



АРАСНЕ КАФКА

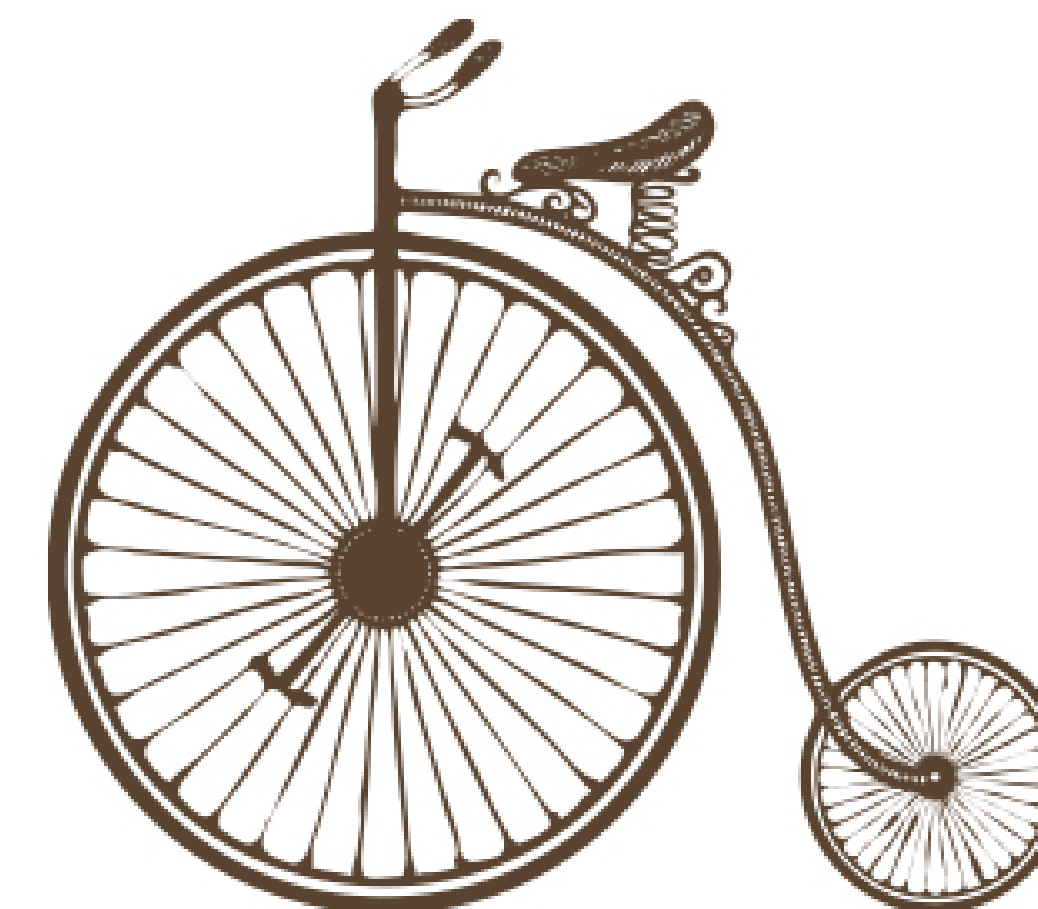
ЧТО ЭТО И КАК ОНА ИЗМЕНИТ АРХИТЕКТУРУ

★★★ вашего приложения ★★★

Иван Пономарёв, КУРС/МФТИ

ponomarev@corchestra.ru

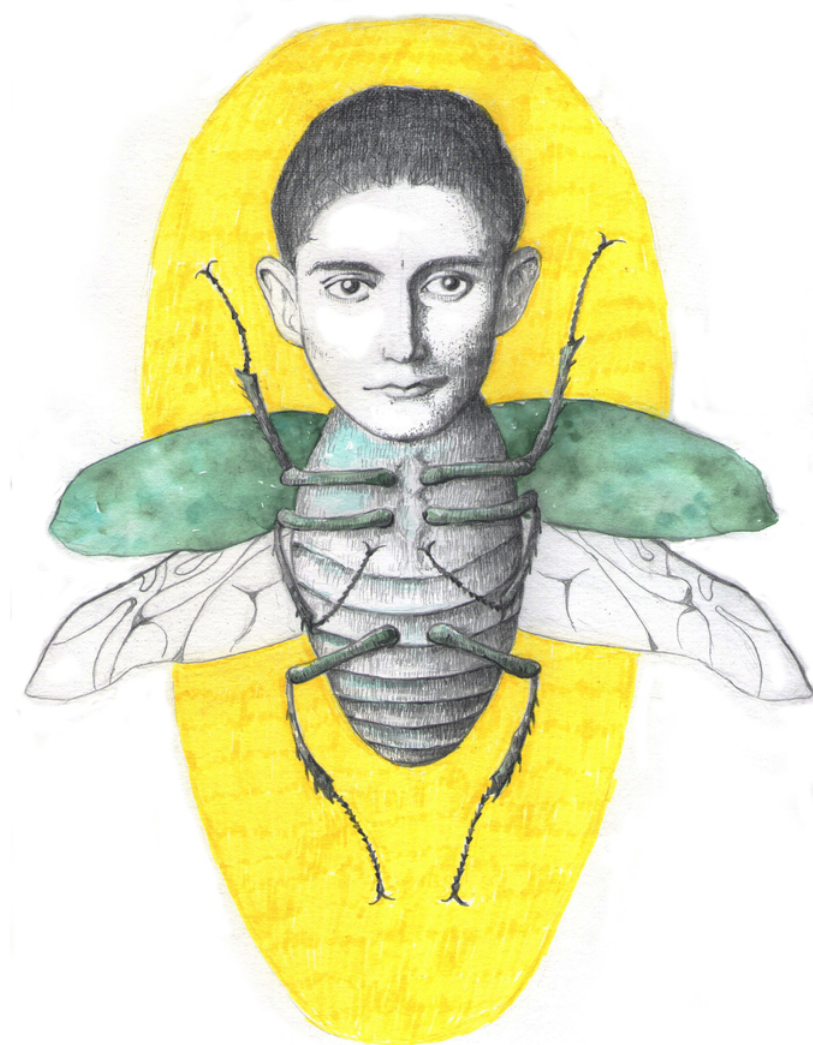
 [@inponomarev](https://twitter.com/inponomarev)



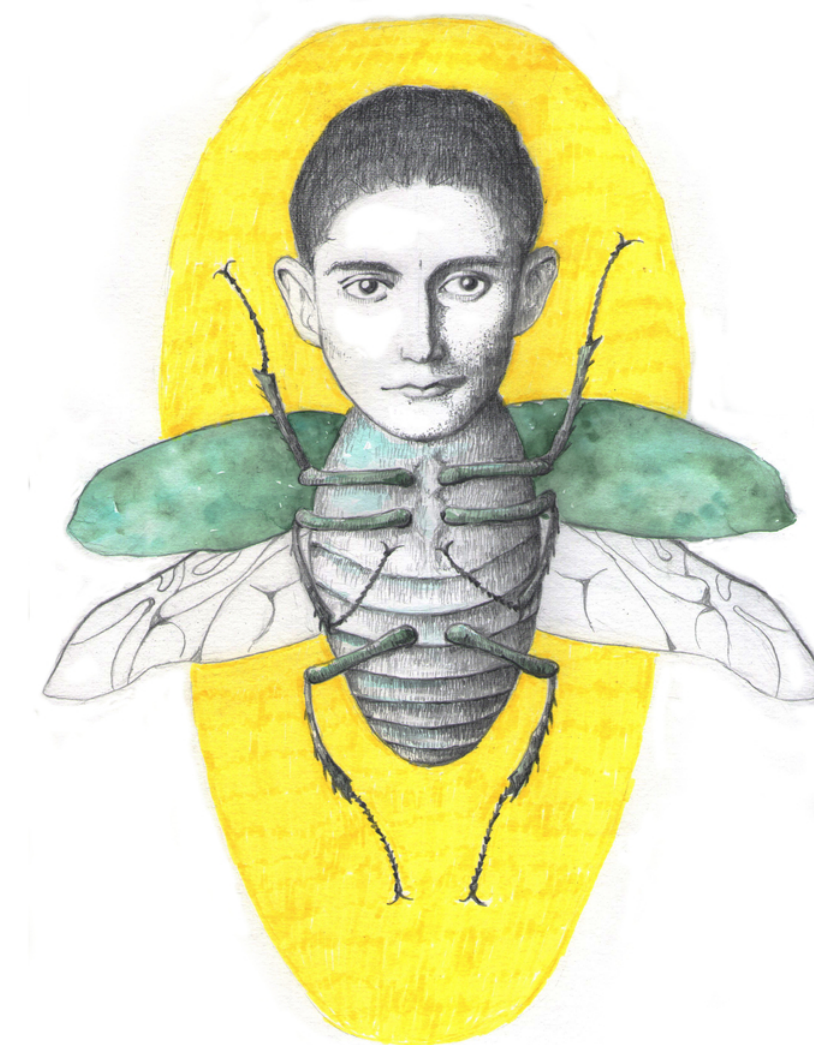


- ★ Tech Lead at KURS
 - ★ ERP systems & Java background
 - ★ Speaker at JPoint, Devoops, Heisenbug, JUG.MSK, etc.
 - ★ Twitter @inponomarev
- ————— ◆

Наш план



1. **Что такое лог и почему это важно**
2. Что такое Kafka и что она умеет
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



Что такое лог?



Что такое лог?



★ Добавляем данные в конец

Что такое лог?



- ★ Добавляем данные в конец
- ★ Записанное не можем изменить

Что такое лог?



- ★ Добавляем данные в конец
- ★ Записанное не можем изменить
- ★ Читаем последовательно

– **Что наша жизнь?**

– Что наша жизнь?



– Лог!

Автоматизировать склад. Как?

Автоматизировать склад. Как?

ТОВАР	ЯЧЕЙКА	КОЛ-ВО
X	A	8
X	B	2
Y	B	1

Всё хорошо, пока...



Что это может быть?

Что это может быть?

- ★ Украла (разбили, выкинули и т. п.)

Что это может быть?

- ★ Украли (разбили, выкинули и т. п.)
- ★ Потеряли: лежит где-то рядом

Что это может быть?

- ★ Украли (разбили, выкинули и т. п.)
- ★ Потеряли: лежит где-то рядом
- ★ **ЭТО ВАША ТУПАЯ ПРОГРАММА НЕ РАБОТАЕТ**



Что будем делать?



Как на самом деле надо проектировать?



★ «База данных — это склад»

Как на самом деле надо проектировать?



★ «База данных — это склад»



- ★ База данных отражает процессы.
- ★ Текущее состояние — результат выполнения процессов

Warehouse Ledger

ДАТА	ТОВАР	ЯЧЕЙКА	КОЛ-ВО	ОПИСАНИЕ
02.04.2020	X	A	10	начальный остаток
02.04.2020	X	B	2	начальный остаток
02.04.2020	Y	B	1	начальный остаток

Warehouse Ledger

ДАТА	ТОВАР	ЯЧЕЙКА	КОЛ-ВО	ОПИСАНИЕ
02.04.2020	X	A	10	начальный остаток
02.04.2020	X	B	2	начальный остаток
02.04.2020	Y	B	1	начальный остаток
09.04.2020	X	A	-2	Василий по заданию №1

Warehouse Ledger

ДАТА	ТОВАР	ЯЧЕЙКА	КОЛ-ВО	ОПИСАНИЕ
02.04.2020	X	A	10	начальный остаток
02.04.2020	X	B	2	начальный остаток
02.04.2020	Y	B	1	начальный остаток
09.04.2020	X	A	-2	Василий по заданию №1
09.04.2020	X	B	2	Василий по заданию №1

А если мы ошиблись с учётом?

ДАТА	ТОВАР	ЯЧЕЙКА	КОЛ-ВО	ОПИСАНИЕ
9.04.2020	X	A	-2	Василий по заданию №1
9.04.2020	X	B	2	Василий по заданию №1
9.04.2020	X	A	2	Сторно задания №1
9.04.2020	X	B	-2	Сторно задания №1

**На какие вопросы уже можно
ответить?**

На какие вопросы уже можно ответить?

- ★ Сколько у нас всего на складе товара Y?

На какие вопросы уже можно ответить?

- ★ Сколько у нас всего на складе товара Y?
- ★ Что лежит в ячейке B?

На какие вопросы уже можно ответить?

- ★ Сколько у нас всего на складе товара Y?
- ★ Что лежит в ячейке B?
- ★ Сколько товаров переместил Василий 9 апреля?

На какие вопросы уже можно ответить?

- ★ Сколько у нас всего на складе товара Y?
- ★ Что лежит в ячейке B?
- ★ Сколько товаров переместил Василий 9 апреля?
- ★ Какие корректировки вносились в систему?

Расследуем инцидент

- ★ 9 апреля Василий должен был положить товар из А в В.
- ★ Давайте посмотрим, что лежит в А?
- ★ Давайте спросим Василия?

**На какие вопросы можно будет легко
ответить?**

На какие вопросы можно будет легко ответить?

- ★ Нагрузка на полку лимитирована 100 кг

На какие вопросы можно будет легко ответить?

- ★ Нагрузка на полку лимитирована 100 кг
- ★ — Добавляем поле "вес" в Ledger!

На какие вопросы можно будет легко ответить?

- ★ Нагрузка на полку лимитирована 100 кг
- ★ — Добавляем поле "вес" в Ledger!
- ★ Нужно посчитать зарплату работников склада

На какие вопросы можно будет легко ответить?

- ★ Нагрузка на полку лимитирована 100 кг
- ★ — Добавляем поле "вес" в Ledger!
- ★ Нужно посчитать зарплату работников склада
- ★ — Даже добавлять ничего не надо.

