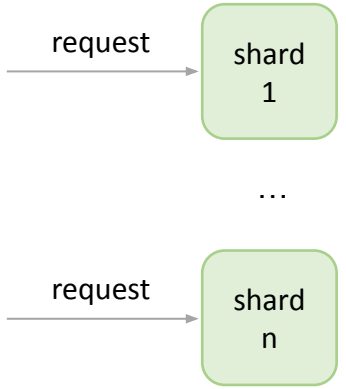
Partitioning in real-life systems

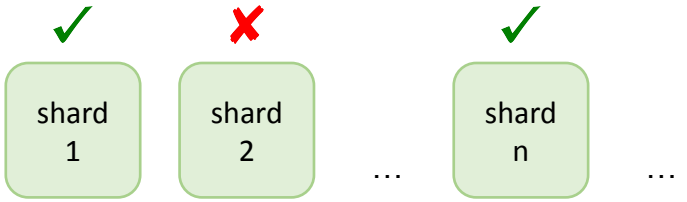
partitioning helps improve

- scalability

- performance

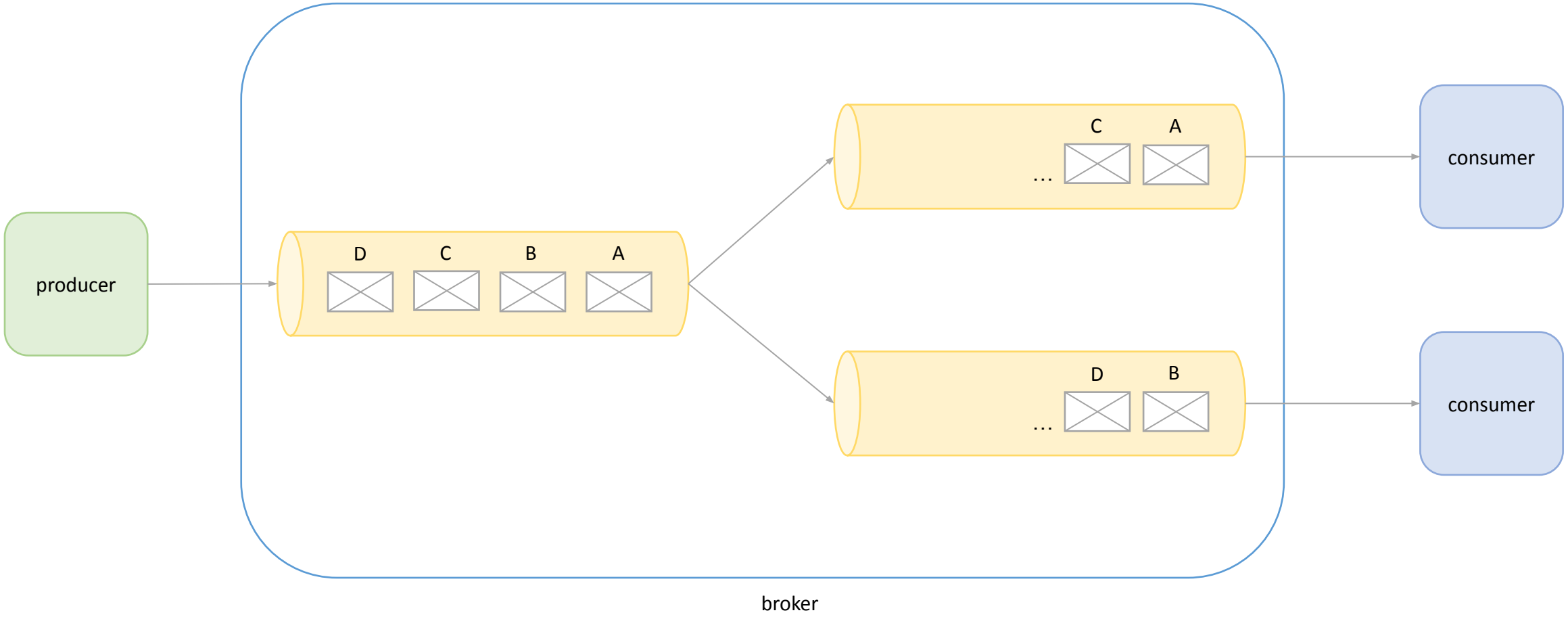


- availability



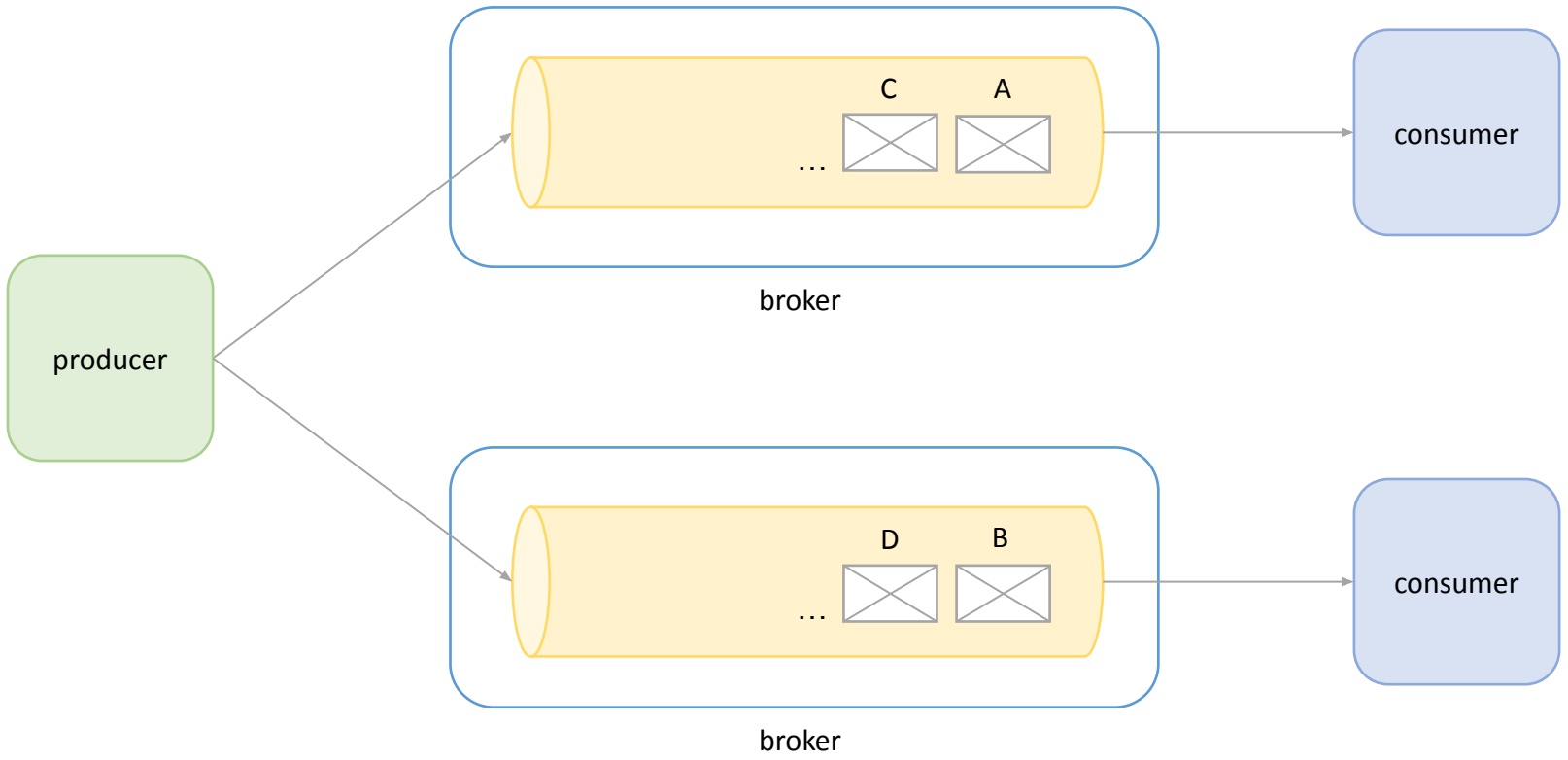
