TGIP-CN 009

Apache Pulsar performance tuning

Penghui Li

Pulsar components

Client	**Broker**	**Bookie**
Pulsar SQL	Function	IO Connector
Pulsar Proxy	Metadata Store	Tiered Storage

Outline

1.Basic concepts
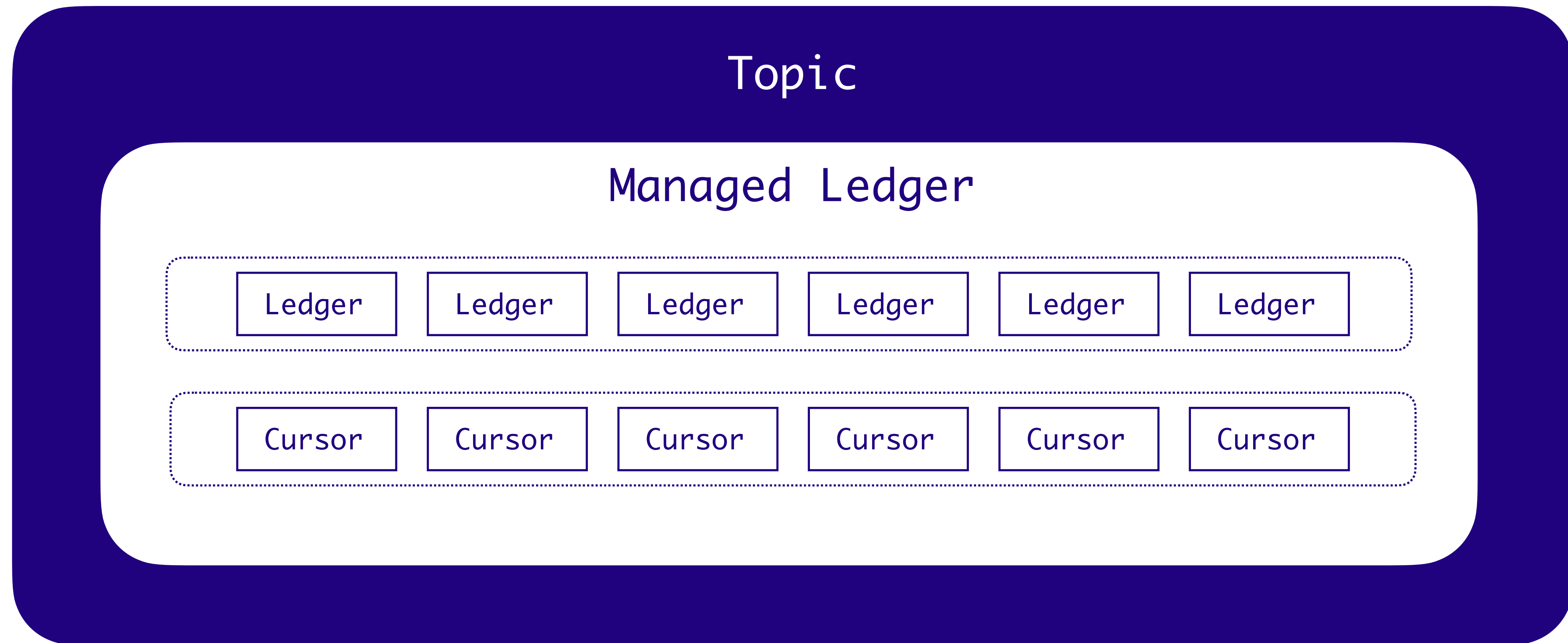
2.Message writing

3.Message reading
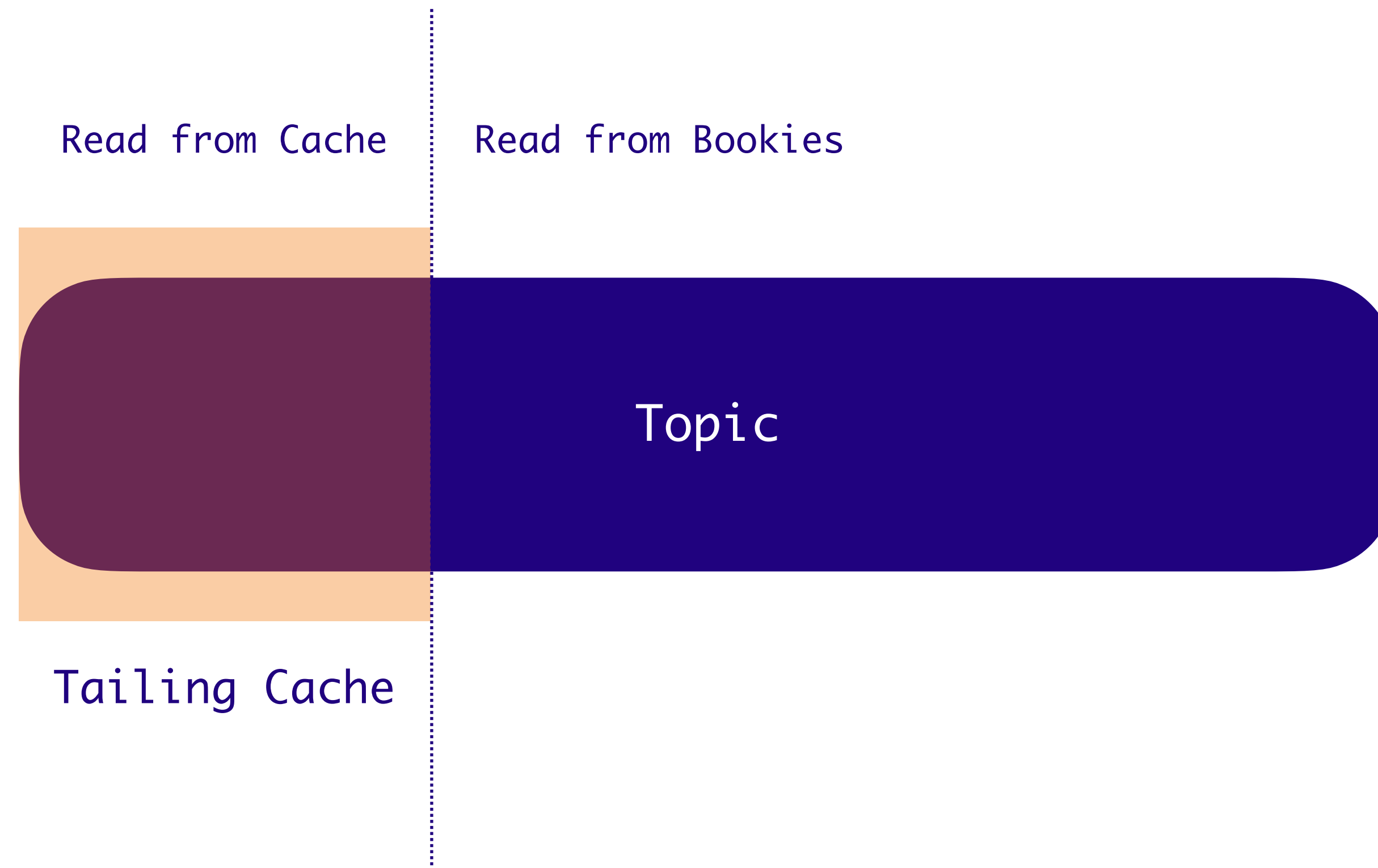
Basic Concepts

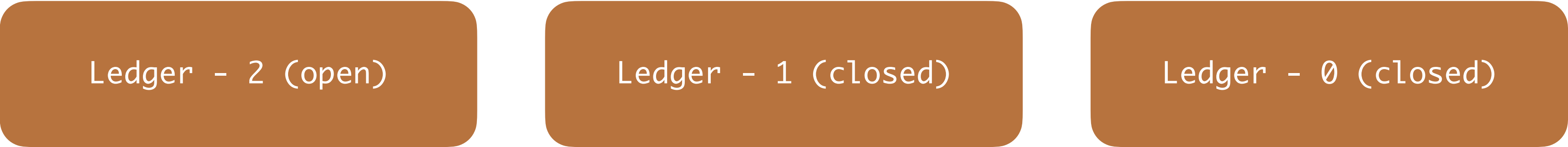# Basic concepts

Basic concepts