Архитектура решения: лог не работает в одиночку



Предварительный итог

Наличие лога позволяет

Предварительный итог

Наличие лога позволяет

- ★ Добавлять новую функциональность

Предварительный итог

Наличие лога позволяет

- ★ Добавлять новую функциональность
- ★ Искать корреляции событий, выявлять и расследовать мошенническое поведение

Предварительный итог

Наличие лога позволяет

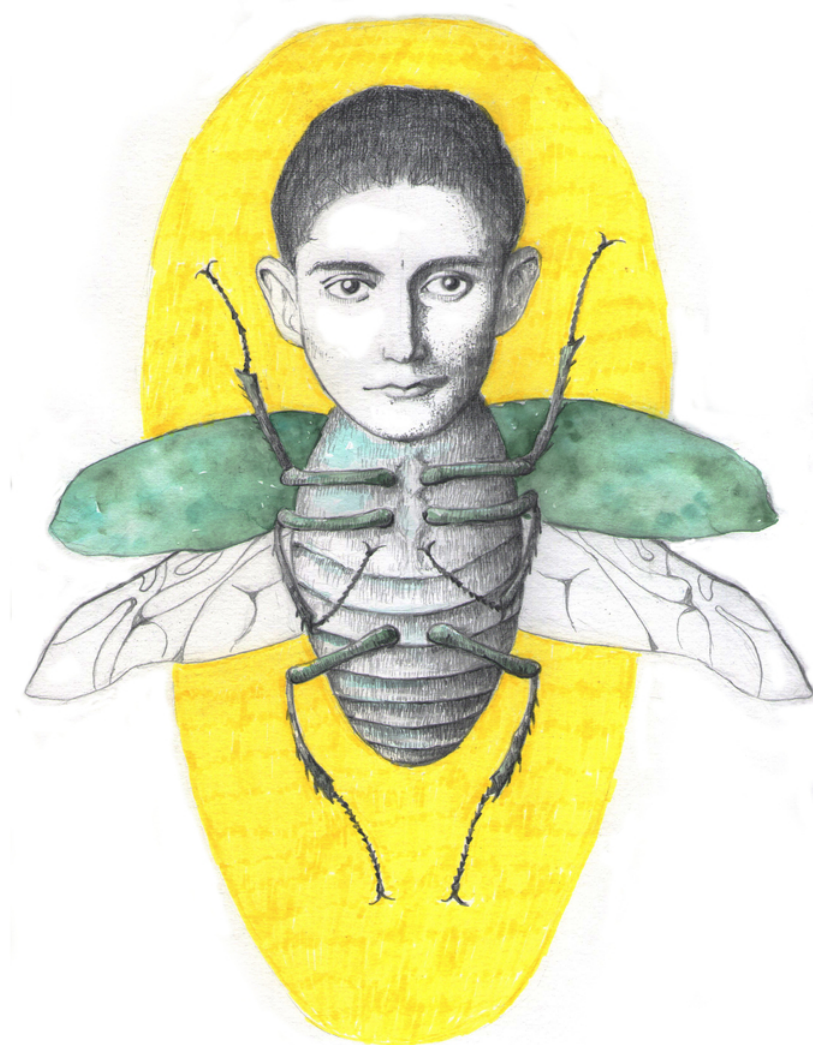
- ★ Добавлять новую функциональность
- ★ Искать корреляции событий, выявлять и расследовать мошенническое поведение
- ★ Исправлять алгоритмические ошибки и пересчитывать данные

Предварительный итог

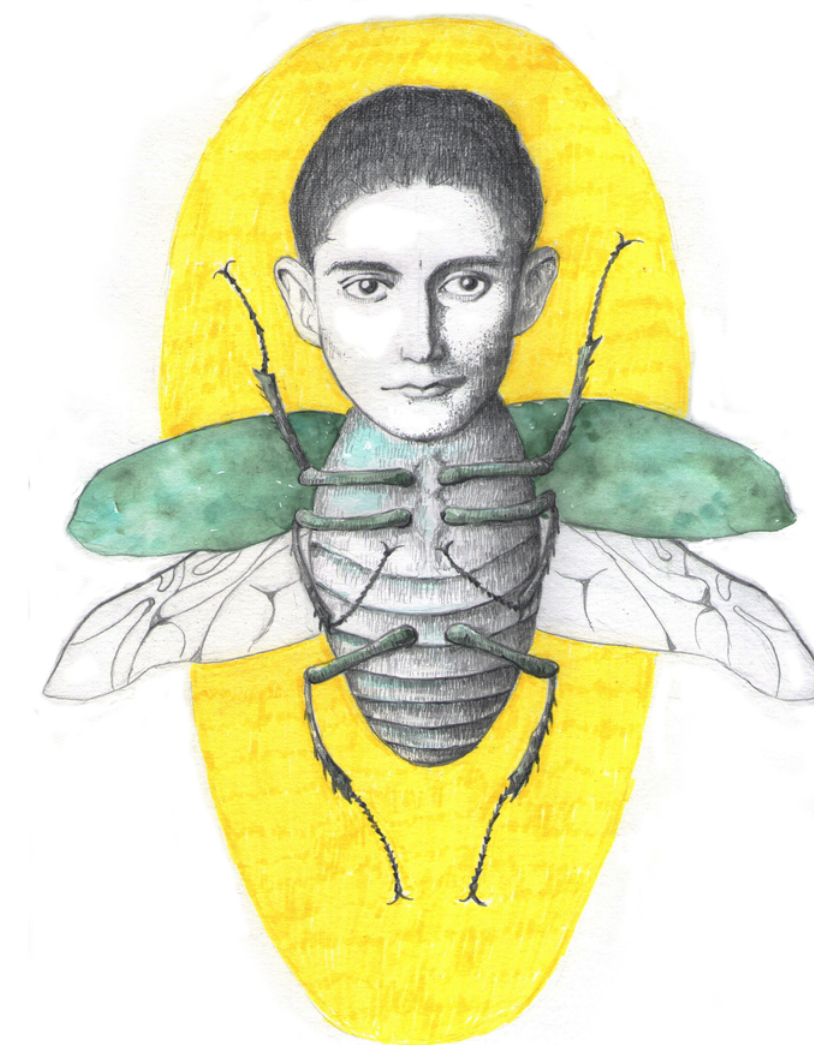
Наличие лога позволяет

- ★ Добавлять новую функциональность
- ★ Искать корреляции событий, выявлять и расследовать мошенническое поведение
- ★ Исправлять алгоритмические ошибки и пересчитывать данные
- ★ Наша жизнь — это append-only log

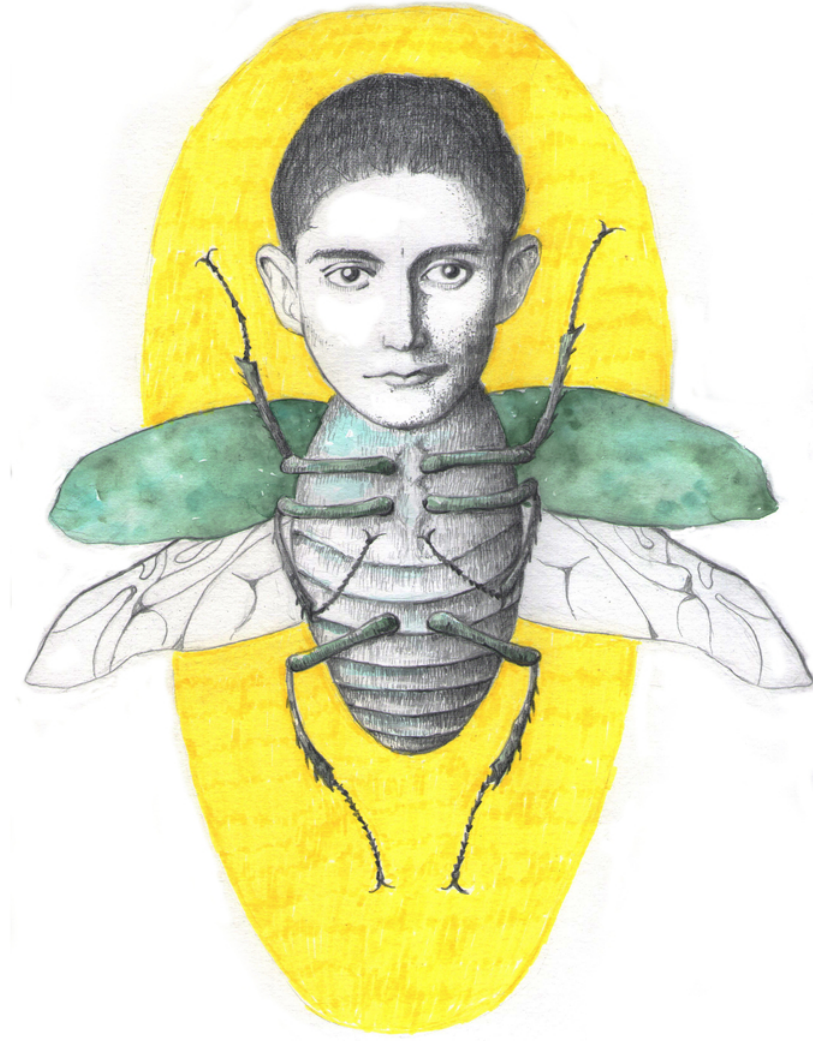
Наш план



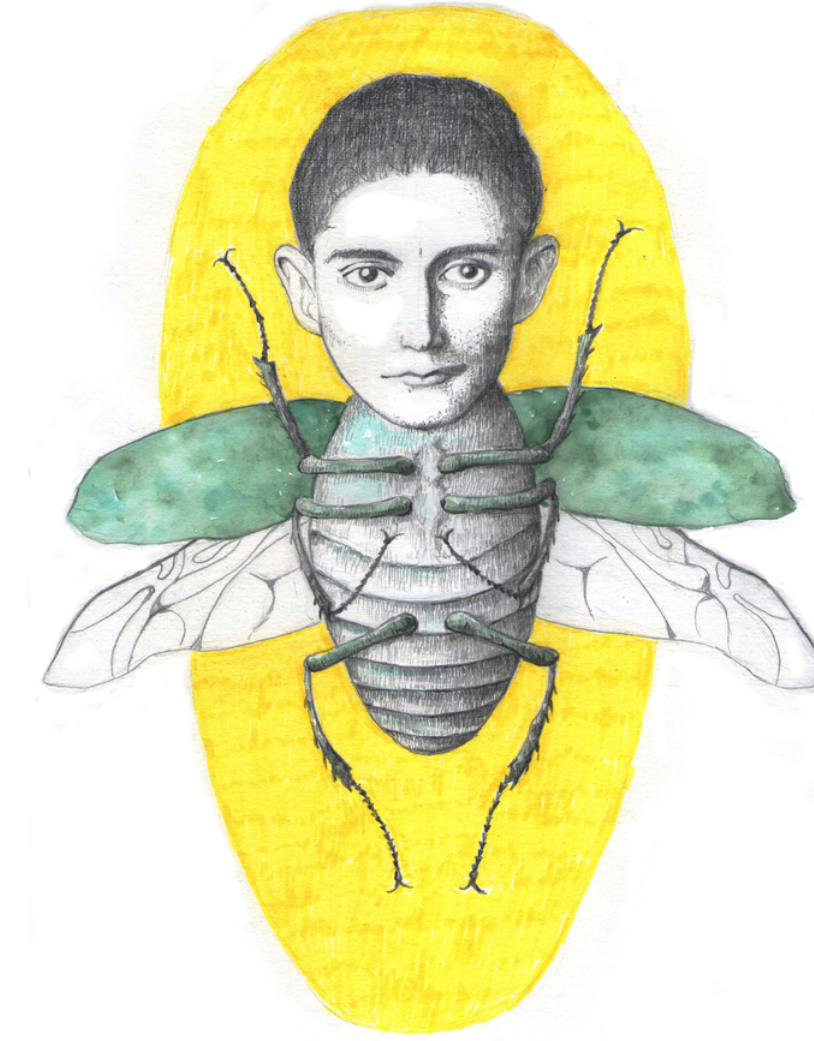
1. ~~Что такое лог и почему это важно~~
2. **Что такое Kafka и что она умеет**
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



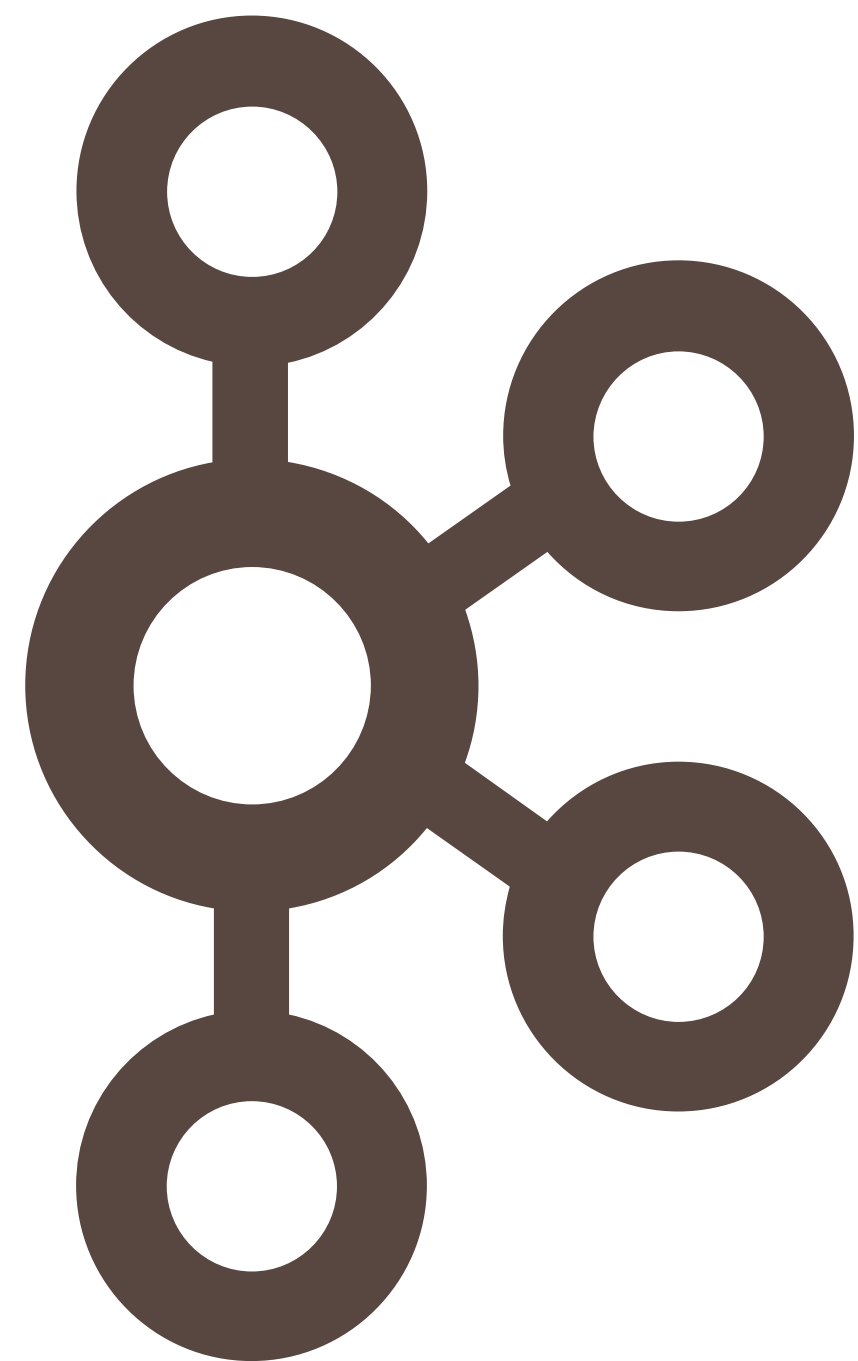
Наш план



1. ~~Что такое лог и почему это важно~~
2. Что такое Kafka и что она умеет
 1. **Общая информация**
 2. Как устроен кластер
 3. Как устроена запись
 4. Как устроено чтение
 5. Retention и компактификация
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



Кafka это



- ★ распределённый
- ★ масштабируемый
- ★ отказоустойчивый лог
- ★ миллионы записей в секунду



