade.
- Princípio FIFO (First-In-First-Out) por partição.
- Cada mensagem possui: Value, Key e Headers.
- Mensagens são entregues na partição baseado na hash da Key.



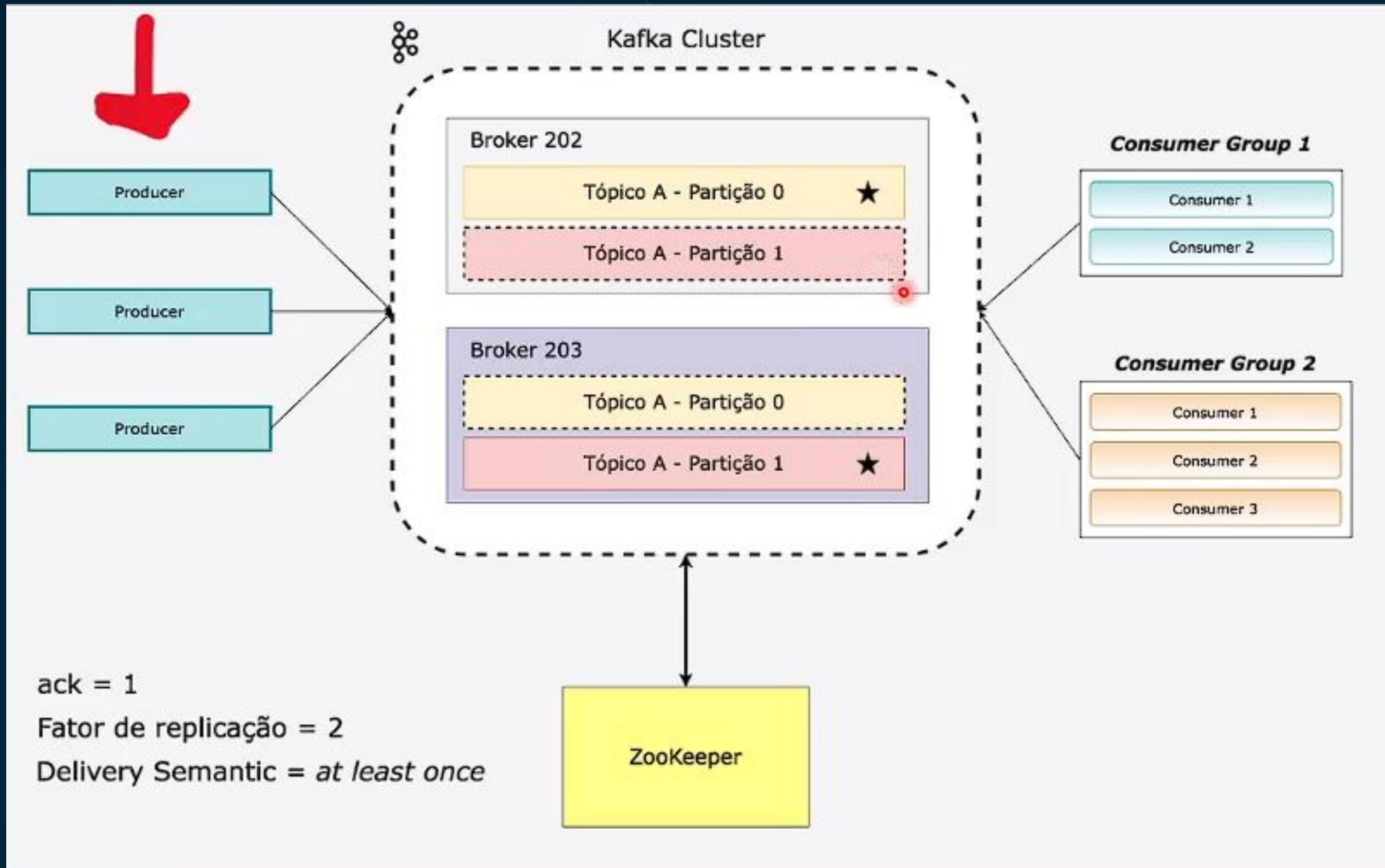
Tópicos e partições

- Partições são redundantes.