Topic

Managed Ledger

| Ledger | Ledger | Ledger | Ledger | Ledger | Ledger |

| Cursor | Cursor | Cursor | Cursor | Cursor | Cursor |

Basic concepts

Read from Cache | Read from Bookies

Topic

Tailing Cache

Basic concepts

Topic		

Ledger - 2 (open)	Ledger - 1 (closed)	Ledger - 0 (closed)	Ledger rollover

Fragment	F - 1	F - 0	Fragment	Bookie failover

E-3	E-2	E-1	E-0	E-3	E-2	E-1	E-0	E-3	E-2	E-1	E-0

Basic concepts

Journal

Entry Log - 0

Index DB

Entry Log - 1

Basic concepts

Single Message

Client   2

Broker   1

Bookie   0

Batched Message

Client   6 7 8

Broker   3 4 5

Bookie   0 1 2
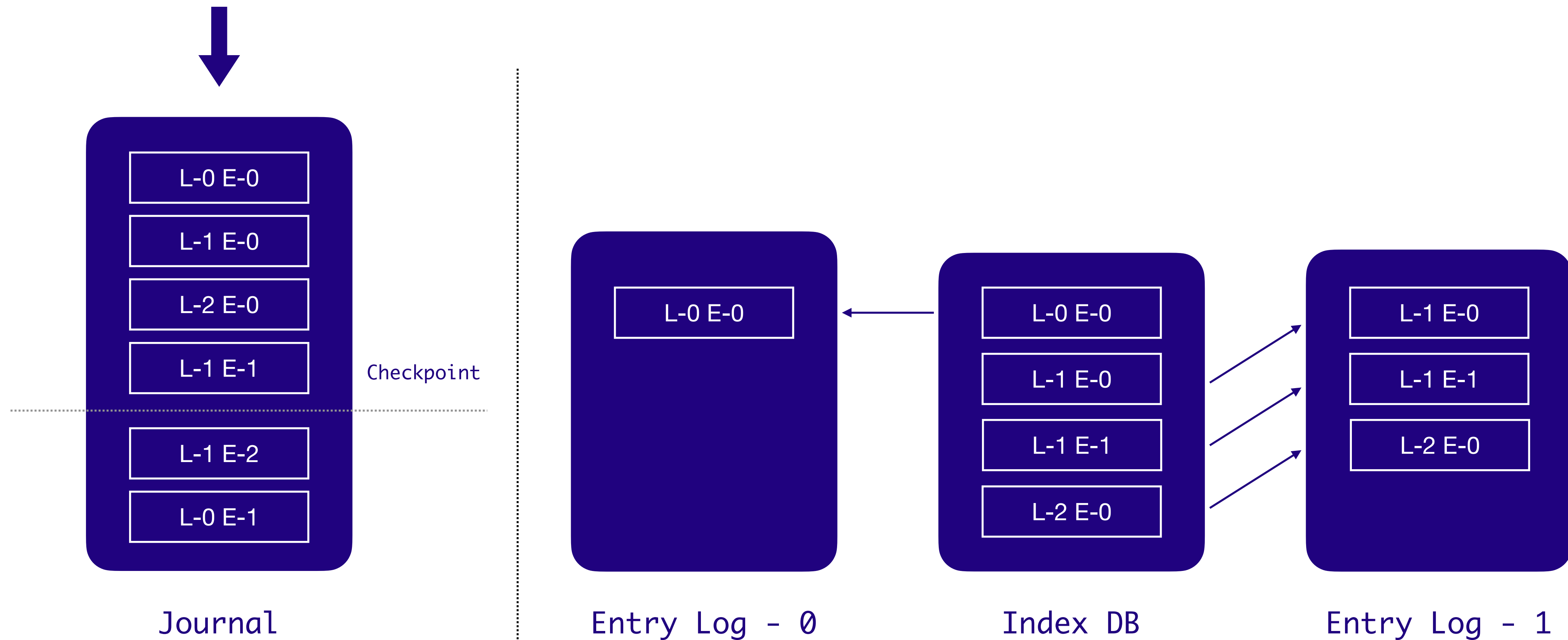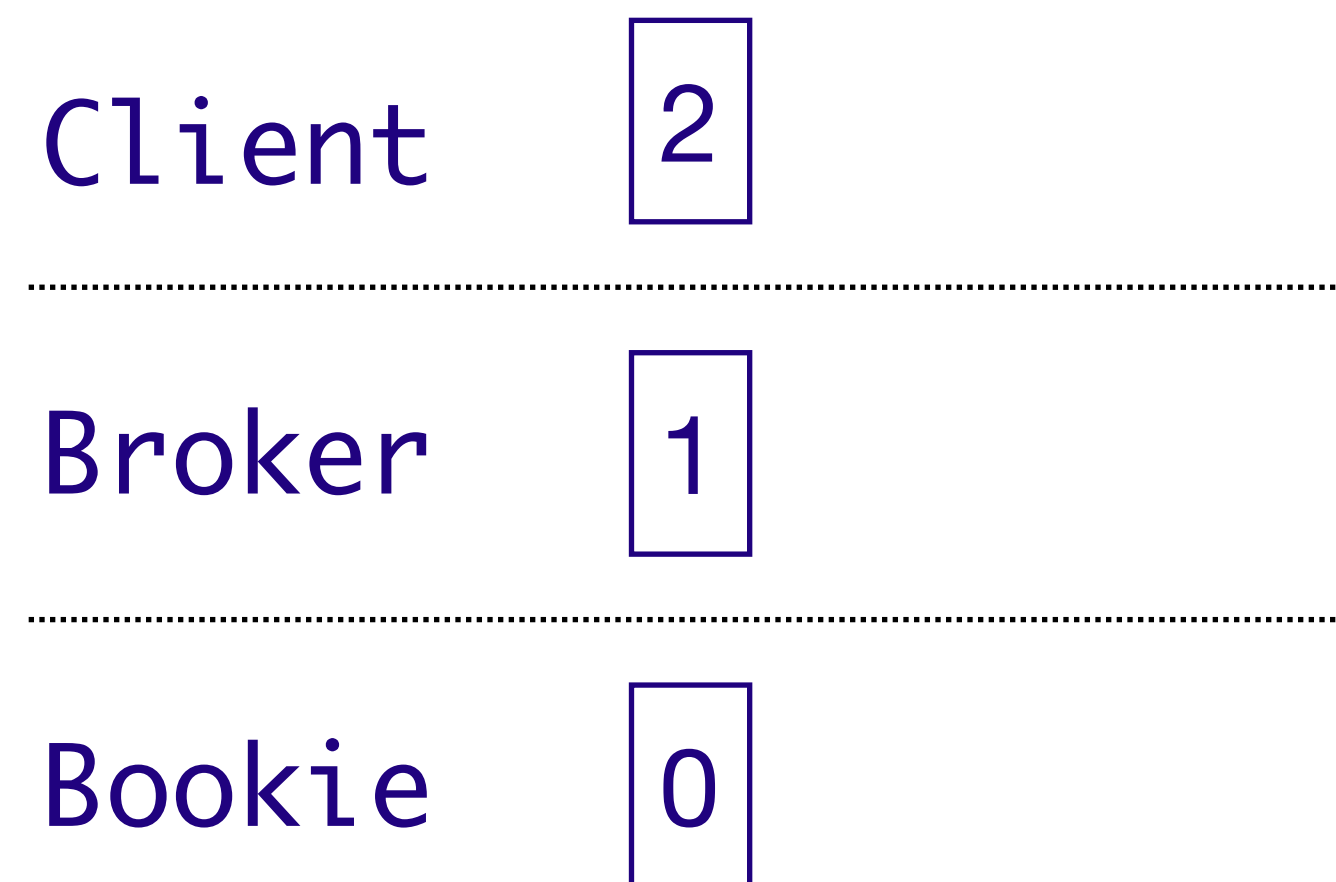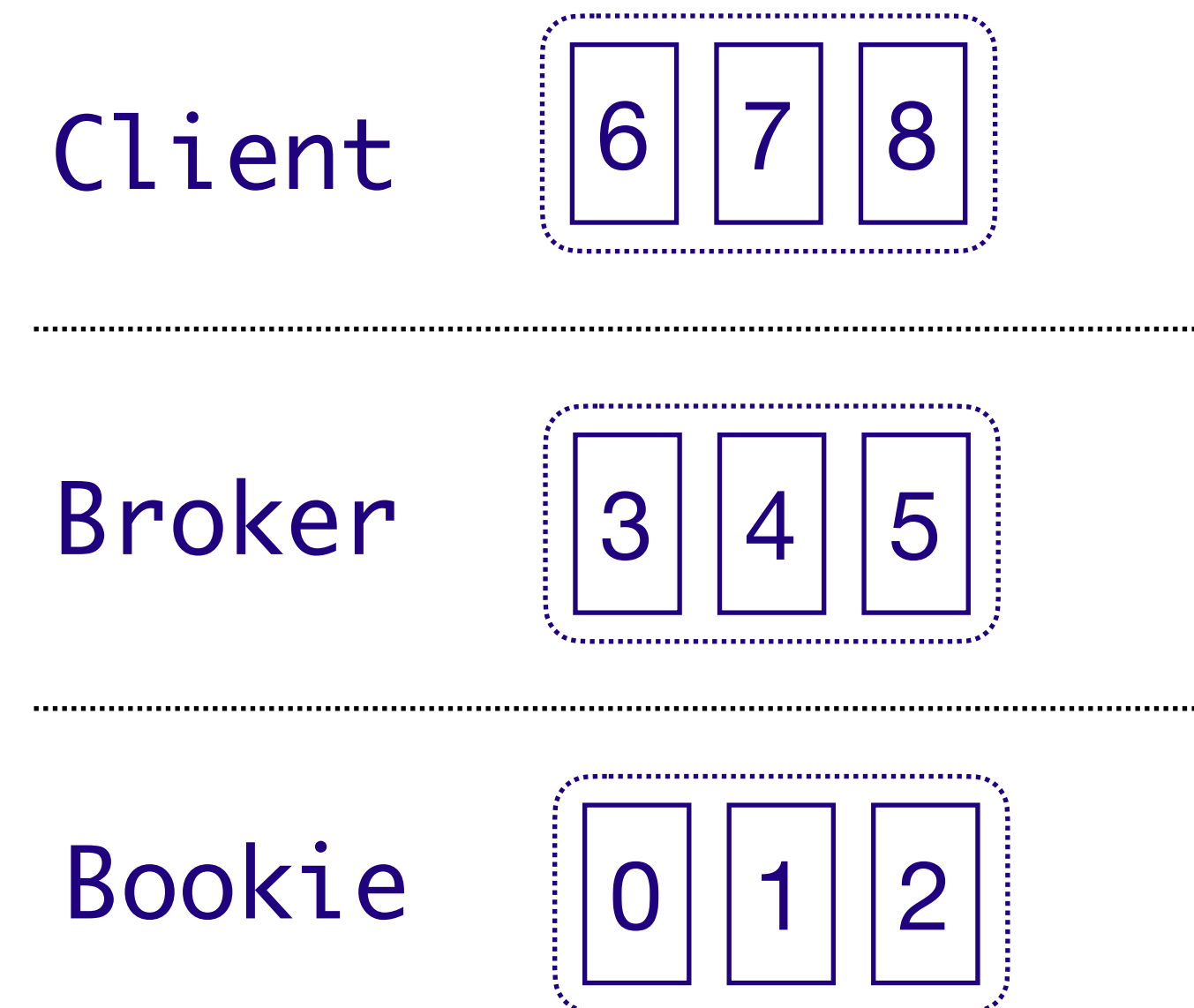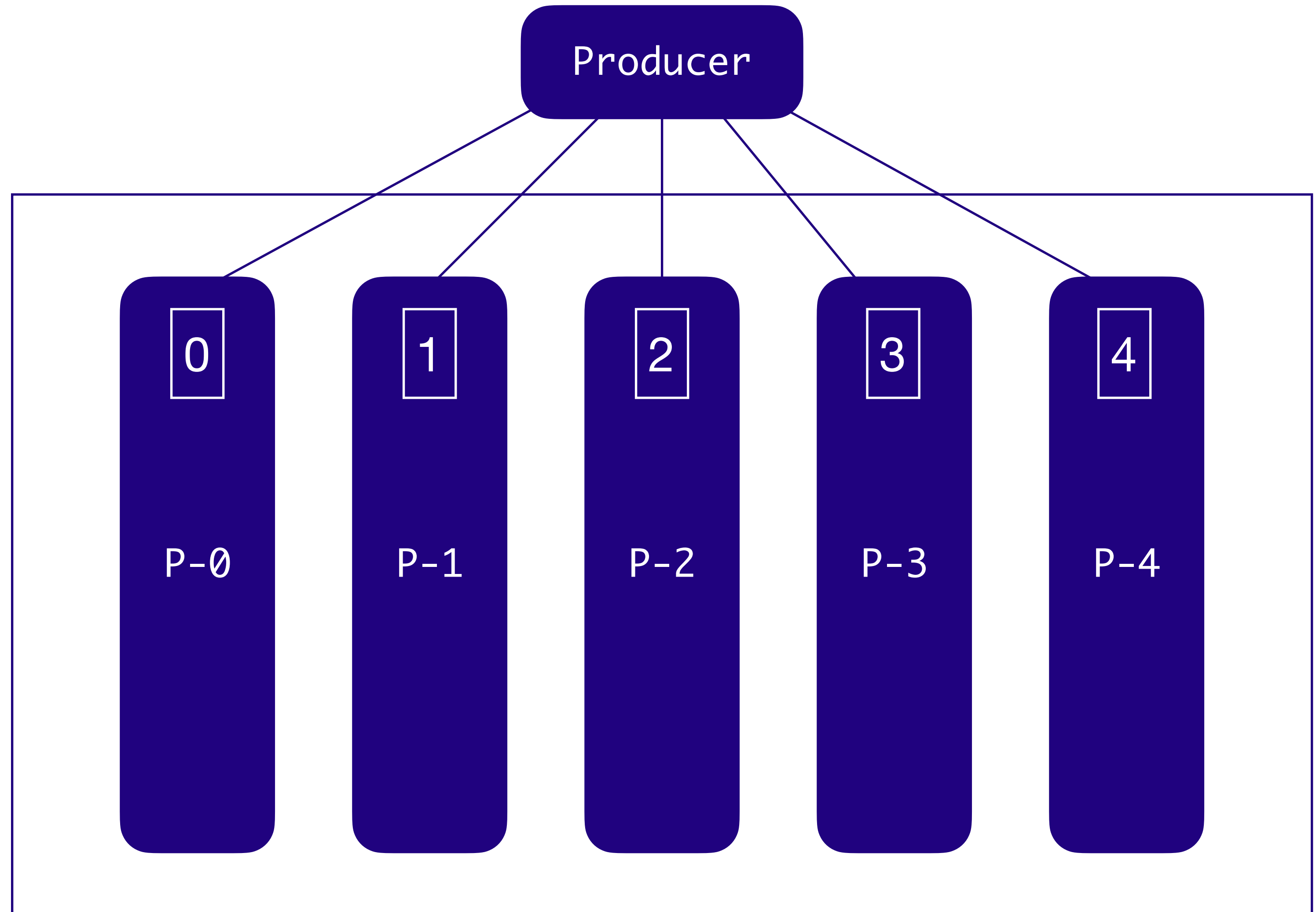
Basic concepts
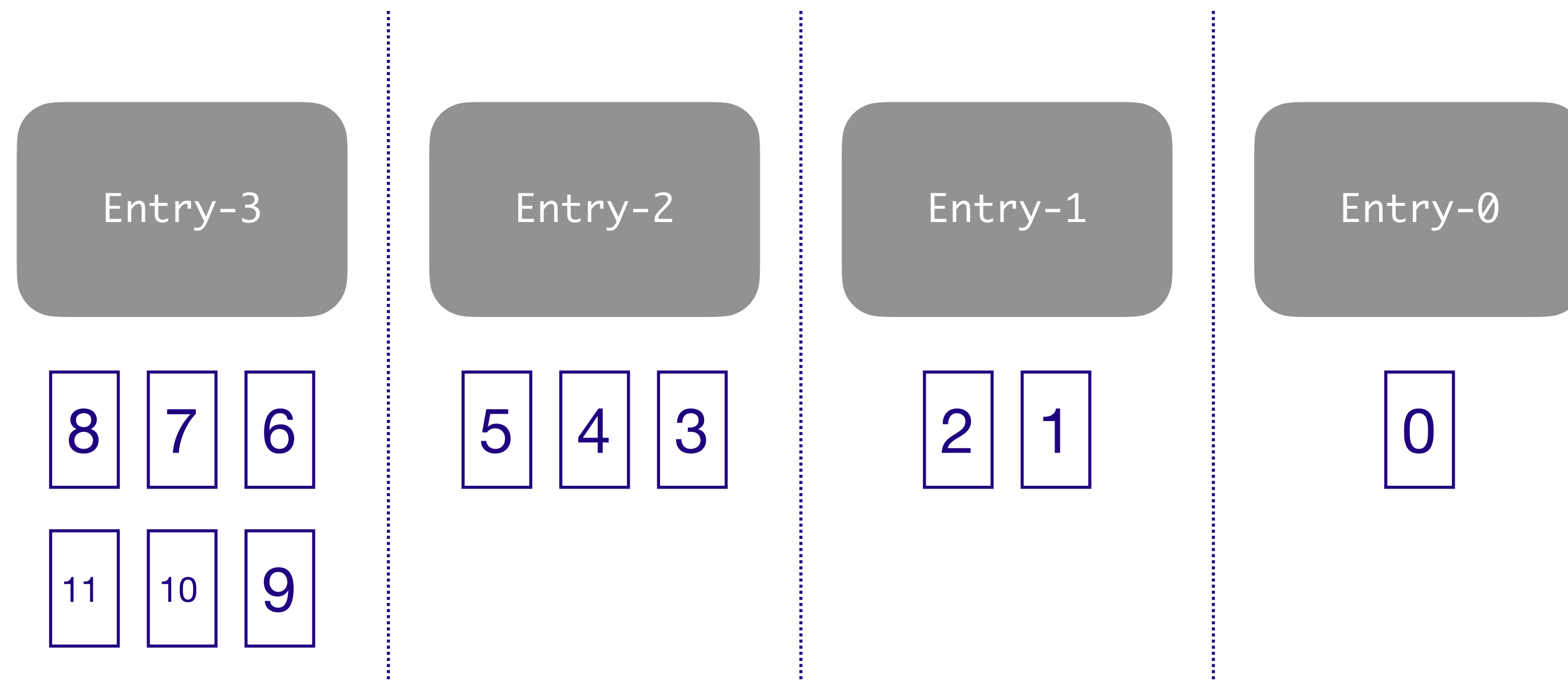
Message writing

