

А вы Кафку пробовали?

Apache Kafka – распределённый брокер сообщений

Григорий Кошелев
СКБ Контур

План

1. Зачем нам Apache Kafka
2. Введение в Кафку
3. Архитектура
4. Неочевидности
5. Выводы

Зачем нам Apache Kafka

Зачем нам Apache Kafka - Vostok Hercules

<https://github.com/vostok>

Зачем нам Apache

Kafka

- Vostok

Hercules

-

Логи

<https://github.com/vostok>

Зачем нам Apache

Kafka

- Vostok

Hercules

- Логи

- Метрики

<https://github.com/vostok>

Зачем нам Apache

Kafka

- Vostok

Hercules

- Логи
- Метрики
- Трассировки

<https://github.com/vostok>

Зачем нам Apache

Kafka

- Vostok

Hercules

- Логи
- Метрики
- Трассировки
- Бизнес-события

<https://github.com/vostok>

Зачем нам Apache Kafka

- Vostok Hercules
- Search & Recommendation Systems (SRS)

Зачем нам Apache Kafka

- Vostok Hercules [Kafka 2.0+]
- Search & Recommendation Systems (SRS)

Зачем нам Apache Kafka

- Vostok Hercules [Kafka 2.0+]
- Search & Recommendation Systems (SRS) [Kafka 0.11.x]

Блиц-опрос

Блиц-опрос

Кто использует Apache Kafka?

Блиц-опрос

Кто использует Apache Kafka?

Версия...

< 0.11?

Блиц-опрос

Кто использует Apache Kafka?

Версия...

< 0.11?

0.11.x?

Блиц-опрос

Кто использует Apache Kafka?

Версия...

< 0.11?

0.11.x?

1.x.x?

Блиц-опрос

Кто использует Apache Kafka?

Версия...

< 0.11?

0.11.x?

1.x.x?

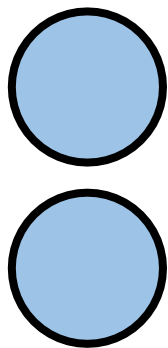
2.x.x?

Введение в Apache Kafka

Введение в Apache Kafka

Kafka Producer

Producer

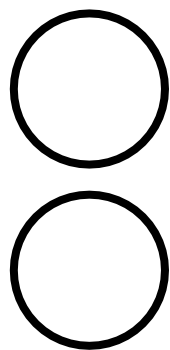


Введение в Apache

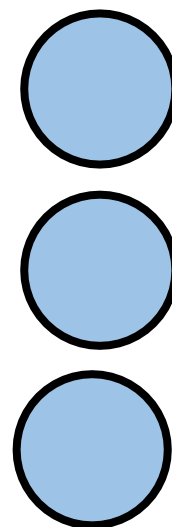
Kafka

Kafka Consumer

Producer

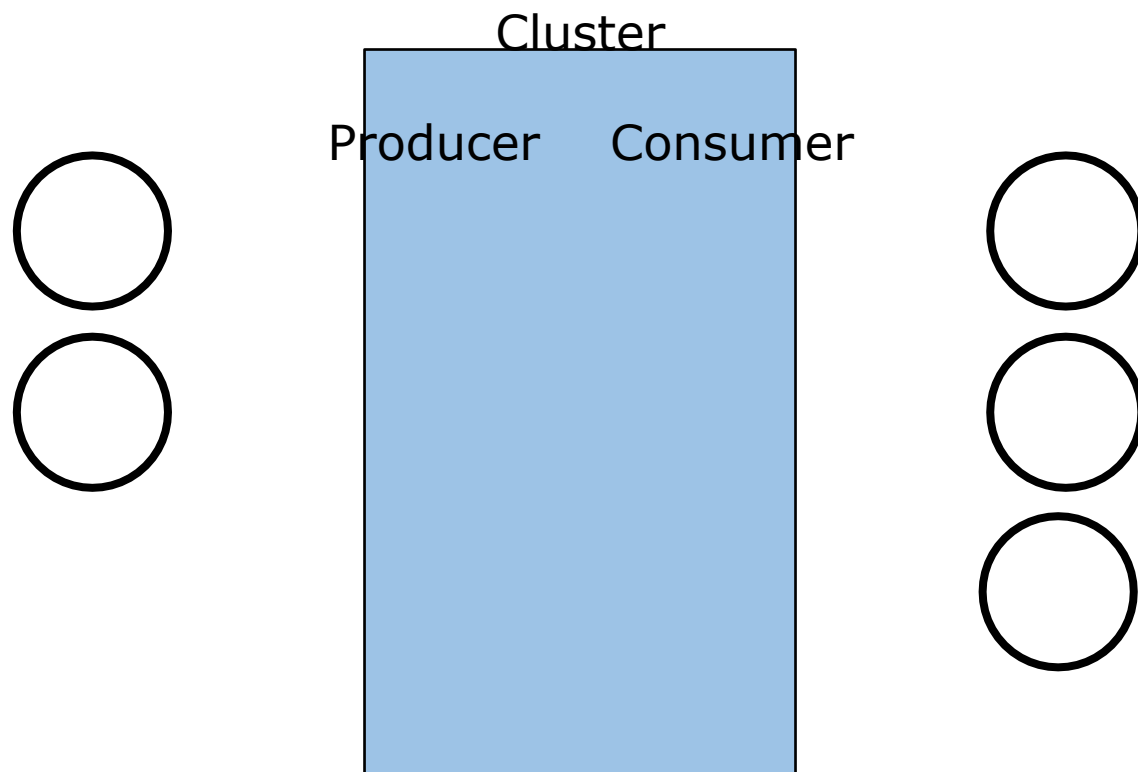


Consumer



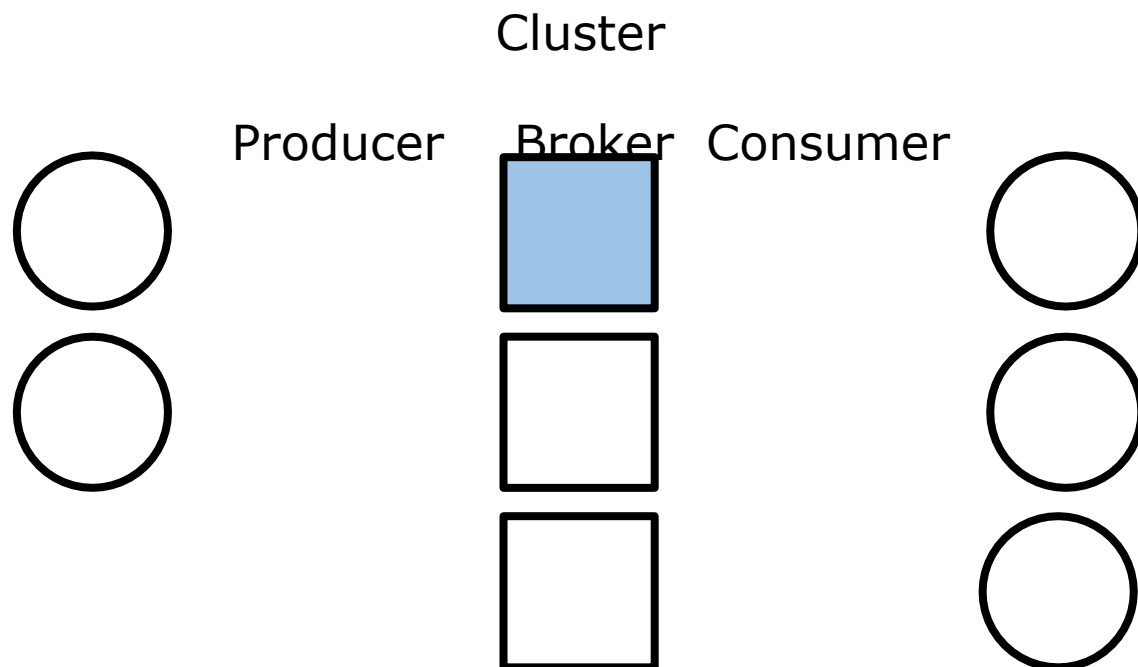
Введение в Apache Kafka

Kafka Cluster

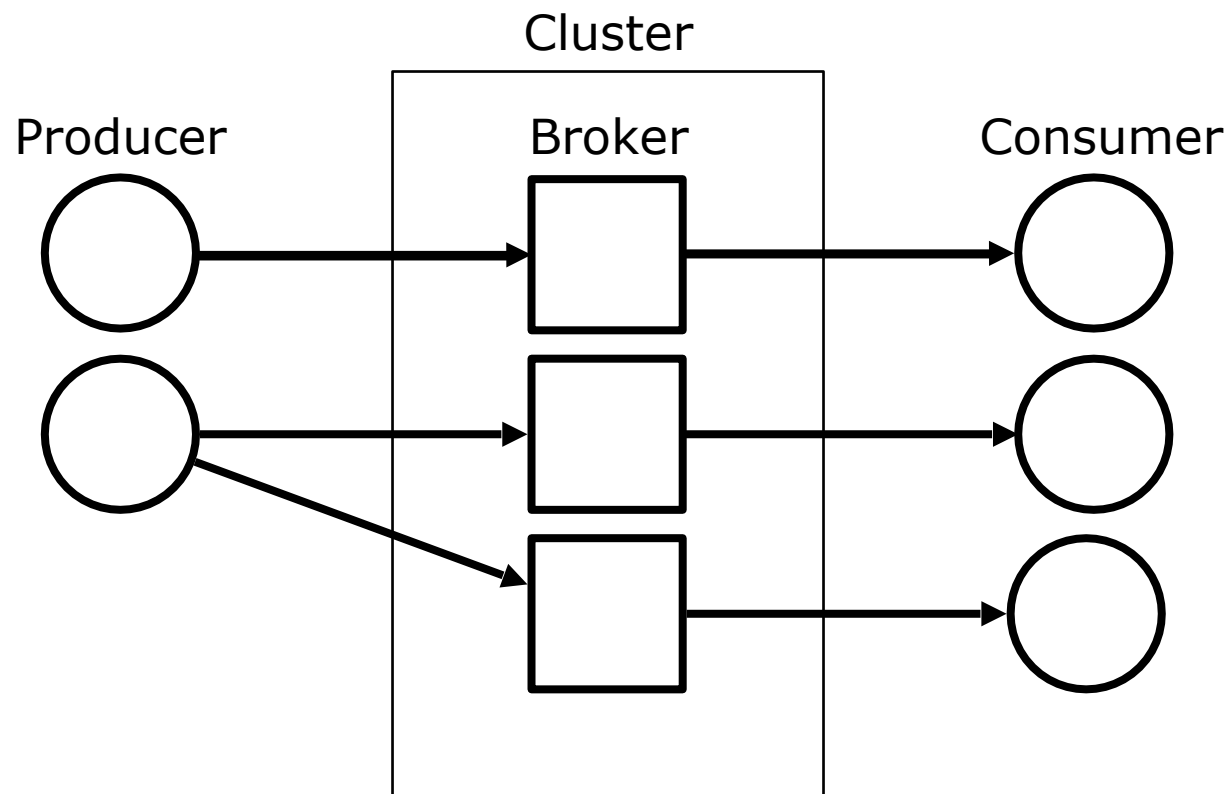


Введение в Apache Kafka

Kafka Broker

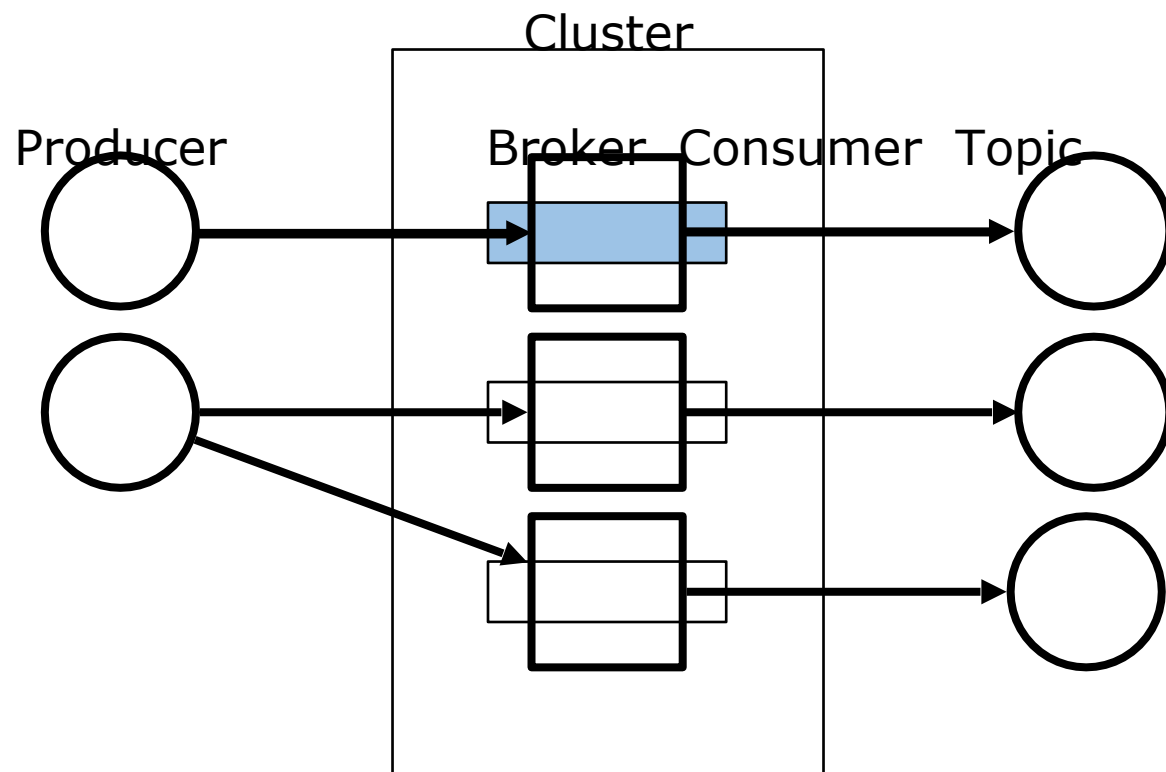


Введение в Apache Kafka



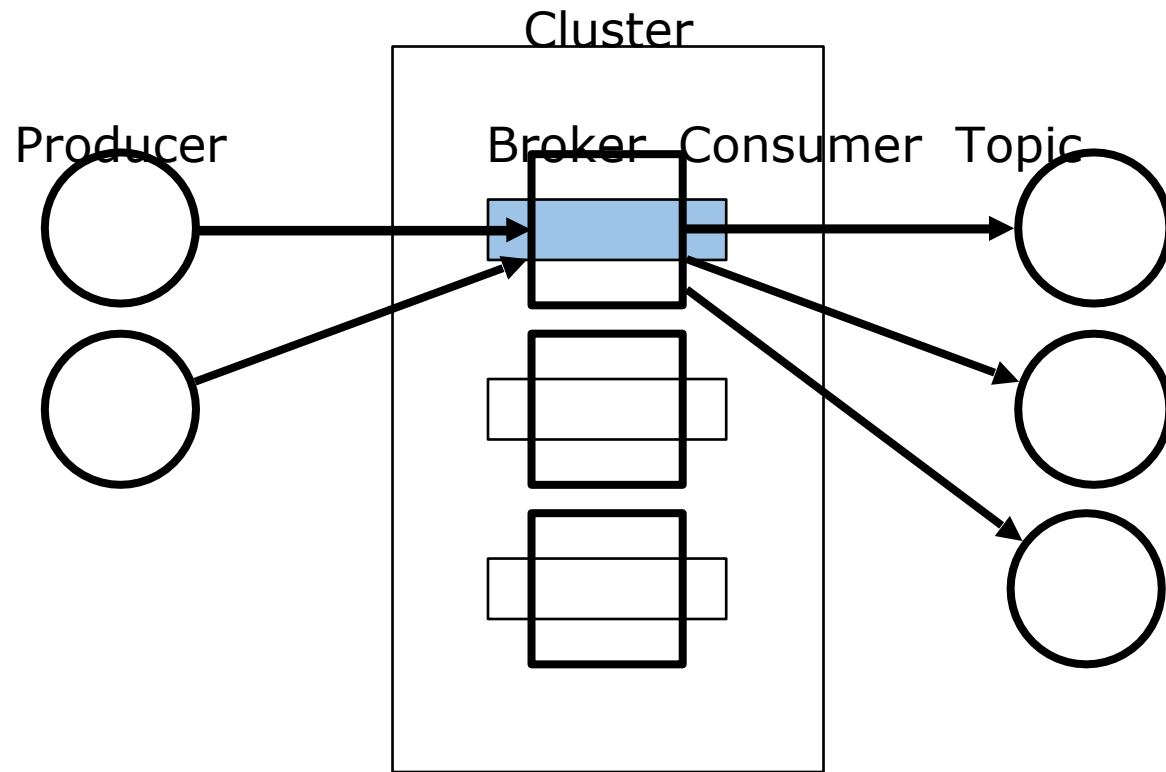
Введение в Apache Kafka

Kafka Topic



Введение в Apache Kafka

Pub-Sub с poll-механикой чтения



Архитектура ApacheKafka

- Topic
- Broker
- Producer
- Consumer

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5
---	---	---	---	---	---

partition 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6
---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

partition 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6
---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

 offset = **0**

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

 offset = **1**

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

 offset = **2**

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

topic = {partition}

partition 0

0	1	2	3	4	5	6
---	---	---	---	---	---	---

offset = **3**

partition 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Архитектура Kafka

Topic

partition = {segment}

Архитектура Kafka

Topic

partition = {segment}

0	1	2	3	4	5	6	7	8	9	n
---	---	---	---	---	---	---	---	---	---	-----	-----	-----	---	-----	-----	-----

Архитектура Kafka

Topic

partition = {segment}

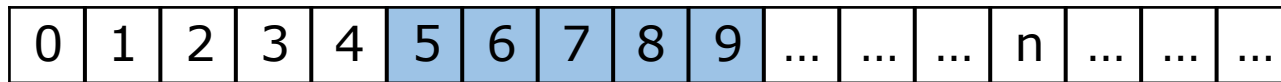
0	1	2	3	4	5	6	7	8	9	n
---	---	---	---	---	---	---	---	---	---	-----	-----	-----	---	-----	-----	-----

segment

Архитектура Kafka

Topic

partition = {segment}

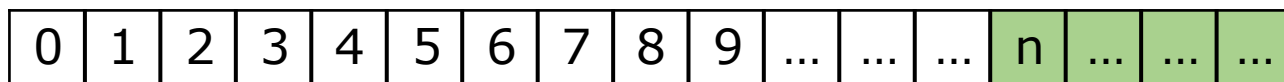


segment

Архитектура Kafka

Topic

partition = {segment}

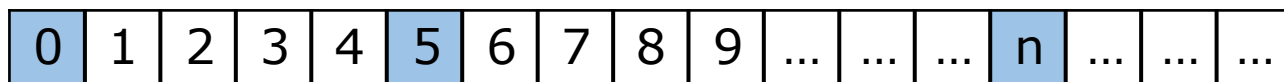


segment

Архитектура Kafka

Topic

partition = {segment}



base offset

Архитектура Kafka

Topic

segment = (base_offset, data, index, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex

Архитектура Kafka

Topic

segment = (**base_offset**, data, index, timeindex)

00000000000**1234567890**.log

00000000000**1234567890**.index

00000000000**1234567890**.timeindex

Архитектура Kafka

Topic

segment = (base_offset, **data**, index, timeindex)

00000000001234567890.log

00000000001234567890.index

00000000001234567890.timeindex

Архитектура Kafka

Topic

segment = (base_offset, **data**, index, timeindex)

00000000001234567890.log

00000000001234567890.index

00000000001234567890.timeindex



log

Архитектура Kafka

Topic

segment = (base_offset, data, **index**, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex



log

Index record = (relative offset, position)

Архитектура Kafka

Topic

segment = (base_offset, data, **index**, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex



Index record = (relative offset, position)

offset = 123456789**0** relative offset =
0

size = 100 position = **0**

Архитектура Kafka

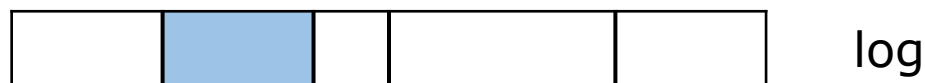
Topic

segment = (base_offset, data, **index**, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex



Index record = (relative offset, position)

offset = 123456789**1** relative offset =
1

size = 100 position = **100**

Архитектура Kafka

Topic

segment = (base_offset, data, **index**, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex



Index record = (relative offset, position)

offset = 123456789**2** relative offset =
2

size = 50 position = **200**

Архитектура Kafka

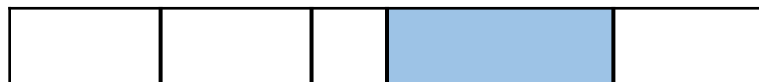
Topic

segment = (base_offset, data, **index**, timeindex)

000000000001234567890.log

000000000001234567890.index

000000000001234567890.timeindex



log

Index record = (relative offset, position)

offset = 123456789**3** relative offset =
3

size = 150 position = **250**

Архитектура Kafka

Topic

segment = (base_offset, data, index, **timeindex**)

000000000001234567890.log

000000000001234567890.index

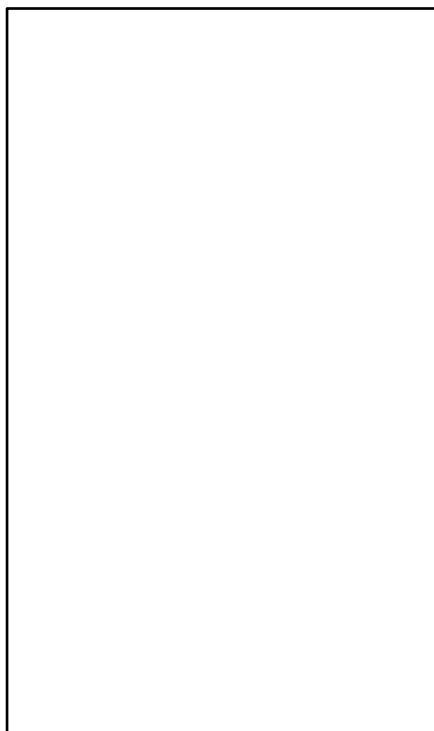
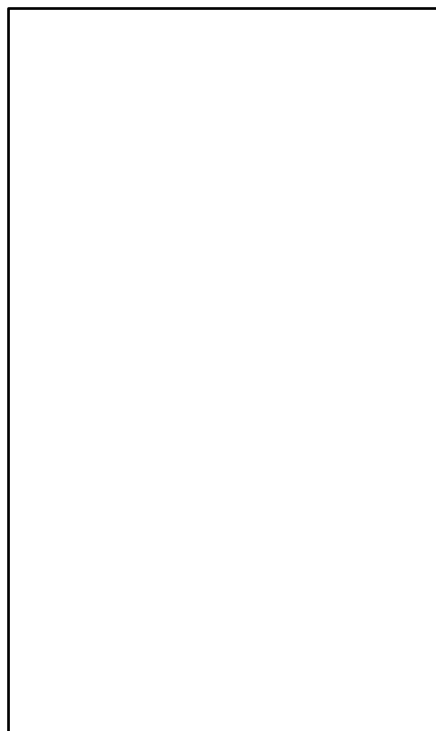
000000000001234567890.timeindex

Архитектура Kafka

Broker

cluster = {broker}

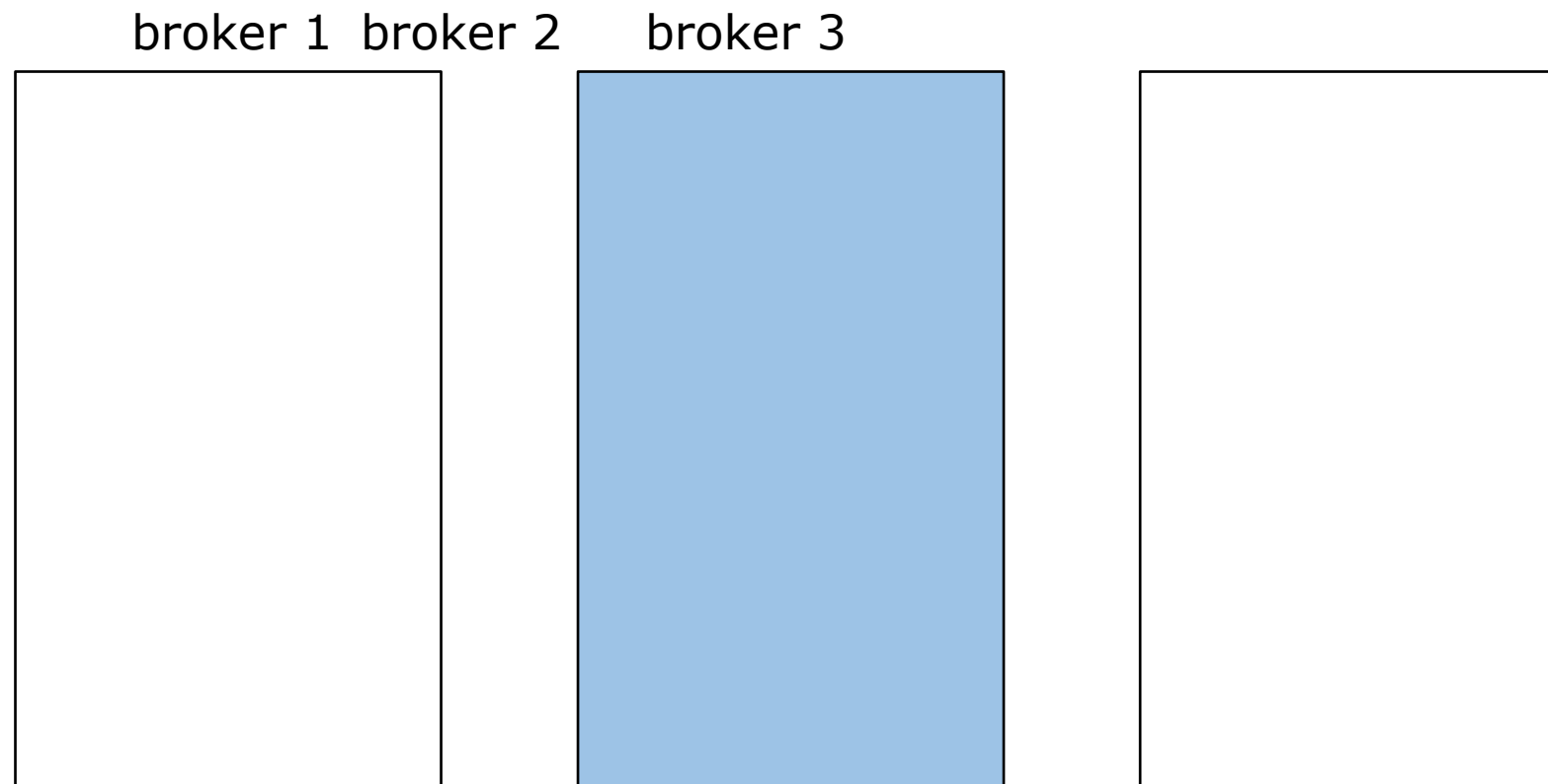
broker 1 broker 2 broker 3



Архитектура Kafka

Broker

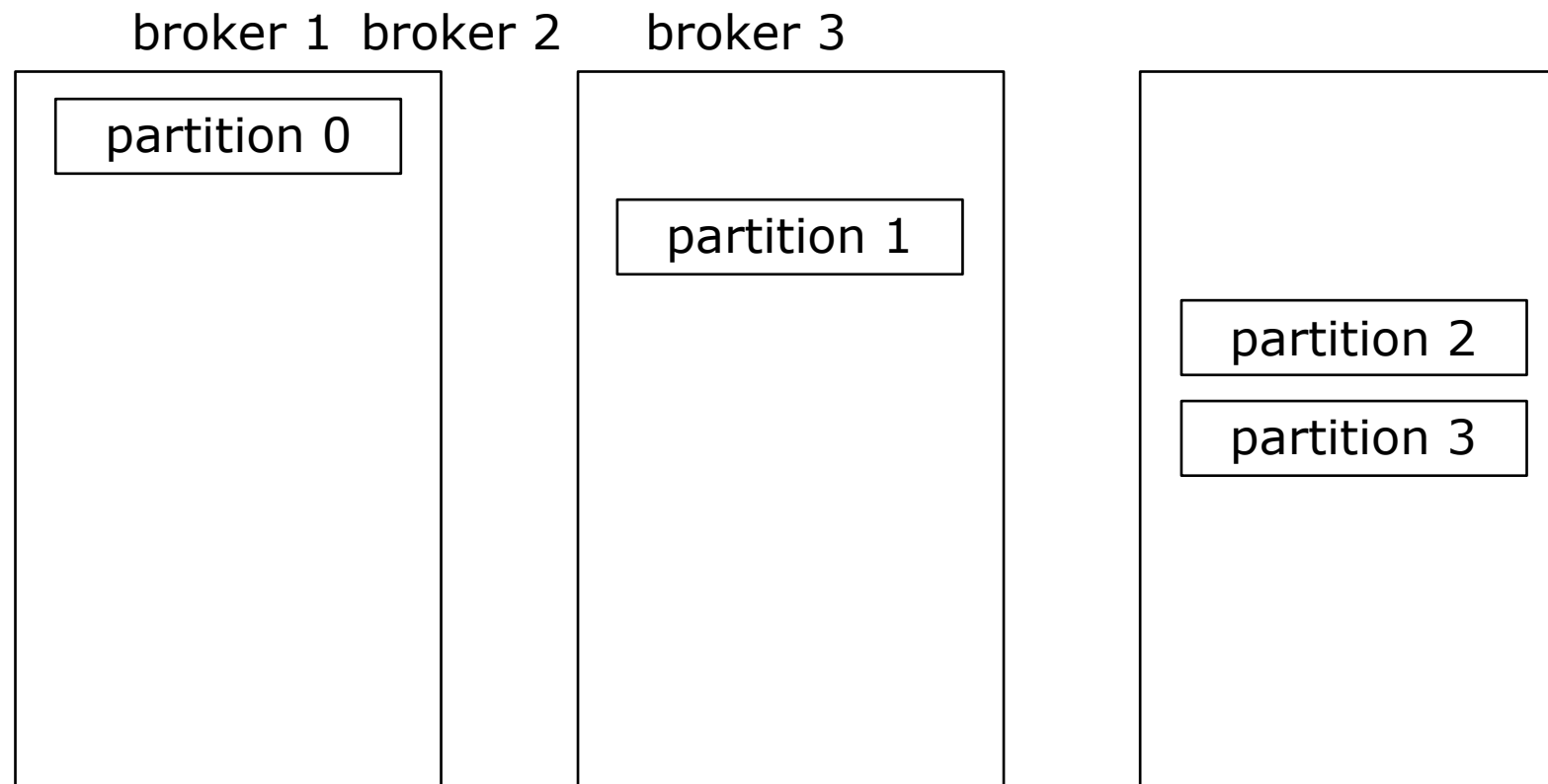
Controller – координирует работу кластера