Partitioning in real-life systems

messaging systems – RabbitMQ – sharded queues



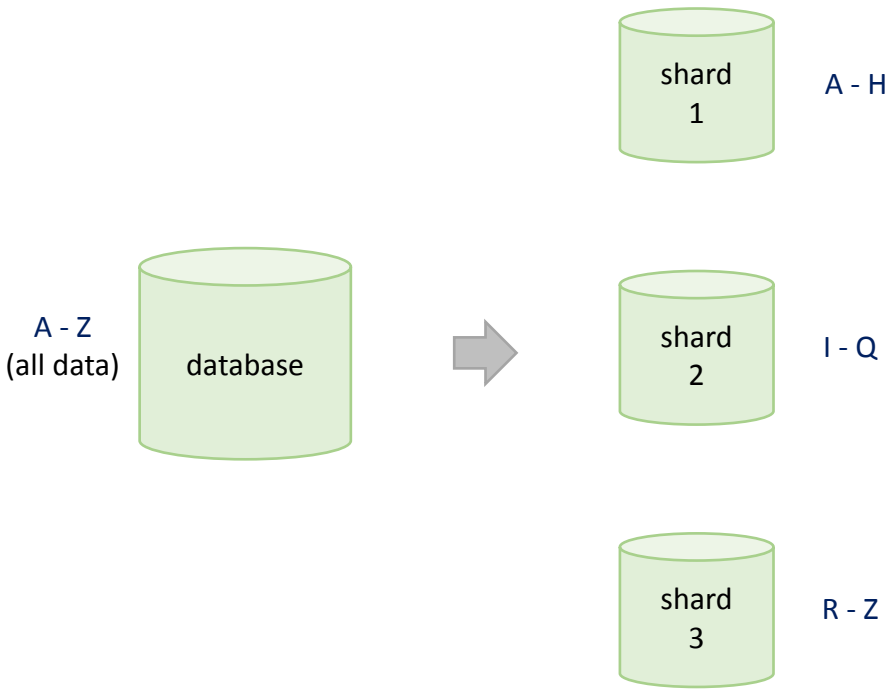
Partitioning in real-life systems

messaging systems – Kafka and Kinesis



Partitioning in real-life systems

databases



Which node should we write data to?