В Кафке можно

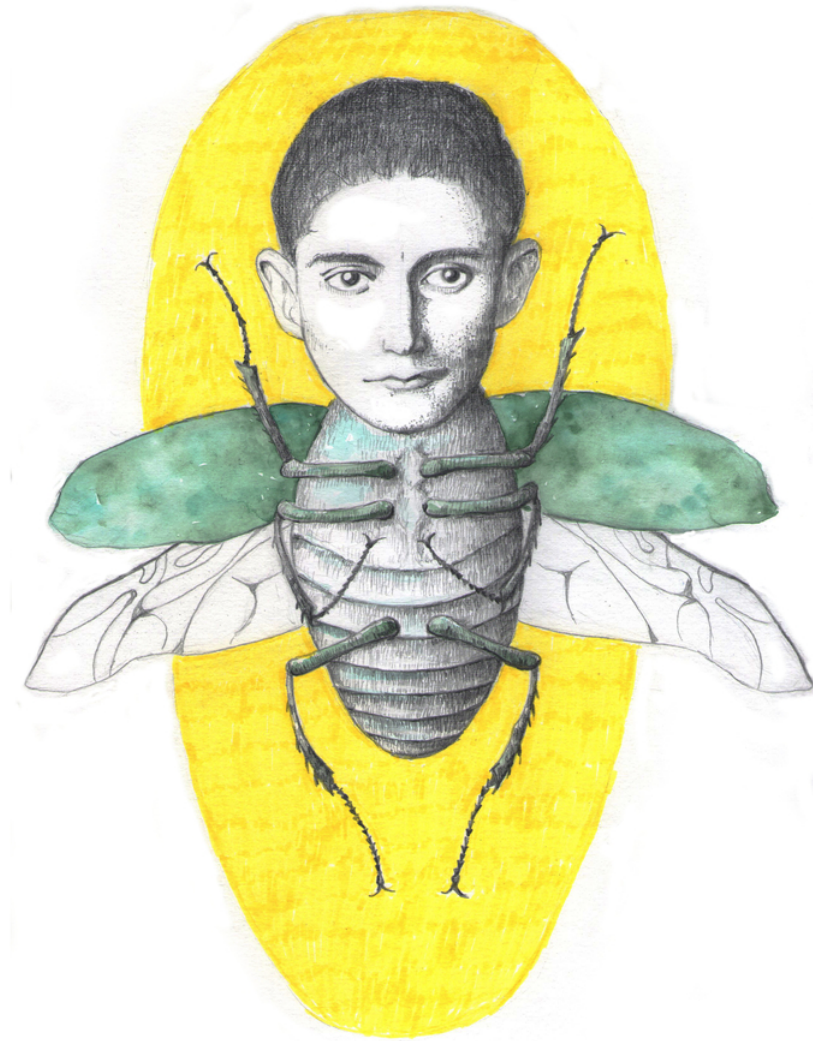
- ★ Записать нечто в именованный лог (topic)
- ★ Прочитать записи из топика в FIFO порядке (в пределах партиции)
- ★ Зафиксировать место, до которого дочитал



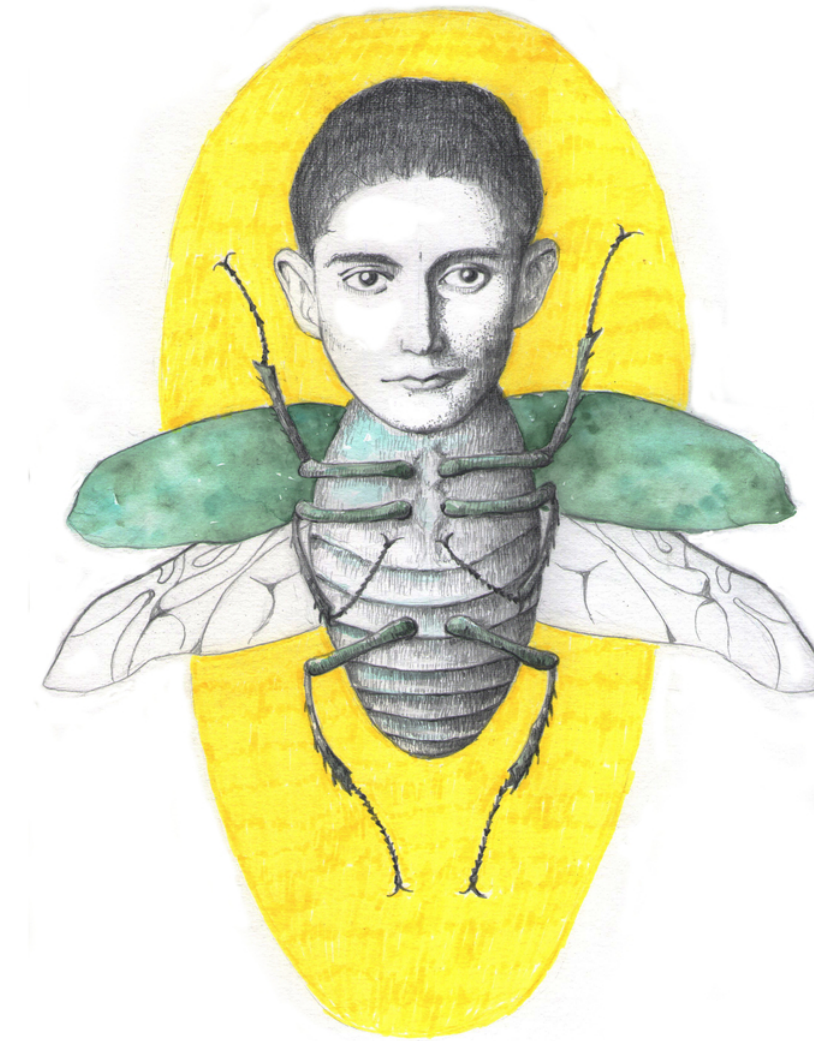
В Кафке нельзя

- ★ Стереть запись
- ★ Изменить запись
- ★ Найти в логе запись иначе, как по её порядковому номеру

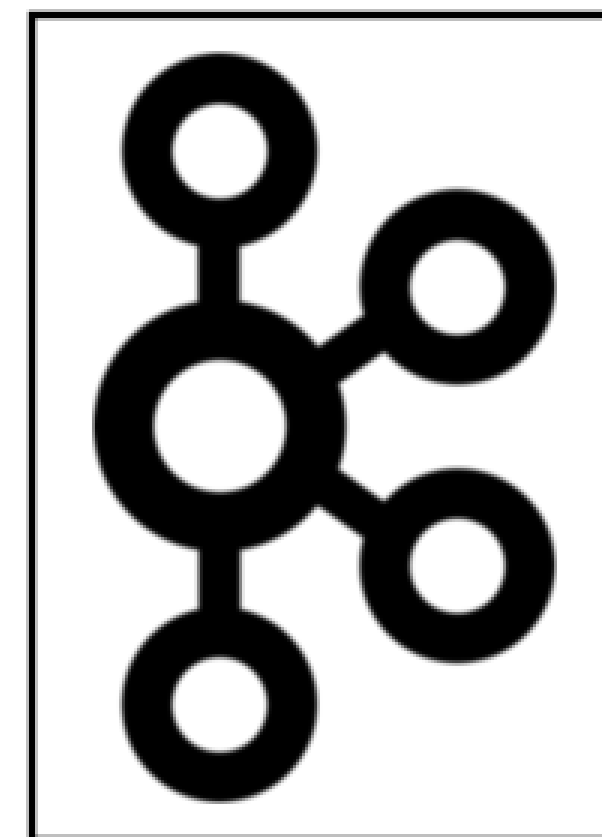
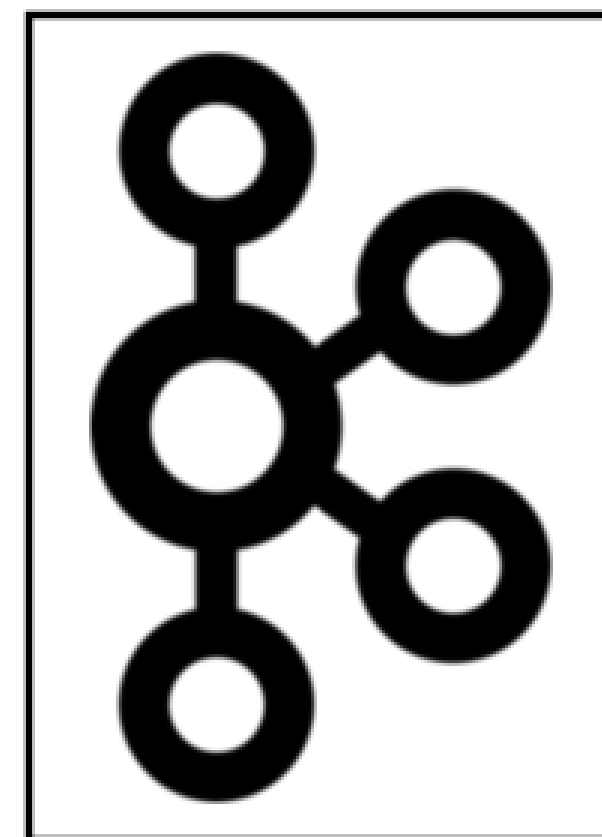
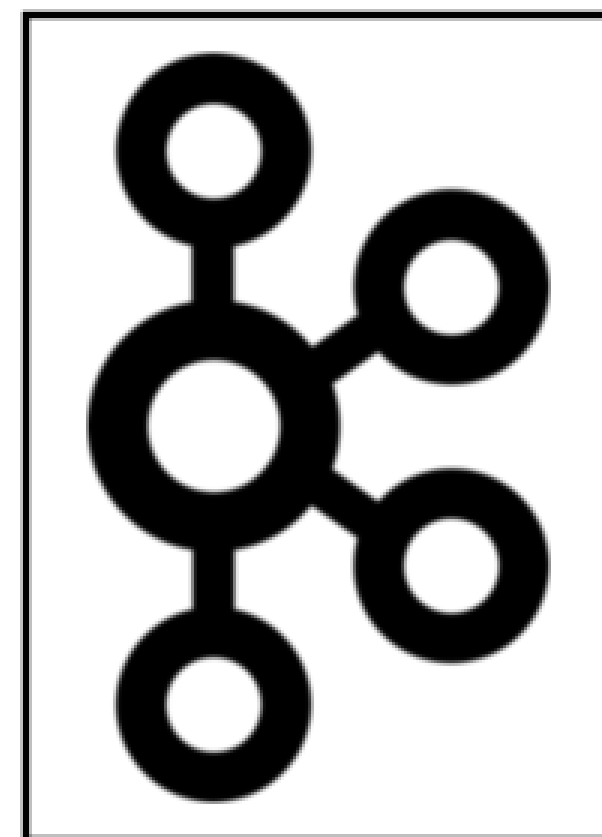
Наш план



1. ~~Что такое лог и почему это важно~~
2. Что такое Kafka и что она умеет
 1. ~~Общая информация~~
 2. **Как устроен кластер**
 3. Как устроена запись
 4. Как устроено чтение
 5. Retention и компактификация
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