Архитектура Kafka

Broker

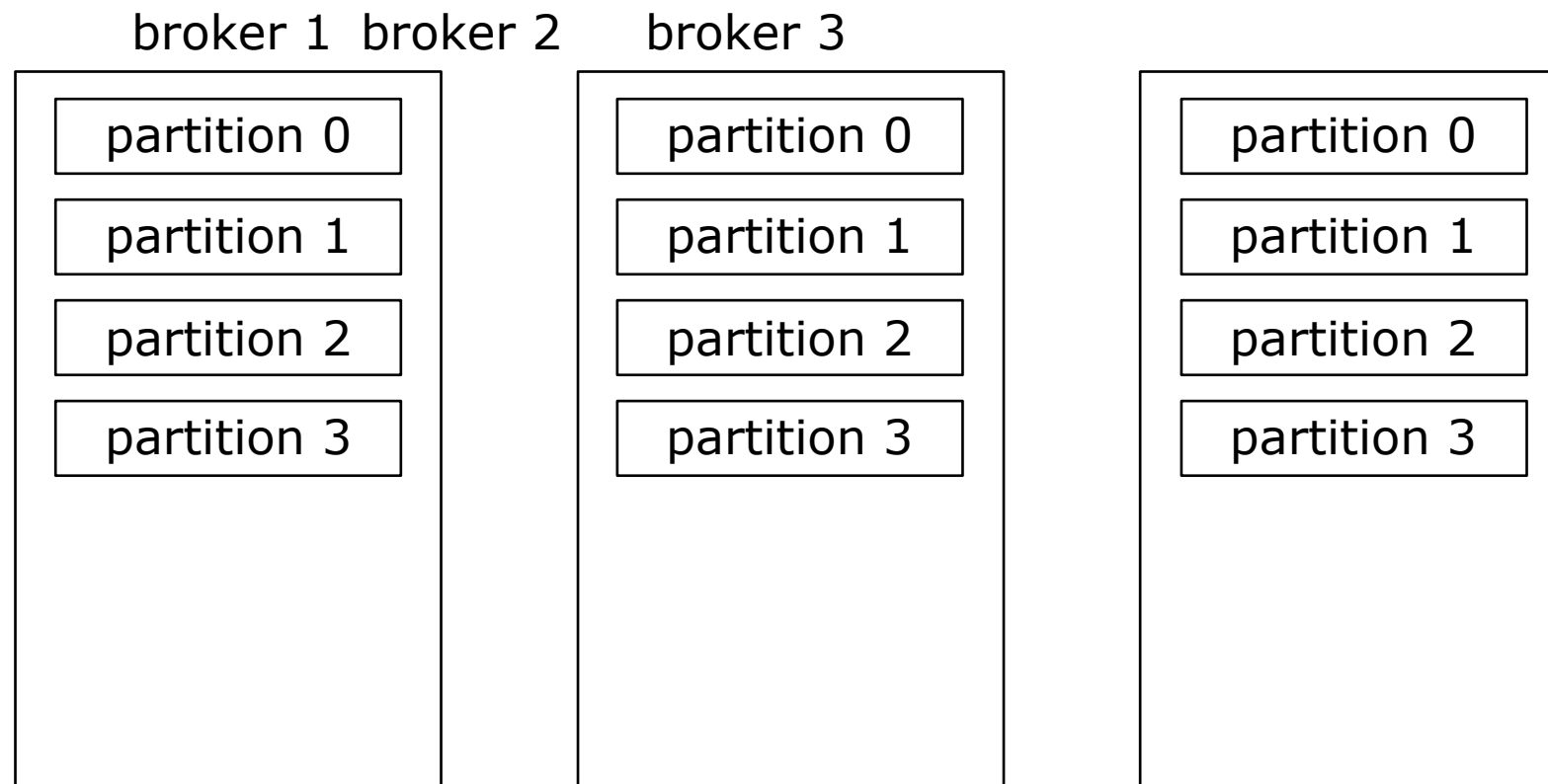
topic = {partition}



Архитектура Kafka

Broker

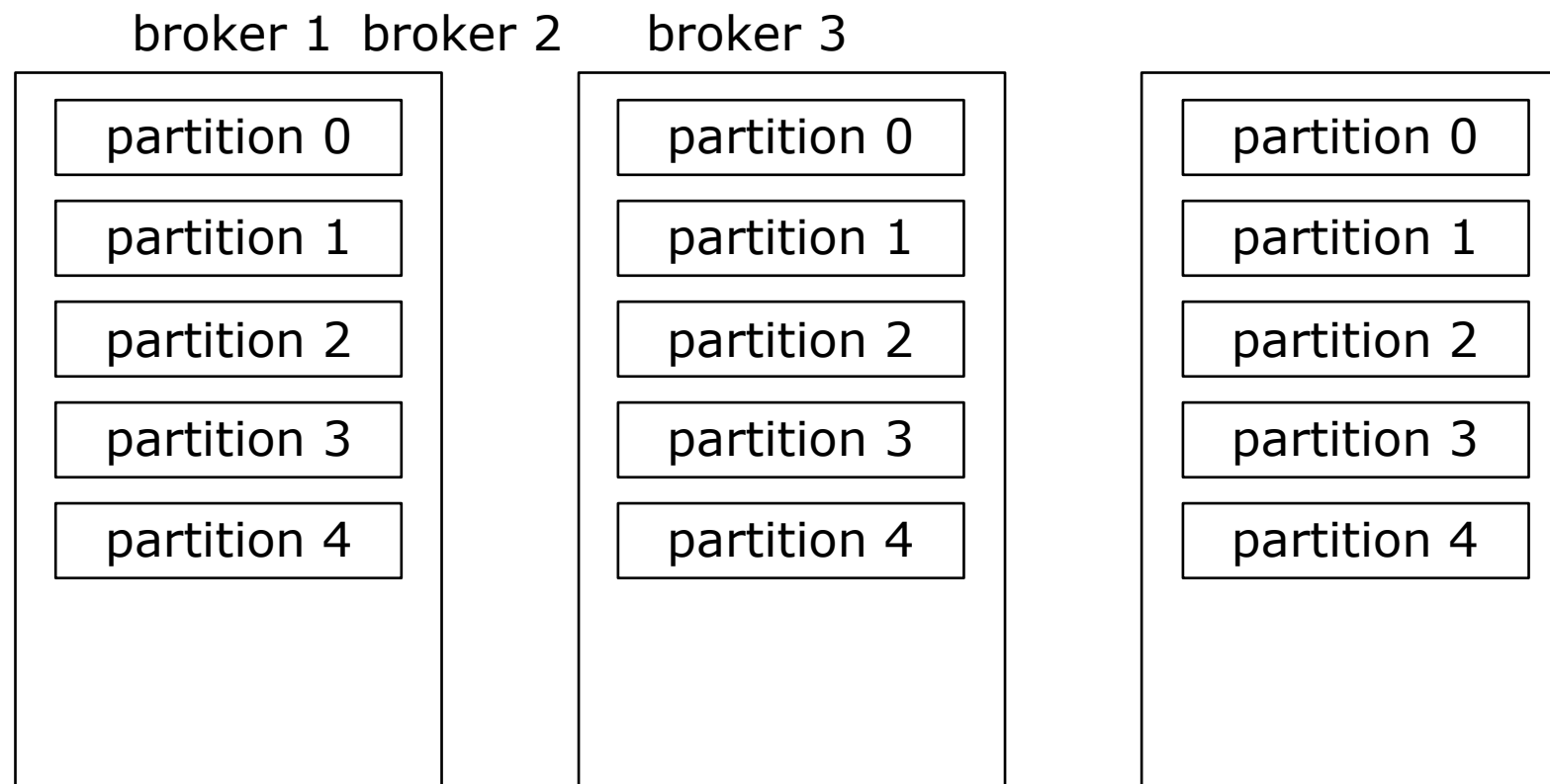
replication factor = 3



Архитектура Kafka

Broker

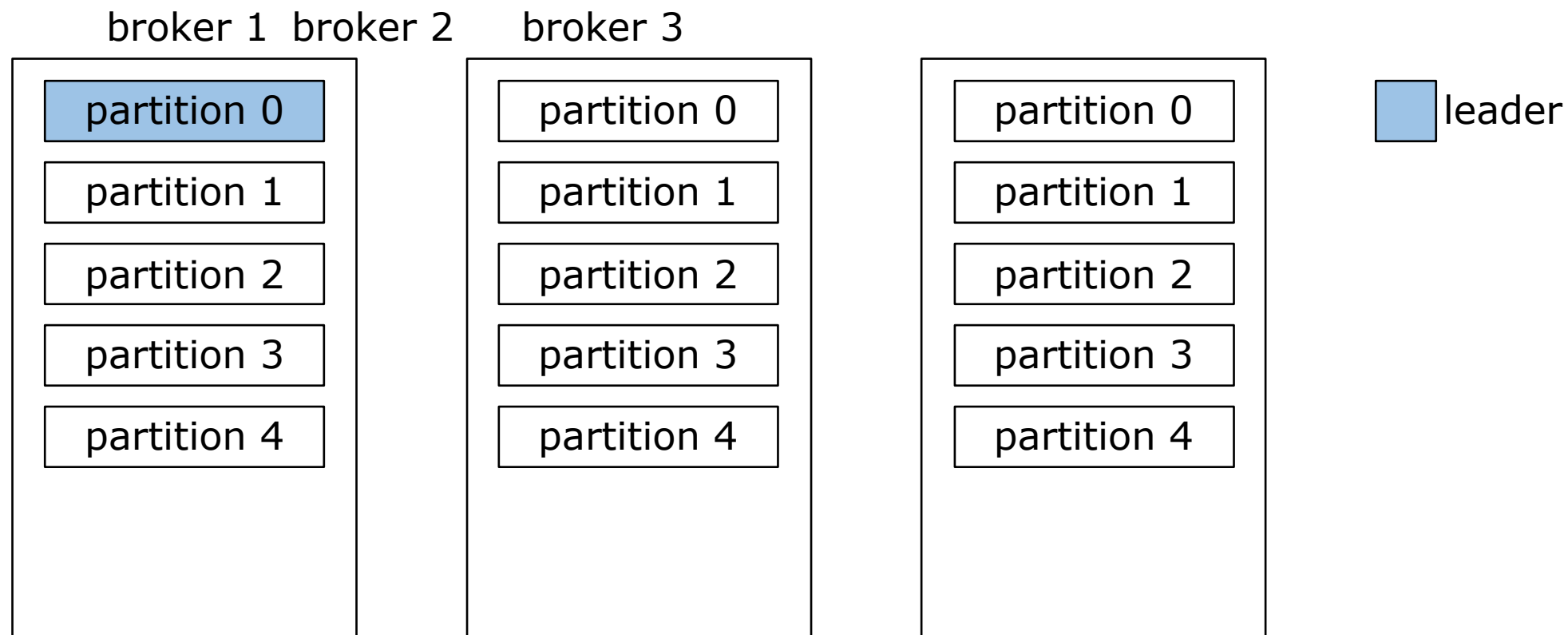
Добавление partition



Архитектура Kafka

Broker

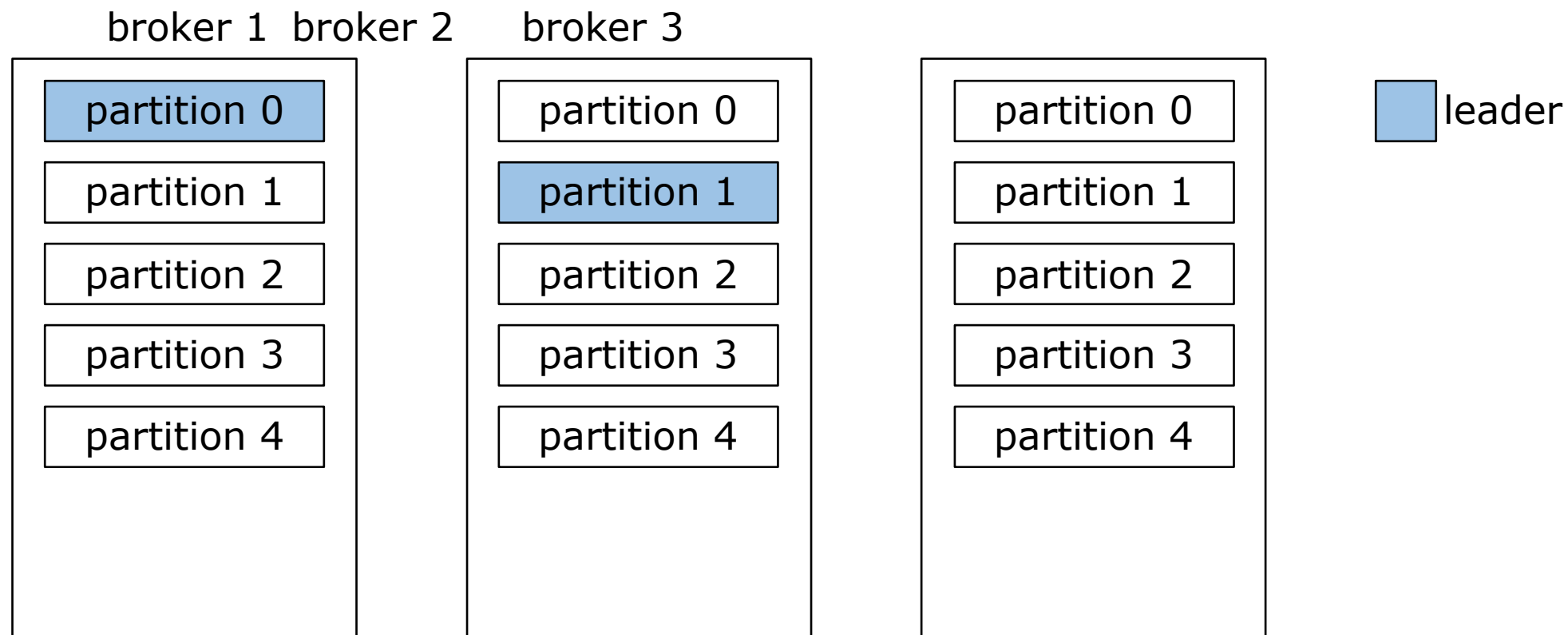
broker 1 – leader для partition 0.



Архитектура Kafka

Broker

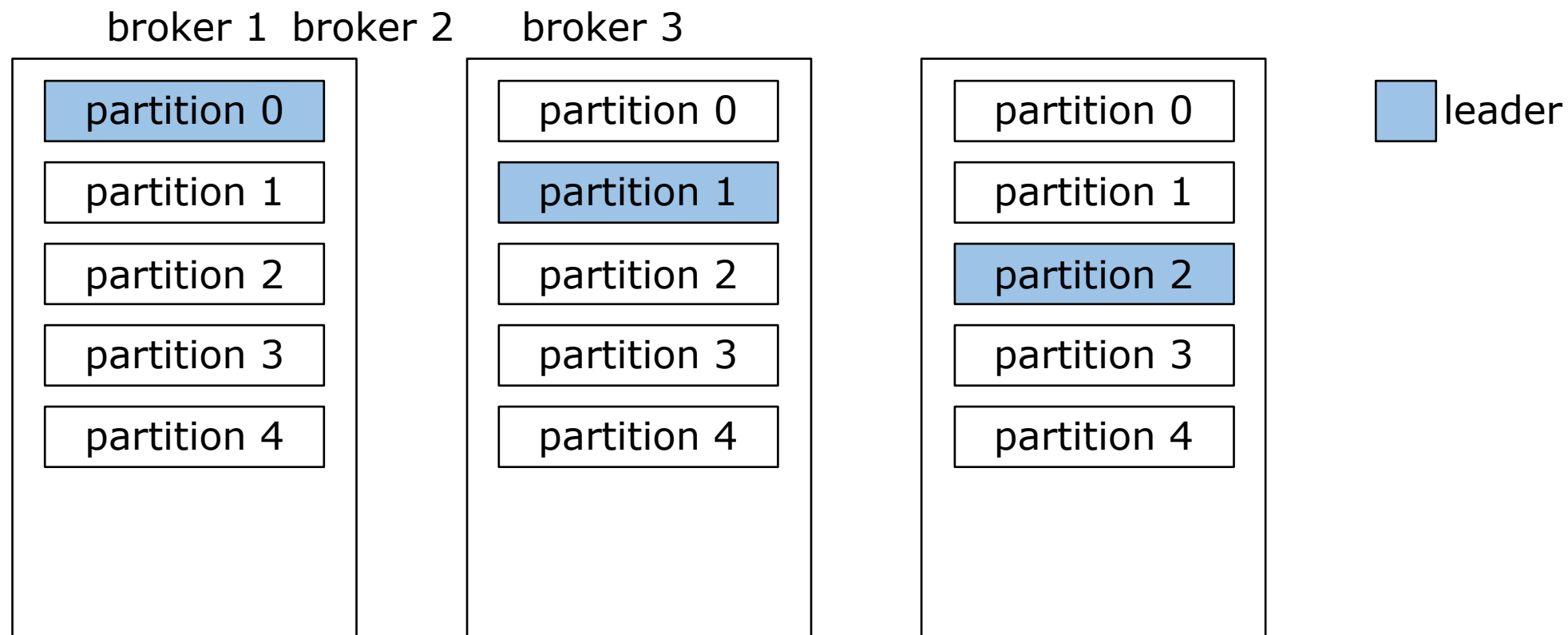
broker 2 – leader для partition 1



Архитектура Kafka

Broker

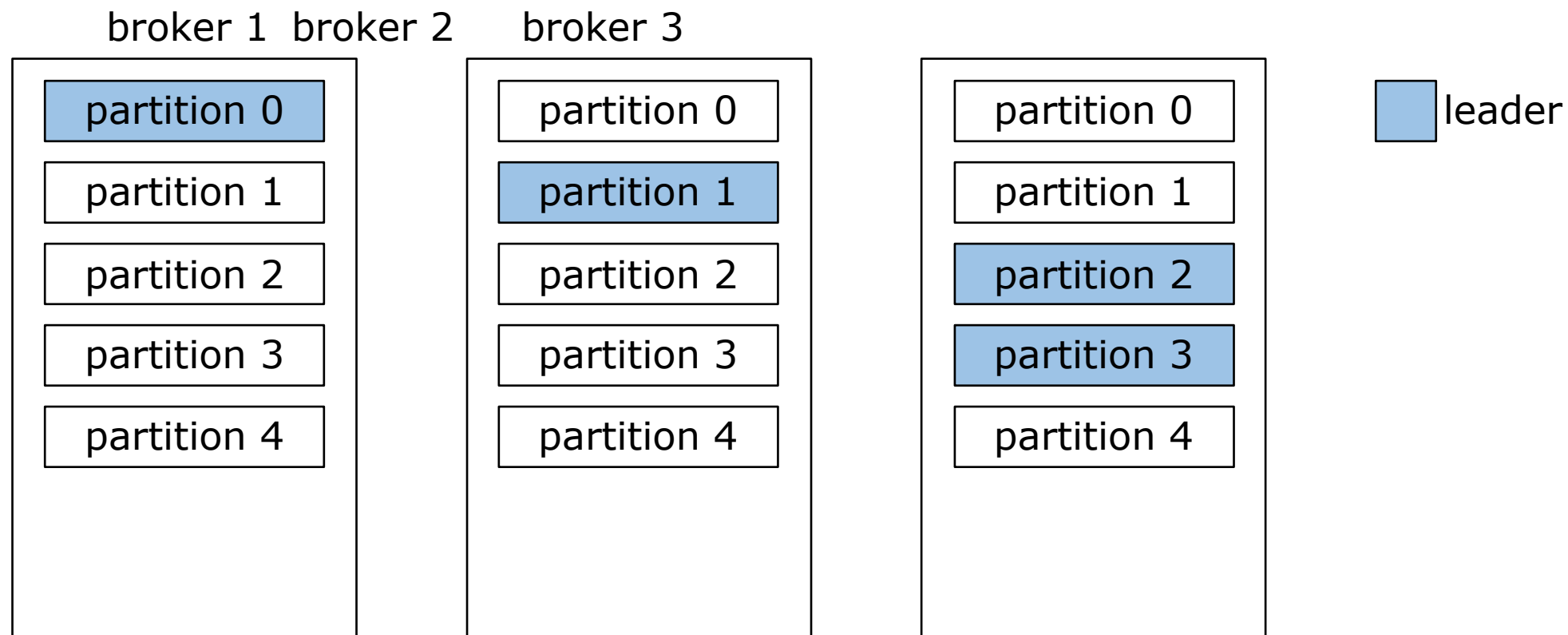
broker 3 – leader для partition 2



Архитектура Kafka

Broker

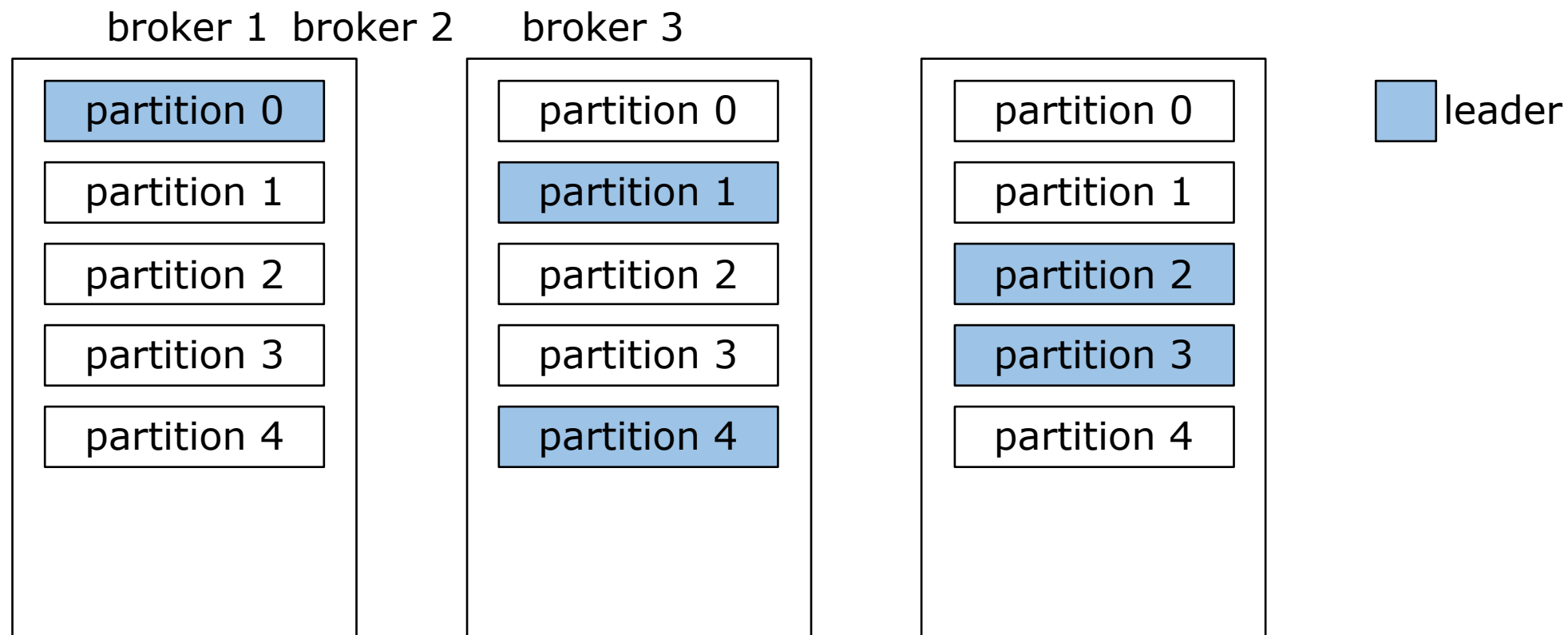
broker 3 – leader для partition 3



Архитектура Kafka

Broker

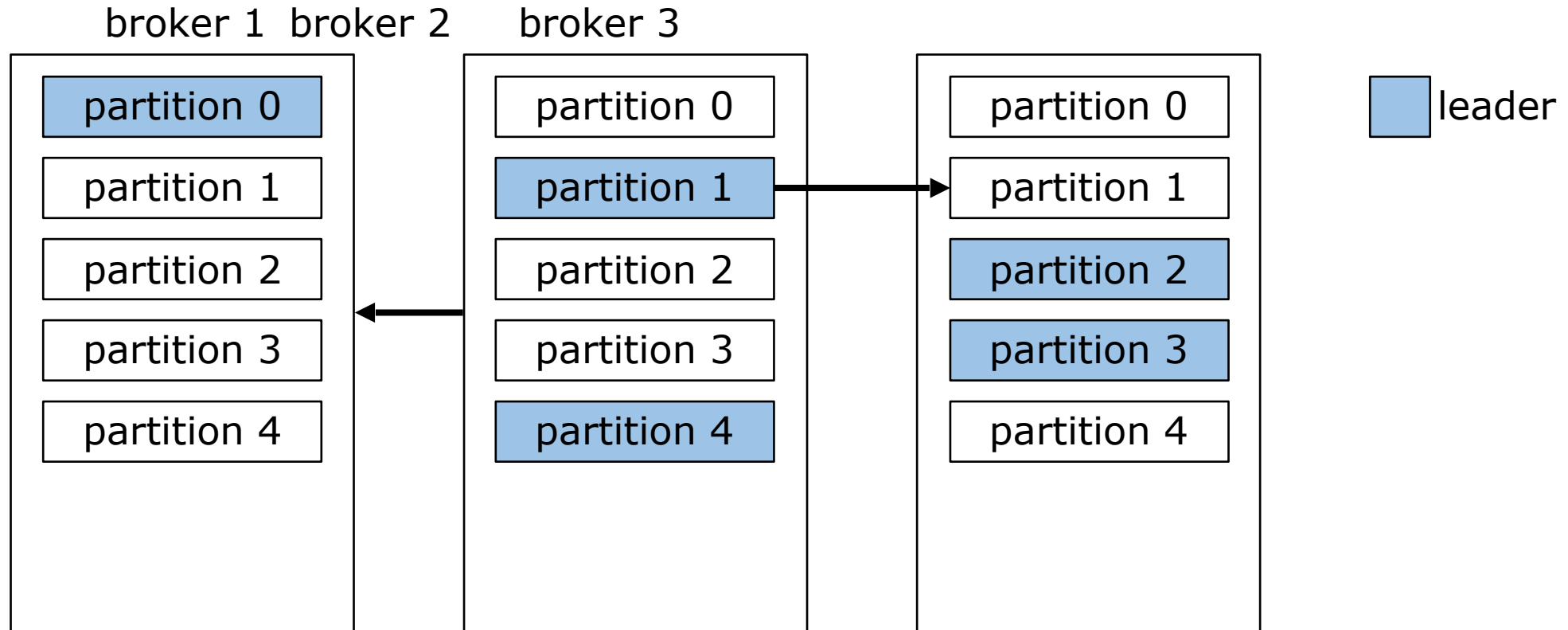
broker 2 – leader для partition 4



Архитектура Kafka

Broker

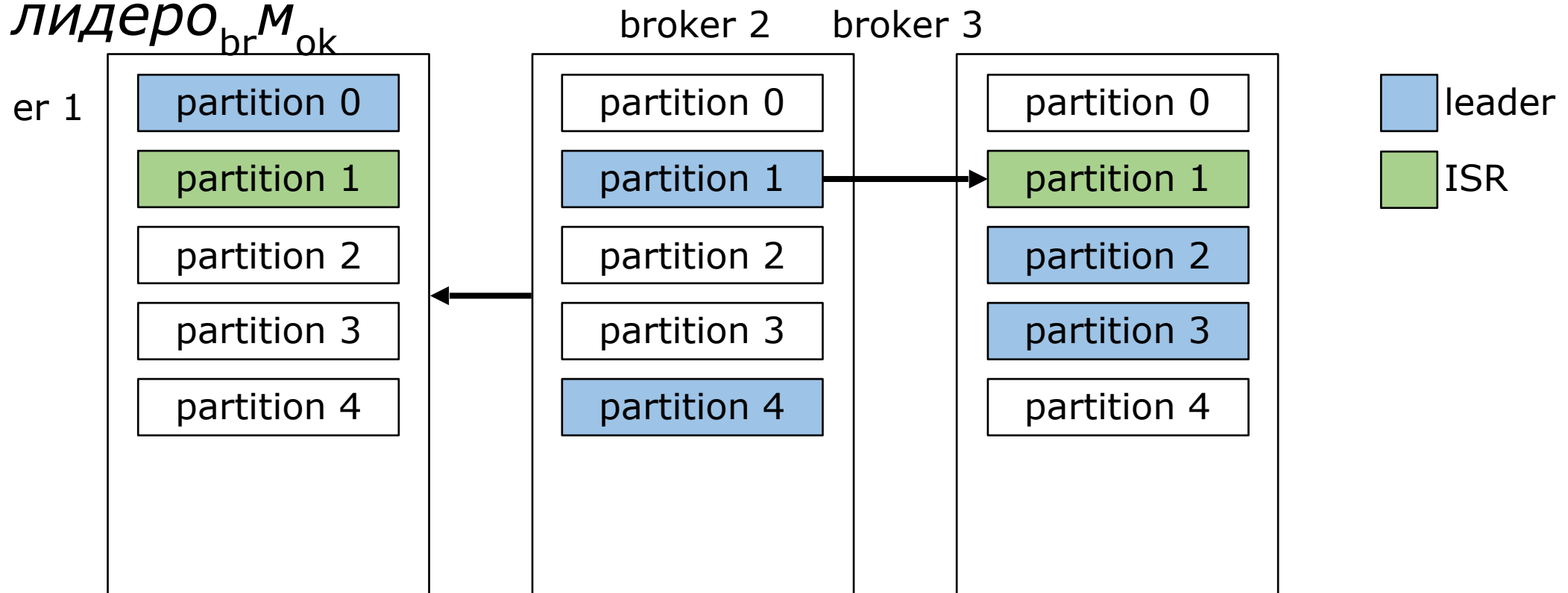
Репликация с *лидера* на другие брокеры



Архитектура Kafka

Broker

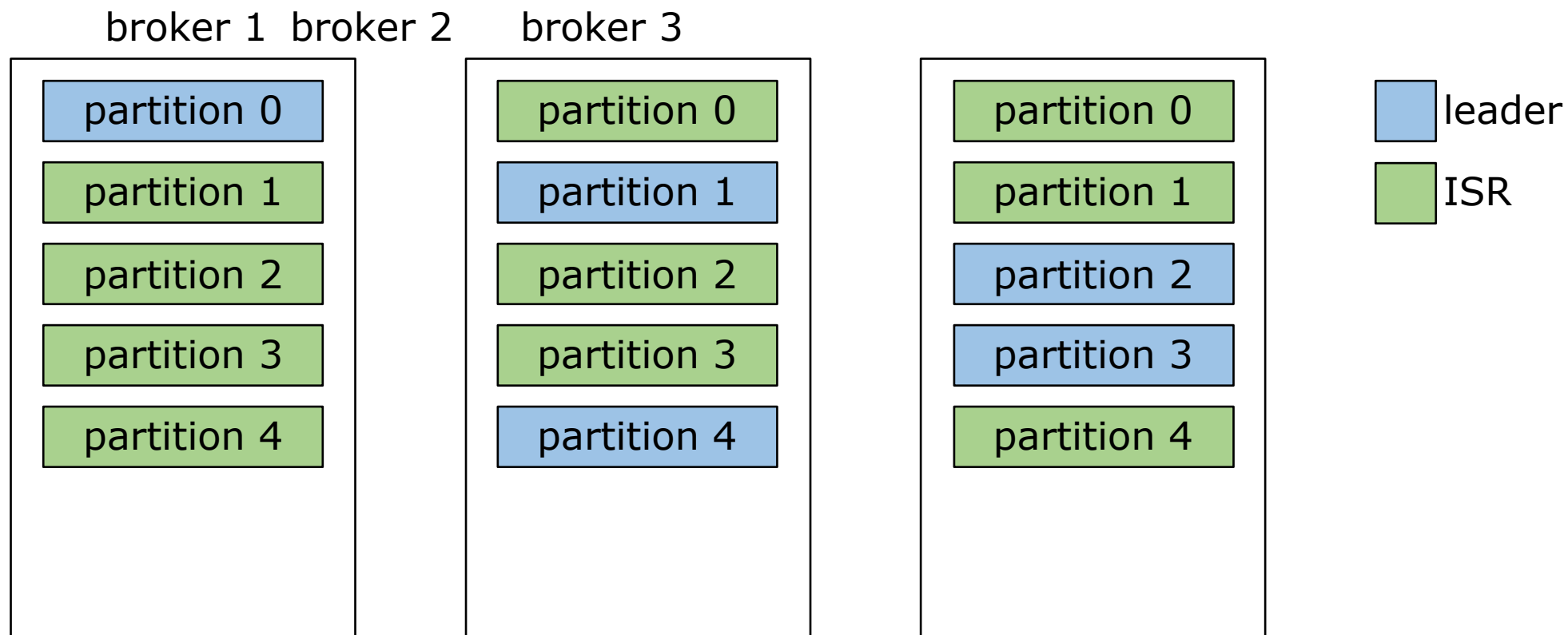
ISR (in sync replica) – *реплика, синхронизированная с лидером*