Topic

Message route mode

Partition switch

Partitioned Topic

Producer pending queue

Producer

Broker

| . | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

→

| 4 | 3 | 2 | 1 | 0 |

Send response to the client

| . | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 |

←
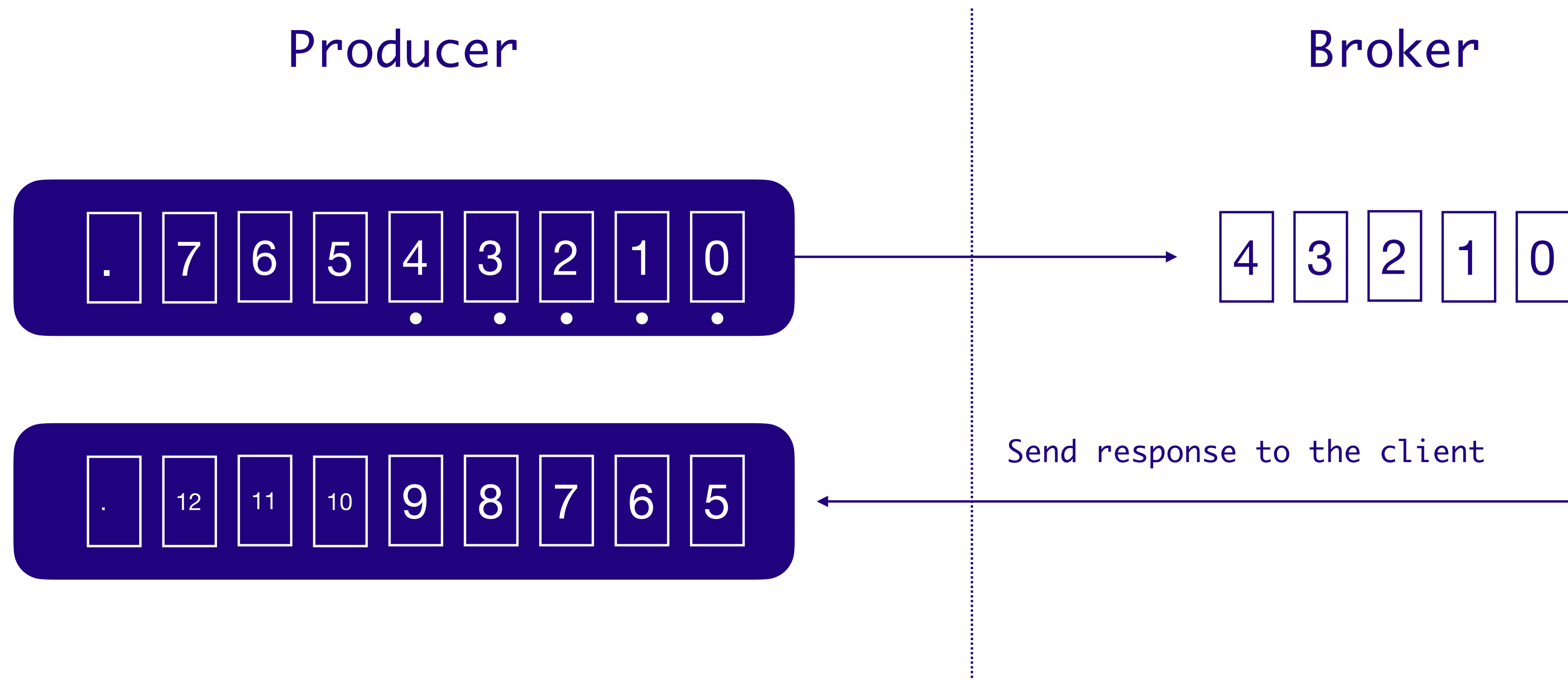
Producer pending queue
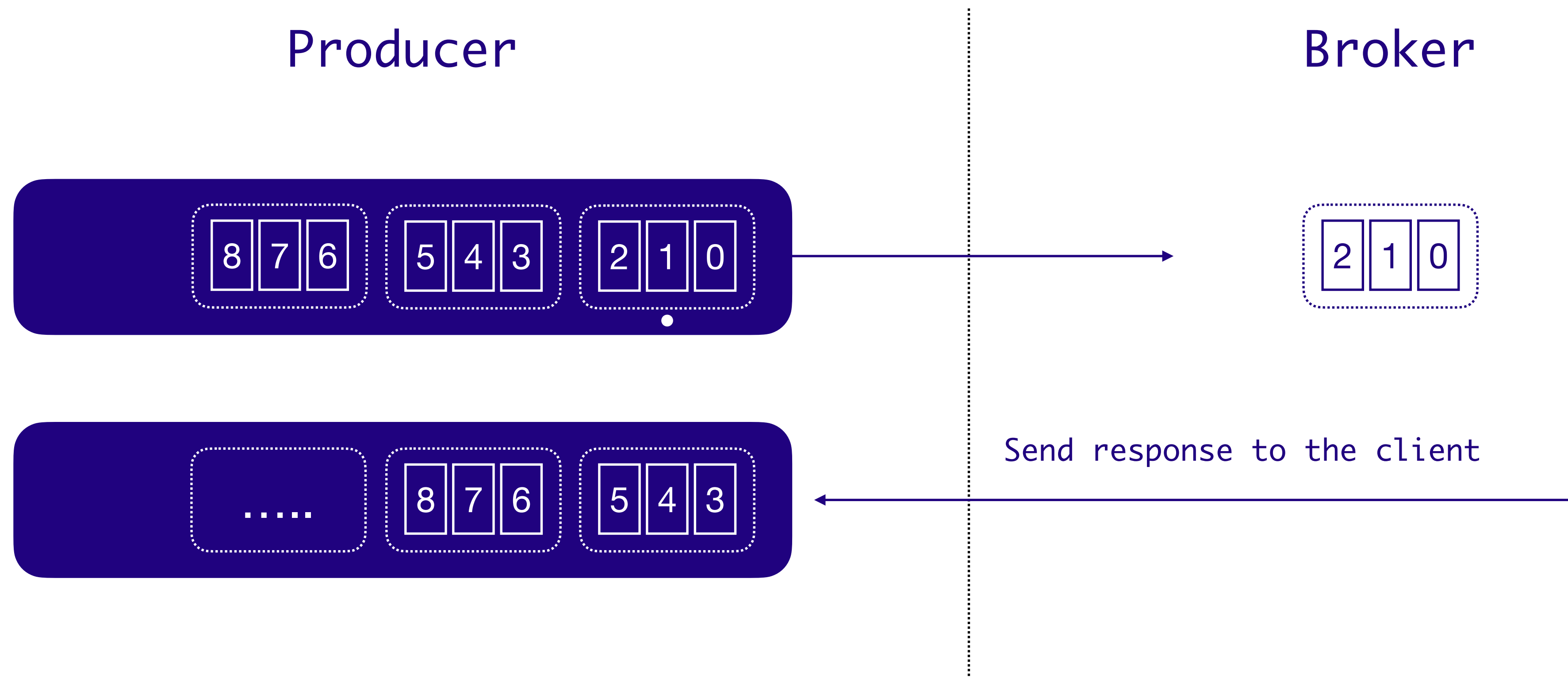
Publish batch message

```java
Producer<byte[]> producer = client.newProducer()
        .topic("my-topic")
        .enableBatching(true)
        .batchingMaxBytes(128 * 1024 * 1024)
        .batchingMaxMessages(1000)
        .batchingMaxPublishDelay(2, TimeUnit.MILLISECONDS)
        .blockIfQueueFull(true)
        .roundRobinRouterBatchingPartitionSwitchFrequency(10)
        .batcherBuilder(BatcherBuilder.DEFAULT)
        .create();

producer.newMessage().value(new byte[0]).send();
producer.newMessage().value(new byte[0]).sendAsync();
producer.send(new byte[0]);
producer.sendAsync(new byte[0]);
```