Кластер Kafka: брокеры и Zookeeper



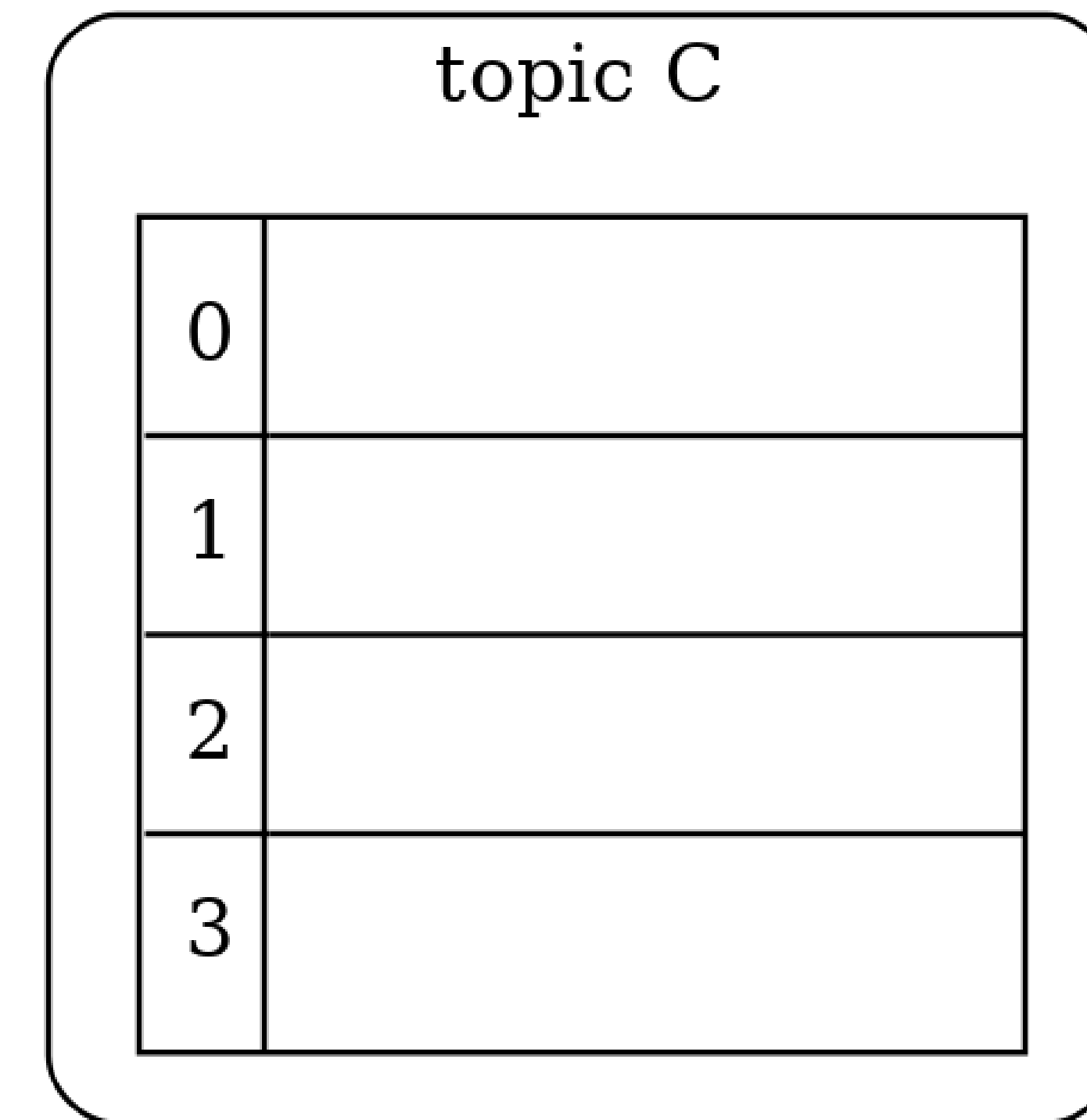
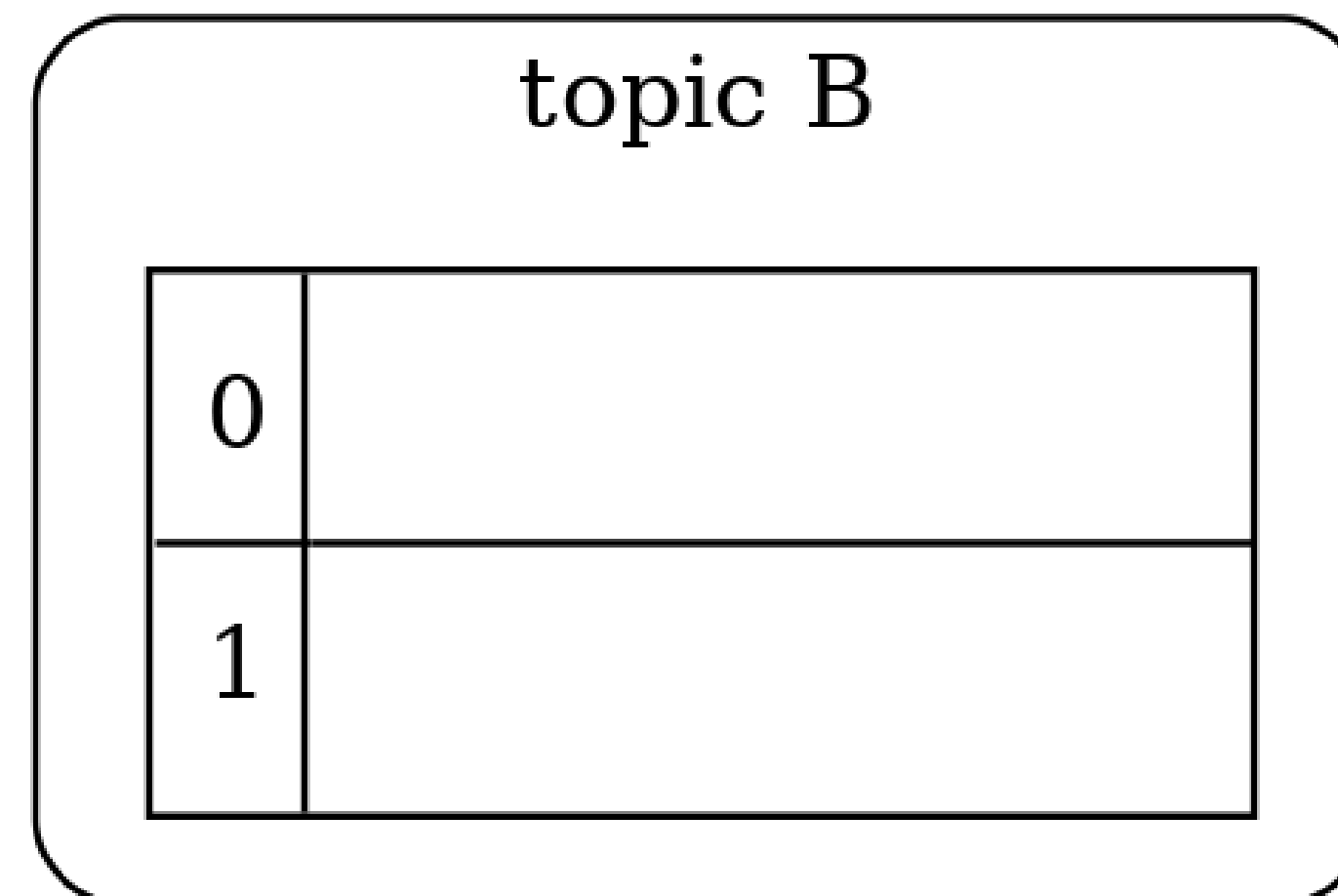
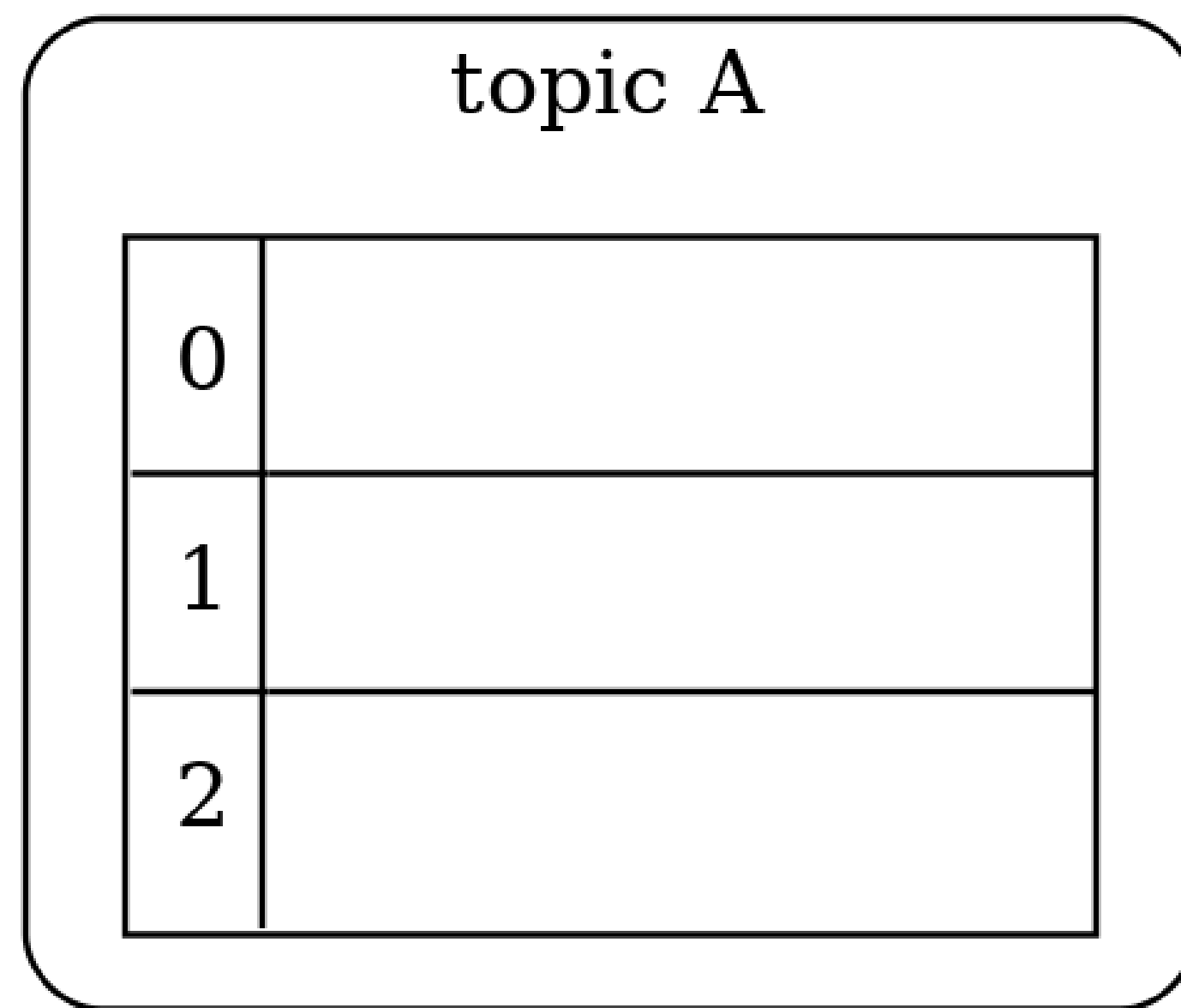
Топики, партиции и сообщения

topic A

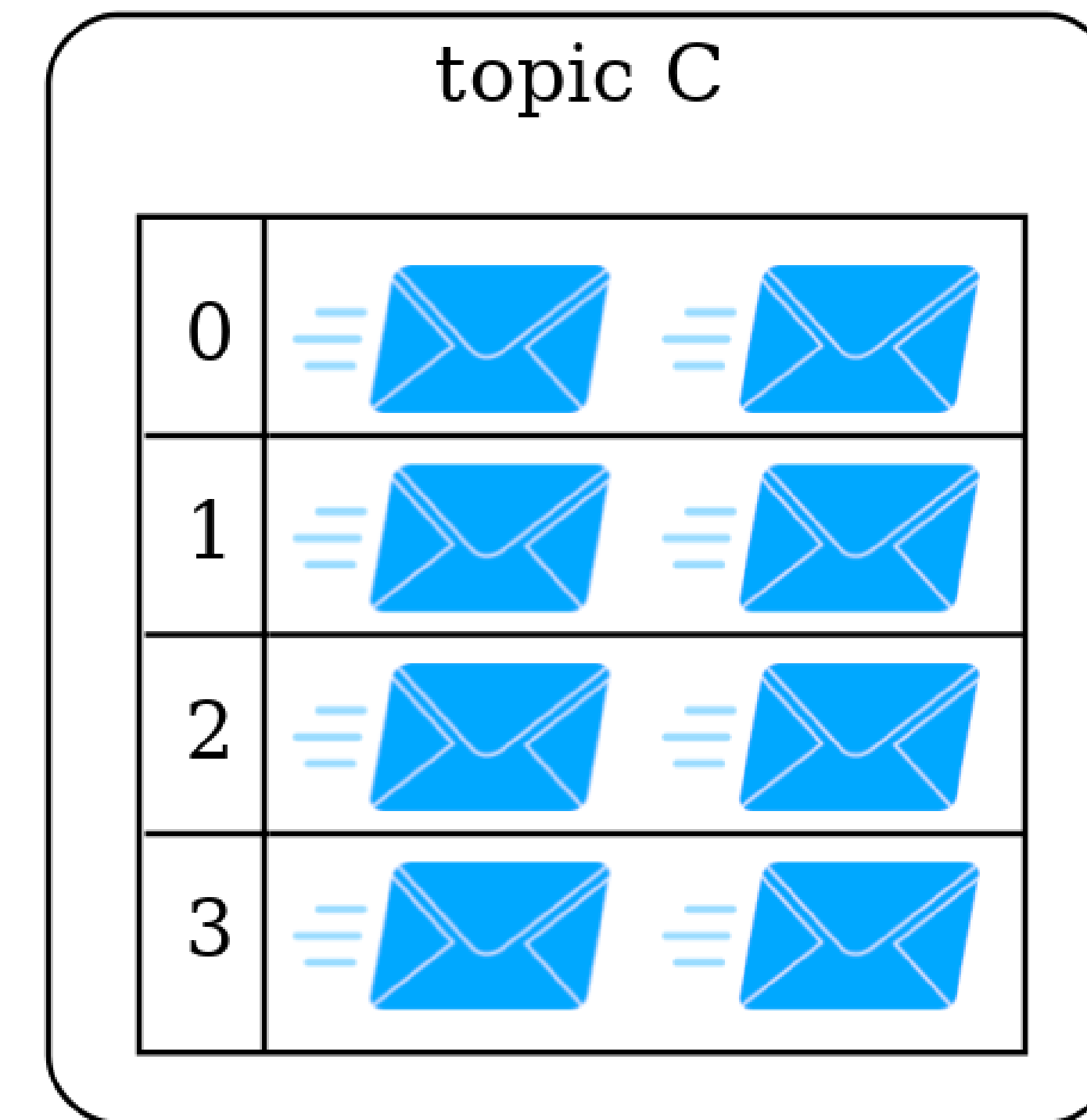
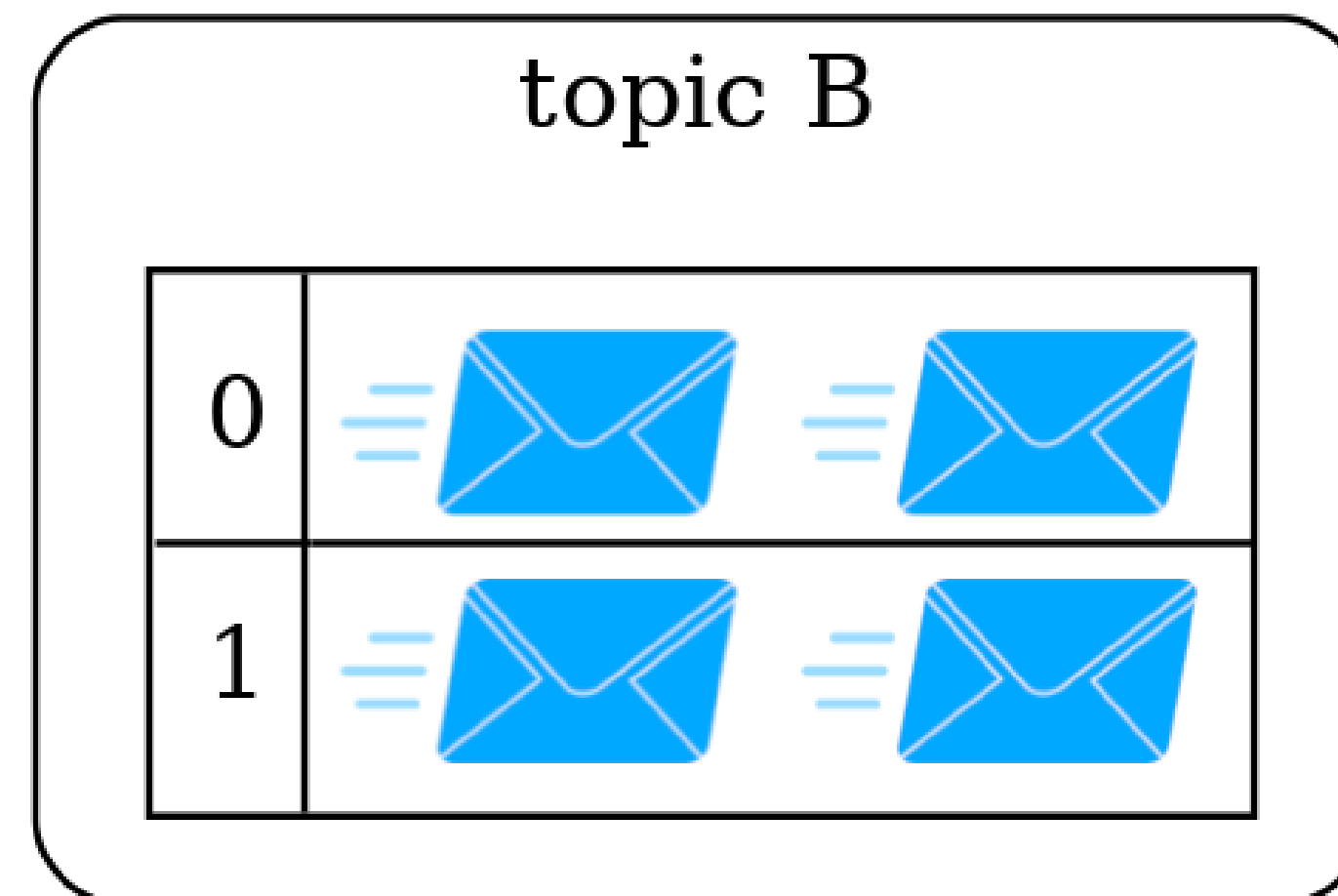
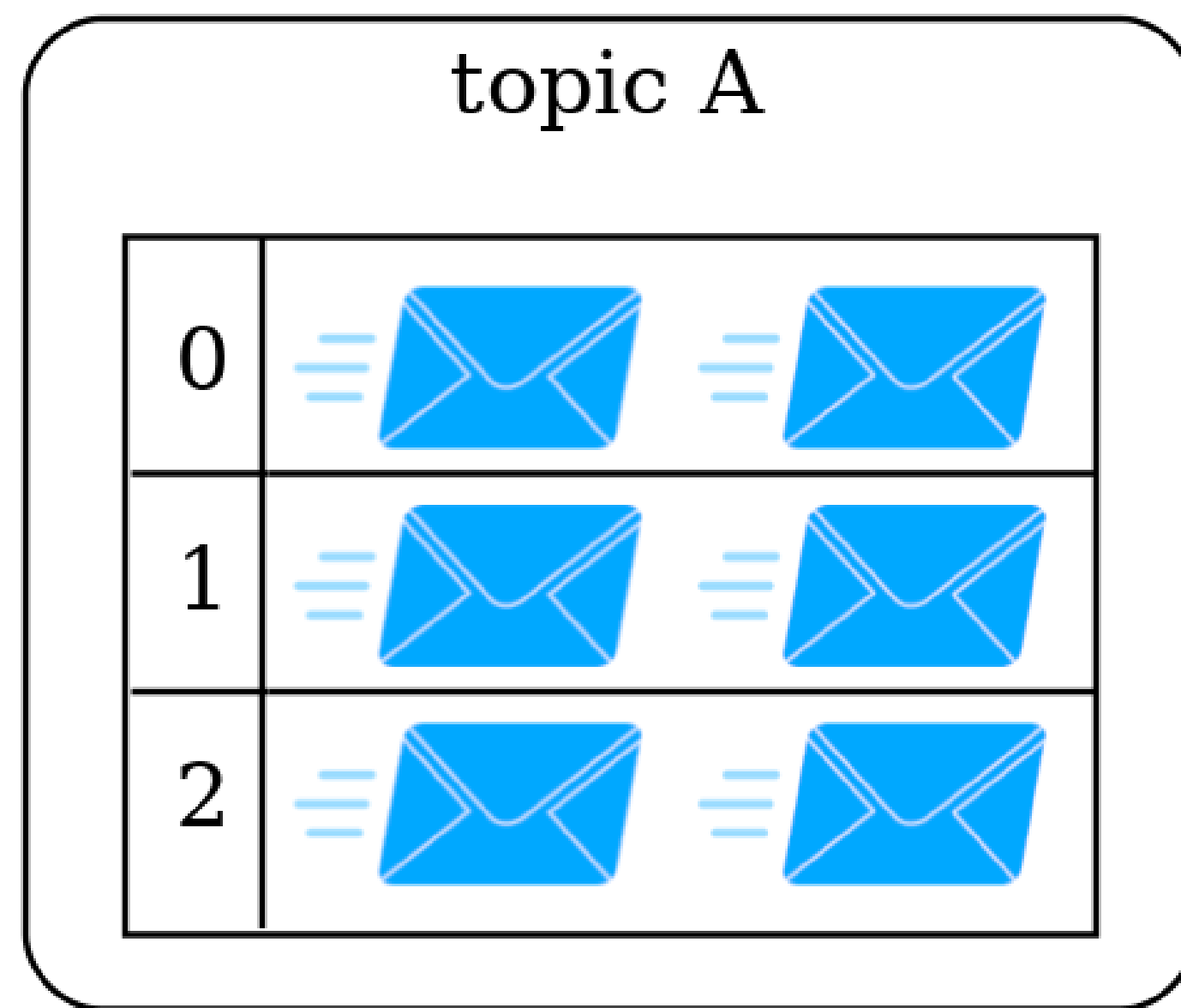
topic B