Архитектура Kafka

Broker

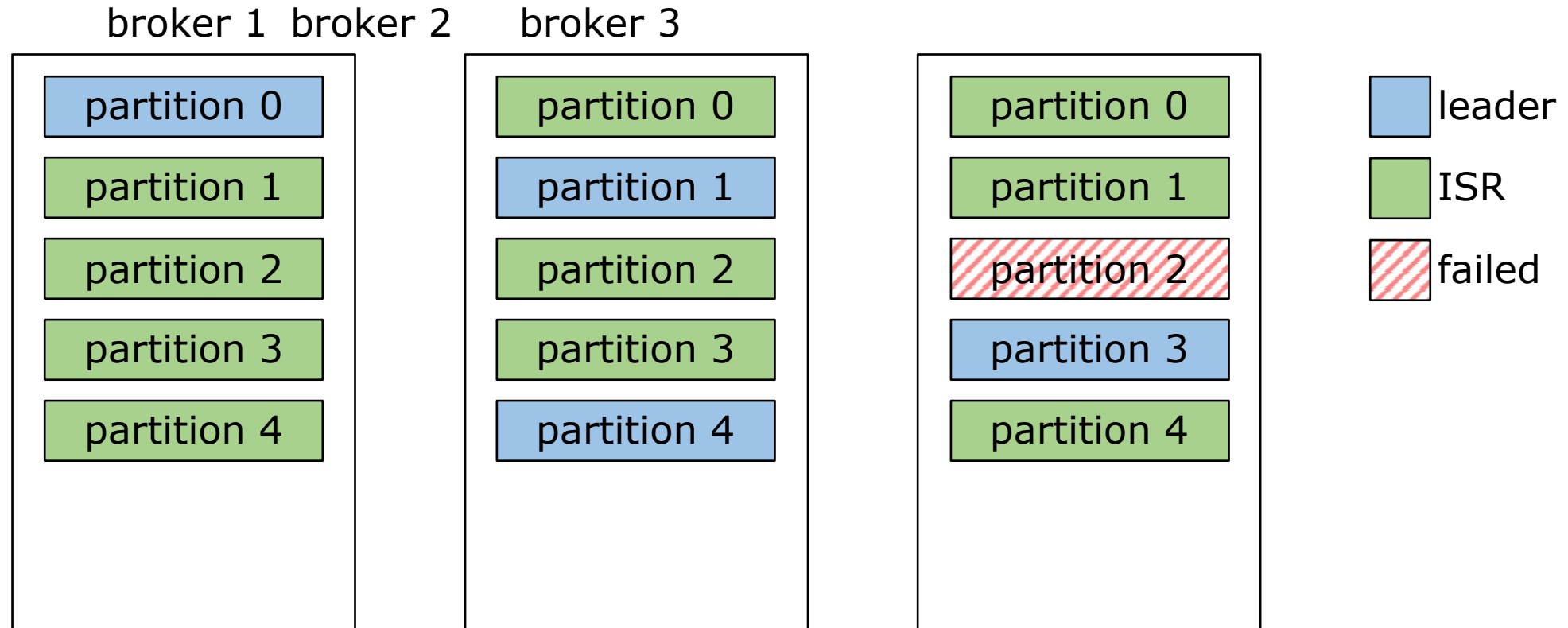
Все реплики синхронизированы



Архитектура Kafka

Broker

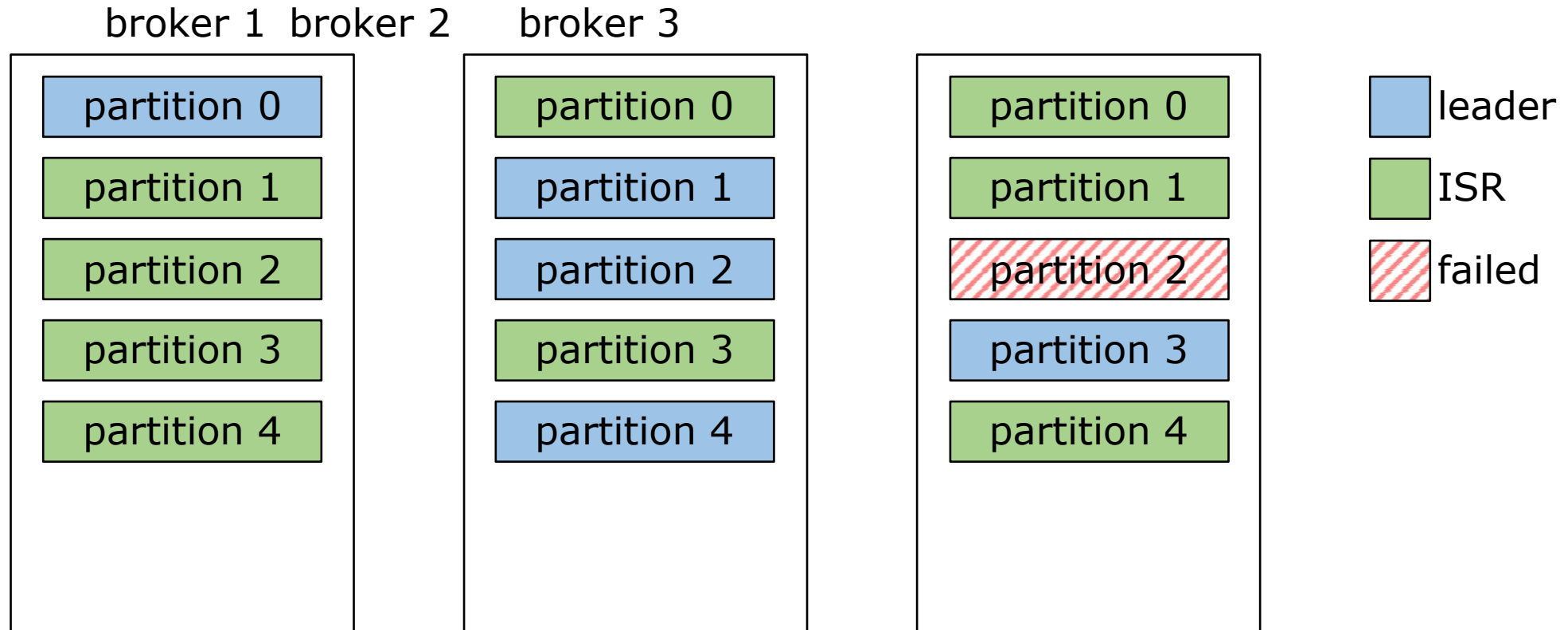
Недоступность *лидера* у partition 2



Архитектура Kafka

Broker

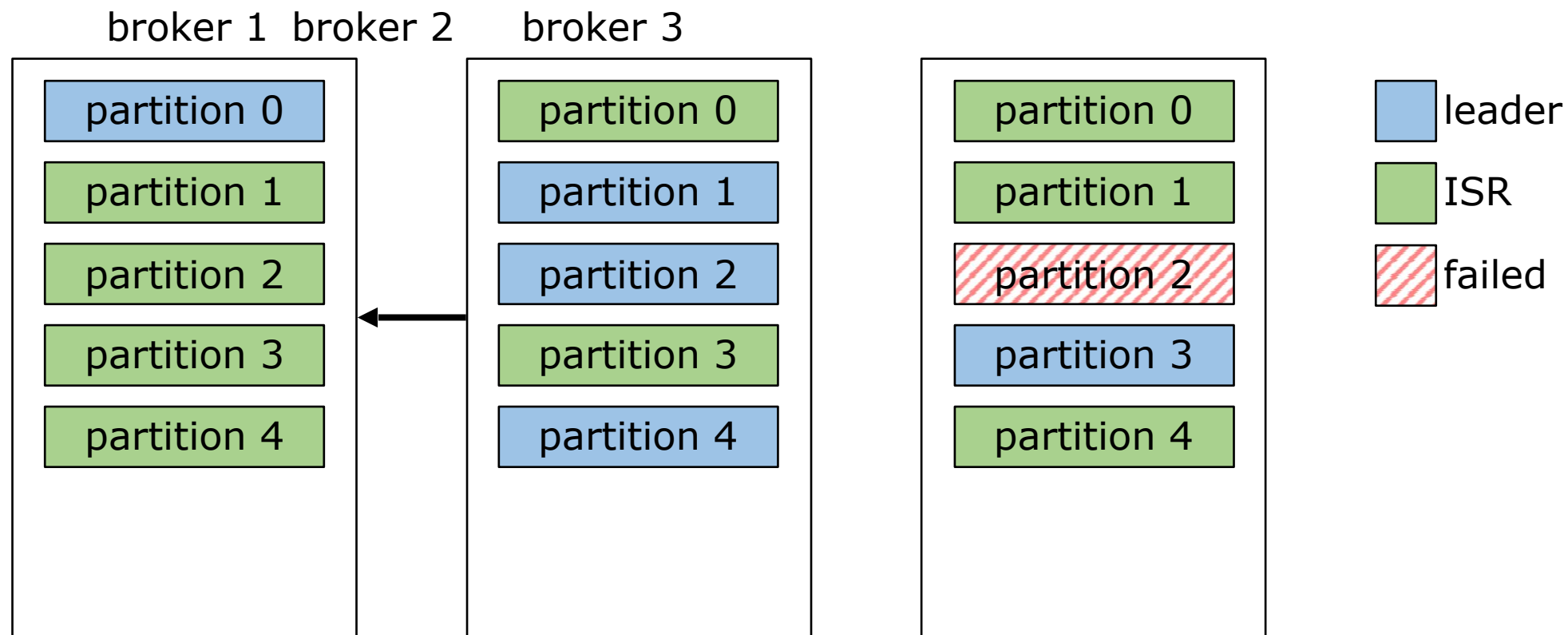
Выбор нового *лидера* в случае недоступности



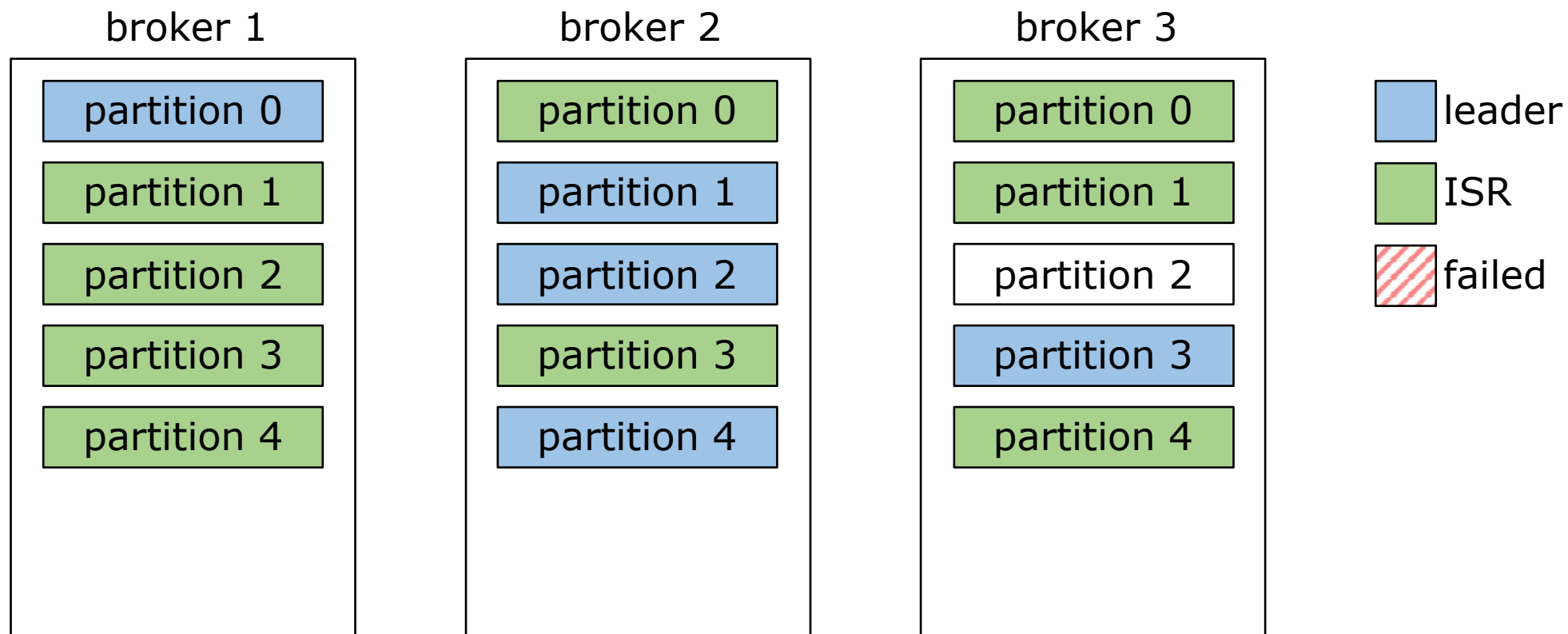
Архитектура Kafka

Broker

Репликация с нового *лидера*



Архитектура Kafka Broker



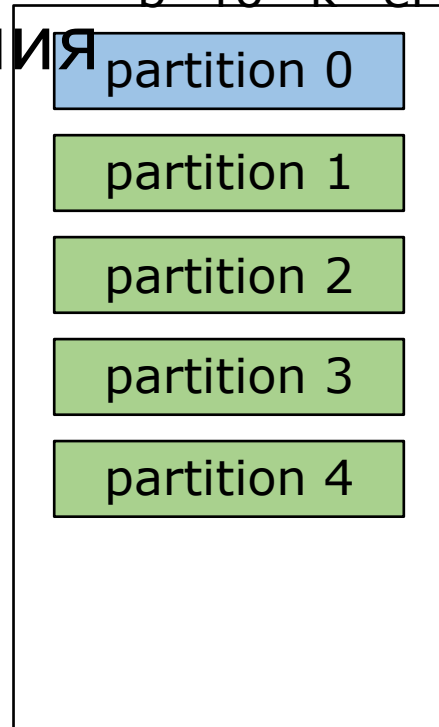
Архитектура Kafka

Broker

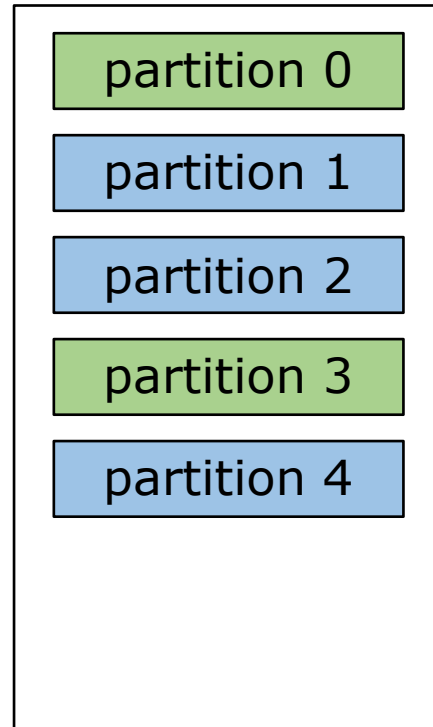
Синхронизация реплики с *лидером* после

восстановления

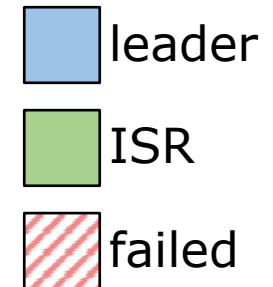
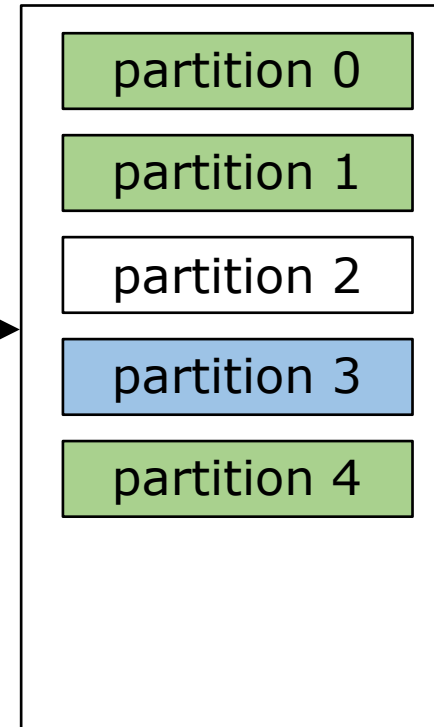
1



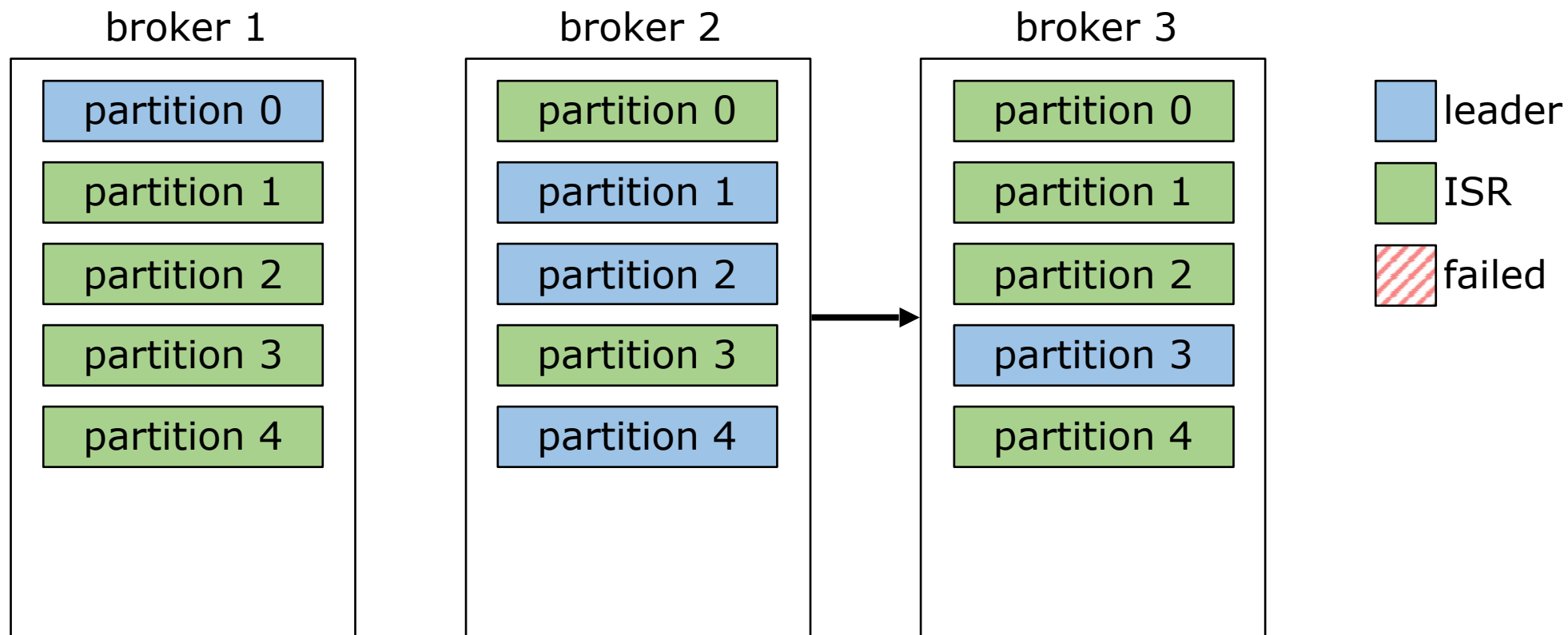
broker 2



broker 3



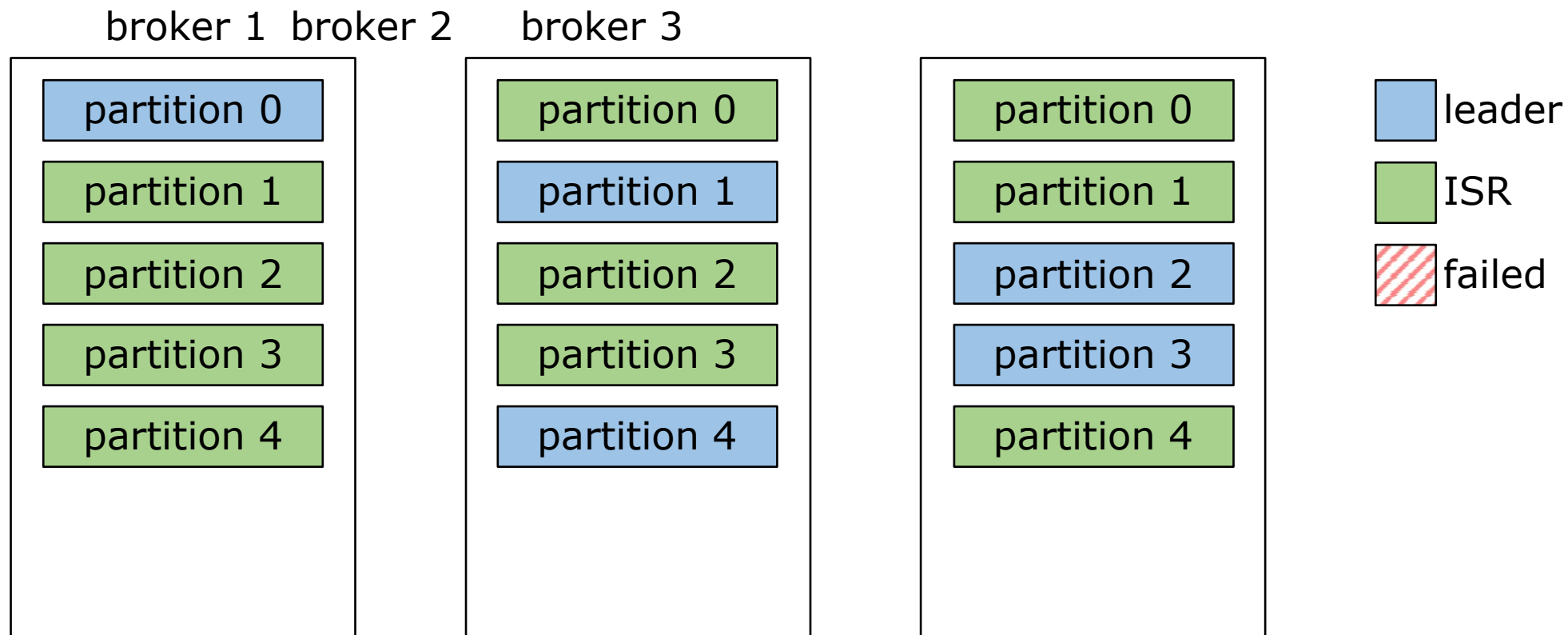
Архитектура Kafka Broker



Архитектура Kafka

Broker

Перебалансировка *лидеров*



Архитектура Kafka Producer

Архитектура Kafka

Producer

message = (key,
value)

Архитектура Kafka

Producer

message = (**key**, value)

Архитектура Kafka

Producer

message = (**key**, value)

partition = murmur2(key) % partitions

Архитектура Kafka

Producer

message = (**key**, value)

partition = murmur2(key) % partitions // key != null

Архитектура Kafka

Producer

message = (**key**, value)

partition = murmur2(key) % partitions // key != null

<https://ru.wikipedia.org/wiki/MurmurHash2>

Архитектура Kafka

Producer

message = (**key**, value)

partition = murmur2(key) % partitions // key != null

partition = round_robin(partitions)

Архитектура Kafka

Producer

message = (**key**, value)

partition = murmur2(key) % partitions // key != null

partition = round_robin(partitions) // key == null

Архитектура Kafka

Producer

message = (key,
value)

Архитектура Kafka Producer

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Архитектура Kafka Producer

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

 leader

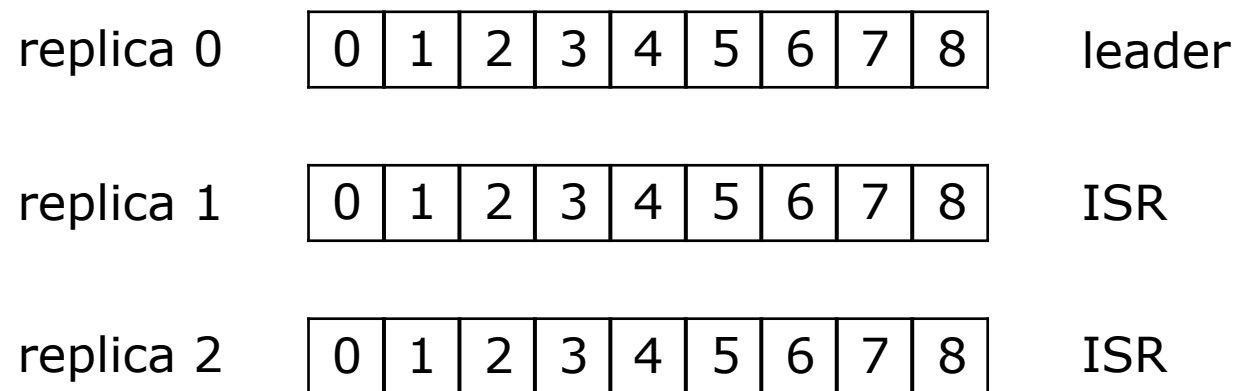
replica 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Архитектура Kafka Producer



Архитектура Kafka

Producer

Acknowledgement (ack) – подтверждение записи

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 2

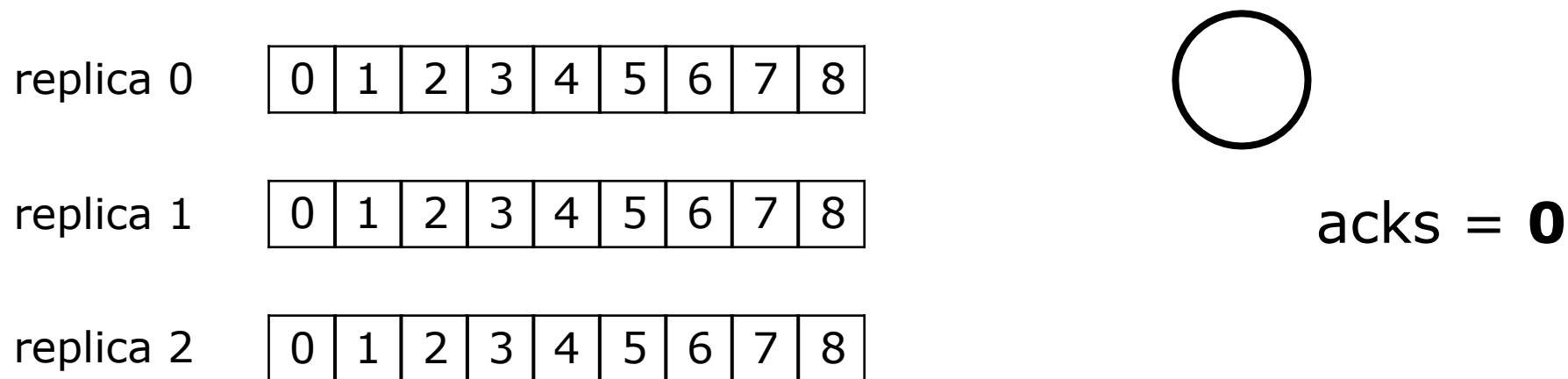
0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

acks = **0**

Архитектура Kafka

Producer

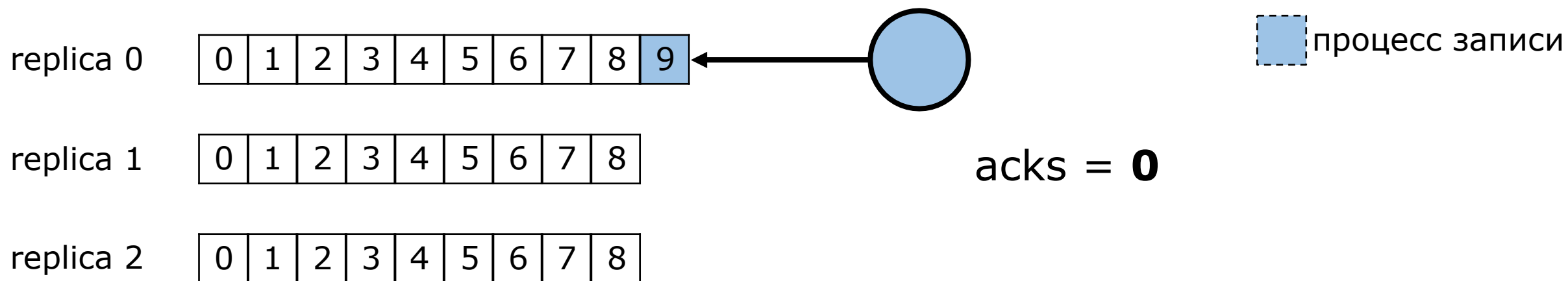
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

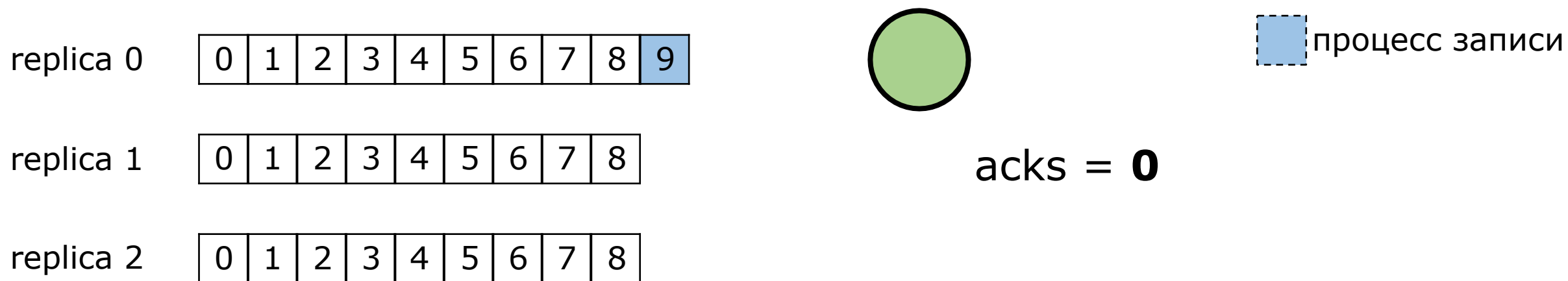
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

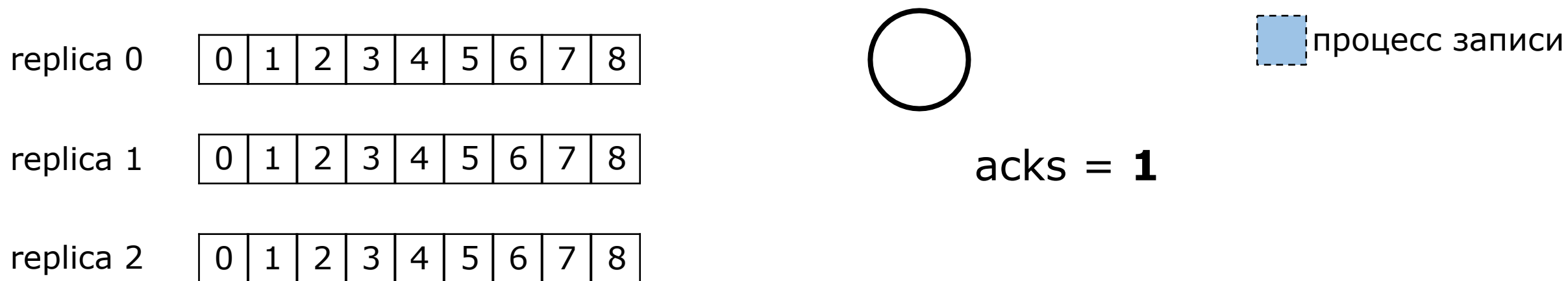
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