How to pick a node when reading data?

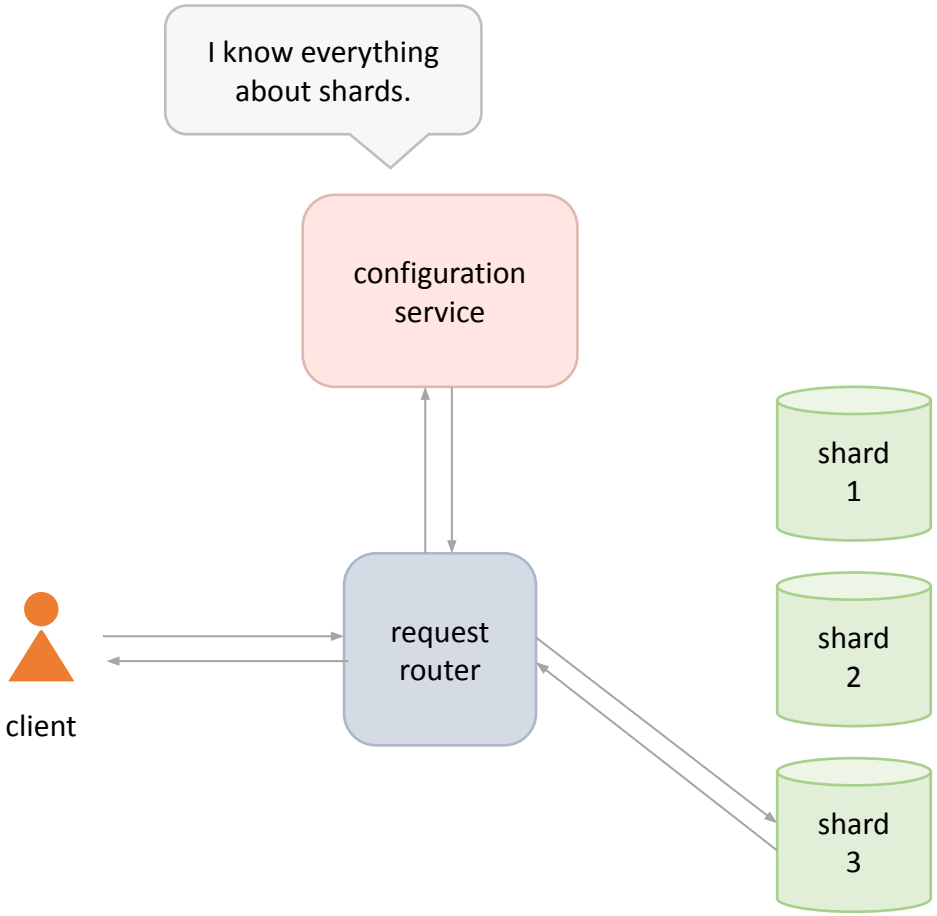
How to ensure that data is evenly distributed among the nodes?

Does “evenly distributed” mean that the nodes store the same volume of data?

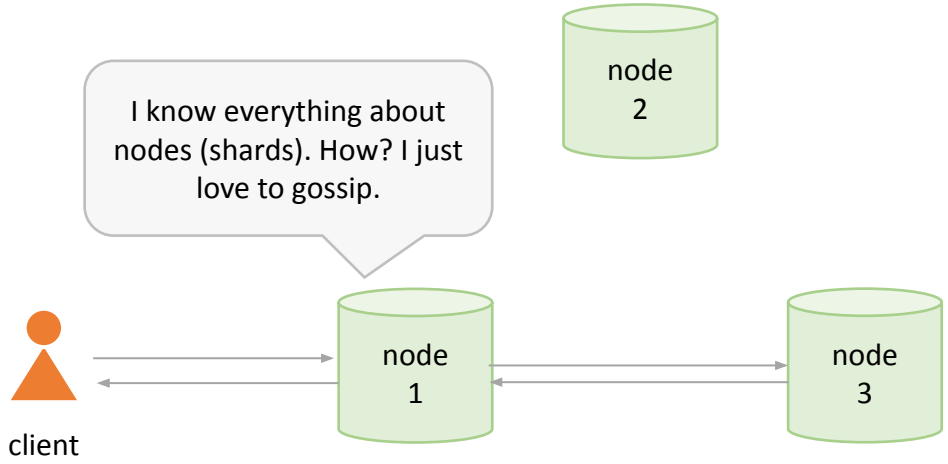
Or does it mean that each node processes the same number of read and write requests?

Partitioning in real-life systems

databases



example
SQL database cluster, MongoDB



example
Dynamo, Cassandra

Partitioning in real-life systems

object storages