topic C

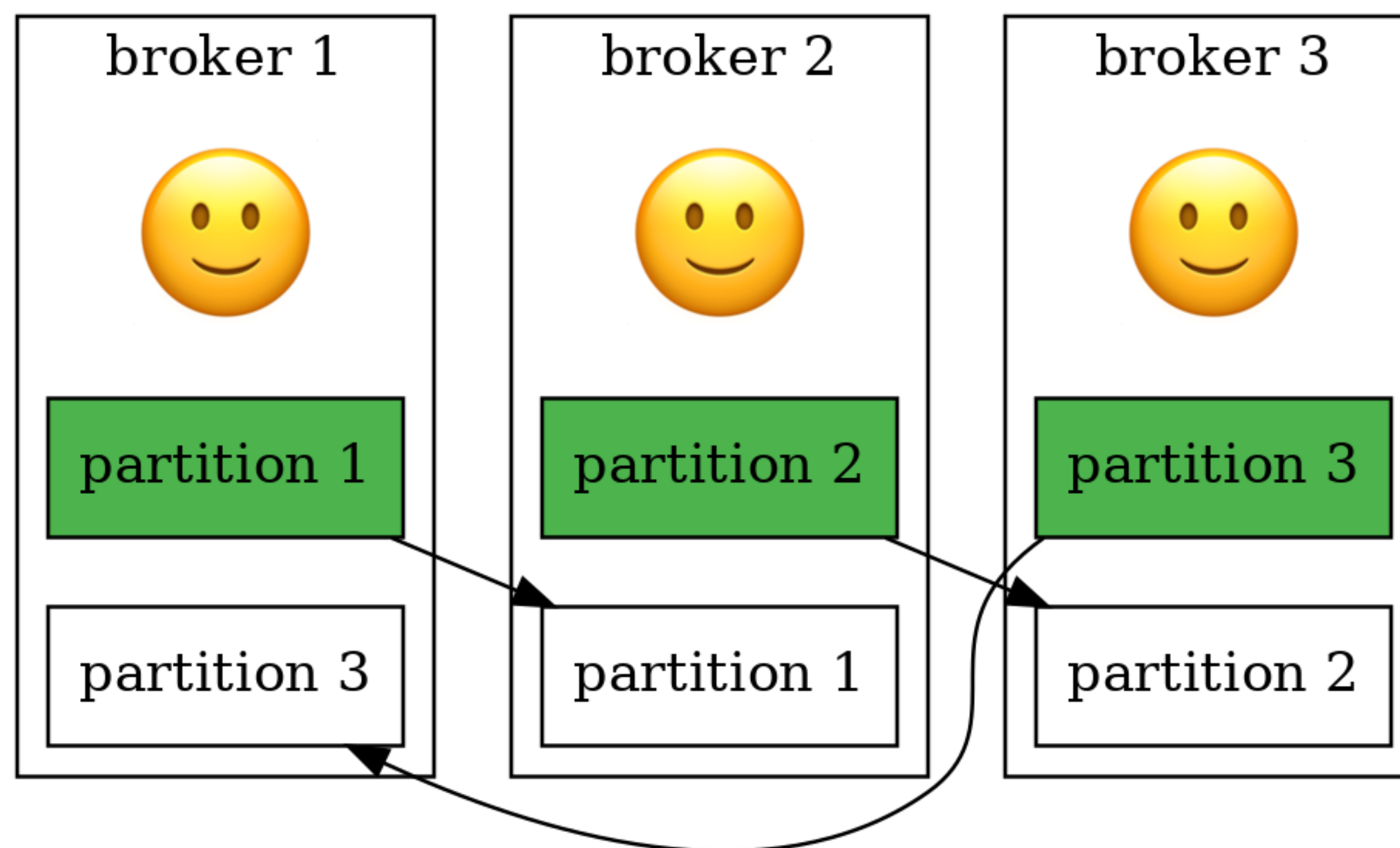
Топики, партиции и сообщения



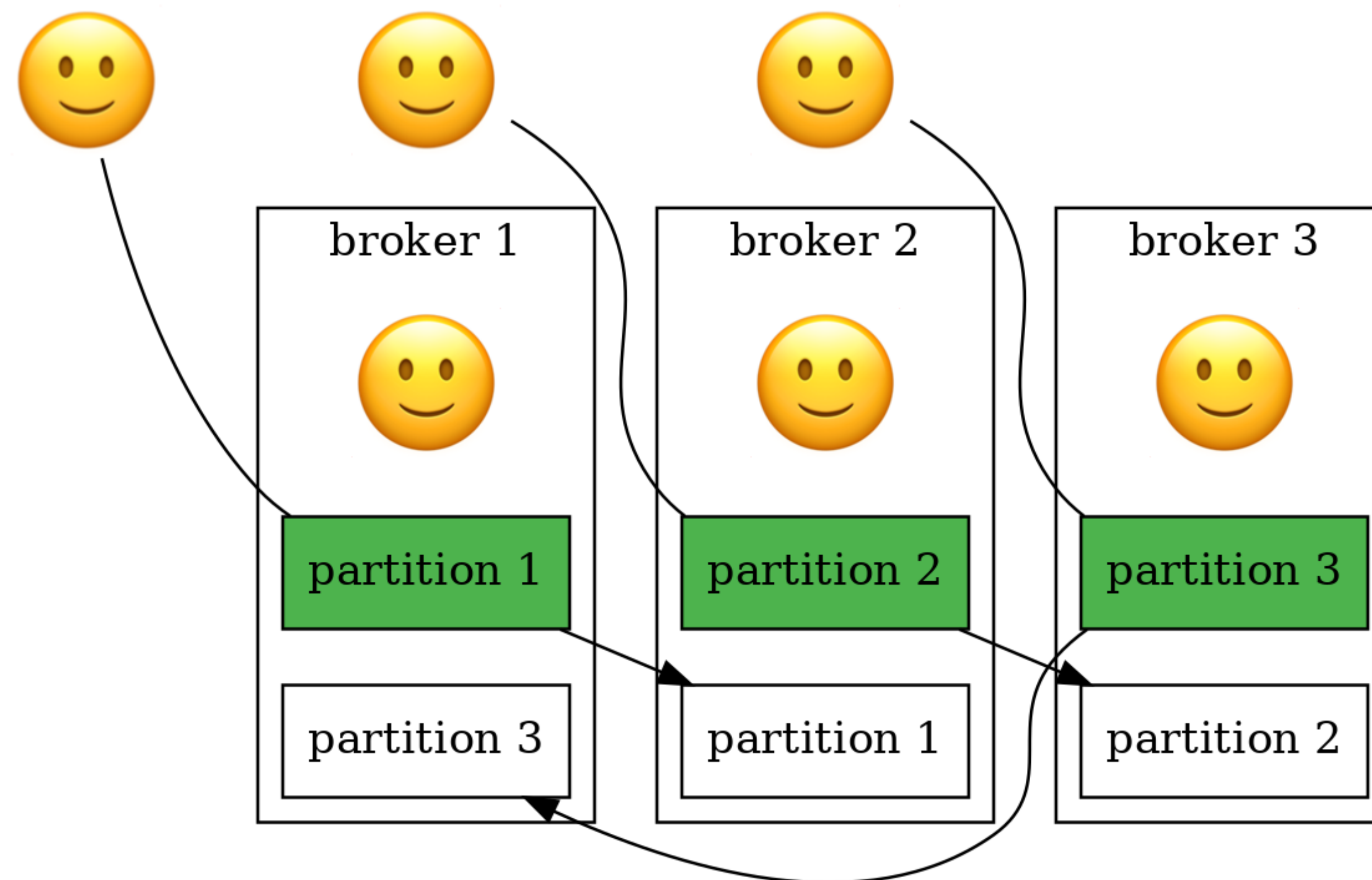
Топики, партиции и сообщения



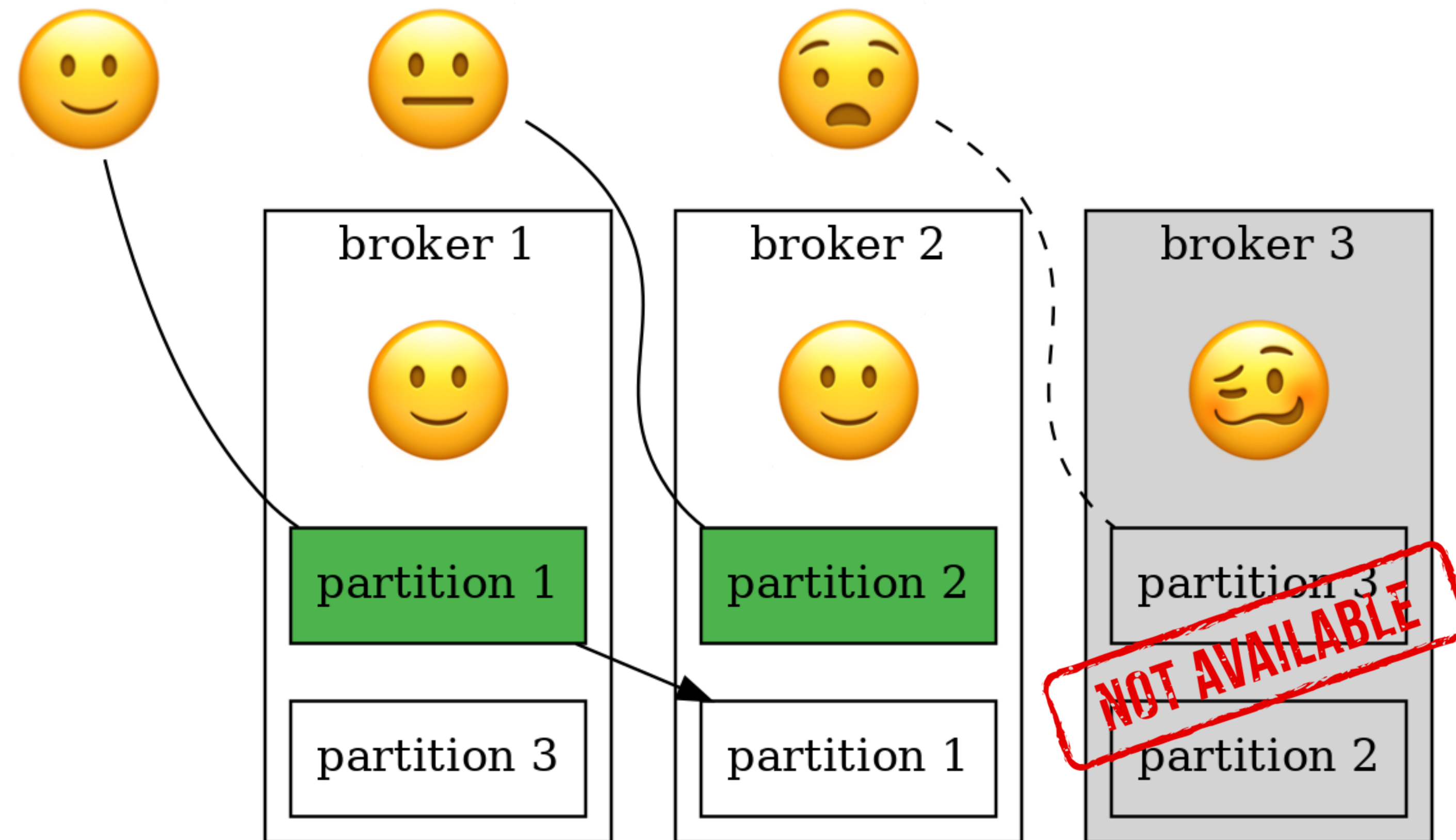
Репликация партиций



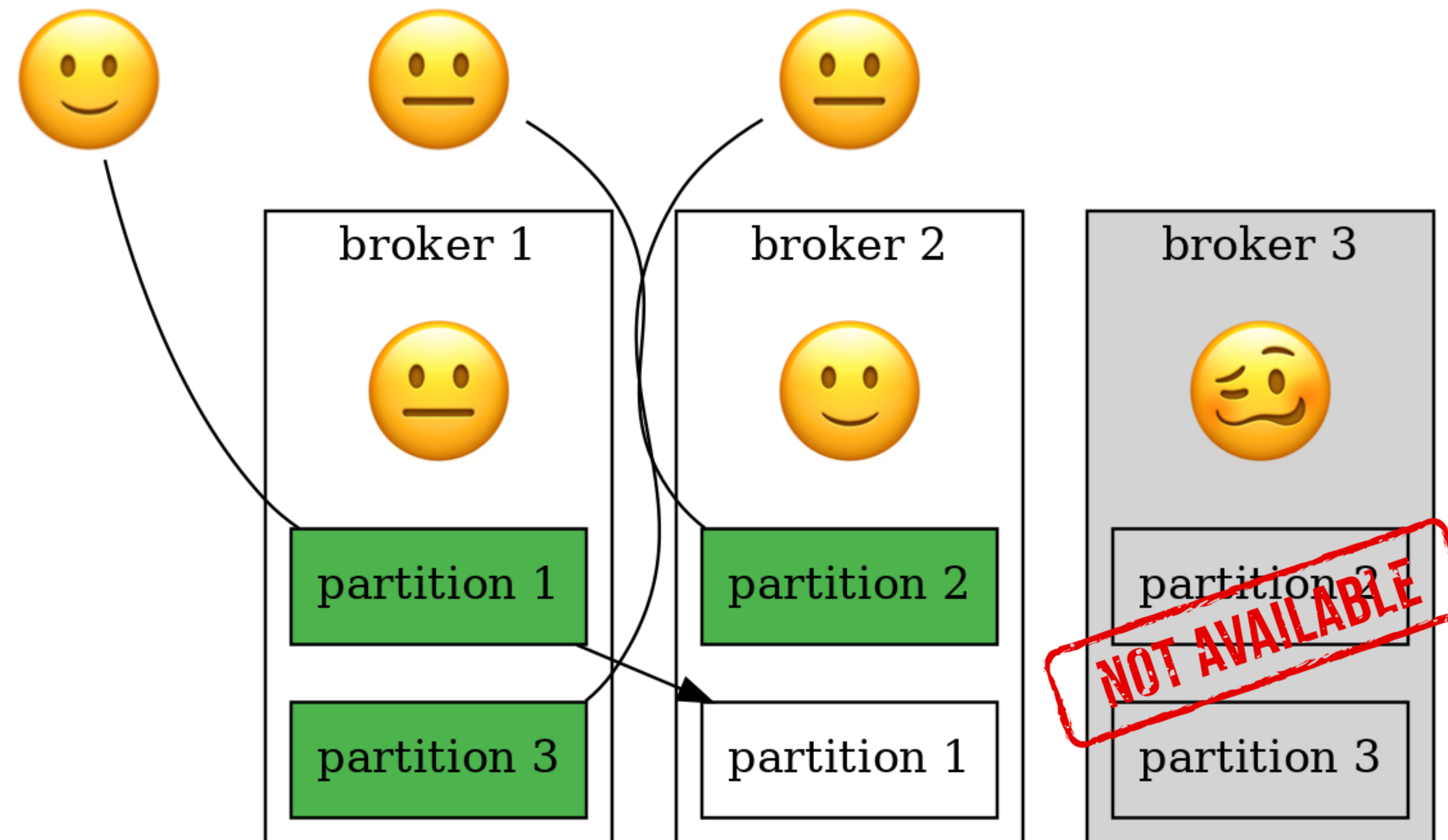
Репликация партиций



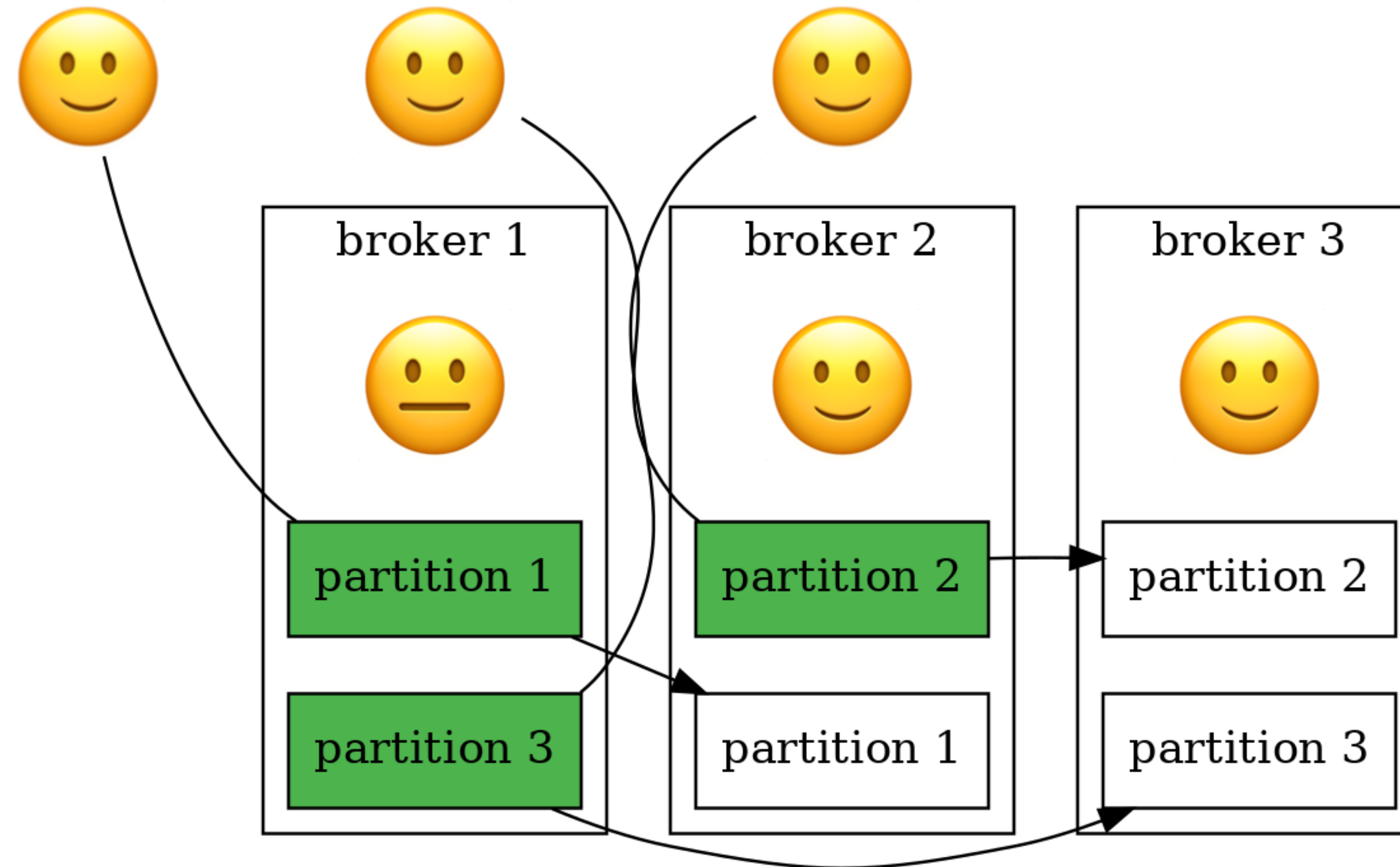
Репликация партиций



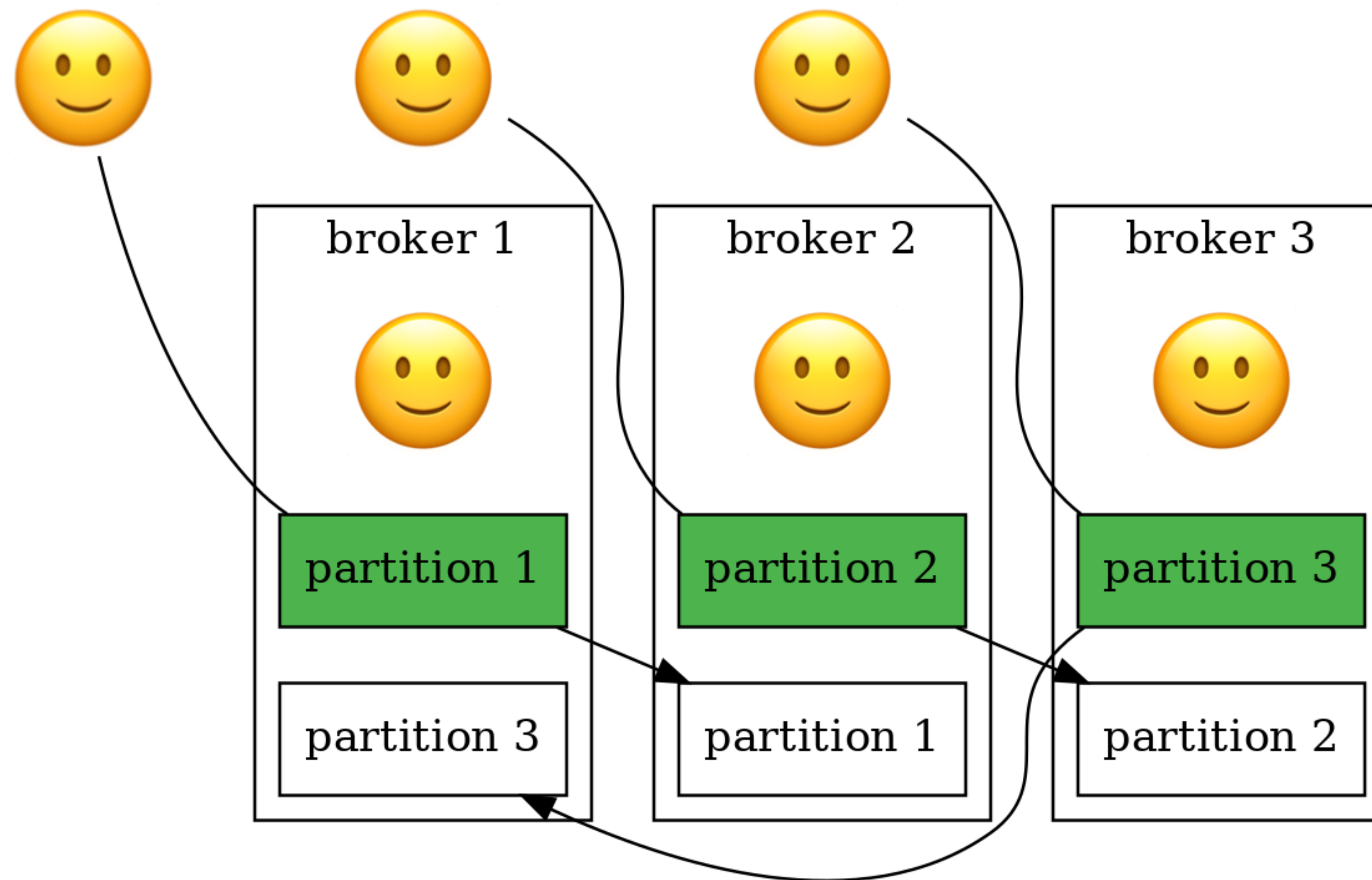
Репликация партиций