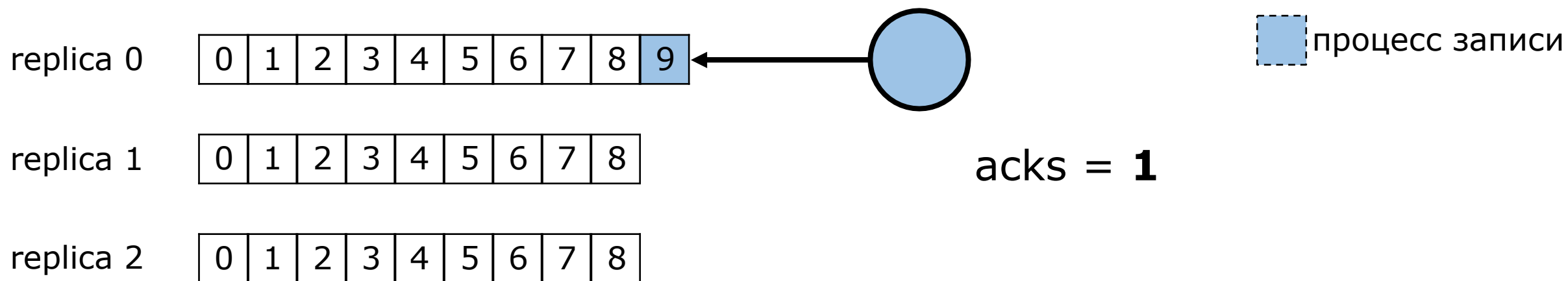
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

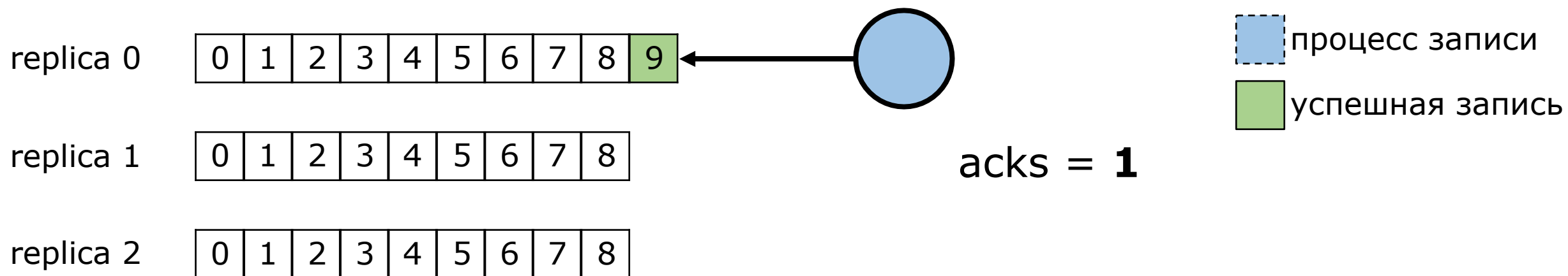
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

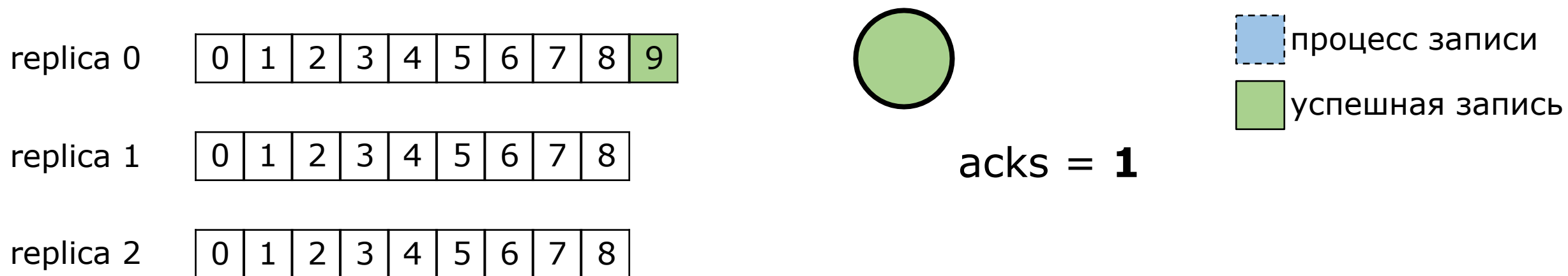
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

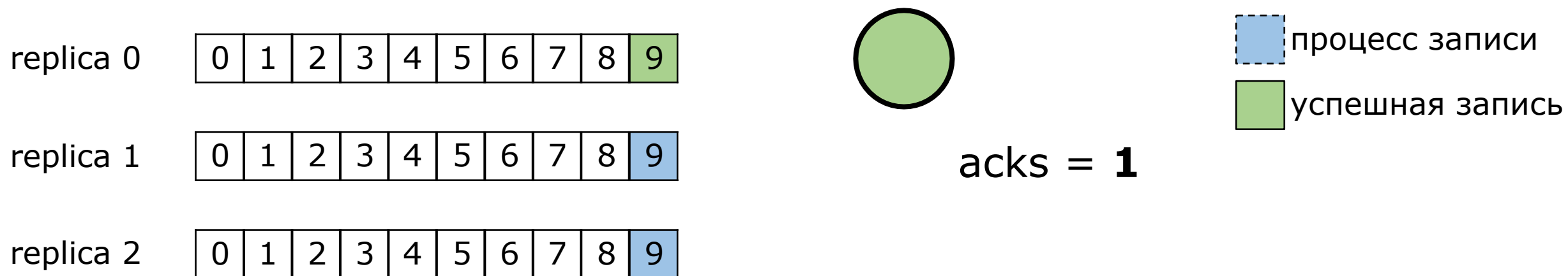
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

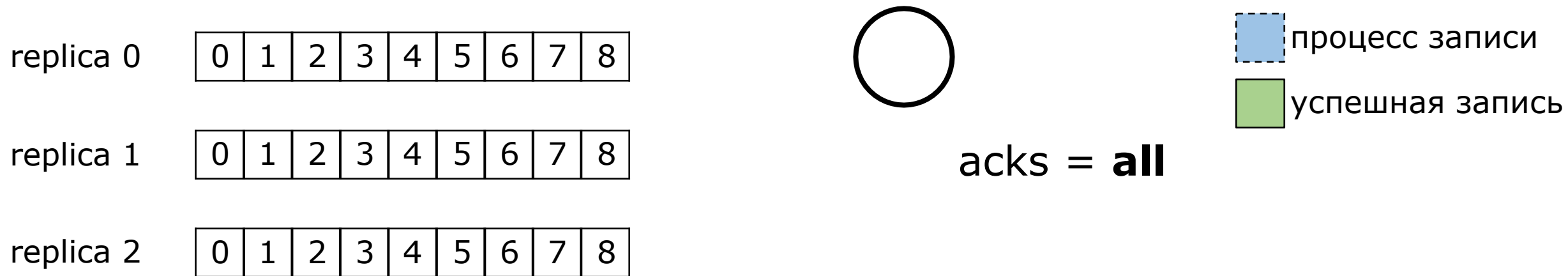
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

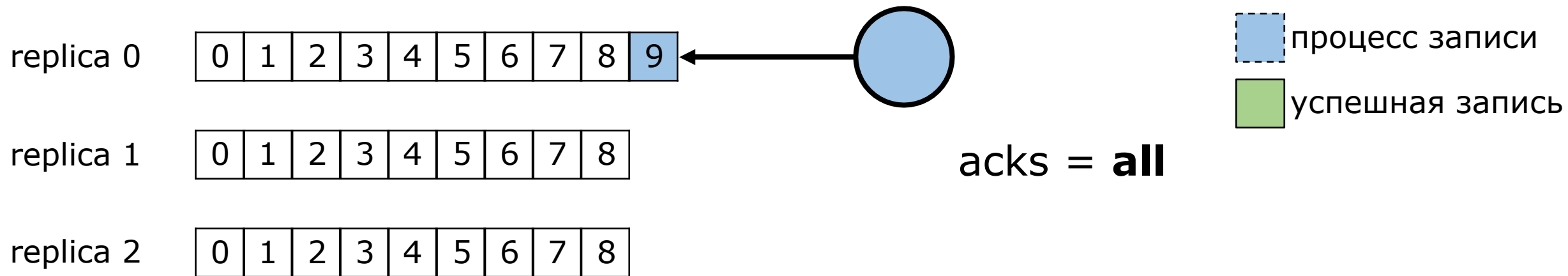
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

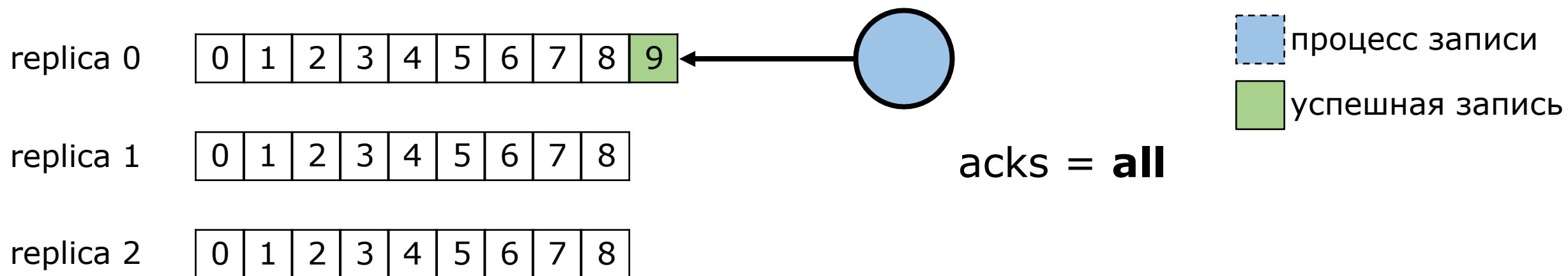
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

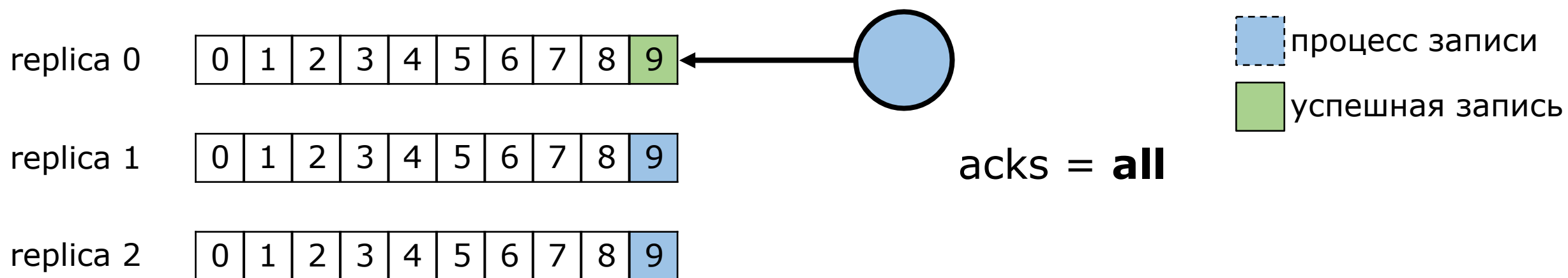
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

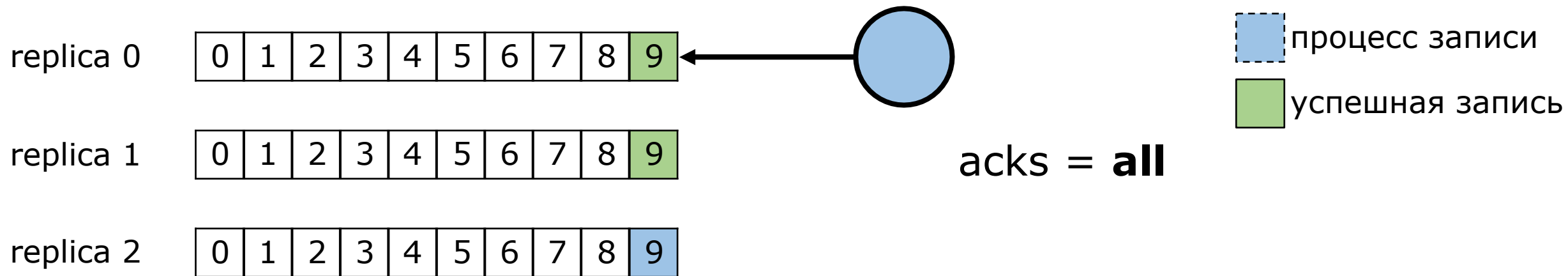
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

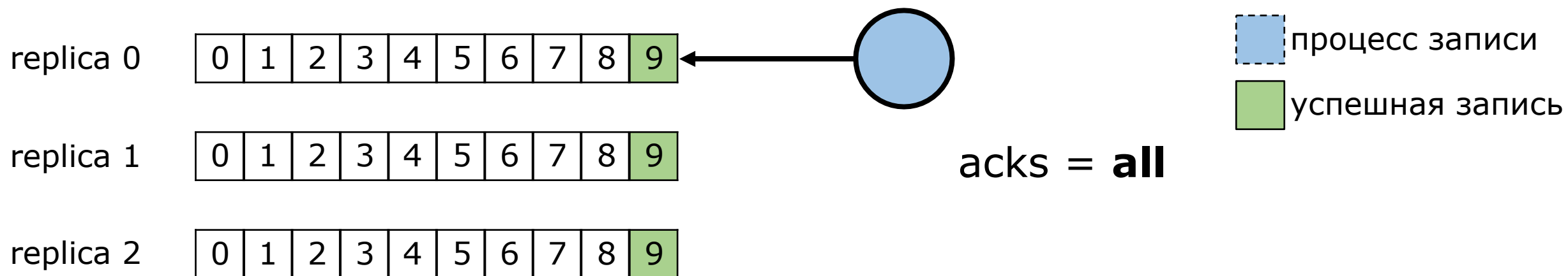
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

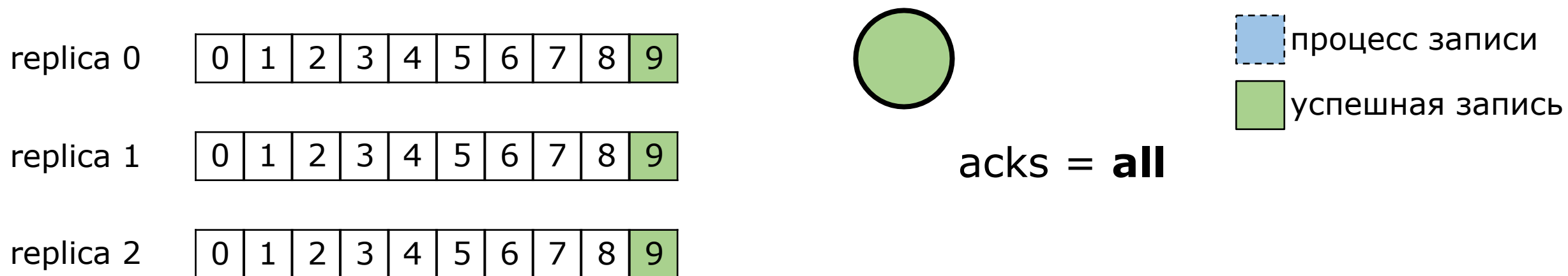
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

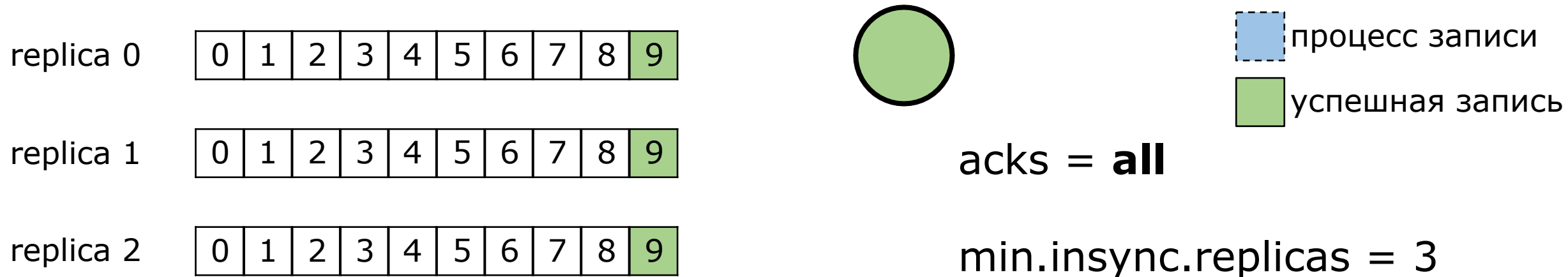
Acknowledgement (ack) – подтверждение записи



Архитектура Kafka

Producer

Acknowledgement (ack) – подтверждение записи



Архитектура Kafka Consumer

Архитектура Kafka Consumer

partition 0

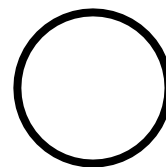
0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

partition 1

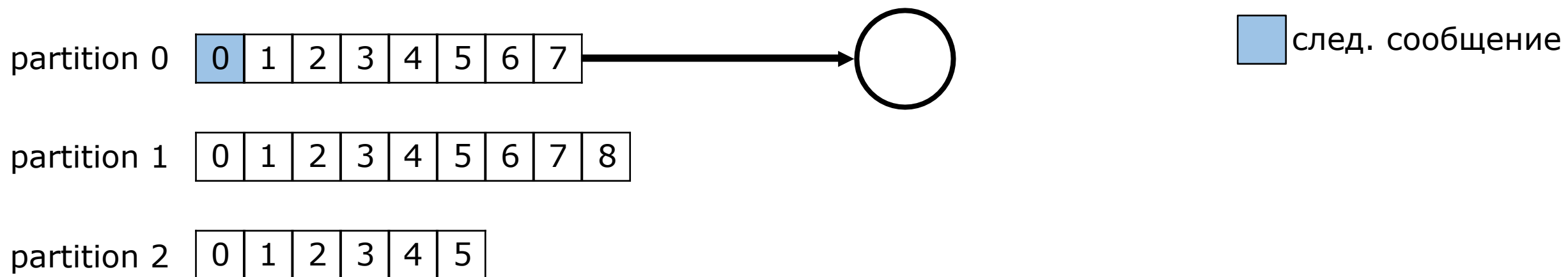
0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