Message compression

```java
Producer<byte[]> producer = client.newProducer()
        .topic("my-topic")
        .compressionType(CompressionType.LZ4)
        .create();
```
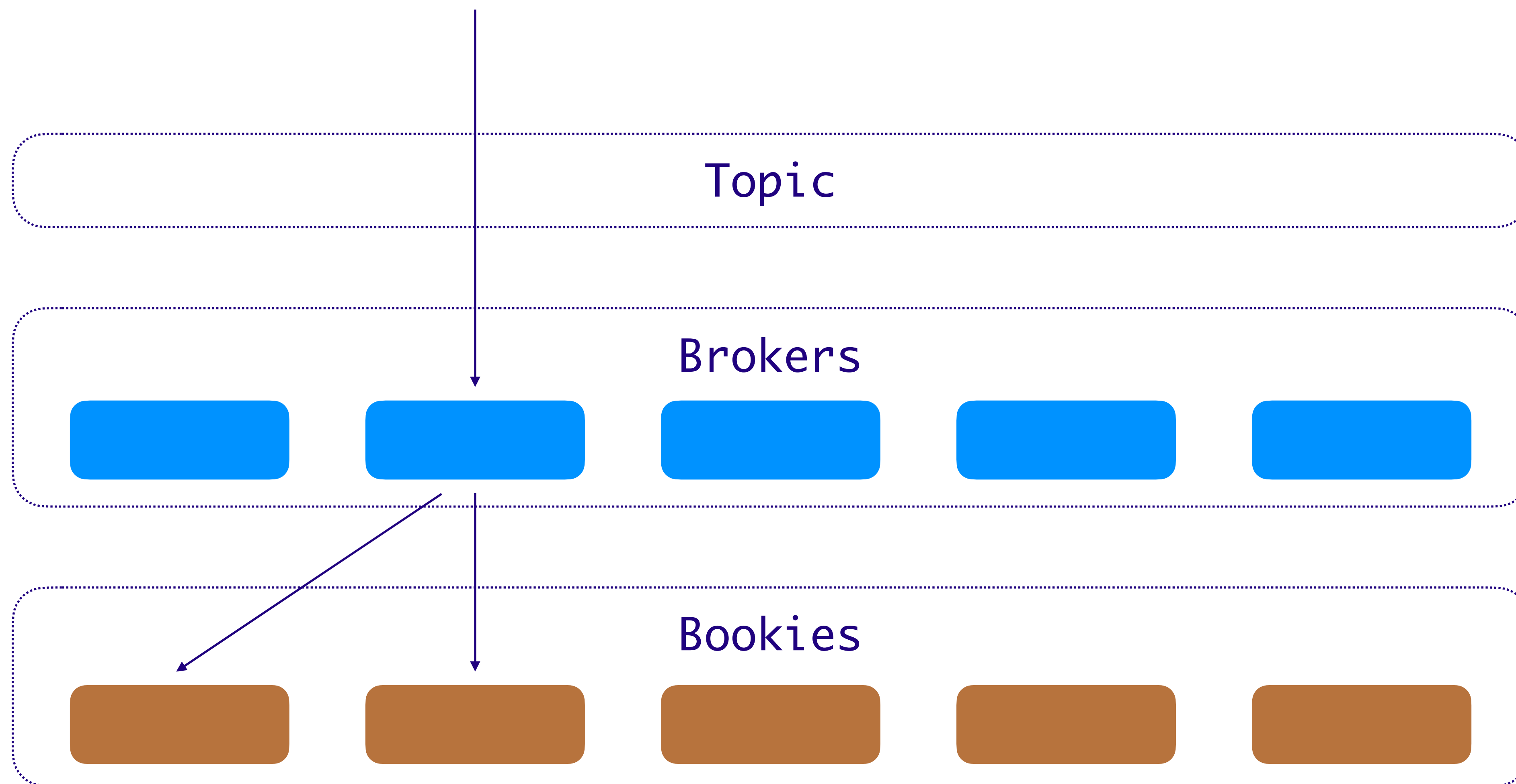
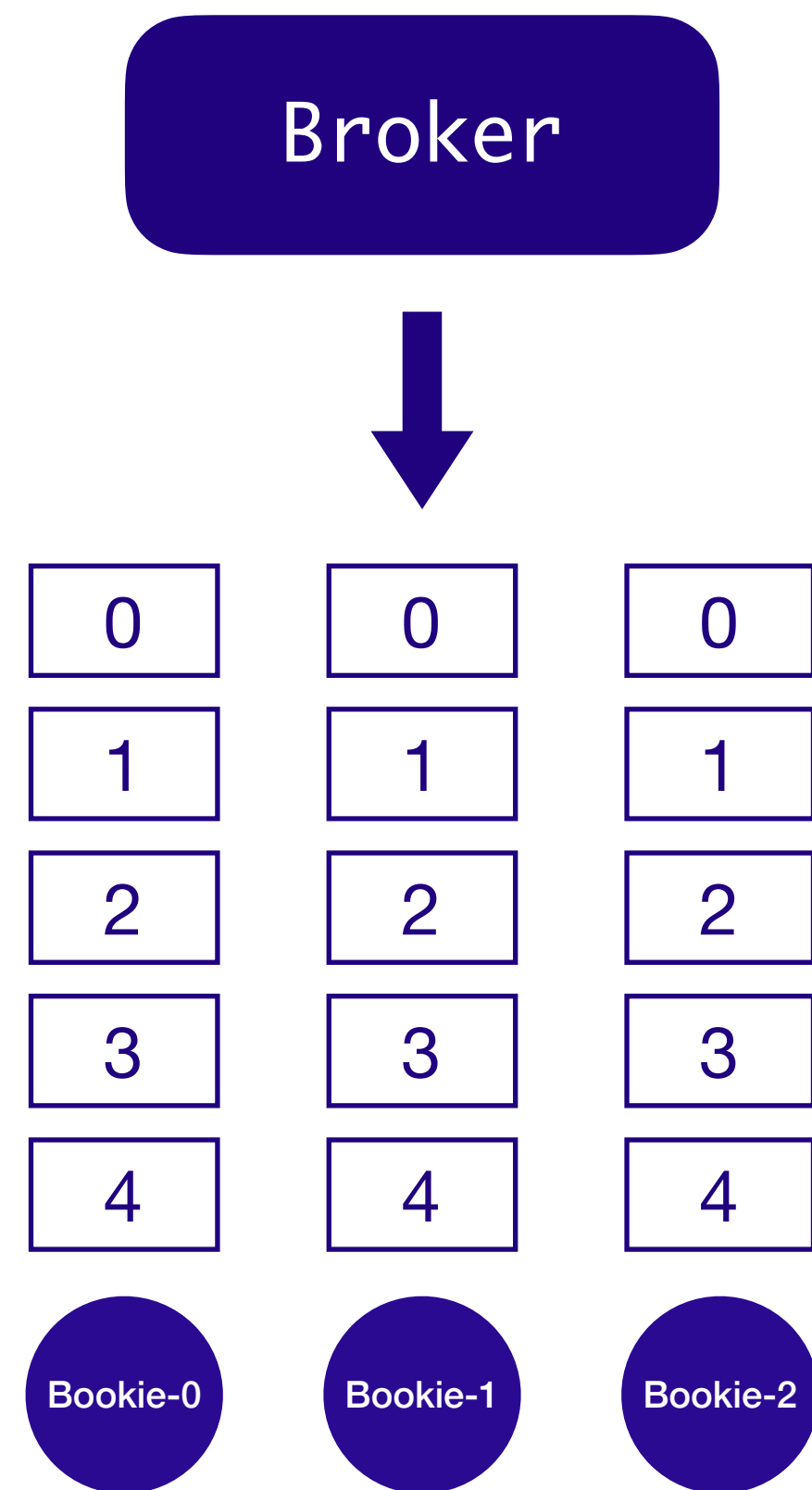| LZ4	ZLIB	ZSTD	SNAPPY
All versions		>= 2.3.0	>= 2.4.0