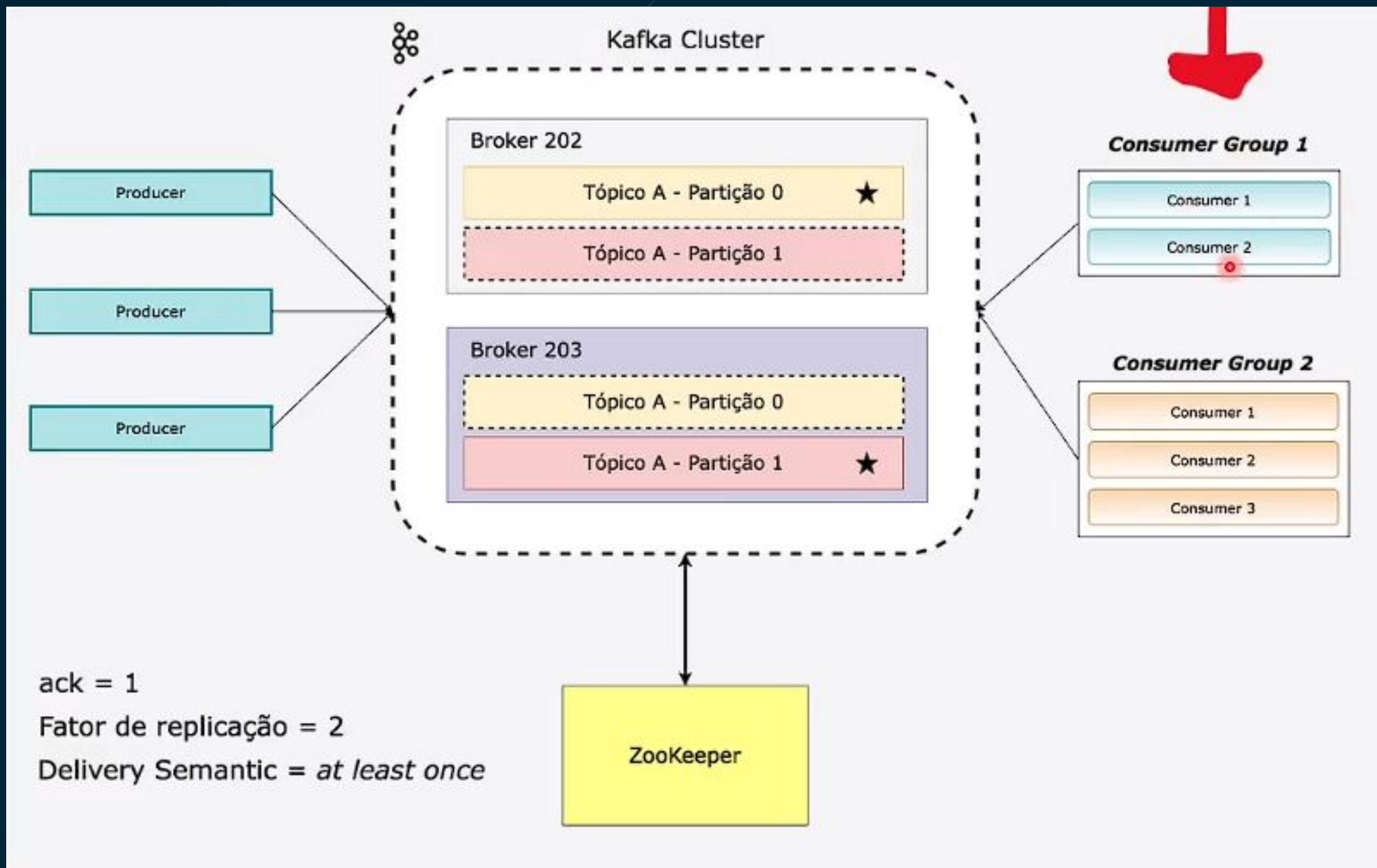
Produtores

- Responsável por produzir mensagens na partição.
- Define como será a entrega da mensagem baseado no ACK = 0,1 ou all.
- ACK = 0 . Não espera resposta, fire-and-forget.
- ACK = 1 . Espera a resposta apenas no líder.
- ACK = all . Espera a resposta de todos os brokers configurados no min.insync.replica.



Consumidores

- São logicamente distribuídos em consumer groups.
- Cada consumer dentro do mesmo consumer group recebe apenas uma vez uma mensagem de uma partição.
- Consumers groups diferentes podem receber a mesma mensagem.
- Cada partição não pode ter mais de um consumidor para o mesmo consumer group.



Consumidores (cluster de 3 nós)

Semântica	COMMIT	Idempotente?
At most once	Antes do Proc.	Não
At least once	Depois do Proc.	Não
Exactly Once	Depois do Proc.	Sim



Principais linguagens e drivers

- Exemplos de produtores e consumidores em várias linguagens no [Github](#). (Java, Golang, Python, .NET, NodeJs entre outras).
- A maioria das linguagens usam o [librdkafka](#) escrito em C como wrapper para comunicação de baixo nível com o Kafka. Com exceção de Java, Scala, Clojure, Kotlin e o próprio C.