partition 2

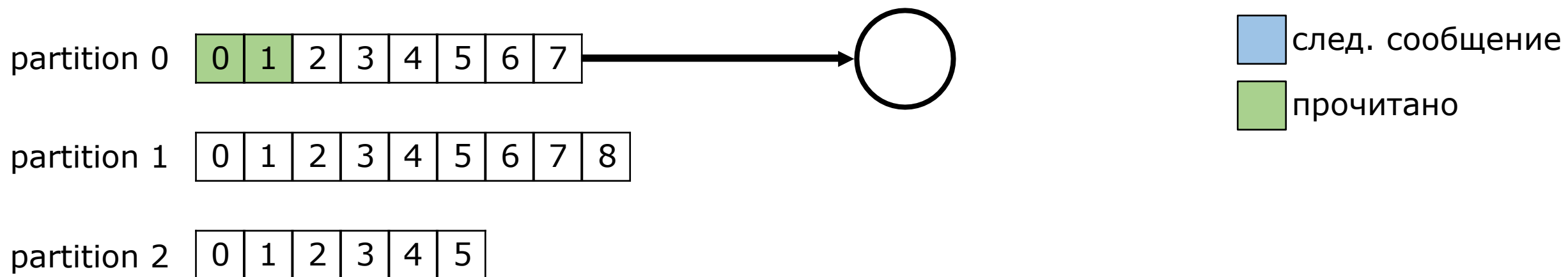
0	1	2	3	4	5
---	---	---	---	---	---



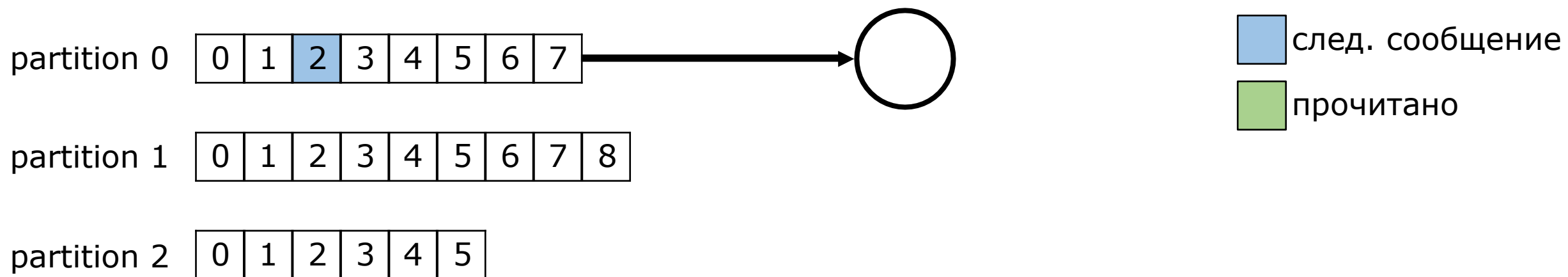
Архитектура Kafka Consumer



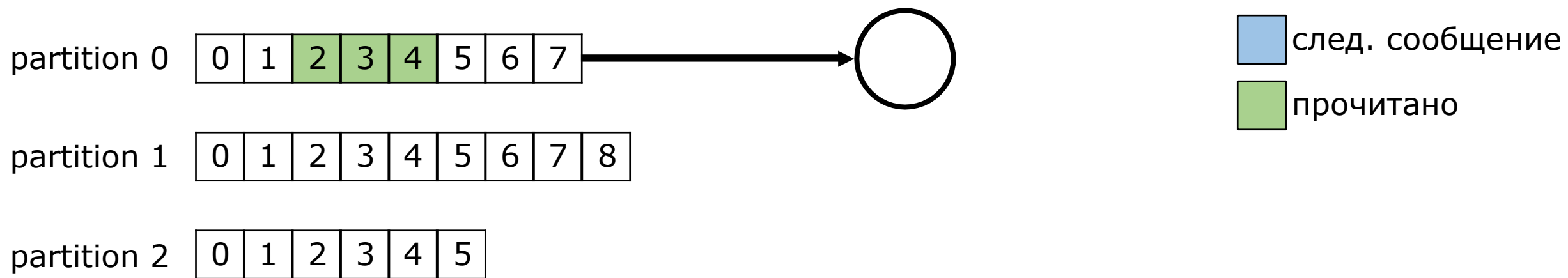
Архитектура Kafka Consumer



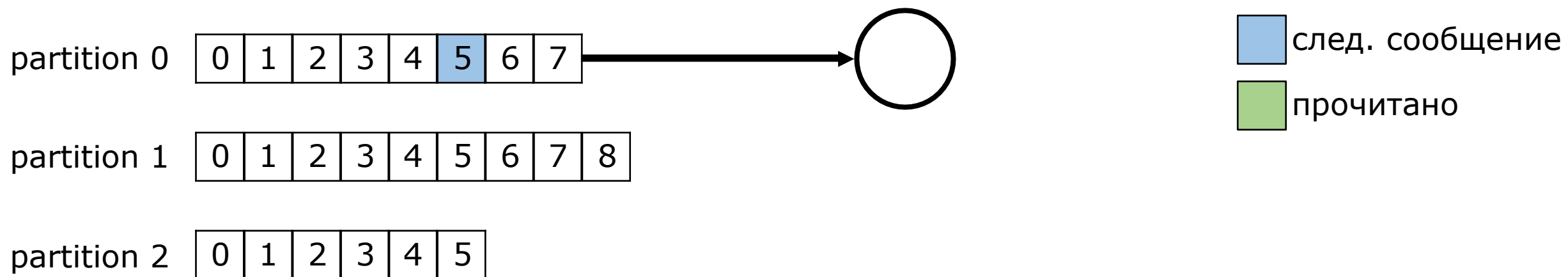
Архитектура Kafka Consumer



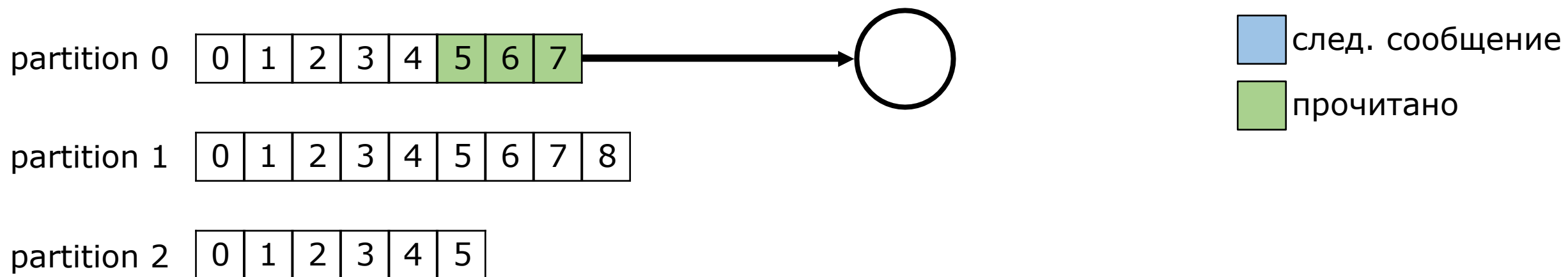
Архитектура Kafka Consumer



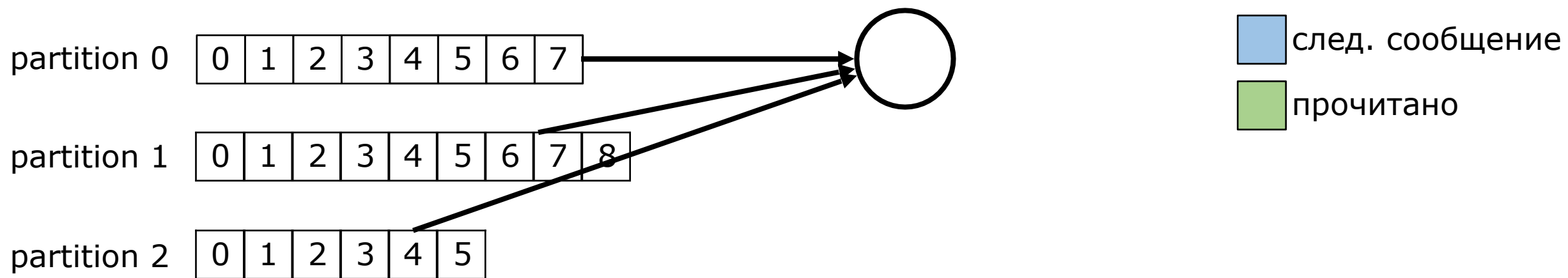
Архитектура Kafka Consumer



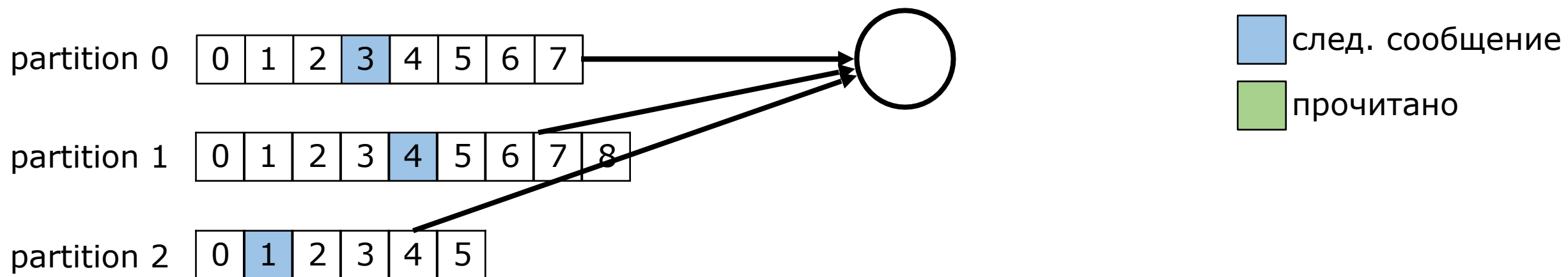
Архитектура Kafka Consumer



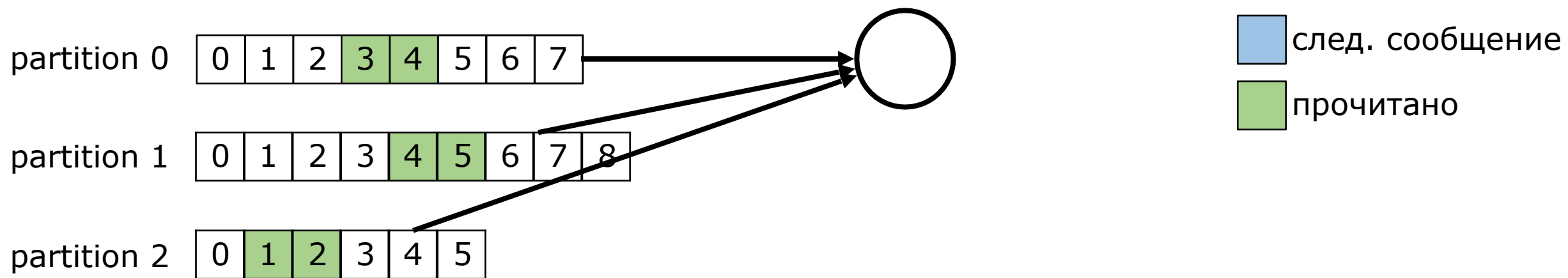
Архитектура Kafka Consumer



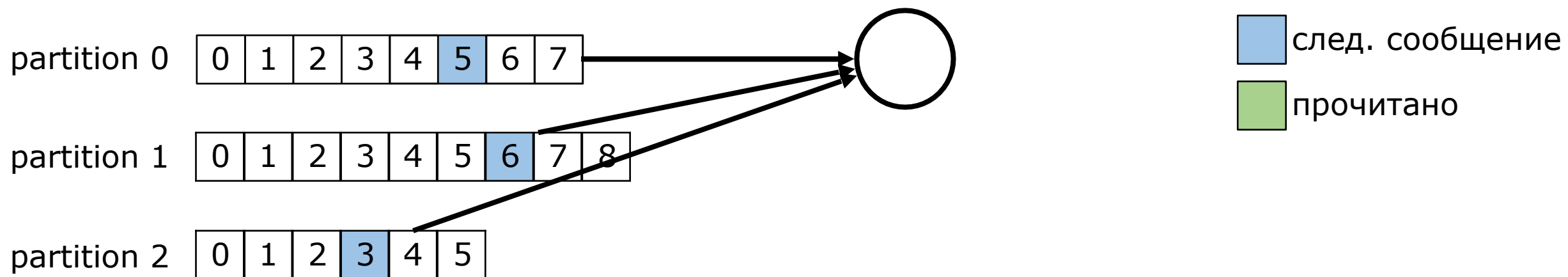
Архитектура Kafka Consumer



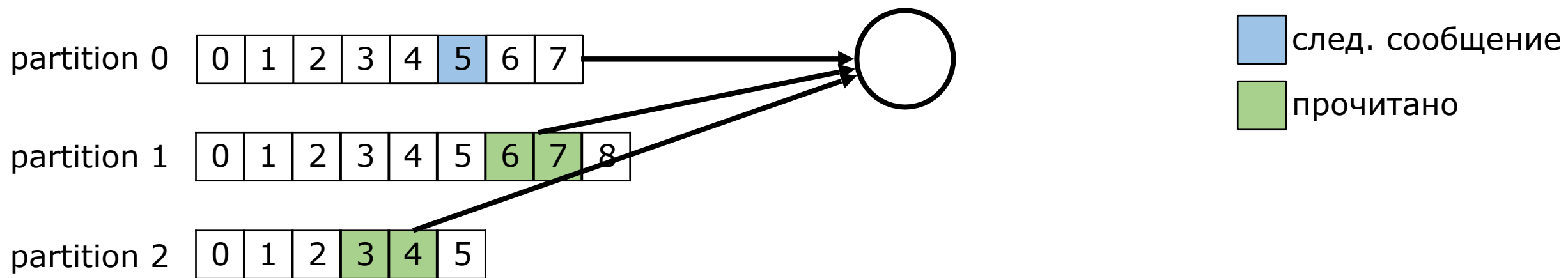
Архитектура Kafka Consumer



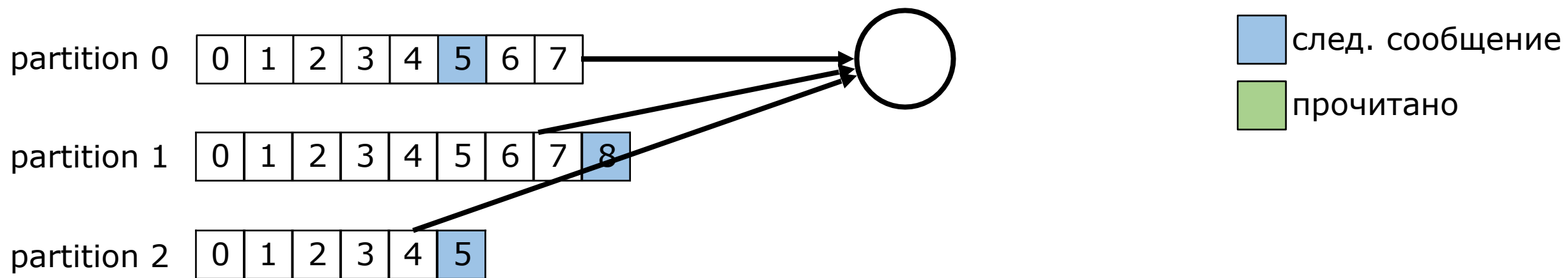
Архитектура Kafka Consumer



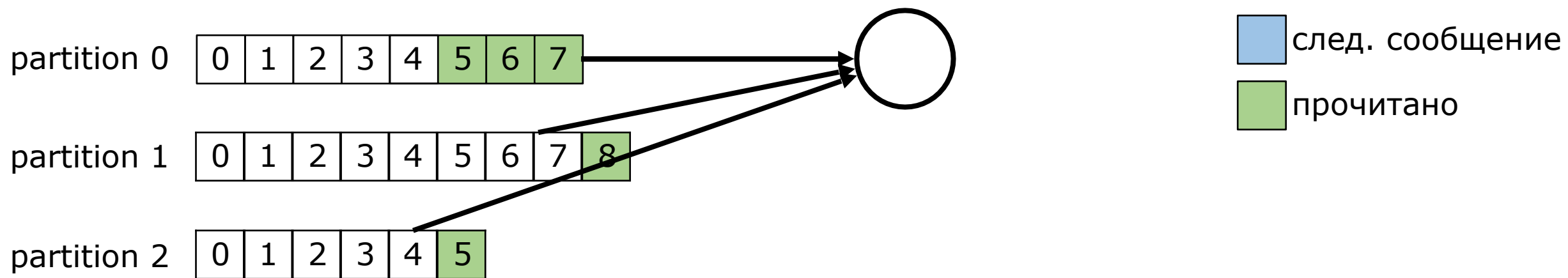
Архитектура Kafka Consumer



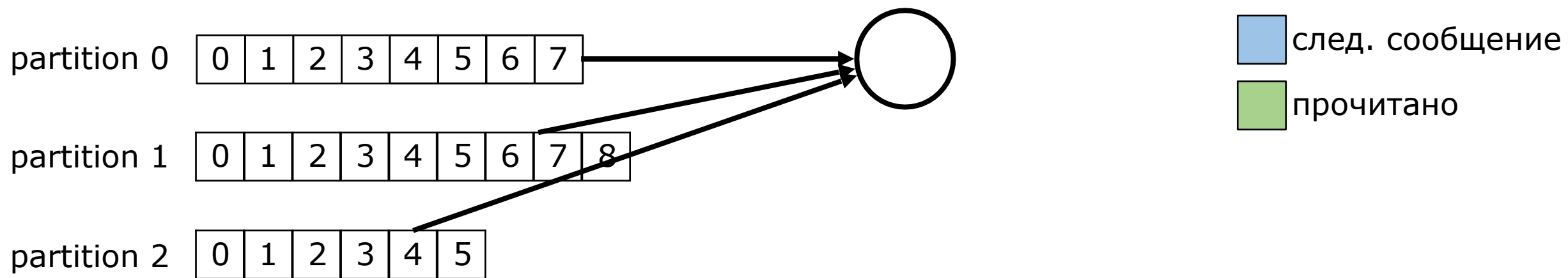
Архитектура Kafka Consumer



Архитектура Kafka Consumer



Архитектура Kafka Consumer



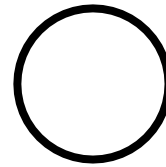
Архитектура Kafka

Consumer

Commit offset

partition 0

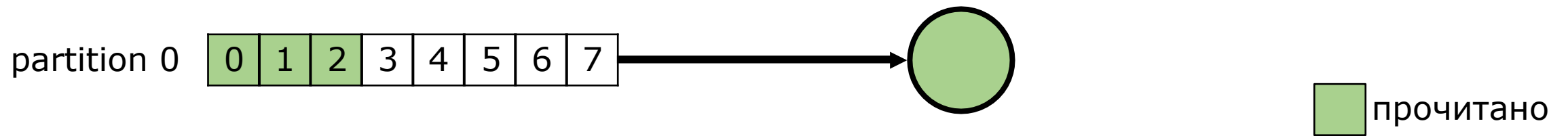
0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---



Архитектура Kafka

Consumer

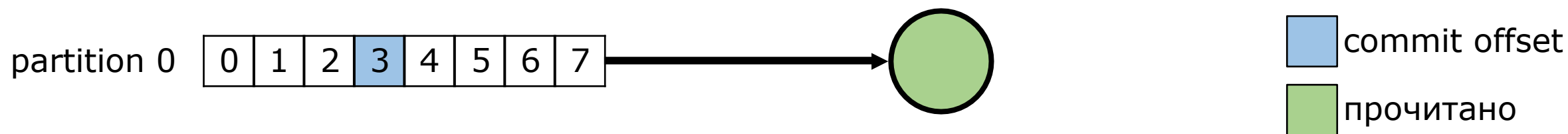
Commit offset



Архитектура Kafka

Consumer

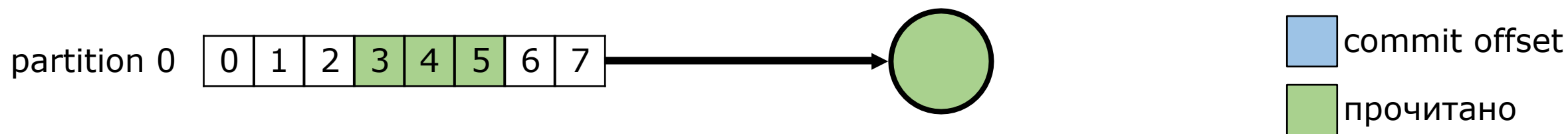
Commit offset



Архитектура Kafka

Consumer

Commit offset



Архитектура Kafka

Consumer

Commit offset



Архитектура Kafka

Consumer

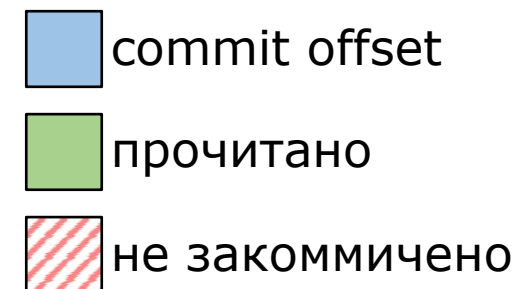
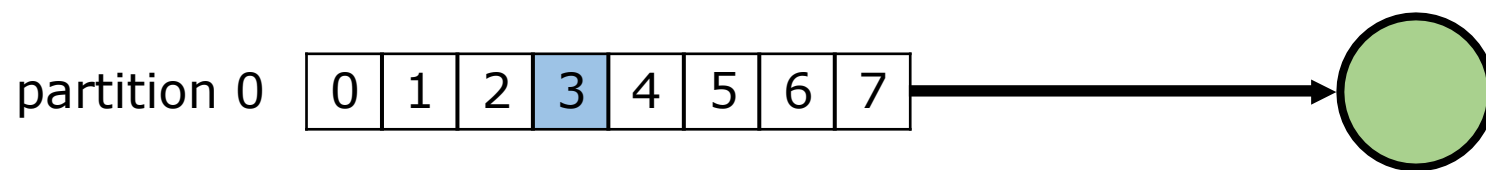
Commit offset



Архитектура Kafka

Consumer

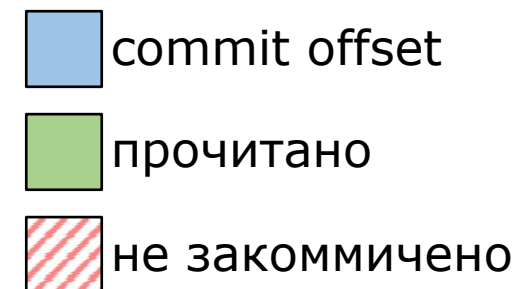
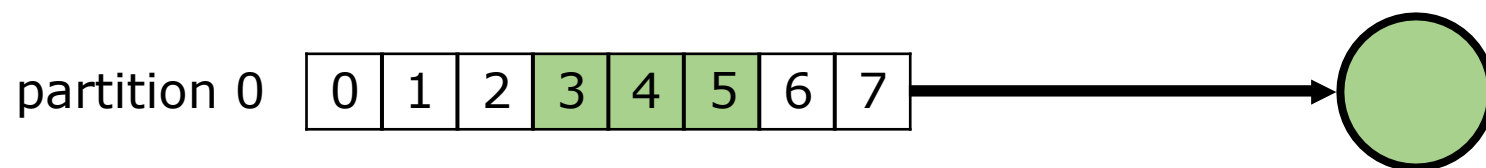
Commit offset