Message writing overview

E Qw Qa

3-3-2
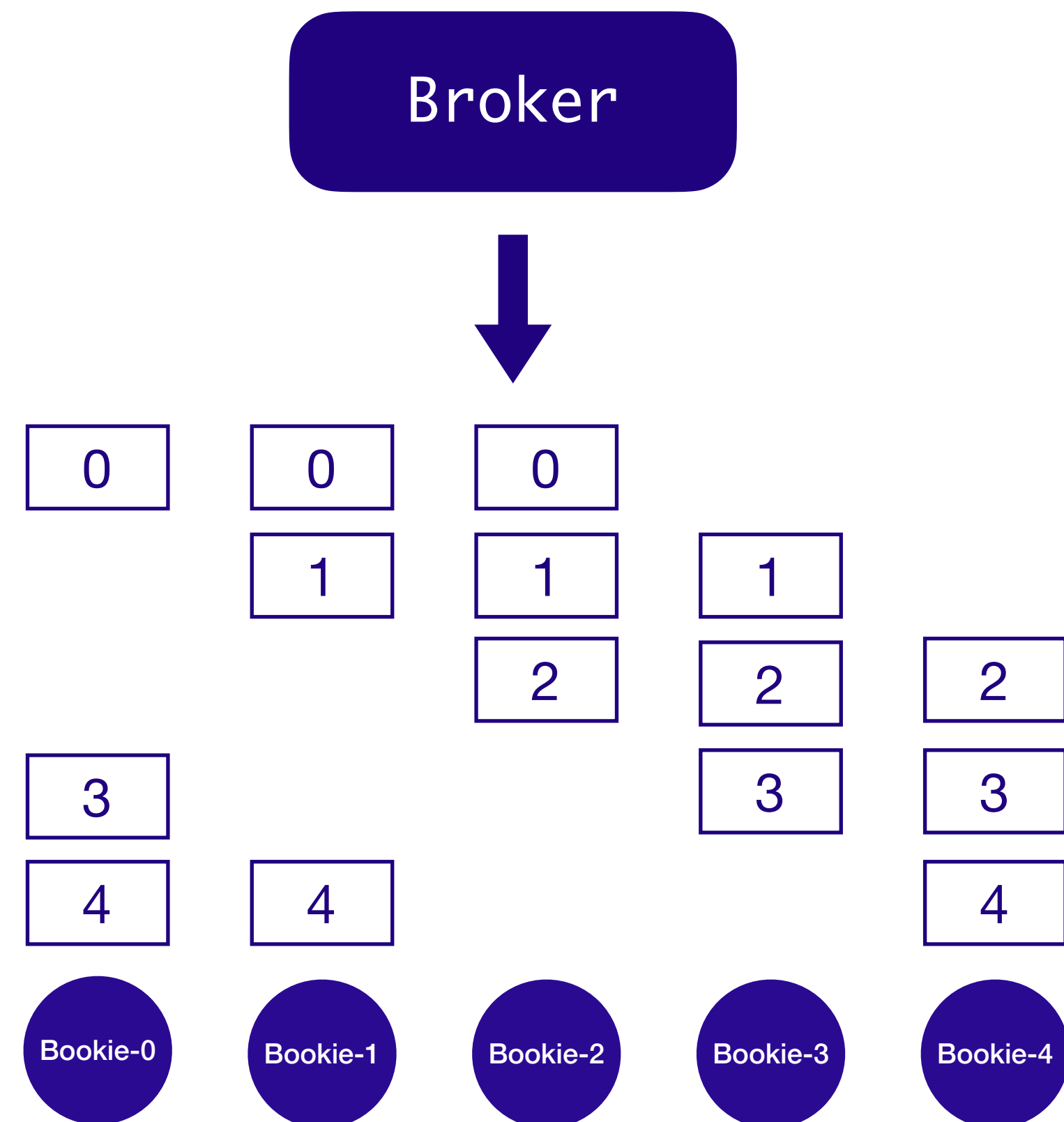
5-3-2

Bookie handle entry add

Request processor

1.Append into Journal

3.Complete add request

Journal

2.Add into Ledger Storage

Memory Table

4.Periodically flush

Entry Log

Ledger Storage
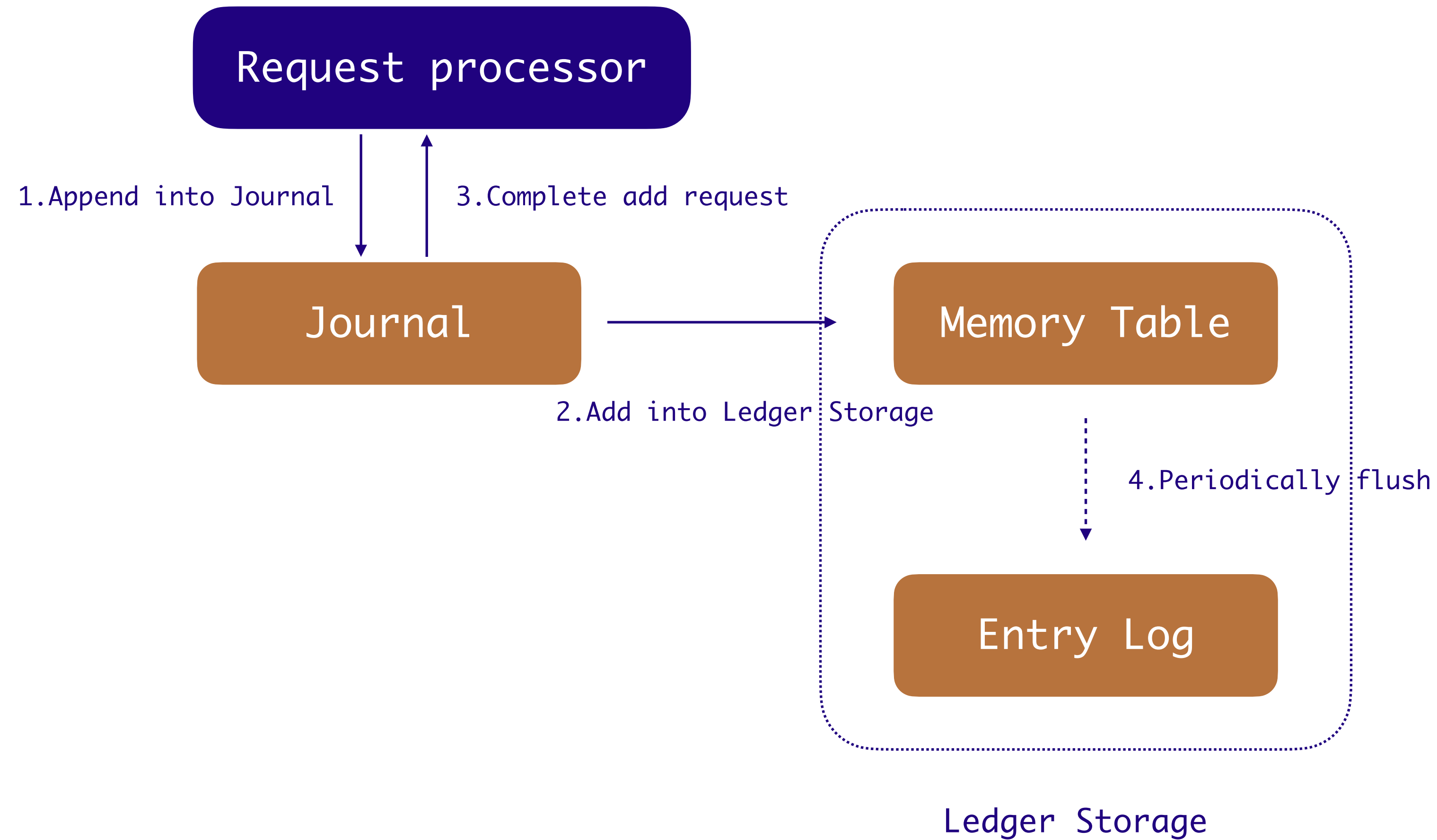
Broker configurations

1. managedLedgerDefaultEnsembleSize

2. managedLedgerDefaultWriteQuorum

3. managedLedgerDefaultAckQuorum

4. managedLedgerNumWorkerThreads

5. numIOThreads

6. -Dorg.apache.bookkeeper.conf.readsystemproperties=true -DnumIOThreads=8
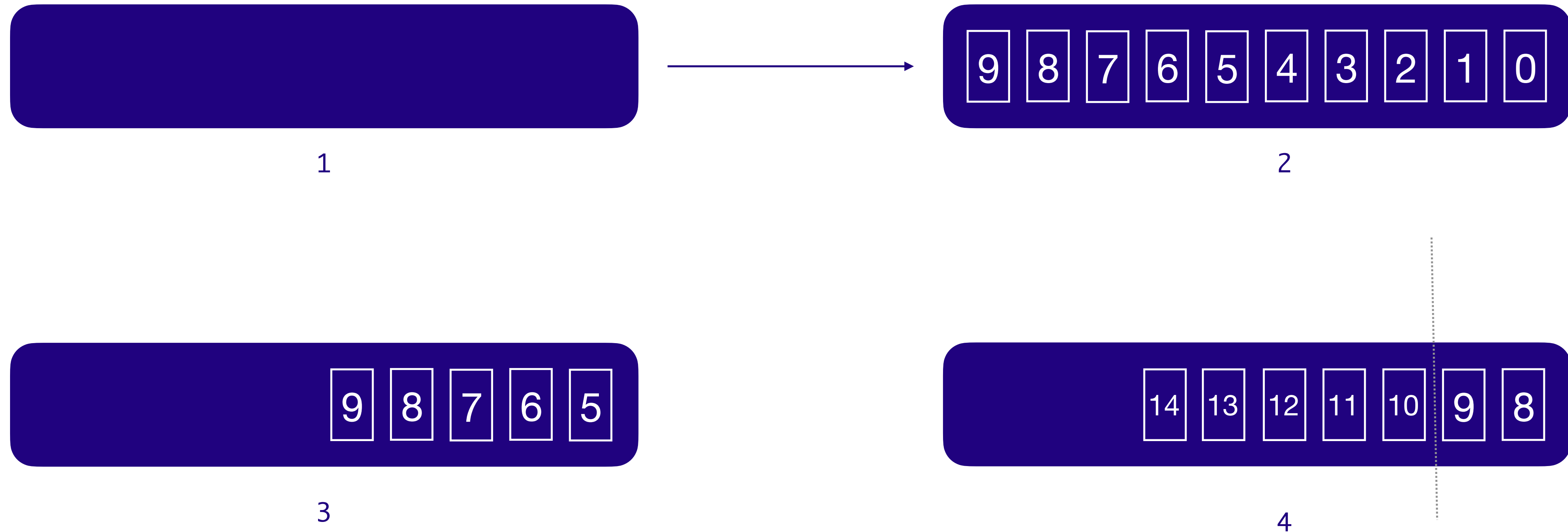
Bookie configurations