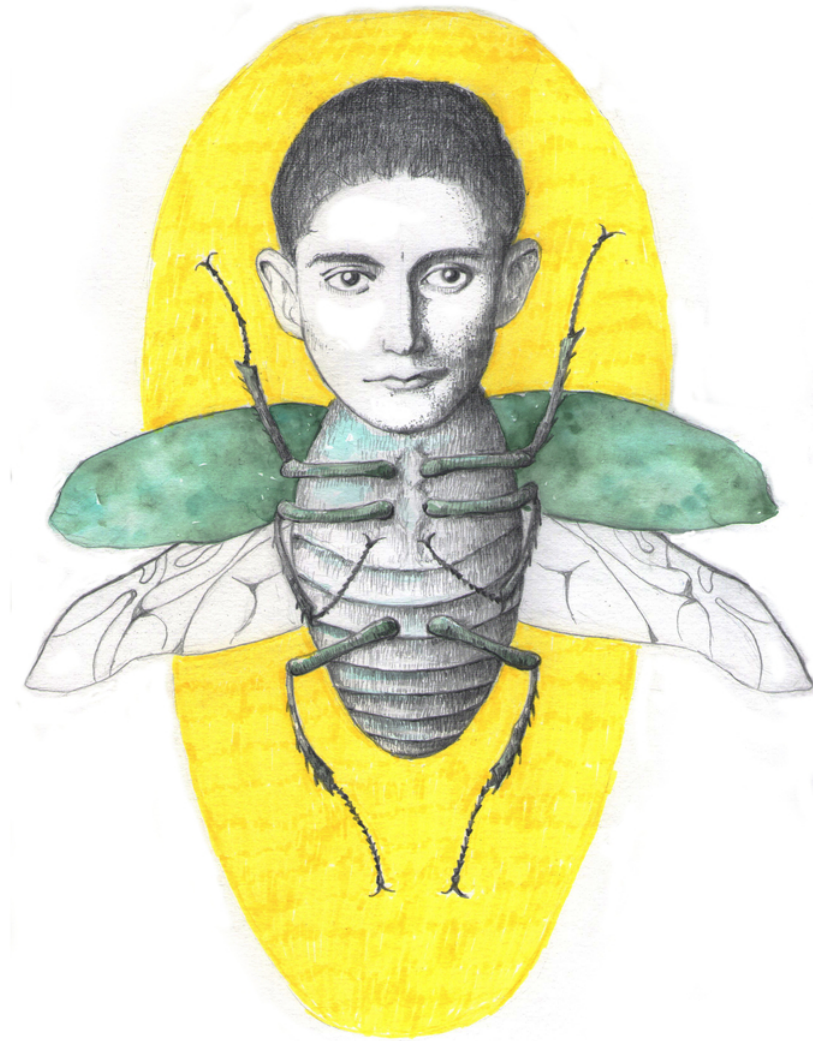
Репликация партиций



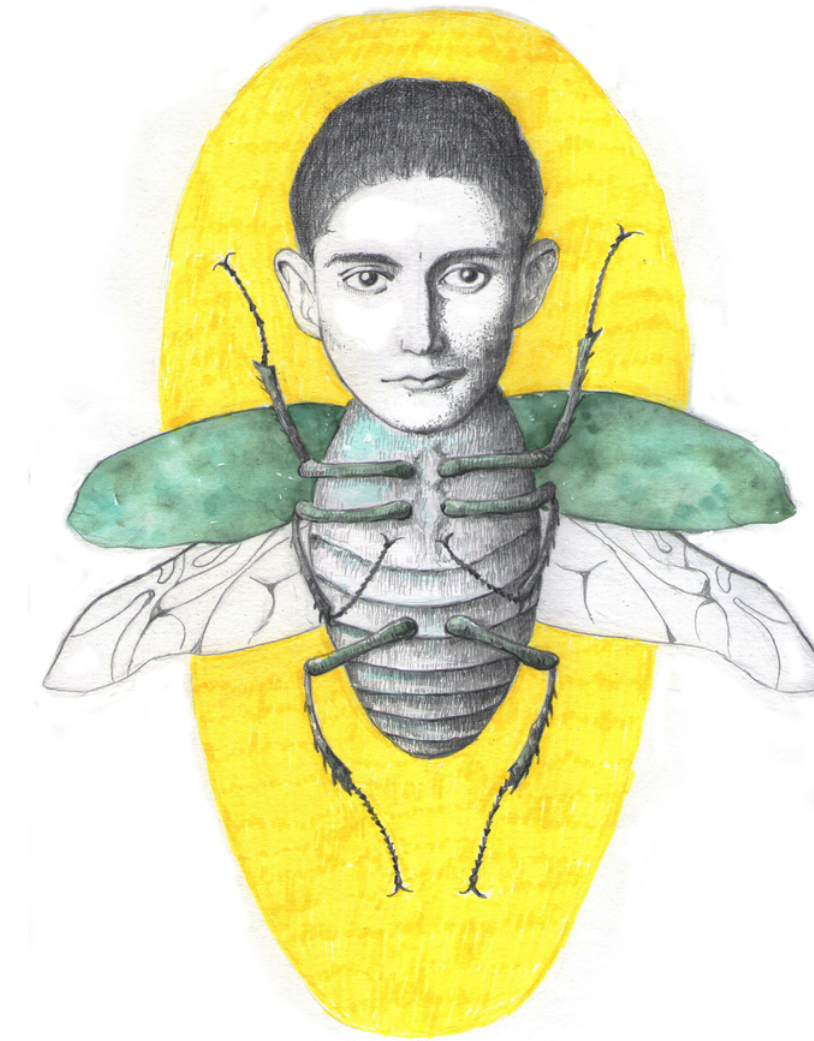
Репликация партиций



Наш план



1. ~~Что такое лог и почему это важно~~
2. Что такое Kafka и что она умеет
 1. ~~Общая информация~~
 2. ~~Как устроен кластер~~
 3. **Как устроена запись**
 4. Как устроено чтение
 5. Retention и компактификация
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