Архитектура Kafka

Consumer

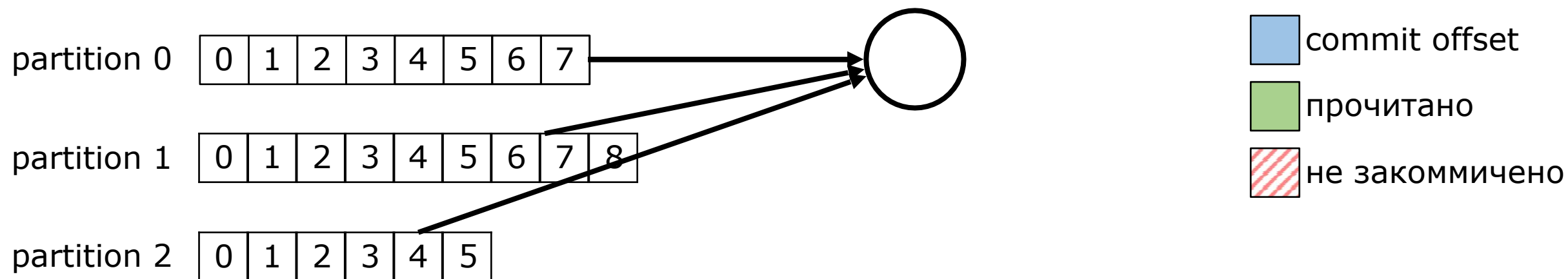
Commit offset



Архитектура Kafka

Consumer

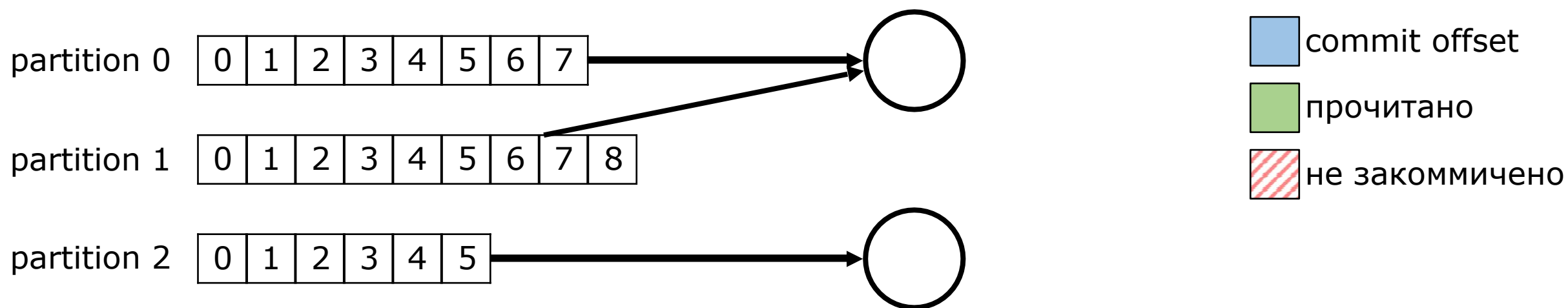
Consumer Group



Архитектура Kafka

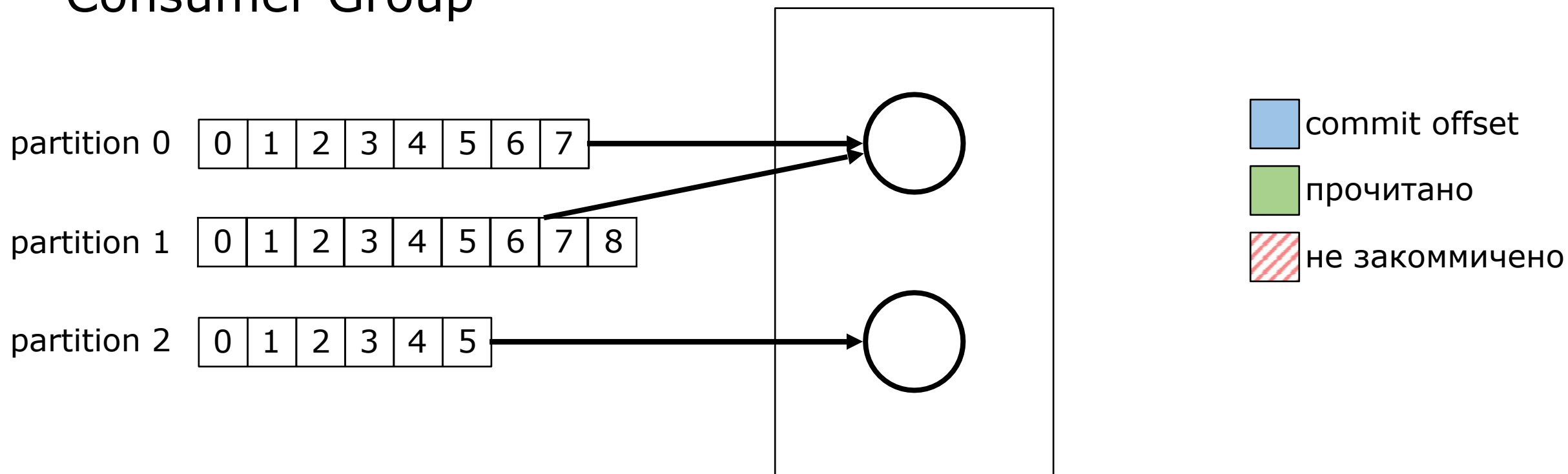
Consumer

Consumer Group



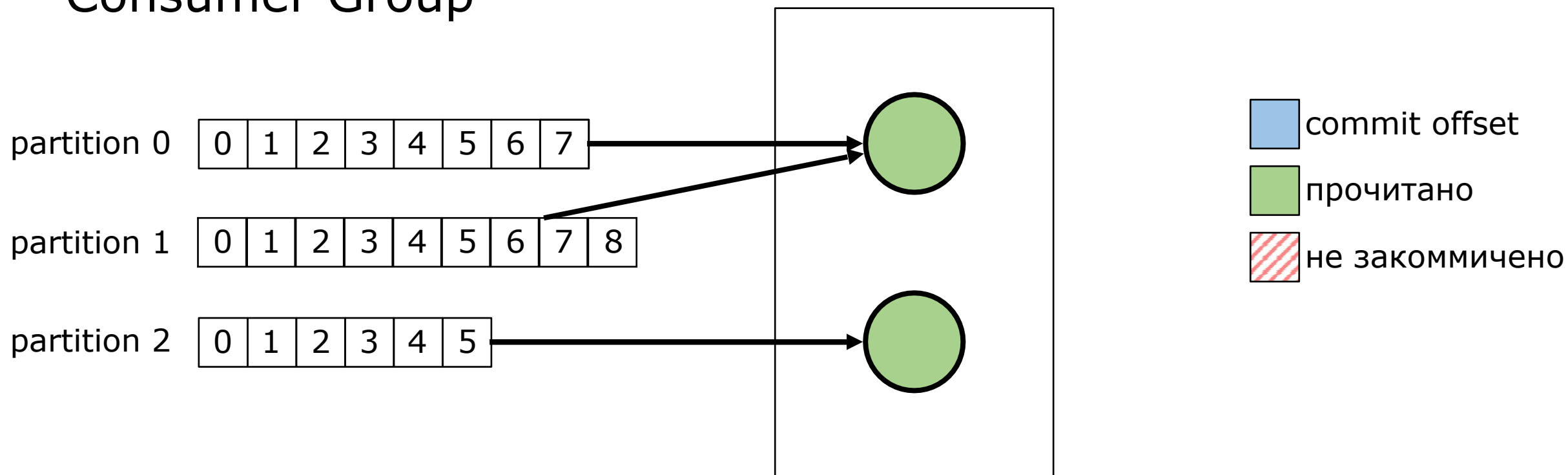
Архитектура Kafka

Consumer Consumer Group



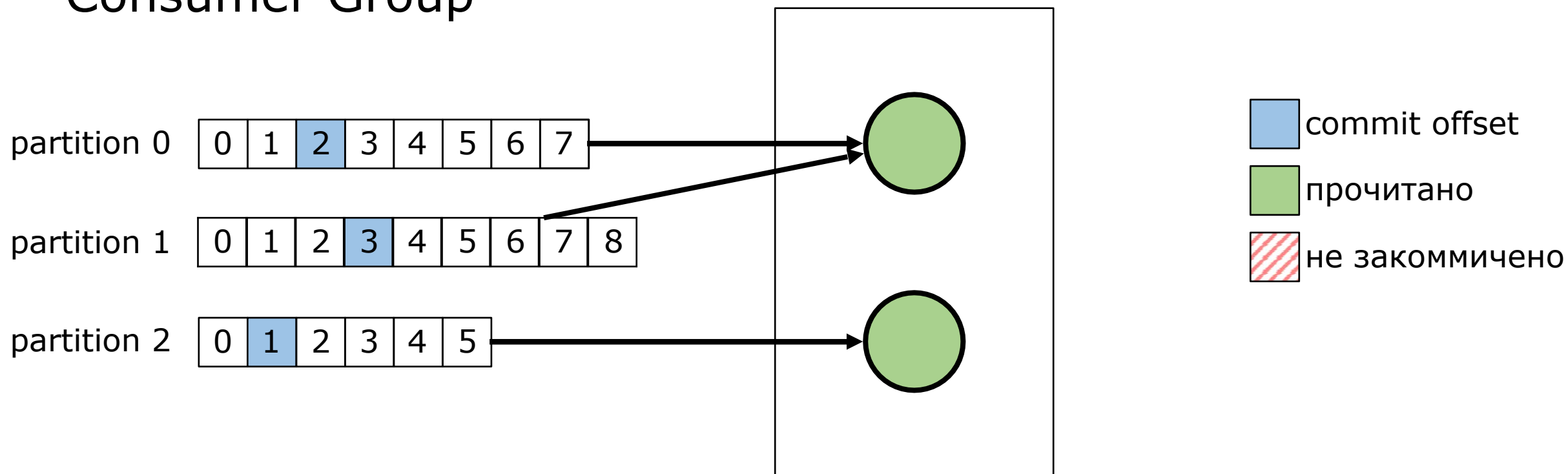
Архитектура Kafka

Consumer Consumer Group



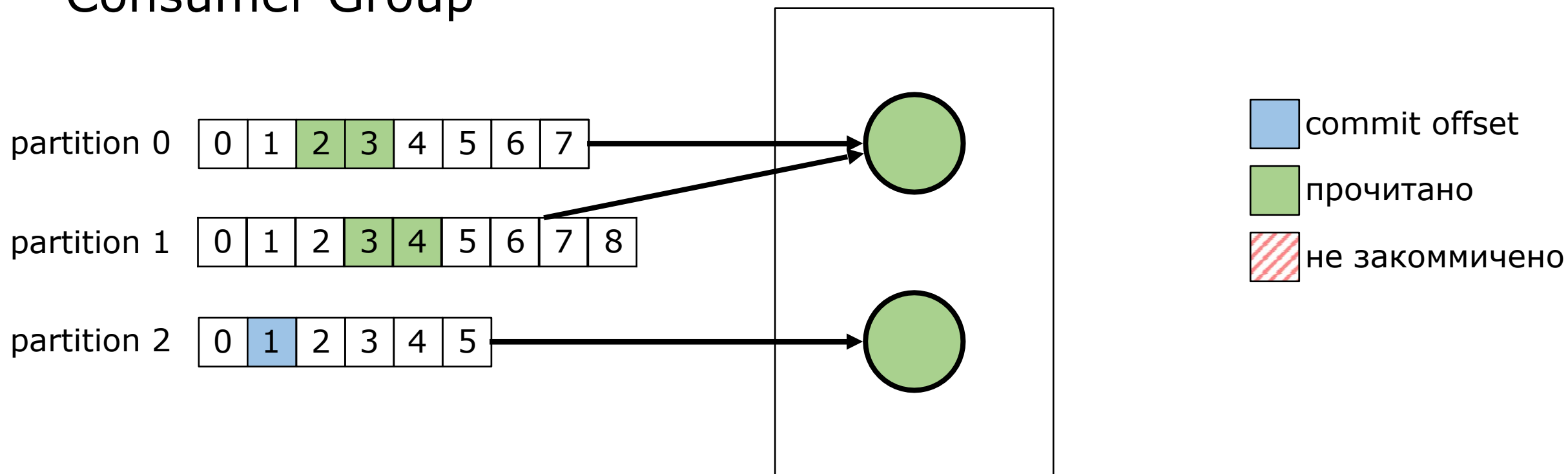
Архитектура Kafka

Consumer Consumer Group



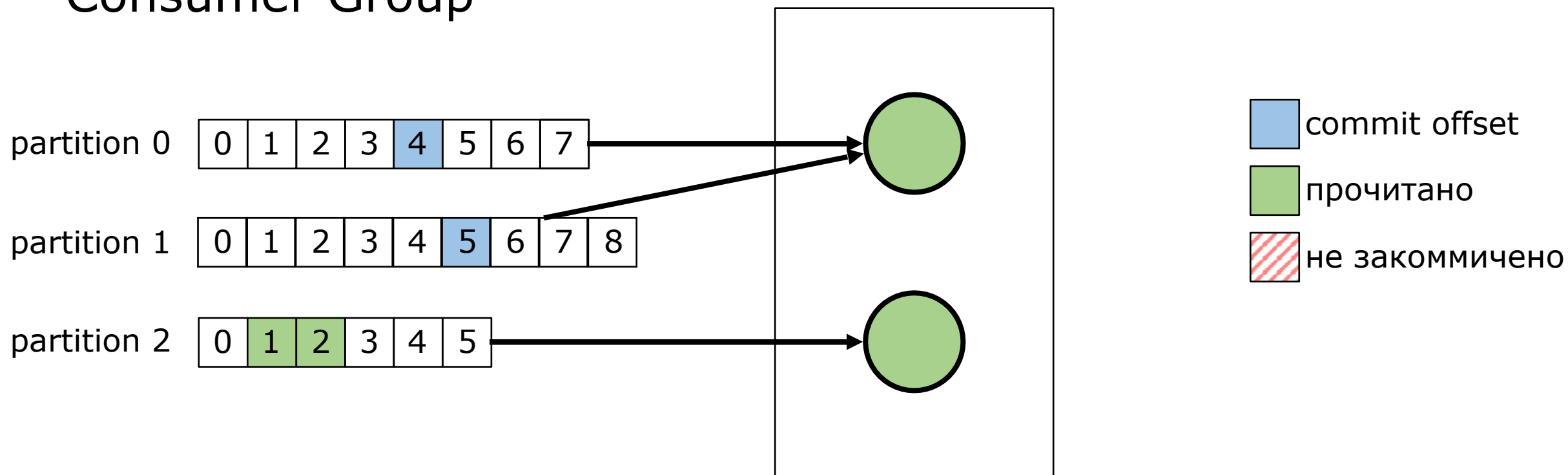
Архитектура Kafka

Consumer Consumer Group



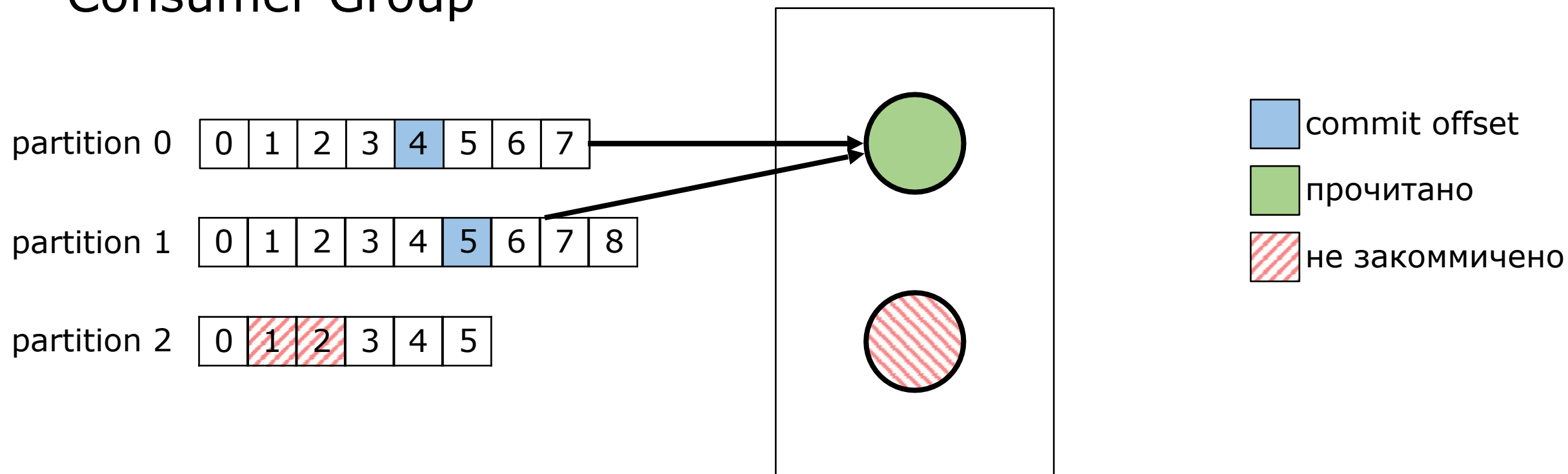
Архитектура Kafka

Consumer Consumer Group



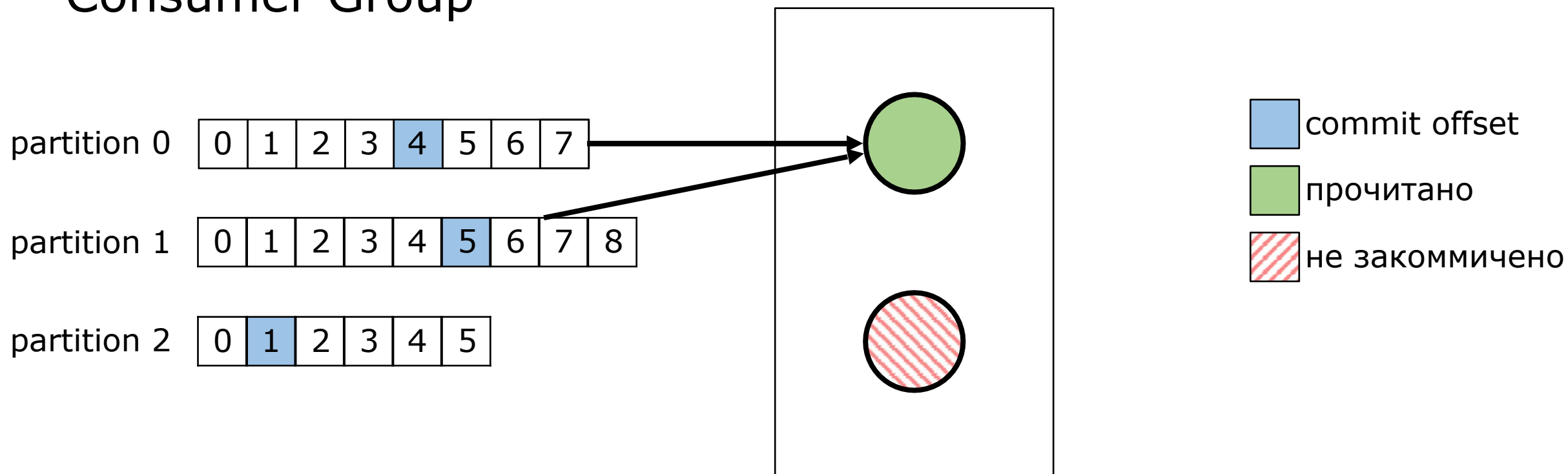
Архитектура Kafka

Consumer Consumer Group



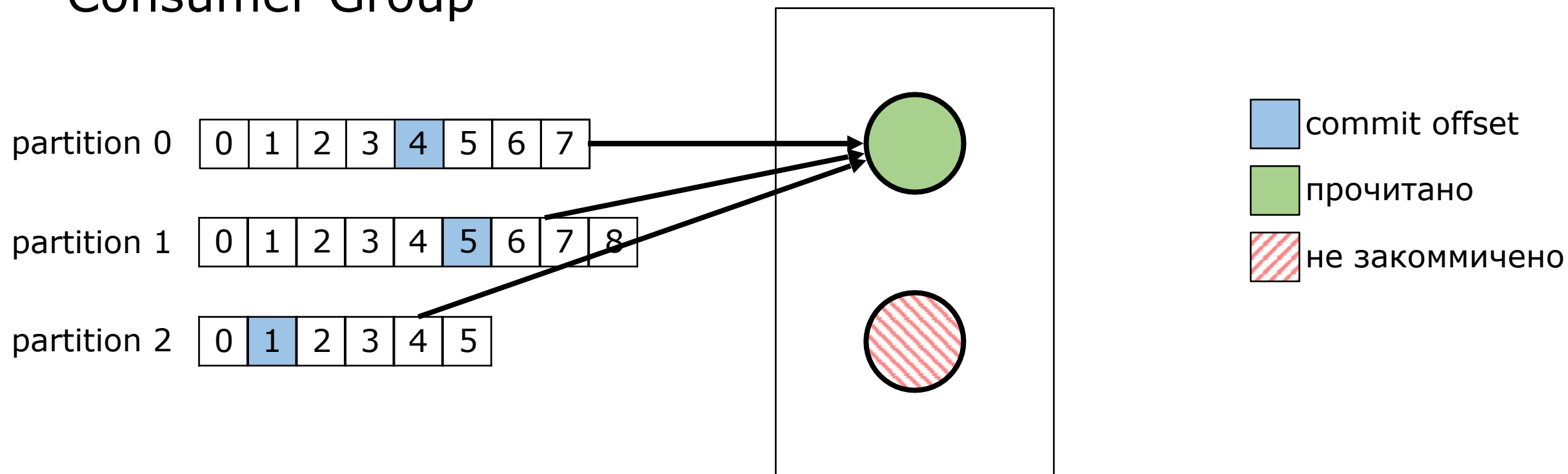
Архитектура Kafka

Consumer Consumer Group



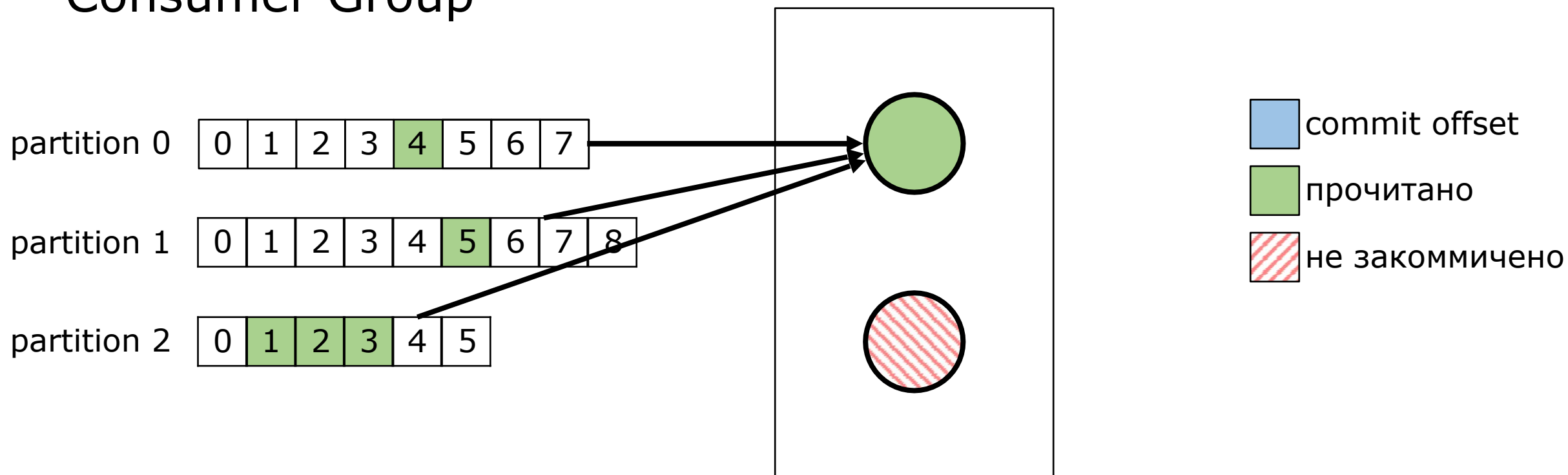
Архитектура Kafka

Consumer Consumer Group



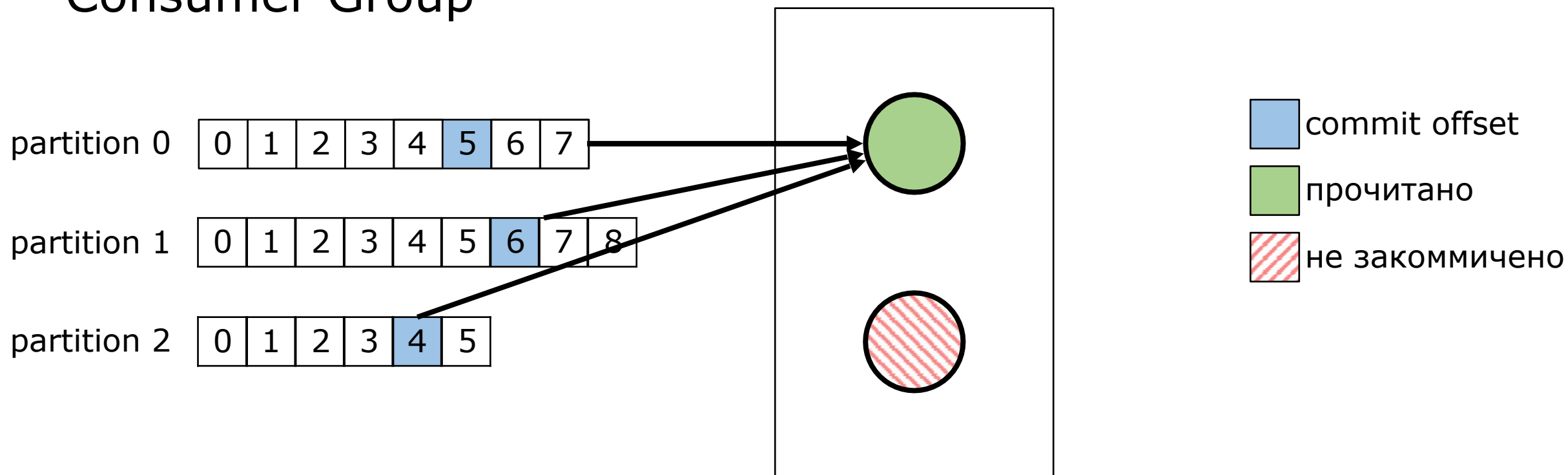
Архитектура Kafka

Consumer Consumer Group



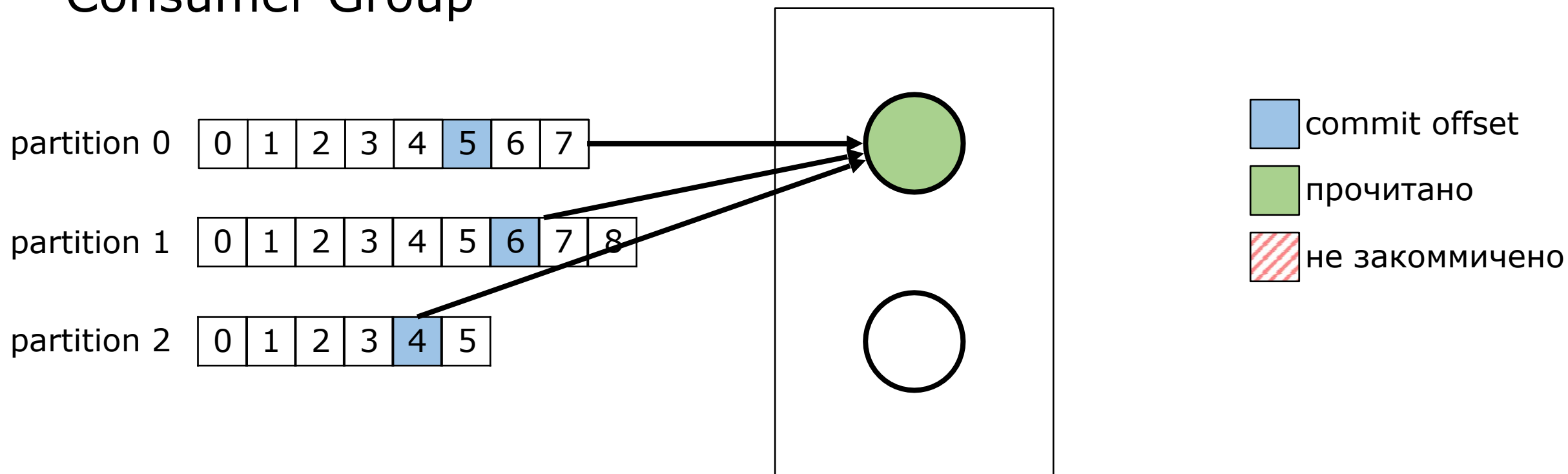
Архитектура Kafka

Consumer Consumer Group



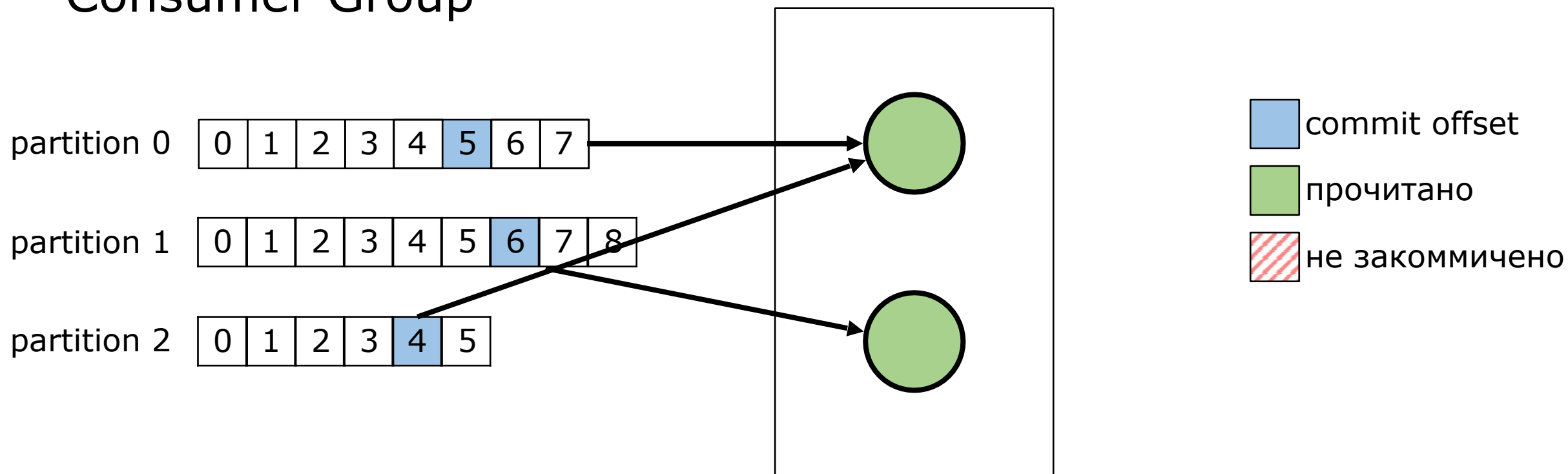
Архитектура Kafka

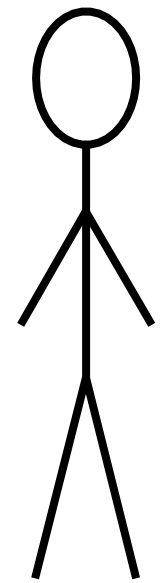
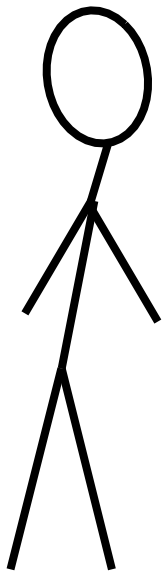
Consumer Consumer Group

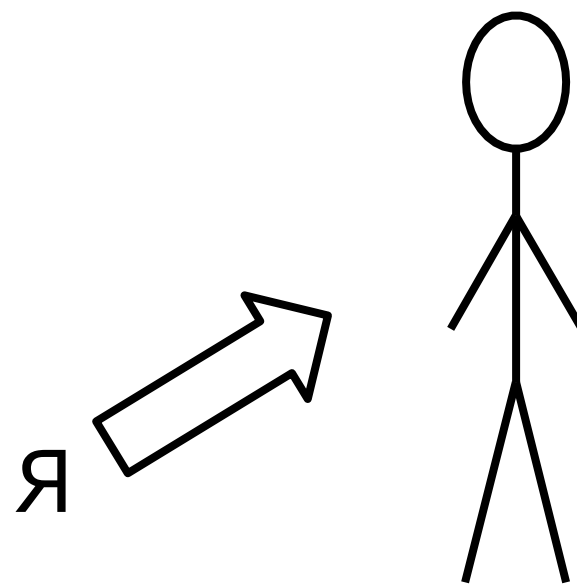
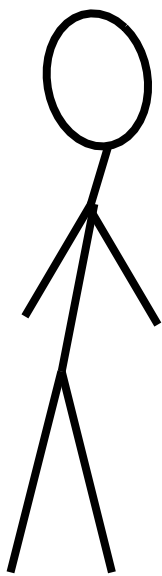


Архитектура Kafka

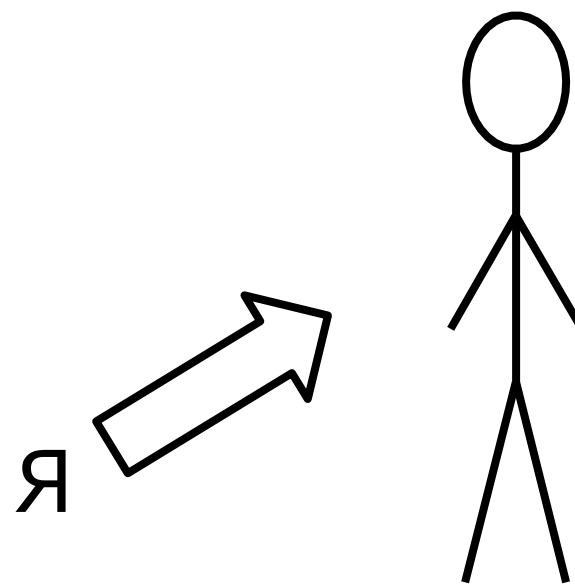
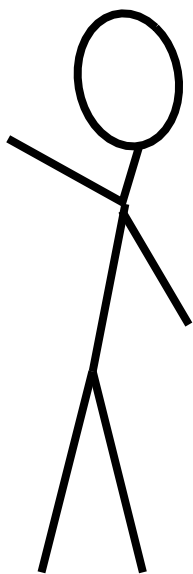
Consumer Consumer Group



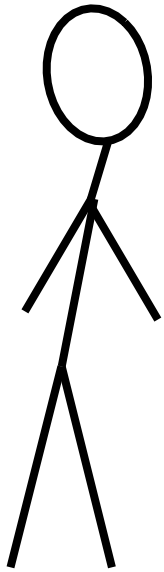




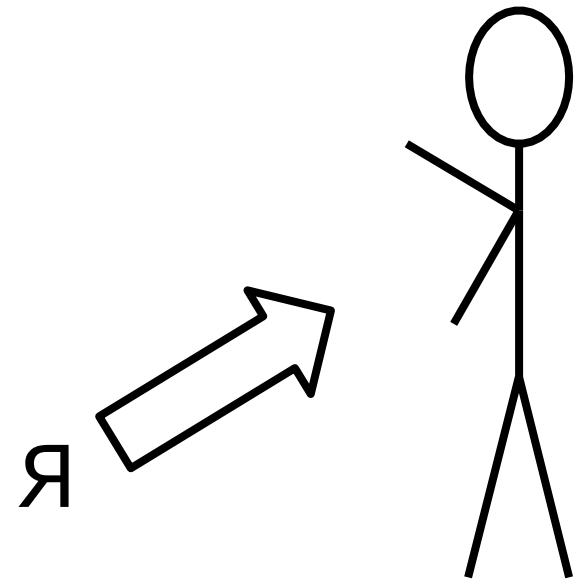
МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...



МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...

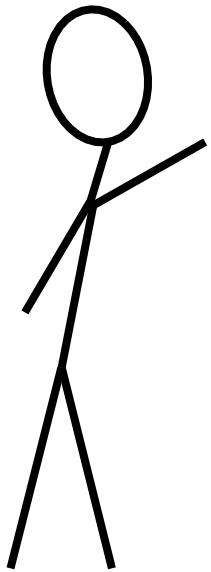


НЕТ!
УЖЕ ЕСТЬ КАФКА!!!

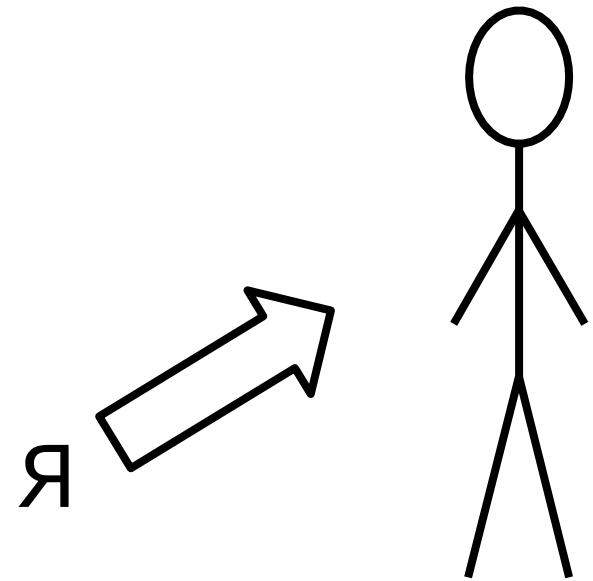


МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...

НО У НАС ВСЕГО
100-500 RPS...

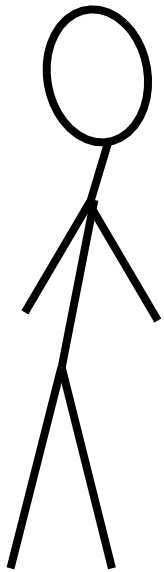


НЕТ!
УЖЕ ЕСТЬ КАФКА!!!



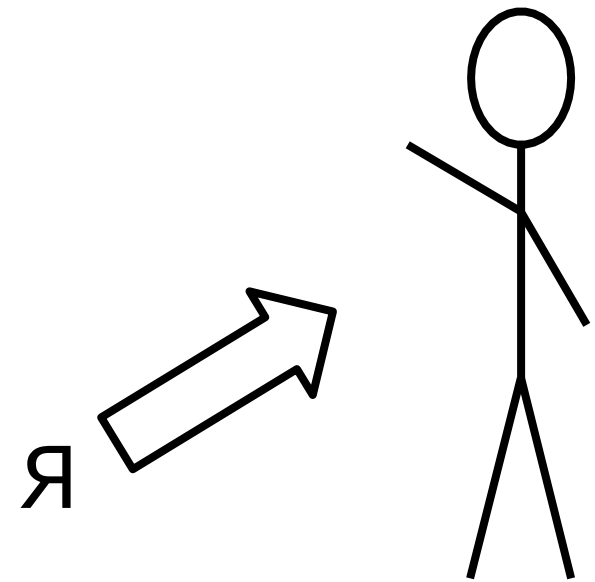
МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...

НО У НАС ВСЕГО
100-500 RPS...



НЕТ!
УЖЕ ЕСТЬ КАФКА!!!

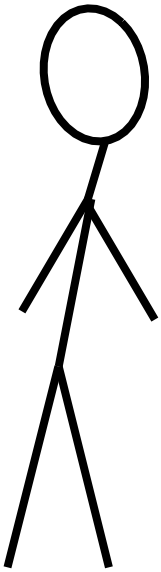
ТОЛЬКО КАФКА!!!
И НЕЧЕГО ДУМАТЬ!!



МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...

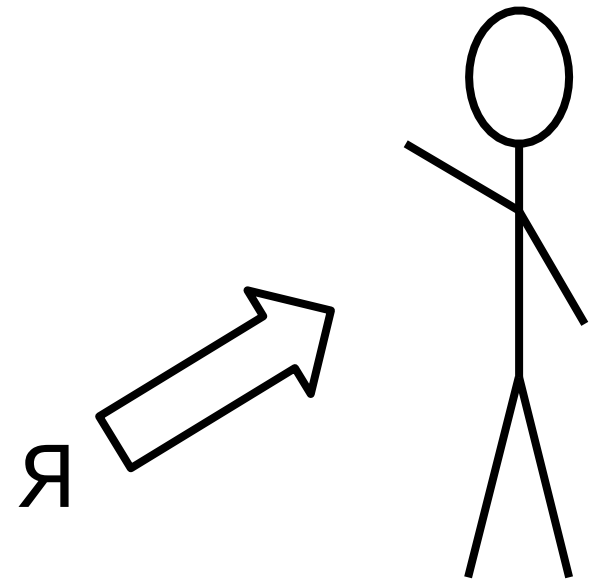
НО У НАС ВСЕГО
100-500 RPS...

... И ДАННЫЕ В ОБЩЕМ-ТО
РЕЛЯЦИОННЫЕ...



НЕТ!
УЖЕ ЕСТЬ КАФКА!!!

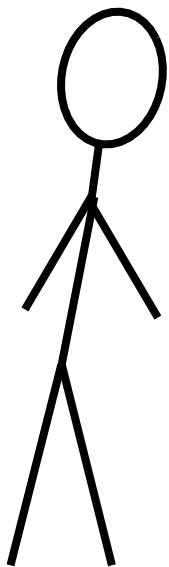
ТОЛЬКО КАФКА!!!
И НЕЧЕГО ДУМАТЬ!!



МЫ ХОТИМ ВЫБРАТЬ
POSTGRESQL ДЛЯ
НОВОГО ПРОЕКТА...

НО У НАС ВСЕГО
100-500 RPS...

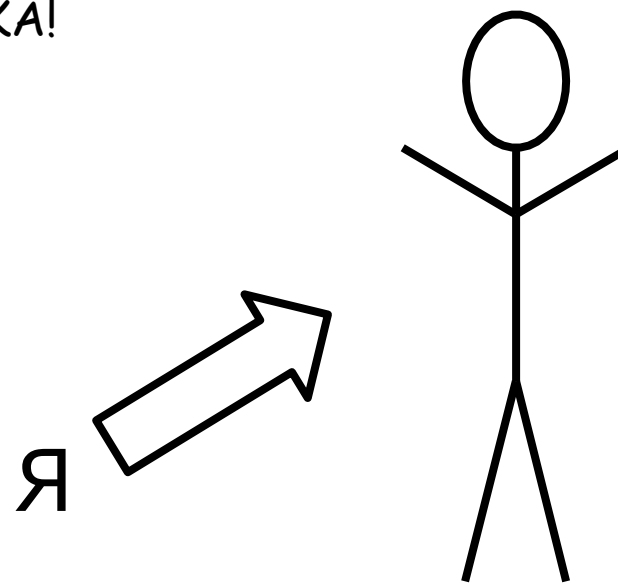
... И ДАННЫЕ В ОБЩЕМ-ТО
РЕЛЯЦИОННЫЕ...



НЕТ!
УЖЕ ЕСТЬ КАФКА!!!

ТОЛЬКО КАФКА!!!
И НЕЧЕГО ДУМАТЬ!!

КАФКА! КАФКА!
КАФКА!



Неочевидности в Kafka

Неочевидности в Kafka

... или что мы пережили за год эксплуатации

Неочевидности в Kafka

... или что мы пережили за год эксплуатации
(в очень кратком изложении)

Настройки – Как разломать кластер

Настройки – Как разломать кластер

log.dirs

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

unclean.leader.election.enable=false

Настройки – Как разломать кластер

log.dirs

replica 0

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

unclean.leader.election.enable=false

[KIP-106 - Change Default](#)

[unclean.leader.election.enabled](#) [from True to False](#)

(0.11)

Настройки – Как разломать кластер

log.dirs

replica 0

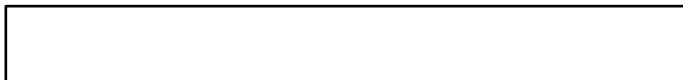
replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Настройки – Как разломать кластер

log.dirs

replica 0



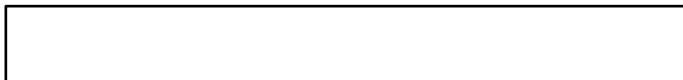
replica 1



Настройки – Как разломать кластер

log.dirs

replica 0



replica 1



Брокер упал 😞
😞

Настройки – Как разломать кластер

log.dirs

replica 0

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

<https://issues.apache.org/jira/browse/KAFKA-3410>

Настройки – Как разломать кластер

log.dirs

replica 0

replica 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

(исправлено в 1.1)

<https://issues.apache.org/jira/browse/KAFKA-3410>

Настройки – Настройки по умолчанию (1)

Настройки – Настройки по умолчанию (1)

- default.replication.factor = 1

Настройки – Настройки по умолчанию (1)

- default.replication.factor = 1
- auto.create.topics.enable = true

Настройки – Настройки по умолчанию (2)

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы
- `message.max.bytes`

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы
- `message.max.bytes` (Broker, 1_000_012)

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы
- message.max.bytes (Broker, 1_000_012)
- max.request.size

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы
- message.max.bytes (Broker, 1_000_012)
- max.request.size (Producer, 1_048_576)

Настройки – Настройки по умолчанию (2)

- Настройки Broker, Consumer и Producer должны быть согласованы
- message.max.bytes (Broker, 1_000_012)
- max.request.size (Producer, 1_048_576)
- max.partition.fetch.bytes (Consumer, 1_048_576)

Настройки – Умножение

- message.max.bytes (Broker, 1_000_012)
- max.request.size (Producer, 1_048_576)
- max.partition.fetch.bytes (Consumer, 1_048_576)

Настройки – Умножение

- message.max.bytes (Broker, 1_000_012)
- max.request.size (Producer, 1_048_576)
- max.partition.fetch.bytes (Consumer, 1_048_576)
- batch.size (Producer, 16_384)

Настройки – Умножение

- message.max.bytes (Broker, 1_000_012)
- max.request.size (Producer, 1_048_576)
- max.partition.fetch.bytes (Consumer, 1_048_576)
- batch.size (Producer, 16_384)
- [KIP-126 - Allow KafkaProducer to split and resend oversized batches](#) (0.11)

API – Блокирующий
send

API – Блокирующий send

- Если мета-данные не доступны –
producer.send()
блокируется

API – Блокирующий send

- Если мета-данные не доступны –
producer.send()
блокируется
- max.block.ms = 60_000

API – Блокирующий send

- Если мета-данные не доступны – `producer.send()` блокируется
- `max.block.ms = 60_000`
- [KIP-286: producer.send\(\) should not block on metadata update](#) (discuss)

API – Бесконечная десериализация

API – Бесконечная десериализация

```
while(true) {  
    ConsumerRecords<Key, Event> records  
        = consumer.poll(1_000);  
    for (var record : records) {  
        /*do something */  
    }  
}
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
consumption.", e);  
}
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
        consumption.", e);  
}
```

```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
consumption.", e);  
}
```

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

```
consumer.seek(partition, offset);
```


API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
        consumption.", e);  
}
```

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
        consumption.", e);  
}
```

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
        consumption.", e);  
}
```



```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
consumption.", e);  
}
```



```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

```
try {  
    /* parsing */  
} catch (RuntimeException e) {  
    throw new SerializationException(  
        "Error deserializing key/value for partition " + partition +  
        " at offset " + record.offset() +  
        ". If needed, please seek past the record to continue  
consumption.", e);  
}
```



```
consumer.seek(partition, offset);
```

API – Бесконечная десериализация

Наш выбор: кастомный десериализатор, который
вернёт null
в случае ошибки

API – Бесконечная

десериализация

Наш выбор: кастомный десериализатор, который вернёт null

в случае ошибки

```
while(true) {  
    ConsumerRecords<Key, Event> records  
        = consumer.poll(1_000);  
    for (var record : records) {  
        if (record.getValue() == null) { continue; }  
        /*do something */  
    }  
}
```

API – Бесконечная

десериализация

Наш выбор: кастомный десериализатор, который вернёт null

в случае ошибки

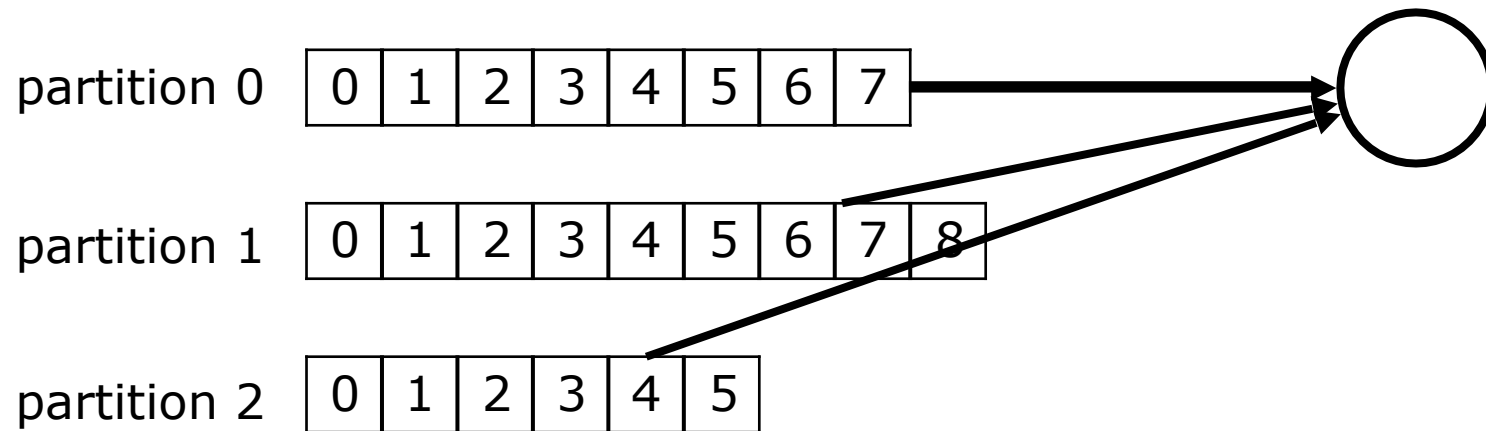
```
while(true) {  
    ConsumerRecords<Key, Event> records  
        = consumer.poll(1 000);  
    for (var record : records) {  
        if (record.getValue() == null) { continue; }  
        /*do something */  
    }  
}
```