1. Journal Directories

2. Ledger Directories

3. Journal sync data

4. Journal group commit

5. Write cache

6. Flush interval

7. Add worker threads and max pending add requests
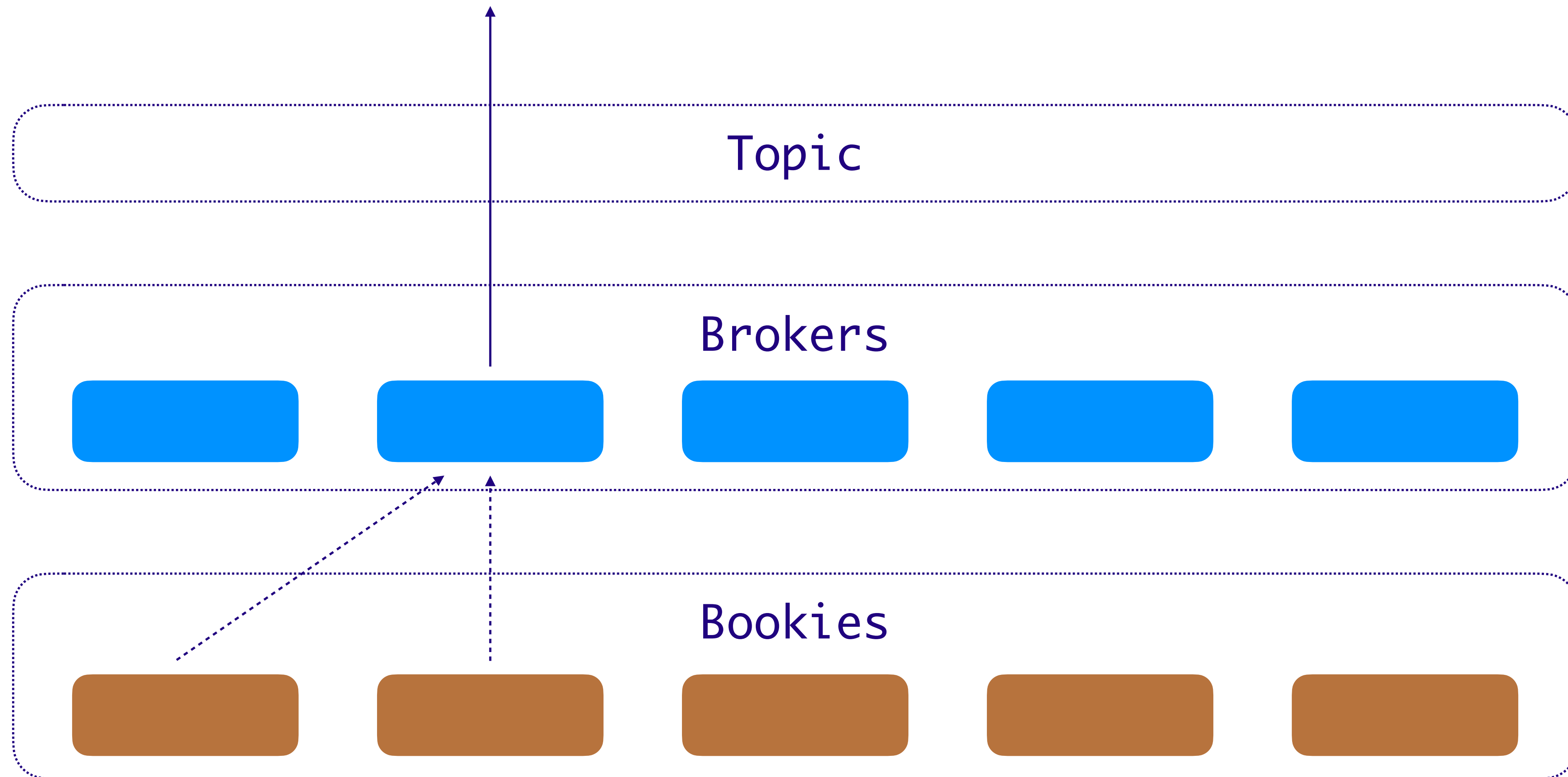
8. Journal pagecache flush interval
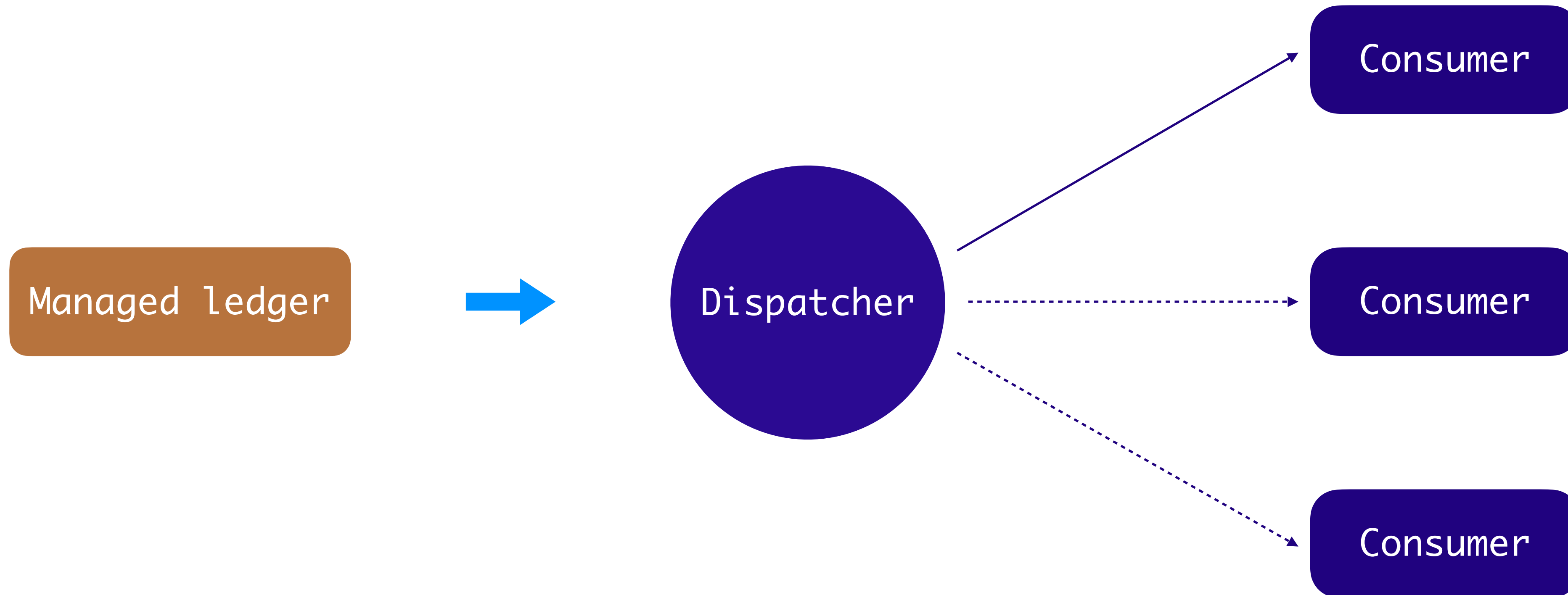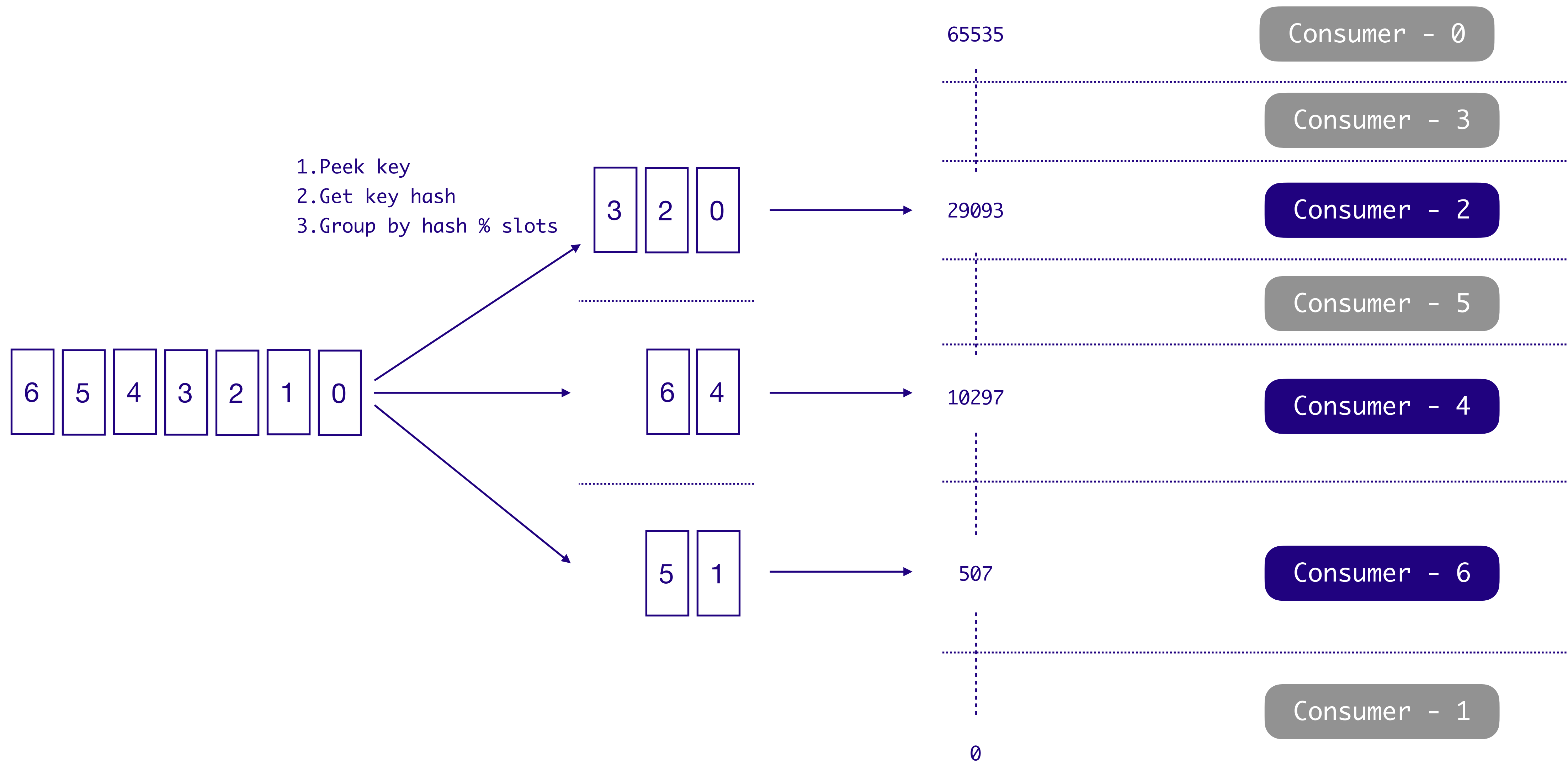
Message reading
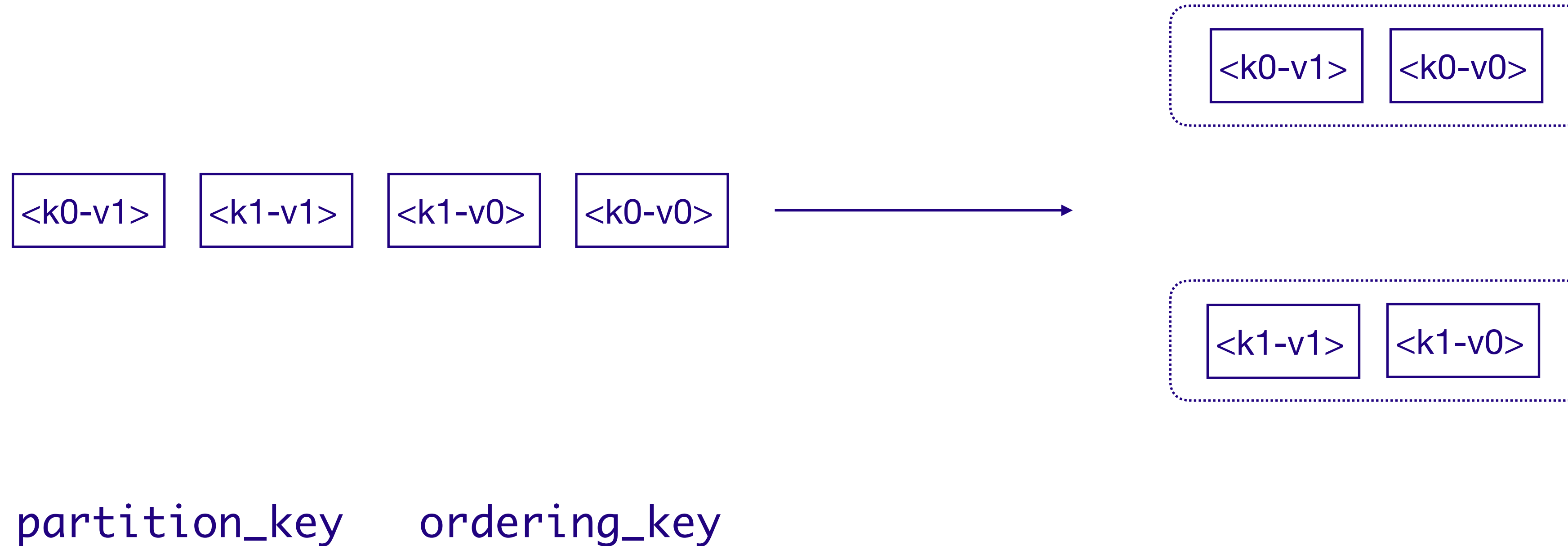
Consumer receiver queue

Message reading overview

Message dispatcher

Key_Shared dispatcher

1. Peek key
2. Get key hash
3. Group by hash % slots

| 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| 3 | 2 | 0 | → 29093

| 6 | 4 | → 10297

| 5 | 1 | → 507

65535

Consumer - 0

Consumer - 3

29093

Consumer - 2

Consumer - 5

10297

Consumer - 4

507

Consumer - 6

Consumer - 1

0

Key based batcher

partition_key   ordering_key

Bookie configurations

1. dbStorage_rocksDB_blockCacheSize

2. dbStorage_readAheadCacheMaxSizeMb

3. dbStorage_readAheadCacheBatchSize

4. Read worker threads

Broker configurations

1. Managed ledger cache

2. Dispatcher max read batch size

3. Bookkeeper sticky reads