Анатомия сообщения



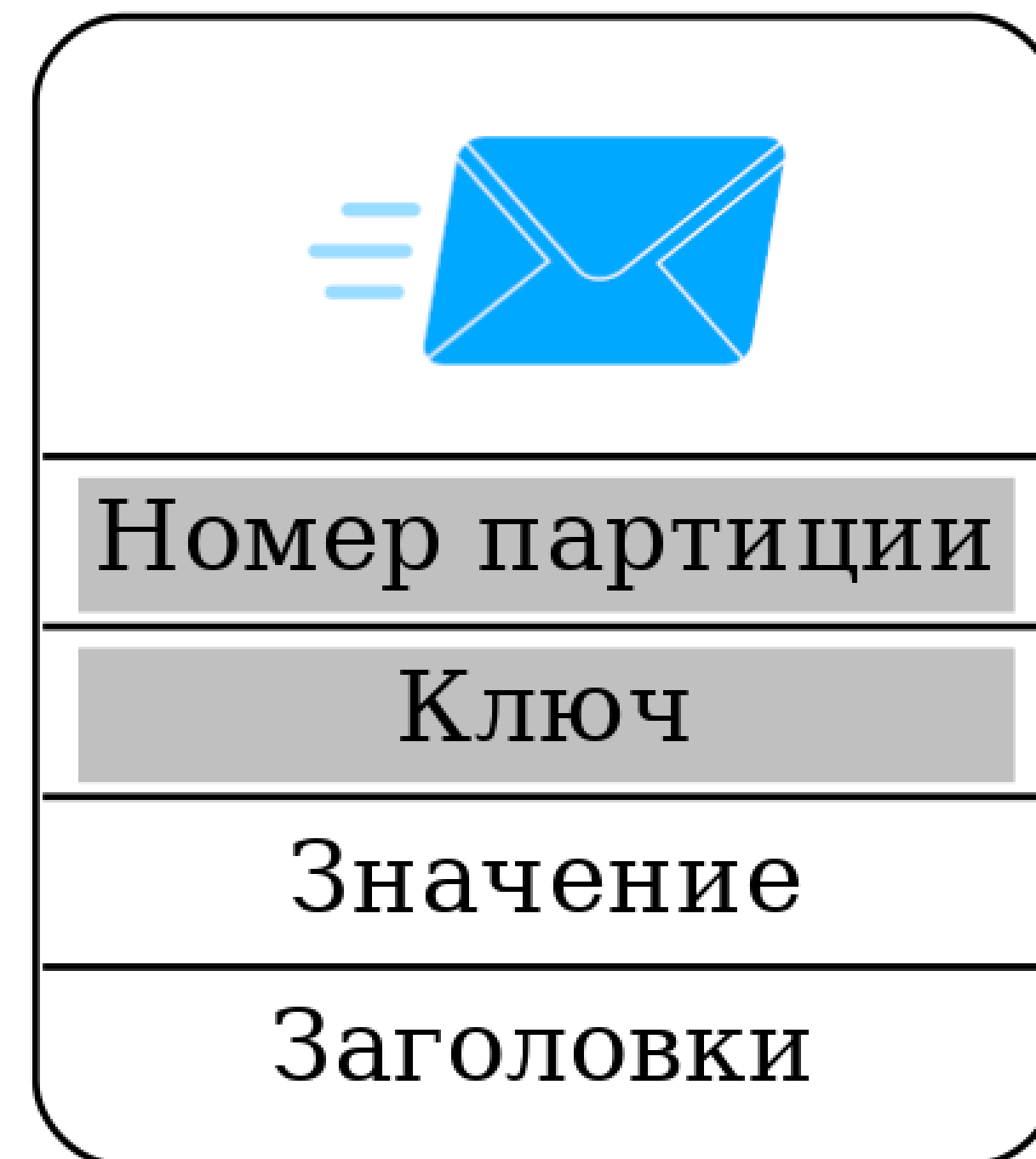
Номер партиции

Ключ

Значение

Заголовки

Анатомия сообщения



```
// hash the keyBytes to choose a partition  
return Utils.toPositive(Utils.murmur2(keyBytes)) % numPartitions;
```

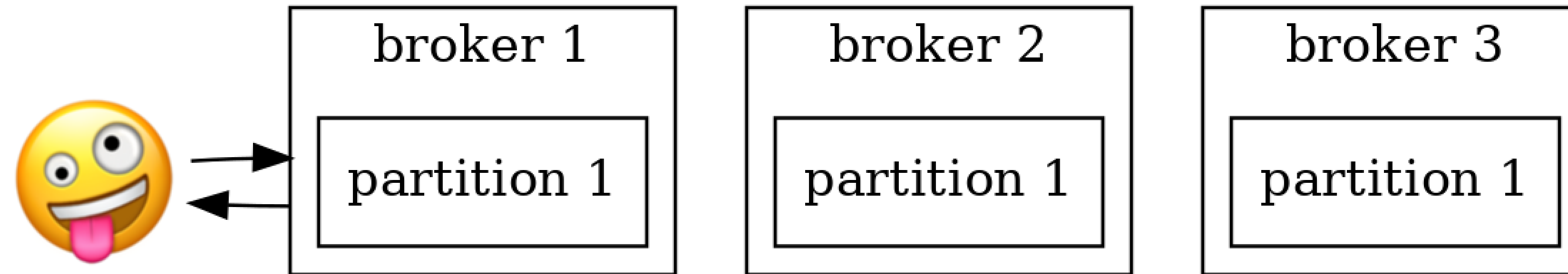
Пропускная способность vs время ожидания