API – Нечестное
чтение

API – НЕчестное

Чтение

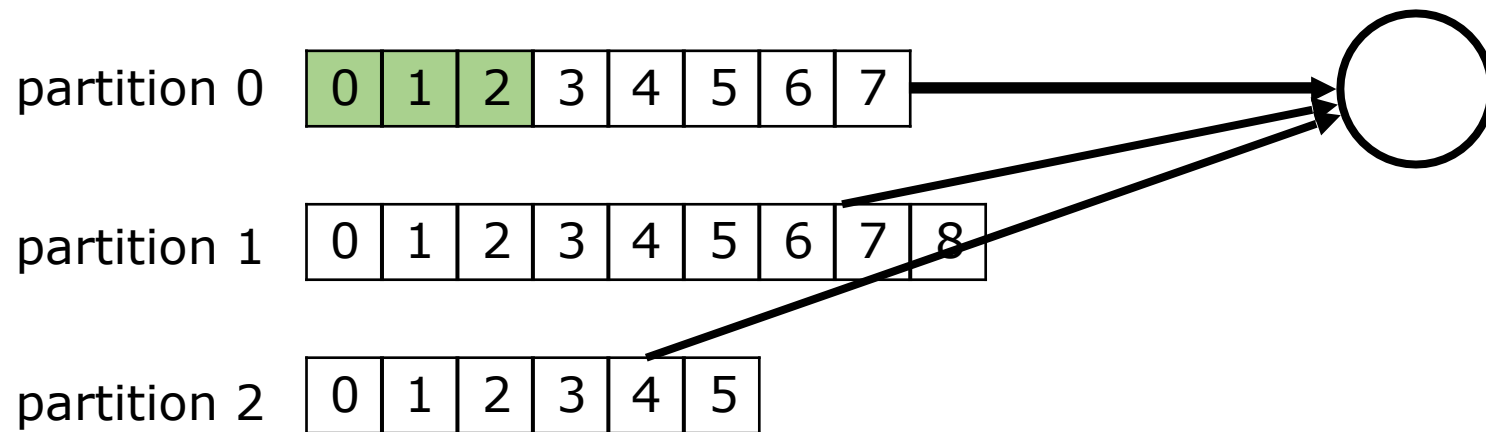
Чтение по 3 сообщения



API – НЕчестное

Чтение

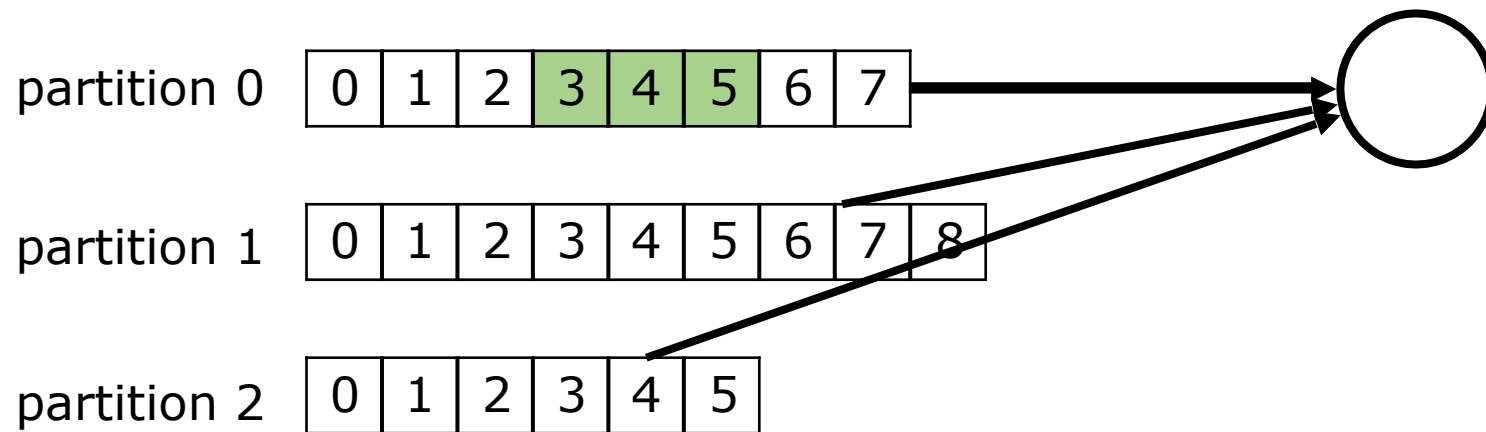
Чтение по 3 сообщения



API – НЕчестное

Чтение

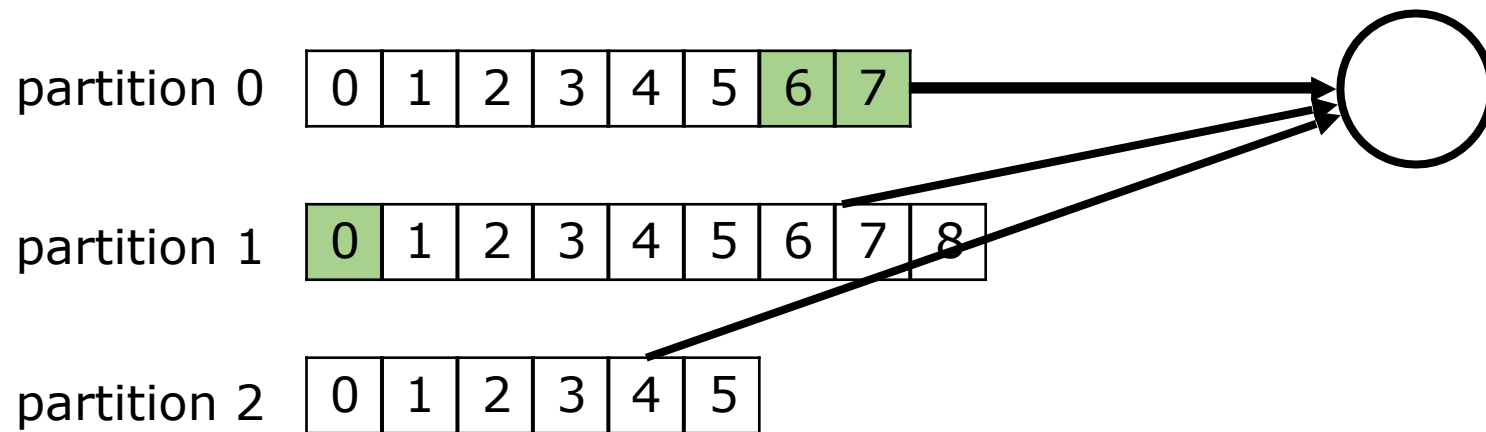
Чтение по 3 сообщения



API – НЕчестное

Чтение

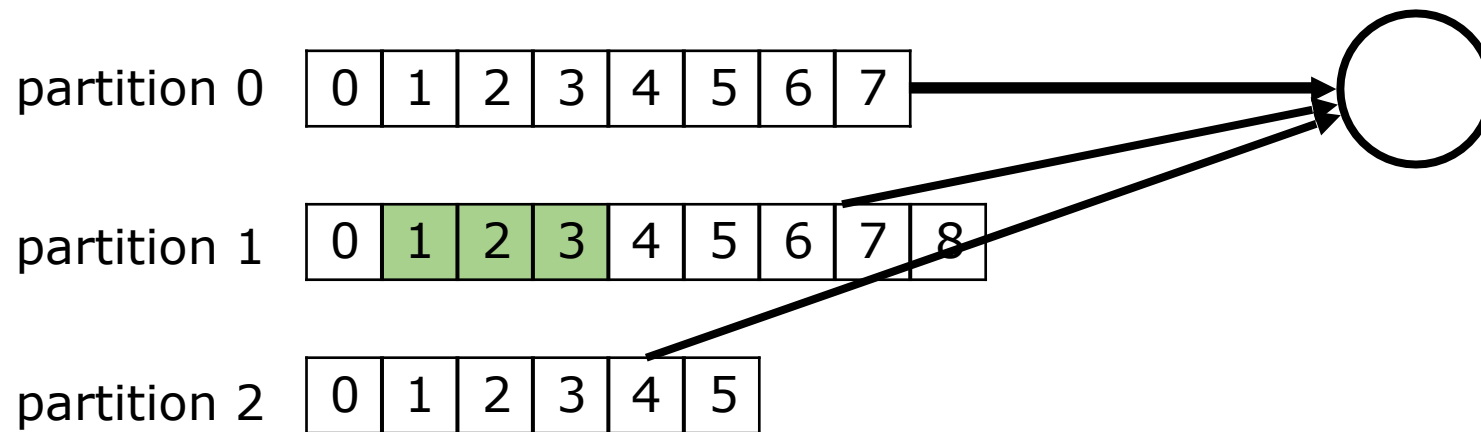
Чтение по 3 сообщения



API – НЕчестное

Чтение

Чтение по 3 сообщения



API – НЕчестное

ЧТЕНИЕ

- [KIP-41: KafkaConsumer Max Records](#)
(0.10)

API – НЕчестное

ЧТЕНИЕ

- [KIP-41: KafkaConsumer Max Records \(0.10\)](#)
- Жадный round-robin

API – НЕчестное

ЧТЕНИЕ

- [KIP-41: KafkaConsumer Max Records](#) (0.10)
- Жадный round-robin
- [KIP-387: Fair Message Consumption Across Partitions in KafkaConsumer](#) (discuss)

Рутина – Ограничение на размер
топика

Рутина – Ограничение на размер ТОПИКА

- log.retention.bytes (Broker,
unlimited)

Рутина – Ограничение на размер ТОПИКА

- log.retention.bytes (Broker, unlimited)
- retention.bytes (Topic)

Рутина – Ограничение на размер ТОПИКА

- log.retention.bytes (Broker, unlimited)
- retention.bytes (Topic) – per partition

Рутина – Место на диске

Рутина – Место на диске

- Нет автораспределения 파티ций по новым дискам

Рутина – Место на диске

- Нет автораспределения партиций по новым дискам
- [KIP-113: Support replicas movement between log directories](#) (1.1)

Рутина – Место на диске

- Нет автораспределения партиций по новым дискам
- [KIP-113: Support replicas movement between log directories](#) (1.1)
- Равномерное распределение партиций **по количеству**

Рутина – Место на диске

- Нет автораспределения партиций по новым дискам
- [KIP-113: Support replicas movement between log directories](#) (1.1)
- Равномерное распределение партиций **по количеству**
- [KIP-178: Size-based log directory selection strategy](#) (discuss)

Рутина – Новый Брокер

Рутина – Новый

Брокер

- Нет автораспределения партиций на нового Брокера

Рутина – Новый

Брокер

- Нет автораспределения партиций на нового Брокера
- Руками делать partition reassignment

Рутина – Новый Брокер

- Нет автораспределения партиций на нового Брокера
- Руками делать partition reassignment

```
{ "version": 1,  
  "partitions": [  
    { "topic": "topic", "partition" : partition,  
      "replicas": [brokerIds]  
    },  
    ...  
  ]  
}
```

Рутина – Новый Брокер

- Нет автораспределения партиций на нового Брокера
- Руками делать partition reassignment

```
{ "version": 1,  
  "partitions": [  
    { "topic": "topic", "partition" : partition,  
      "replicas": [brokerIds]  
    },  
    ...  
  ]  
}
```

Рутина – Новый Брокер

- Нет автораспределения партиций на нового Брокера
- Руками делать partition reassignment

```
{ "version": 1,  
  "partitions": [  
    { "topic": "hg2tg", "partition" : partition,  
      "replicas": [brokerIds]  
    },  
    ...  
  ]  
}
```


Рутина – Новый Брокер

- Нет автораспределения партиций на нового Брокера
- Руками делать partition reassignment

```
{ "version": 1,  
  "partitions": [  
    { "topic": "hg2tg", "partition" : 42,  
      "replicas": [brokerIds]  
    },  
  ],  
}
```

Рутина – Новый Брокер

```
{ "version": 1,  
  "partitions": [  
    { "topic": "hg2tg", "partition" : 42,  
      "replicas": [1, 2, 3]  
    },  
  ],  
}
```

Рутина – Новый Брокер

- Preferred leader – первый брокер в списке реплик

```
{ "version": 1,  
  "partitions": [  
    { "topic": "hg2tg", "partition" : 42,  
      "replicas": [1, 2, 3]  
    },  
    ...  
  ]
```

Выводы

ВЫВОДЫ

- Внимательное отношение к **настройкам**

Выводы

- Внимательное отношение к **настройкам**
- Особенности (недоработки?) клиентского **API**

Выводы

- Внимательное отношение к **настройкам**
- Особенности (недоработки?) клиентского **API**
- Большое количество **рутины**

Выводы

- Внимательное отношение к **настройкам**
- Особенности (недоработки?) клиентского **API**
- Большое количество **рутины**
- Документация о многом умалчивает

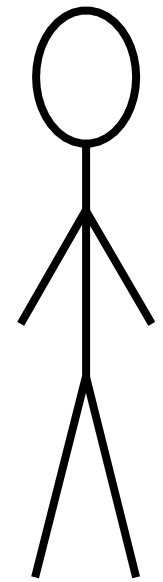
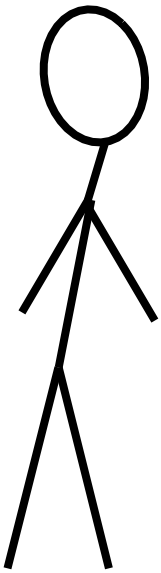
Выводы

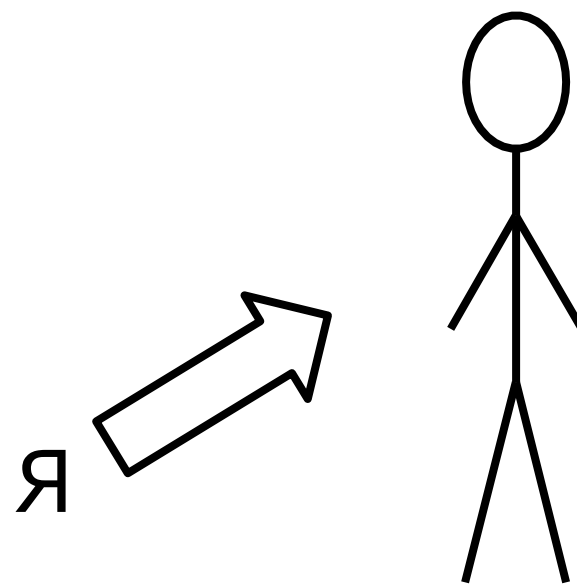
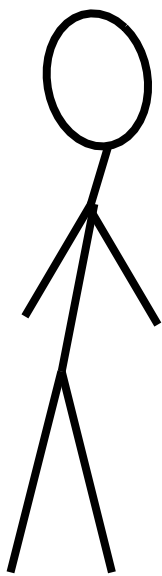
- Внимательное отношение к **настройкам**
- Особенности (недоработки?) клиентского **API**
- Большое количество **рутины**
- Документация о многом умалчивает
- Kafka – лучшее, что есть...

Выводы

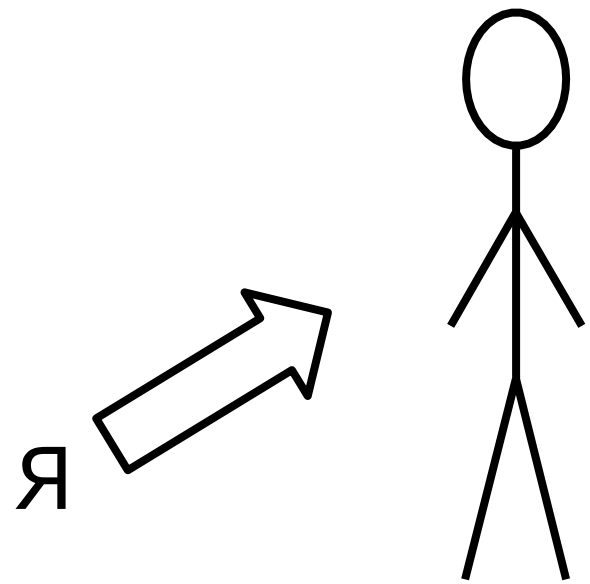
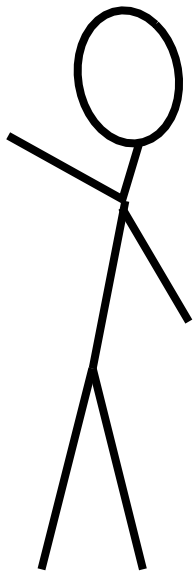
- Внимательное отношение к **настройкам**
- Особенности (недоработки?) клиентского **API**
- Большое количество **рутины**
- Документация о многом умалчивает
- Kafka – лучшее, что есть... И она классная

* В СКОРОМ ВРЕМЕНИ *

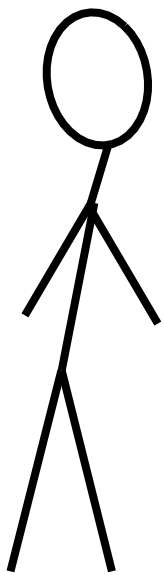




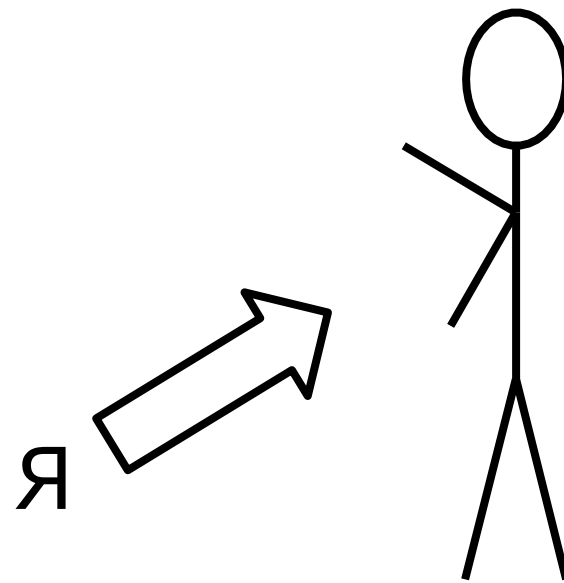
НОВЫЙ ПРОЕКТ НАМЕЧАЕТСЯ,
ДУМАЕМ НАД ВЫБОРОМ
ТЕХНОЛОГИЙ...



НОВЫЙ ПРОЕКТ НАМЕЧАЕТСЯ,
ДУМАЕМ НАД ВЫБОРОМ
ТЕХНОЛОГИЙ...

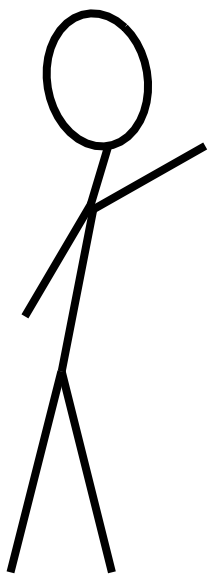


МНОГО ЖЕ ВСЕГО!
ДАВАЙ ДЕТАЛИ!

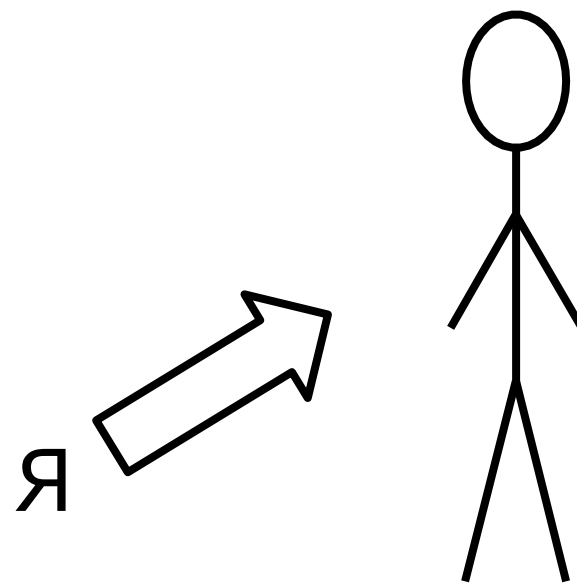


НОВЫЙ ПРОЕКТ НАМЕЧАЕТСЯ,
ДУМАЕМ НАД ВЫБОРОМ
ТЕХНОЛОГИЙ...

ОБЕЩАЮТ ХАЙЛОАД,
БИГ-ДАТУ,
ВОТ ЭТО ВСЁ!

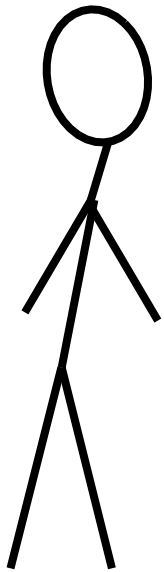


МНОГО ЖЕ ВСЕГО!
ДАВАЙ ДЕТАЛИ!



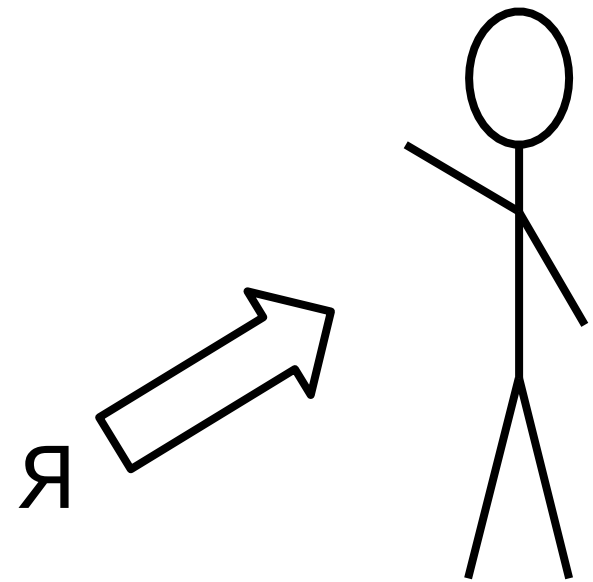
НОВЫЙ ПРОЕКТ НАМЕЧАЕТСЯ,
ДУМАЕМ НАД ВЫБОРОМ
ТЕХНОЛОГИЙ...

ОБЕЩАЮТ ХАЙЛОАД,
БИГ-ДАТУ,
ВОТ ЭТО ВСЁ!



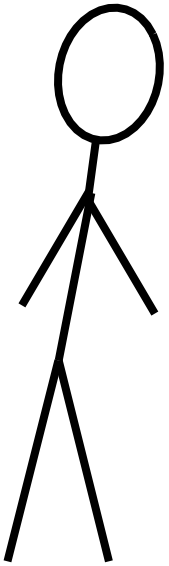
МНОГО ЖЕ ВСЕГО!
ДАВАЙ ДЕТАЛИ!

А-А-А!
К ЧЁРТУ ВСЁ!!!



НОВЫЙ ПРОЕКТ НАМЕЧАЕТСЯ,
ДУМАЕМ НАД ВЫБОРОМ
ТЕХНОЛОГИЙ...

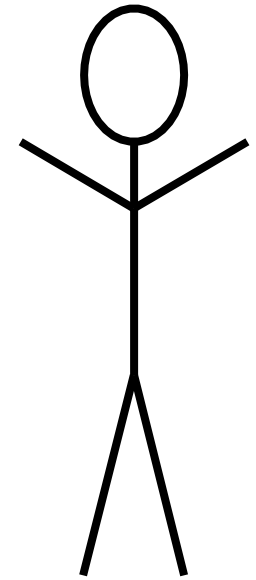
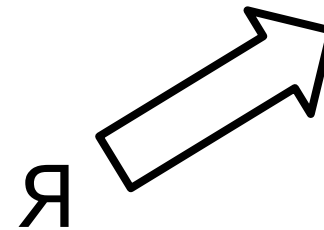
ОБЕЩАЮТ ХАЙЛОАД,
БИГ-ДАТУ,
ВОТ ЭТО ВСЁ!



МНОГО ЖЕ ВСЕГО!
ДАВАЙ ДЕТАЛИ!

А-А-А!
К ЧЁРТУ ВСЁ!!!

КАФКА! КАФКА!
КАФКА!



Консистентность и доступность

Консистентность и доступность

replica 0

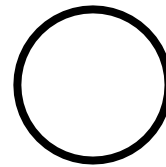
0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---



Консистентность и доступность

replica 0

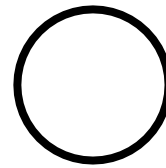
0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

replica 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

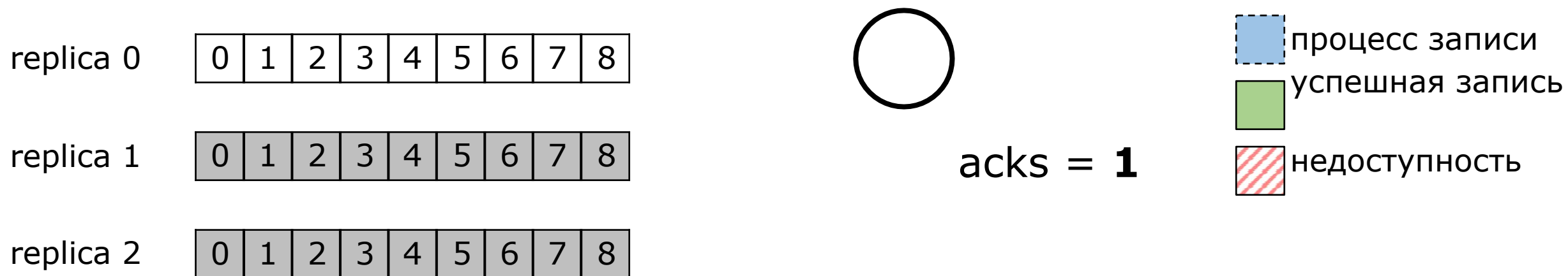
replica 2

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

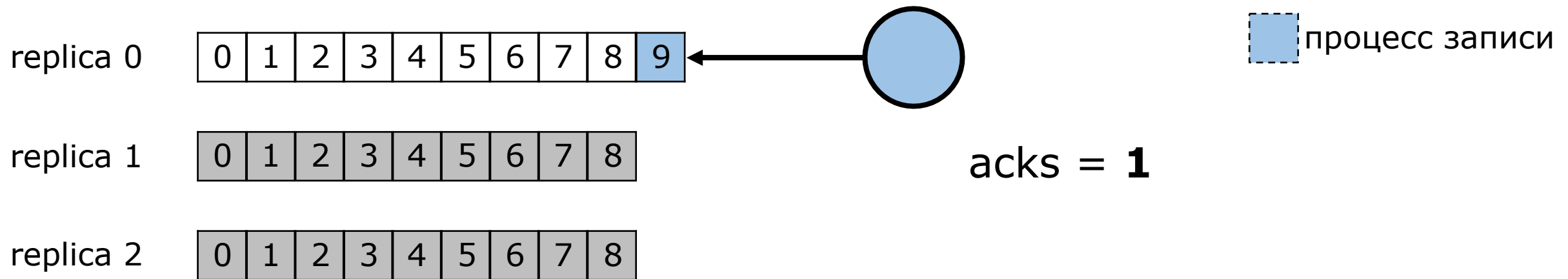


acks = **1**

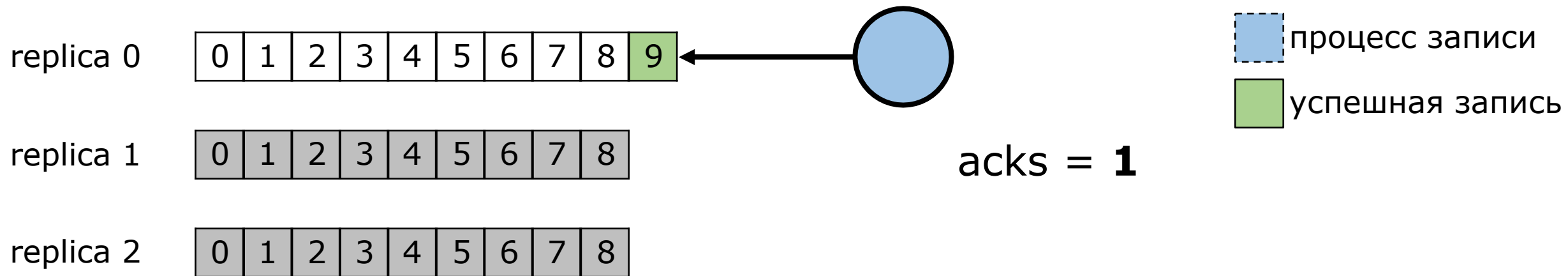
Консистентность и доступность



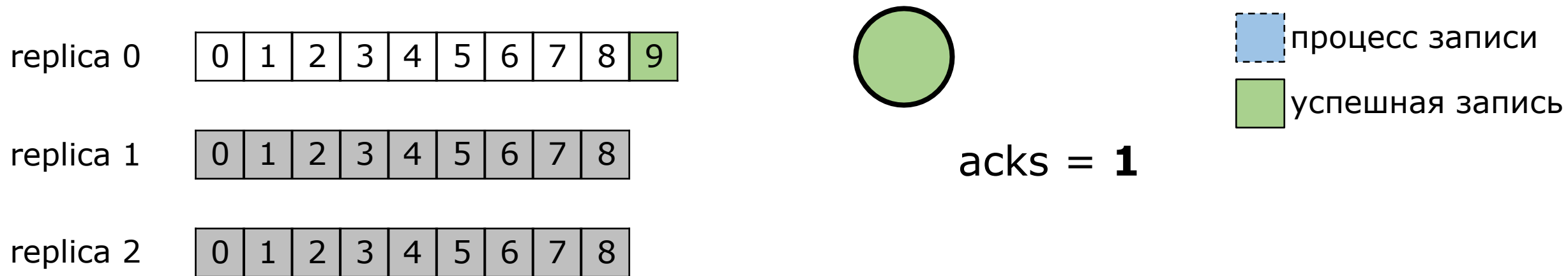
Консистентность и доступность



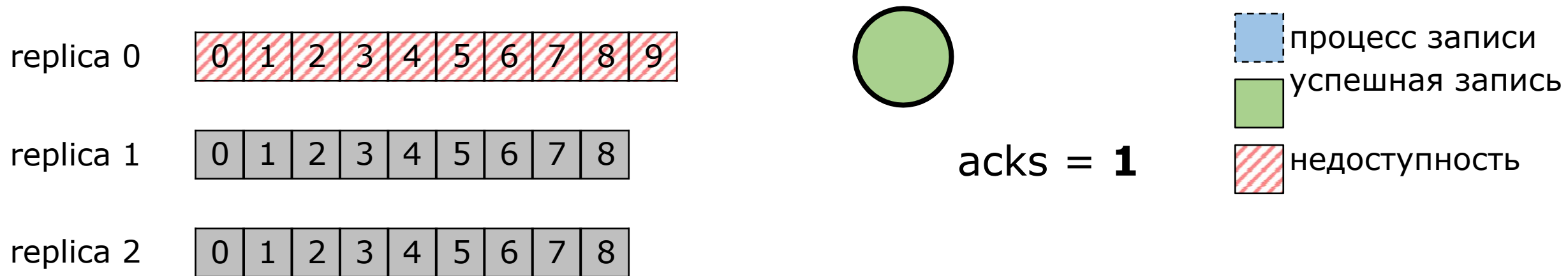
Консистентность и доступность



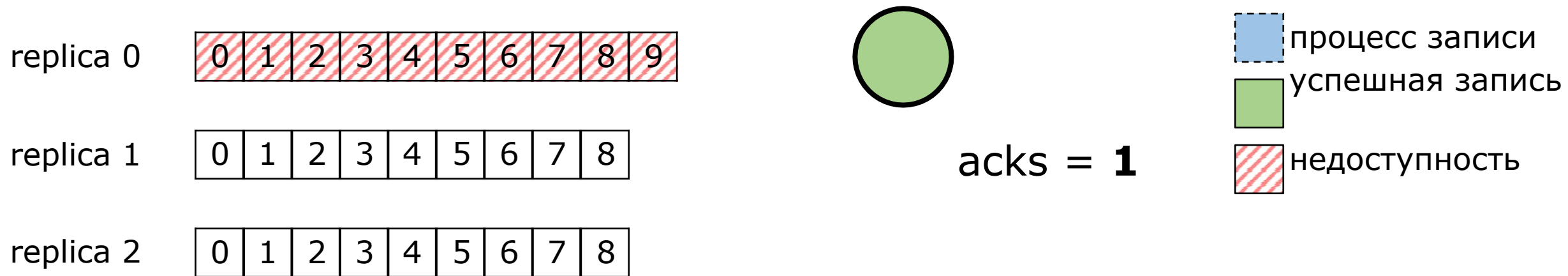
Консистентность и доступность



Консистентность и доступность

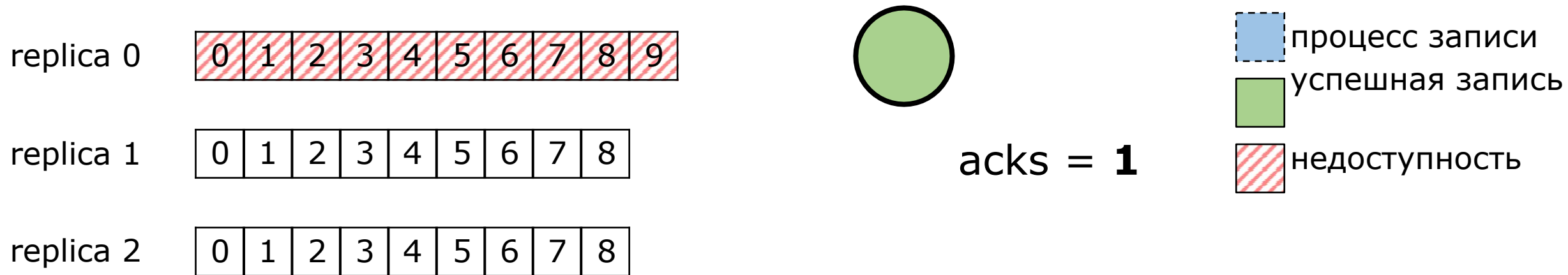


Консистентность и доступность



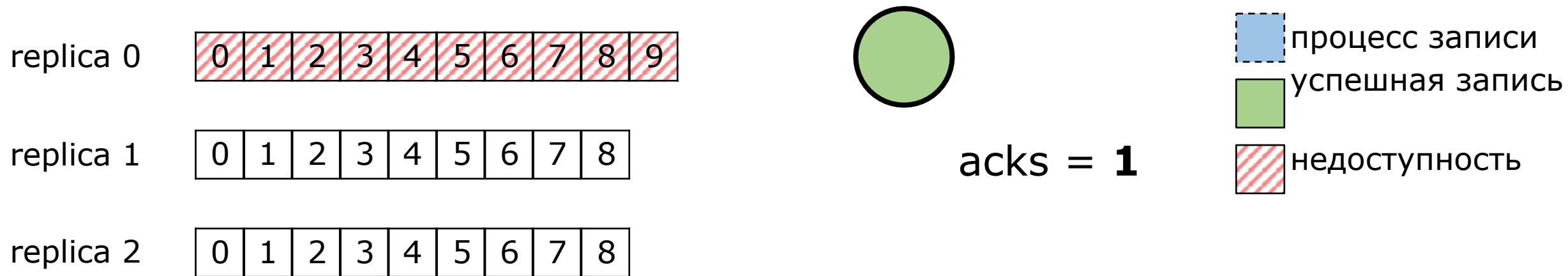
Консистентность и доступность

unclean leader election

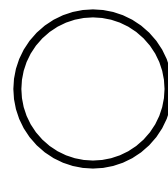
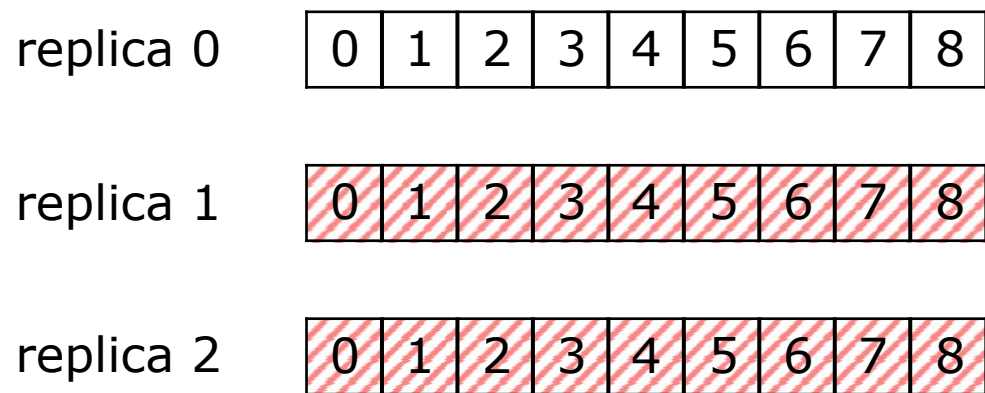


Консистентность и доступность

unclean leader election (но это не точно)



Консистентность и доступность



acks = **all**

min.insync.replicas = 3



процесс записи

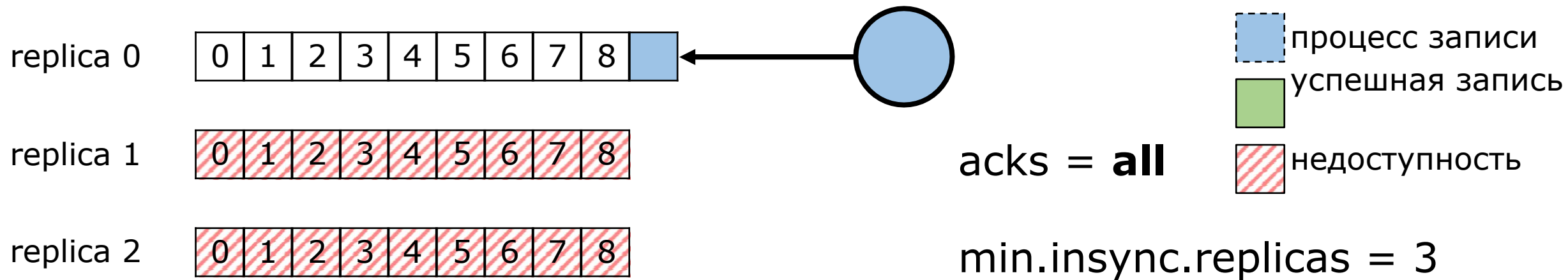


успешная запись

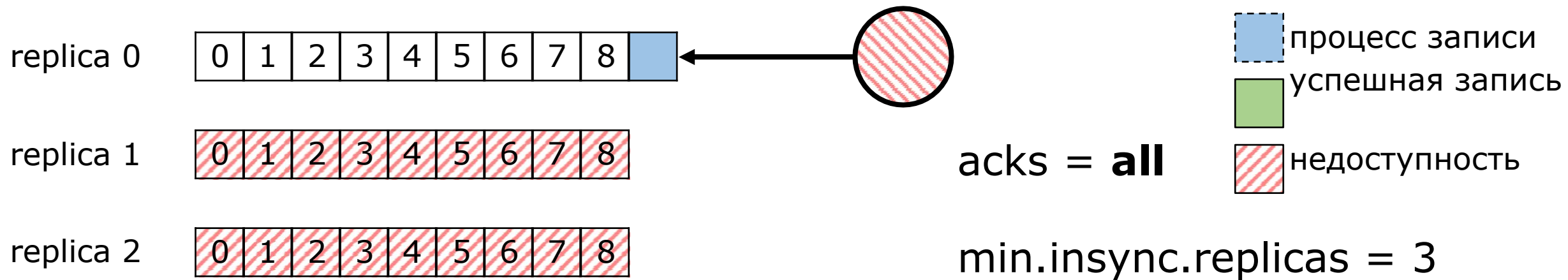


недоступность

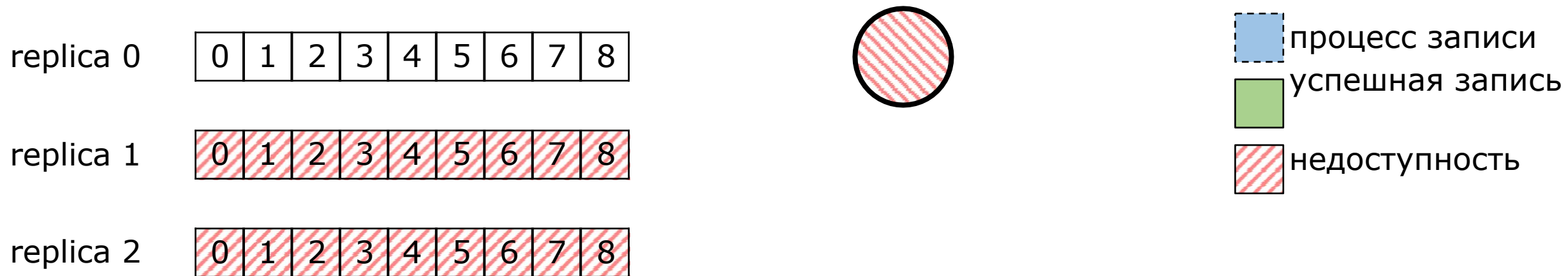
Консистентность и доступность



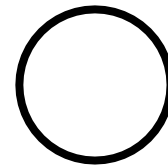
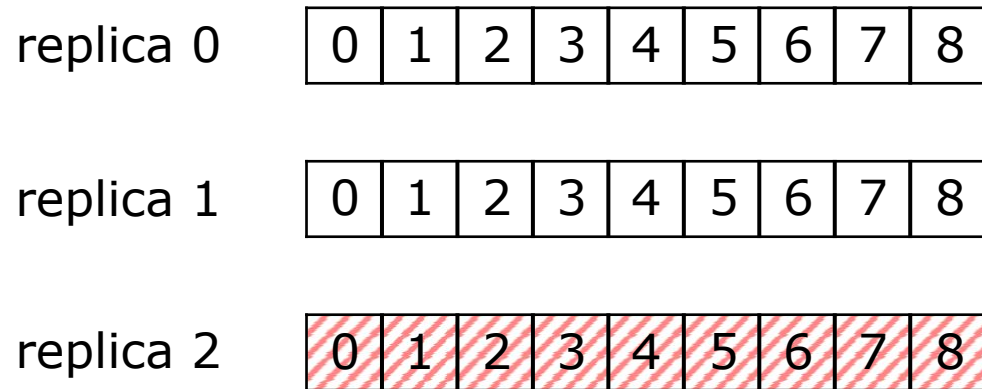
Консистентность и доступность



Консистентность и доступность



Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи

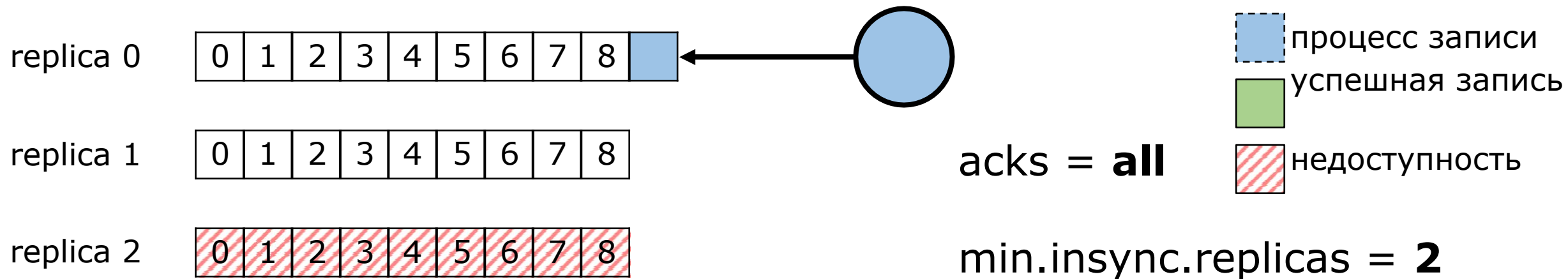


успешная запись

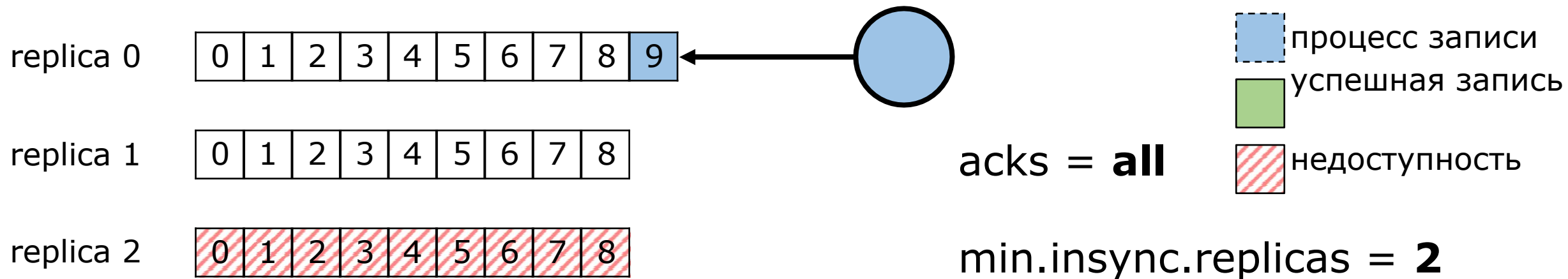


недоступность

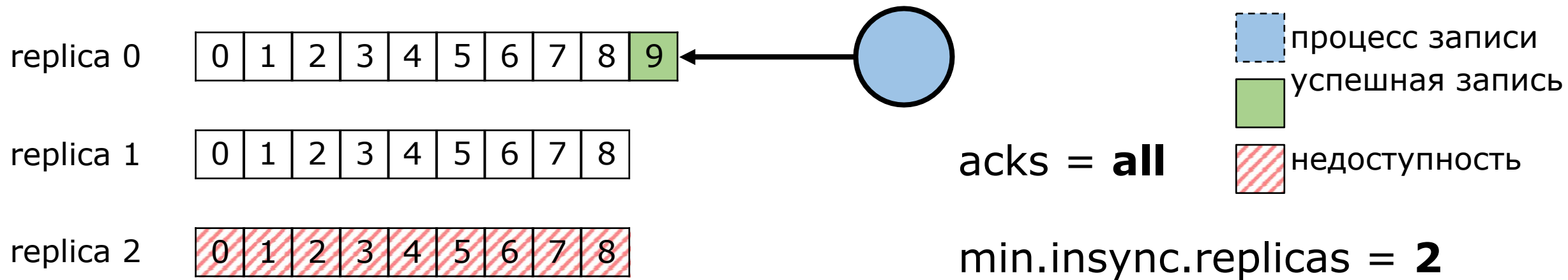
Консистентность и доступность



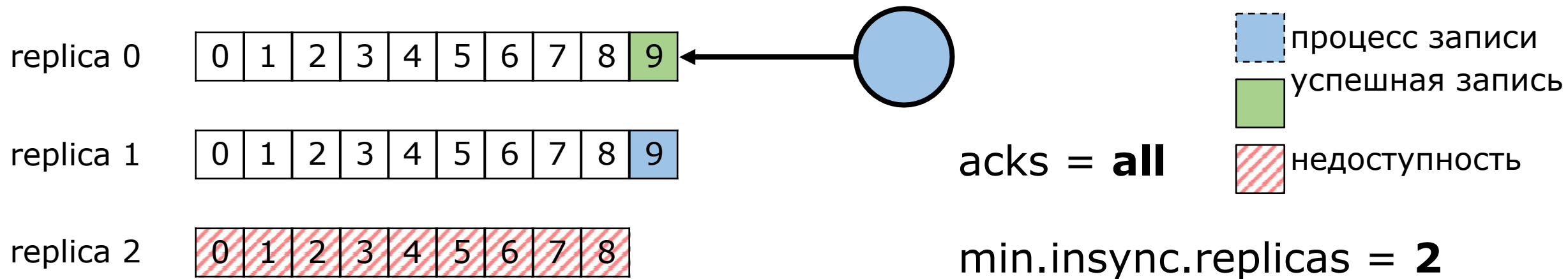
Консистентность и доступность



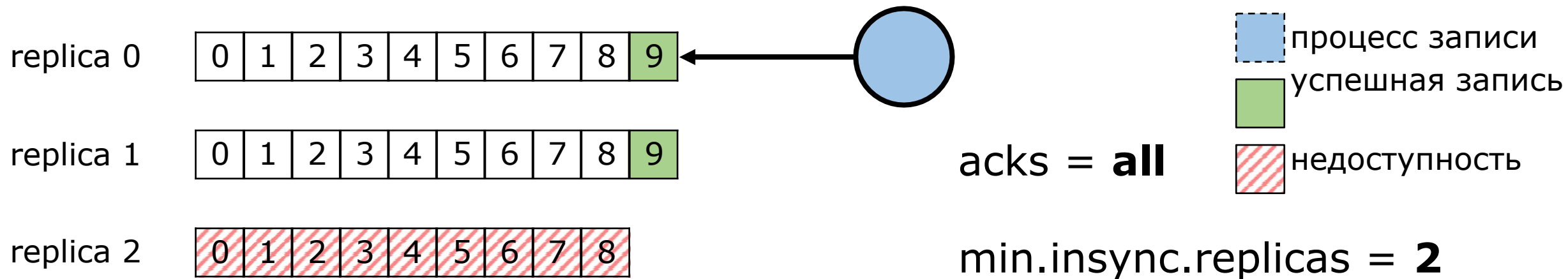
Консистентность и доступность



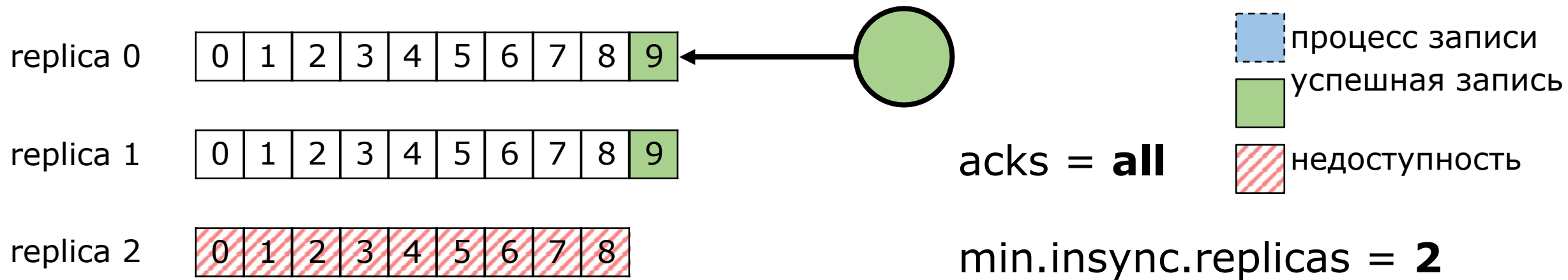
Консистентность и доступность



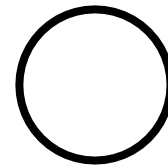
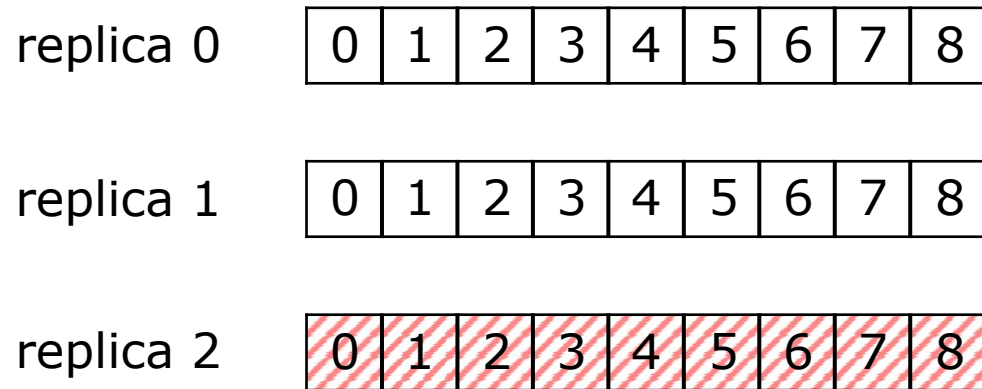
Консистентность и доступность



Консистентность и доступность



Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи

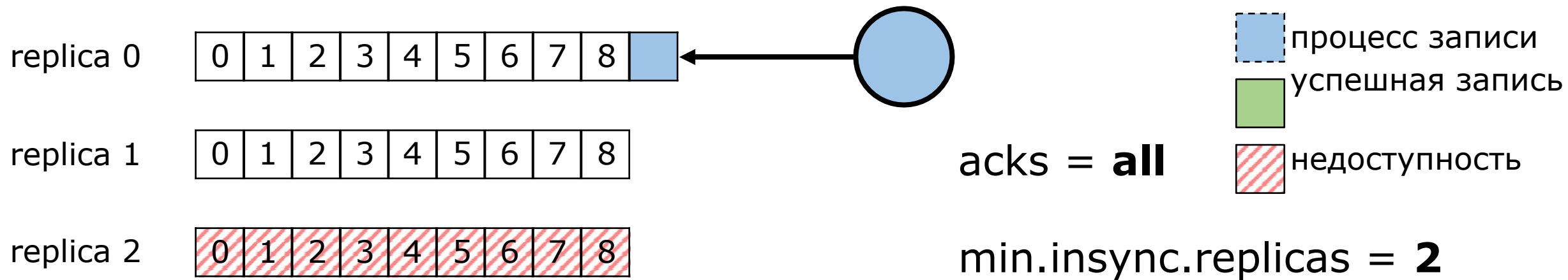


успешная запись

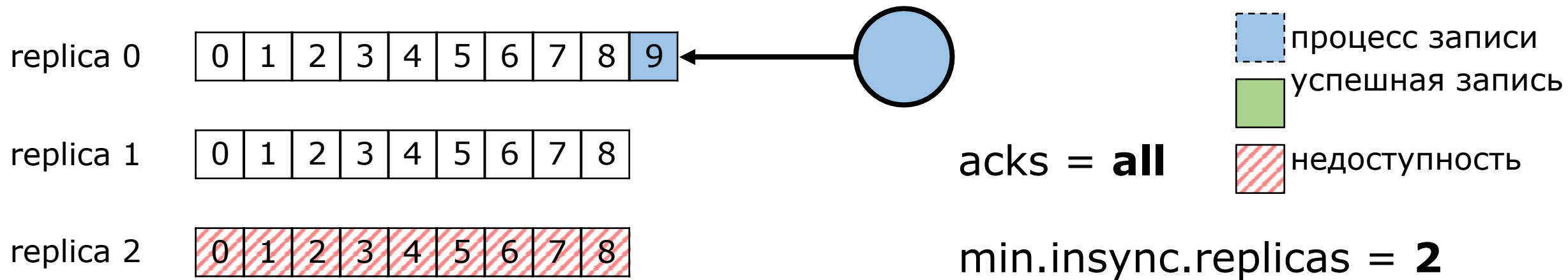


недоступность

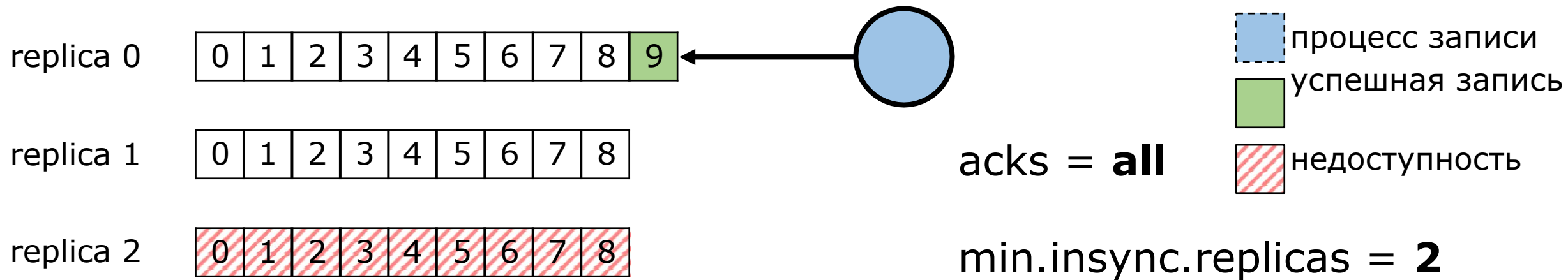
Консистентность и доступность



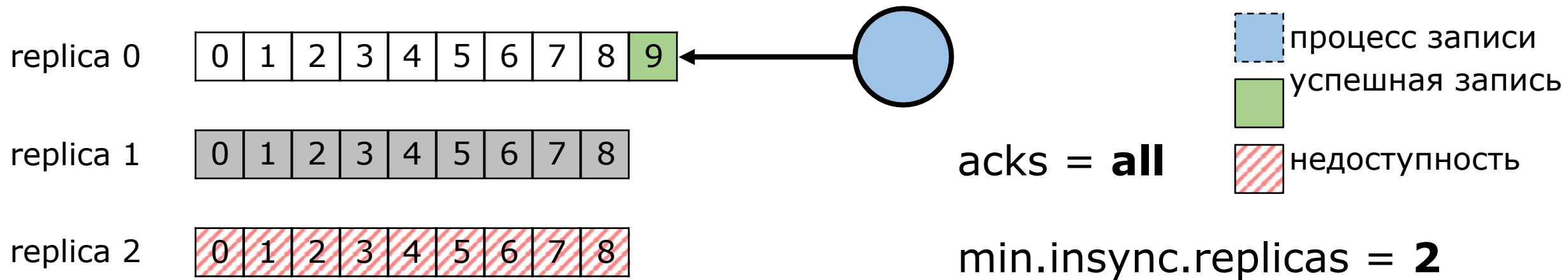
Консистентность и доступность



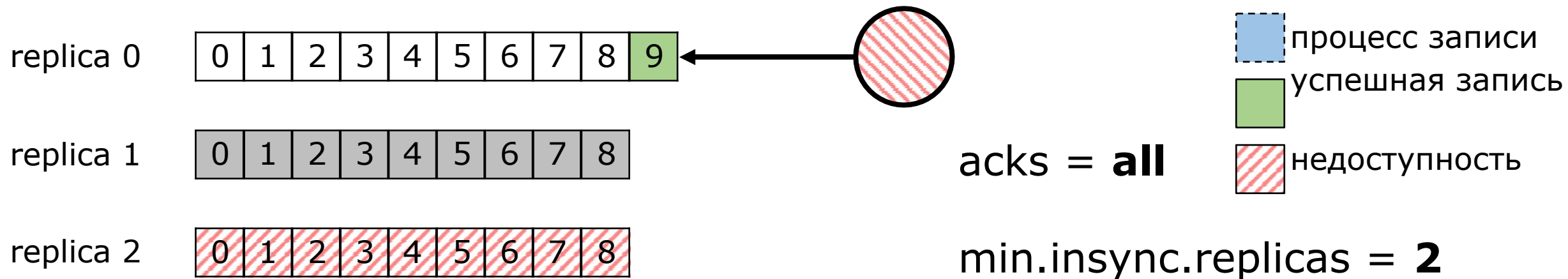
Консистентность и доступность



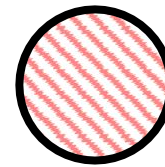
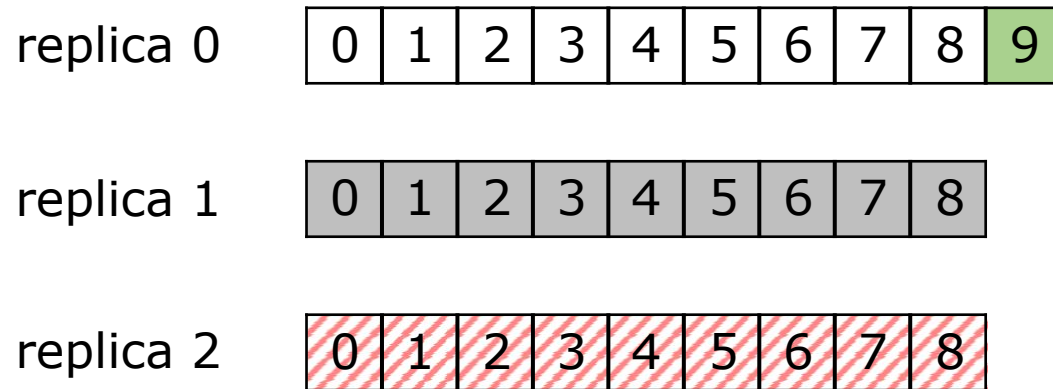
Консистентность и доступность



Консистентность и доступность



Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи

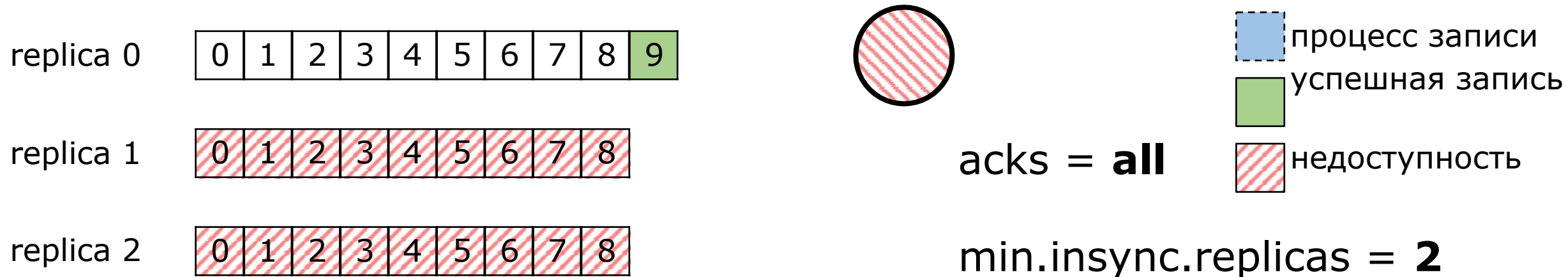


успешная запись

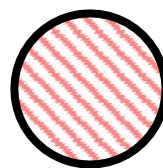
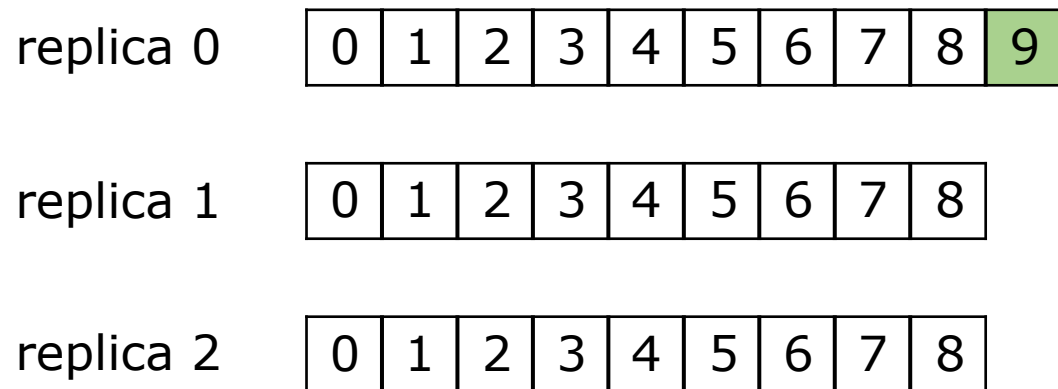


недоступность

Консистентность и доступность



Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи

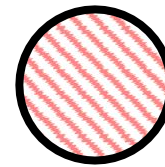
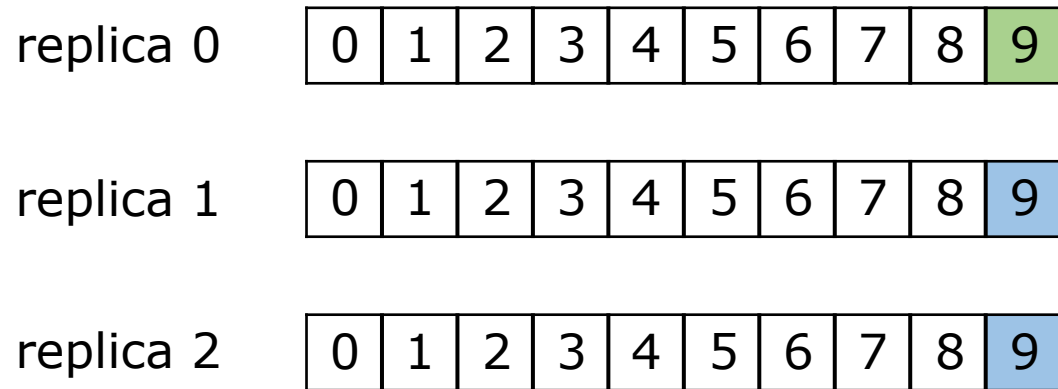


успешная запись



недоступность

Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи

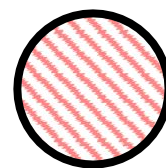
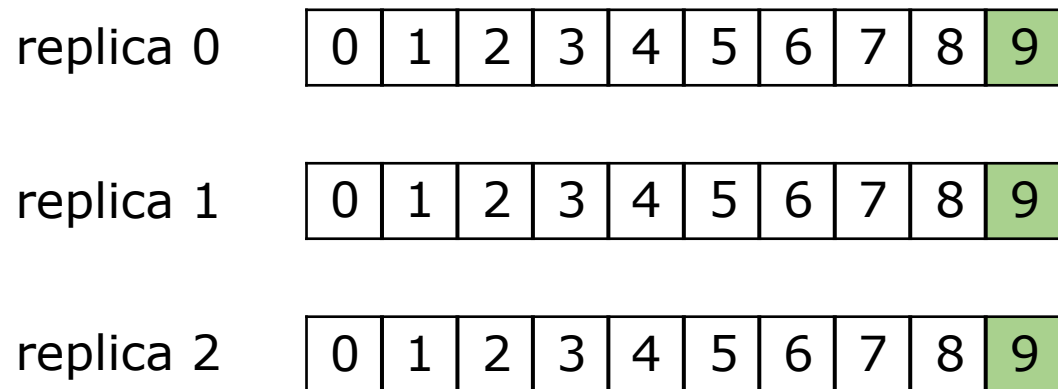


успешная запись



недоступность

Консистентность и доступность



acks = **all**

min.insync.replicas = **2**



процесс записи



успешная запись



недоступность