`batch.size` and `linger.ms`



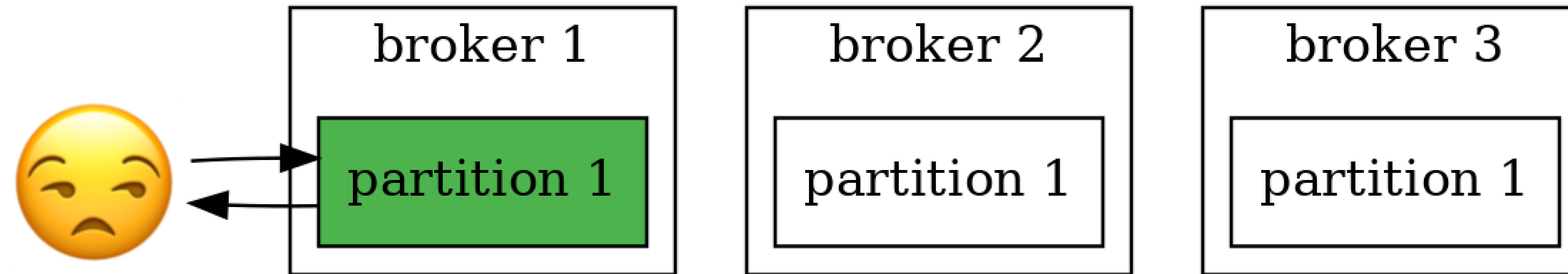
Запись в Кафку

`acks = 0`



Запись в Кафку

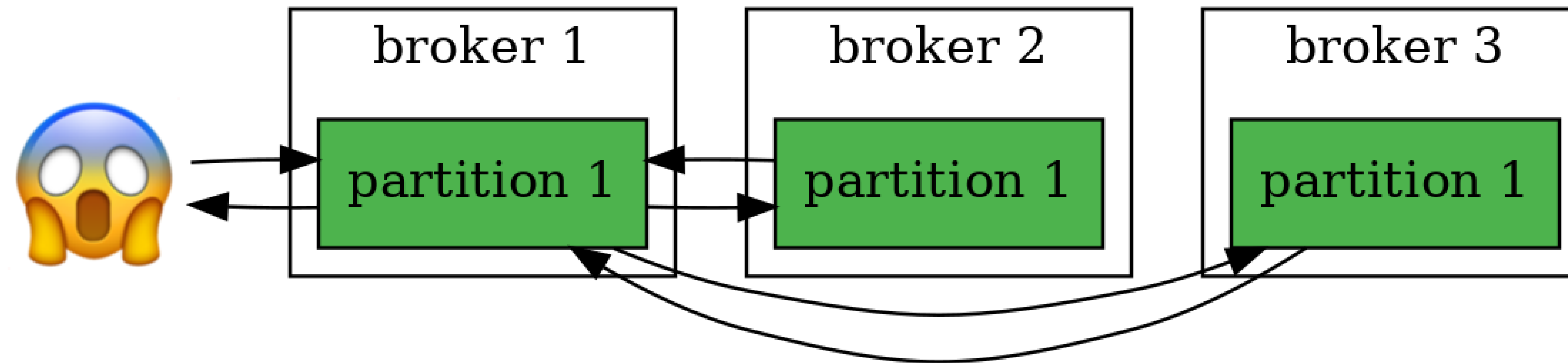
acks = 1



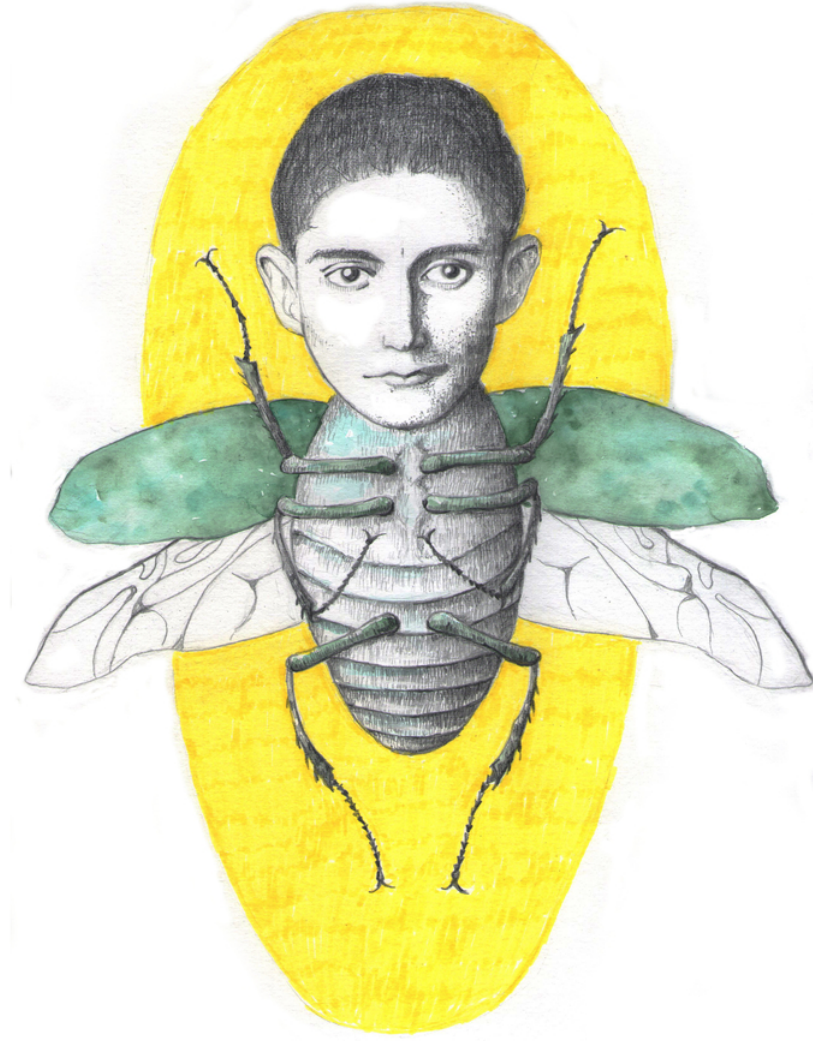
Запись в Кафку

`acks = -1`

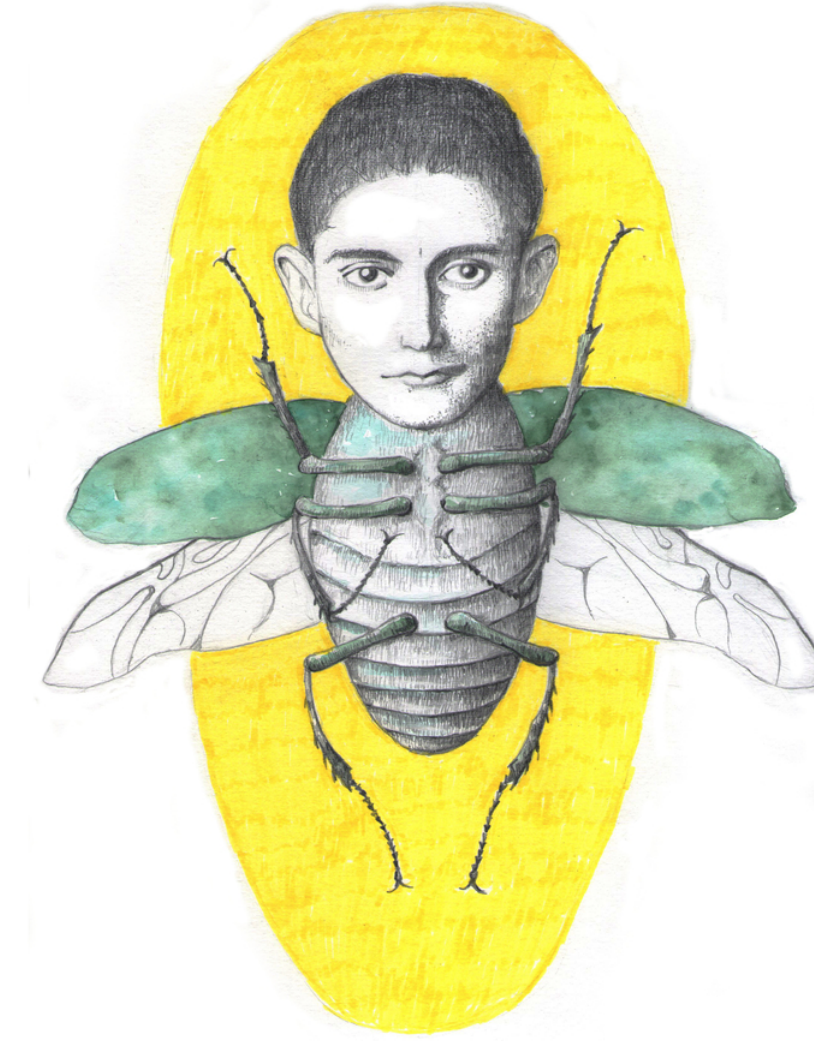
`min.insync.replicas`



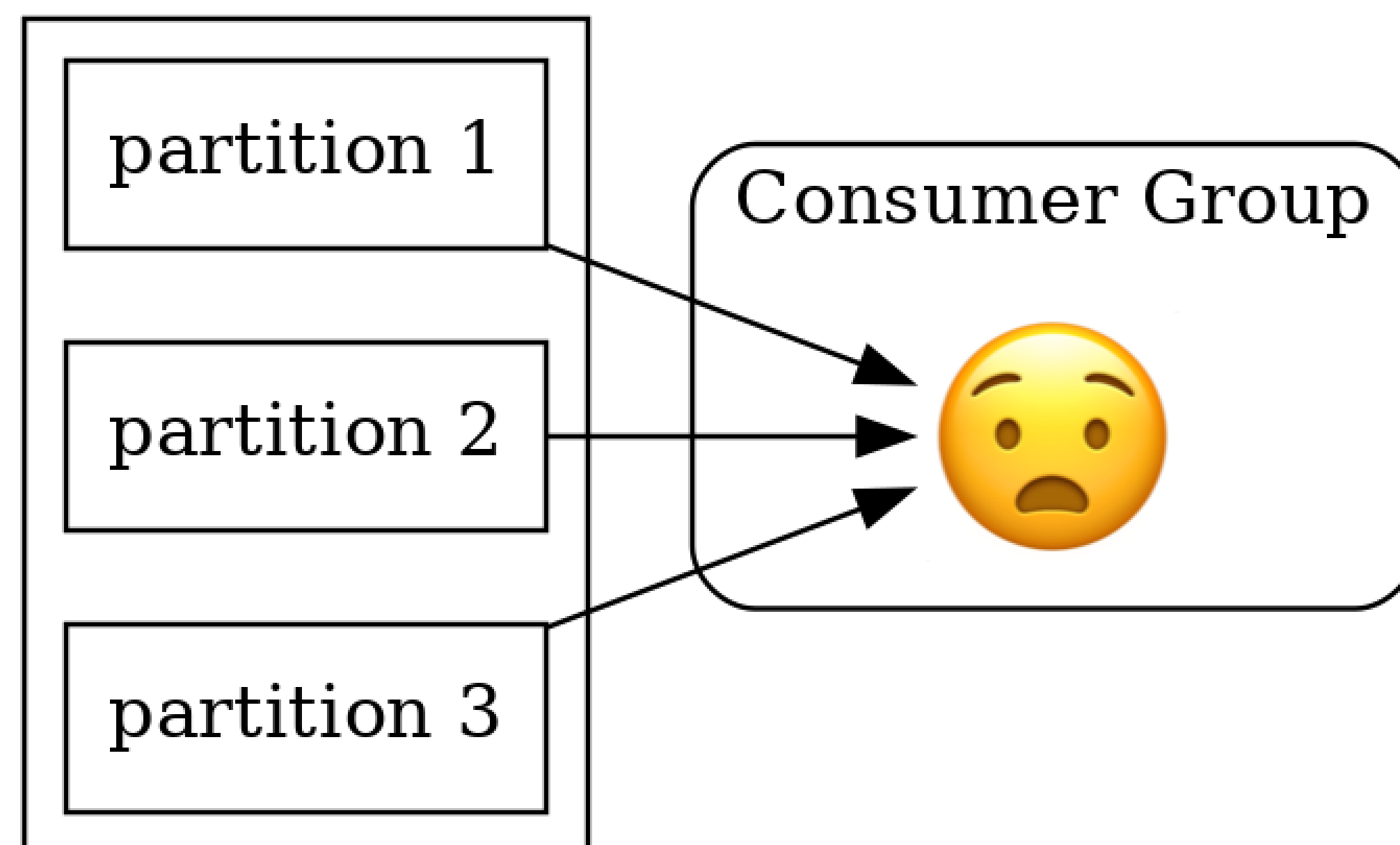
Наш план



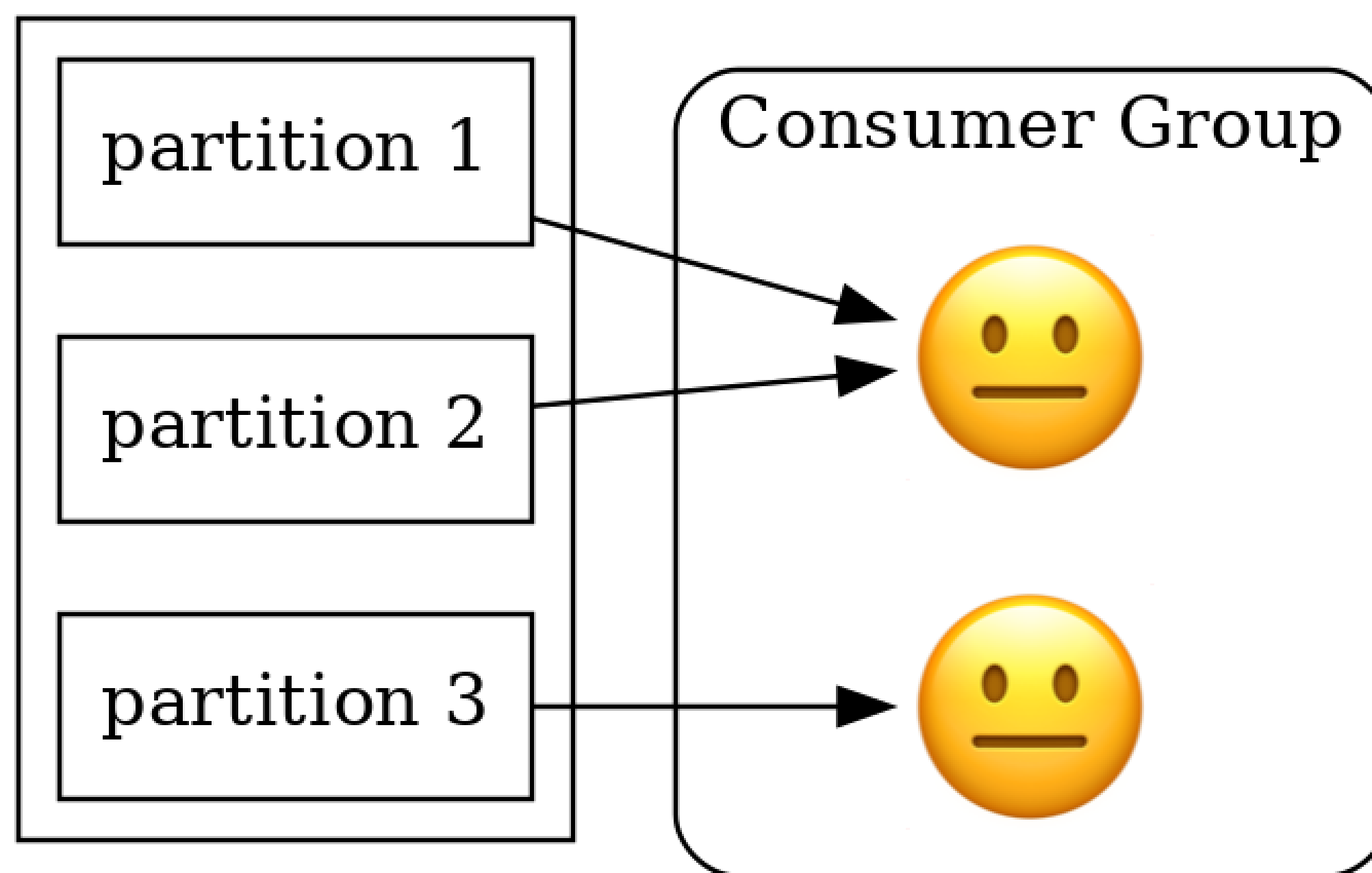
- ~~1. Что такое лог и почему это важно~~
2. Что такое Kafka и что она умеет
 - ~~1. Общая информация~~
 - ~~2. Как устроен кластер~~
 - ~~3. Как устроена запись~~
 - 4. Как устроено чтение**
 5. Retention и компактификация
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



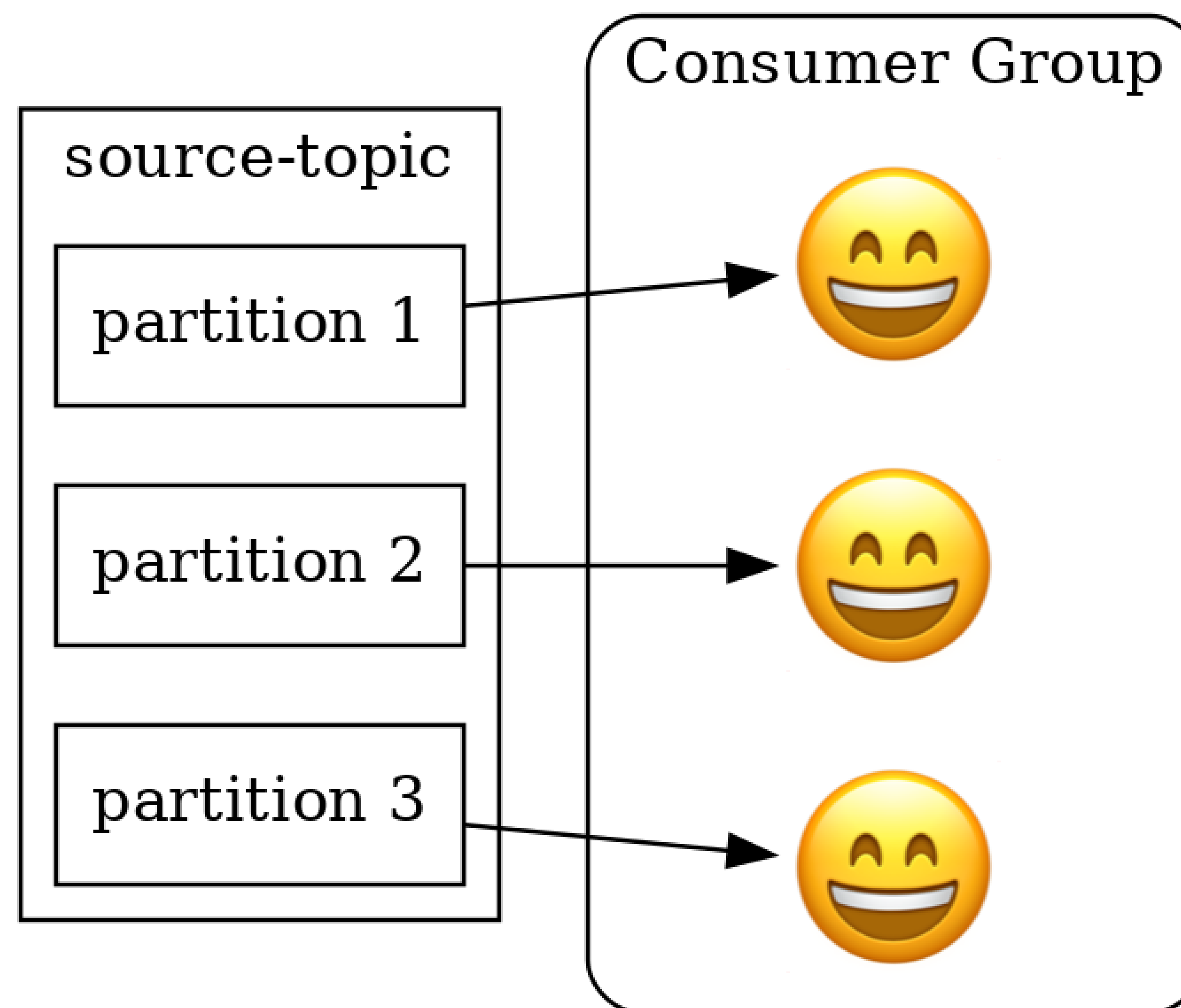
Чтение из Кафки



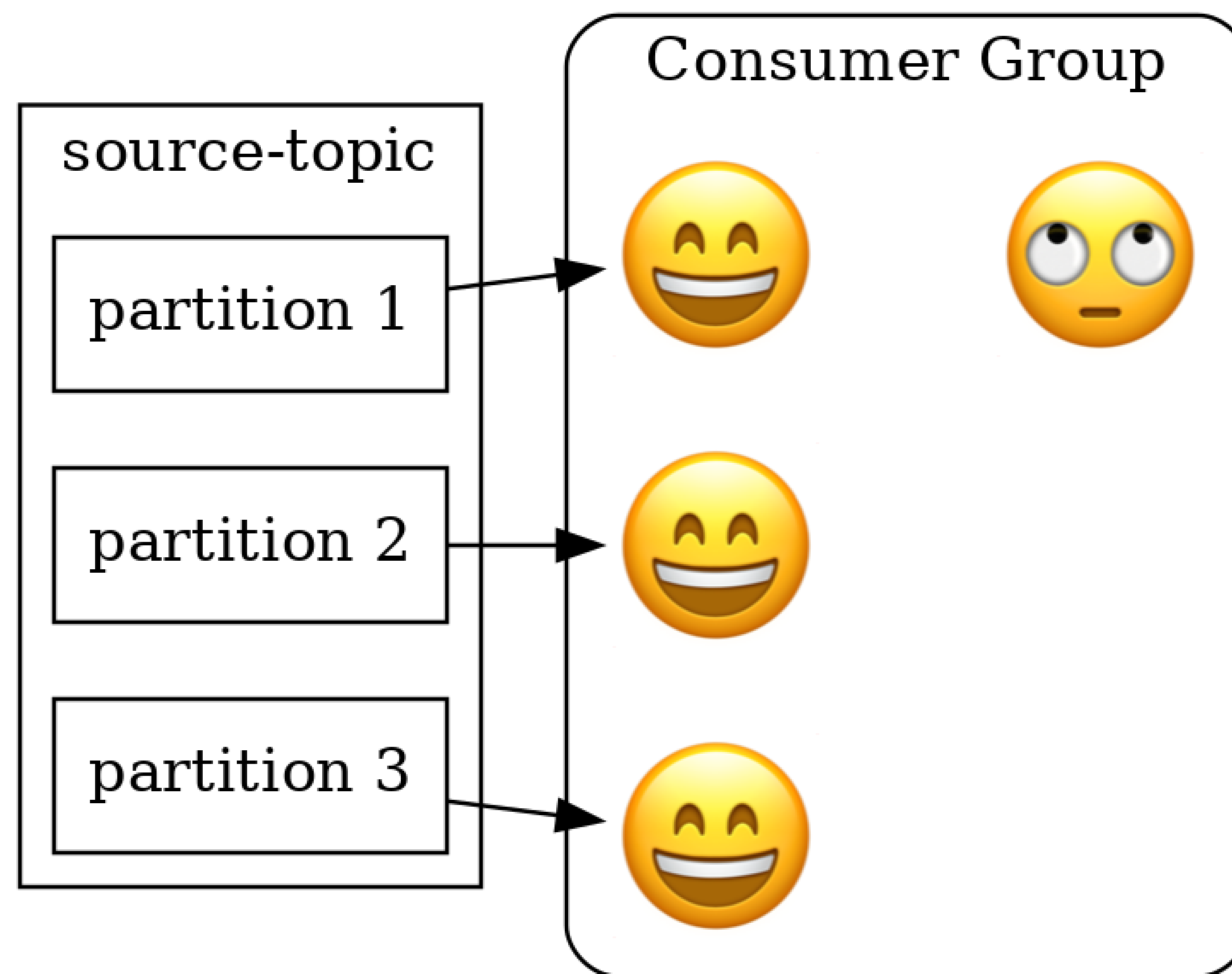
Чтение из Кафки



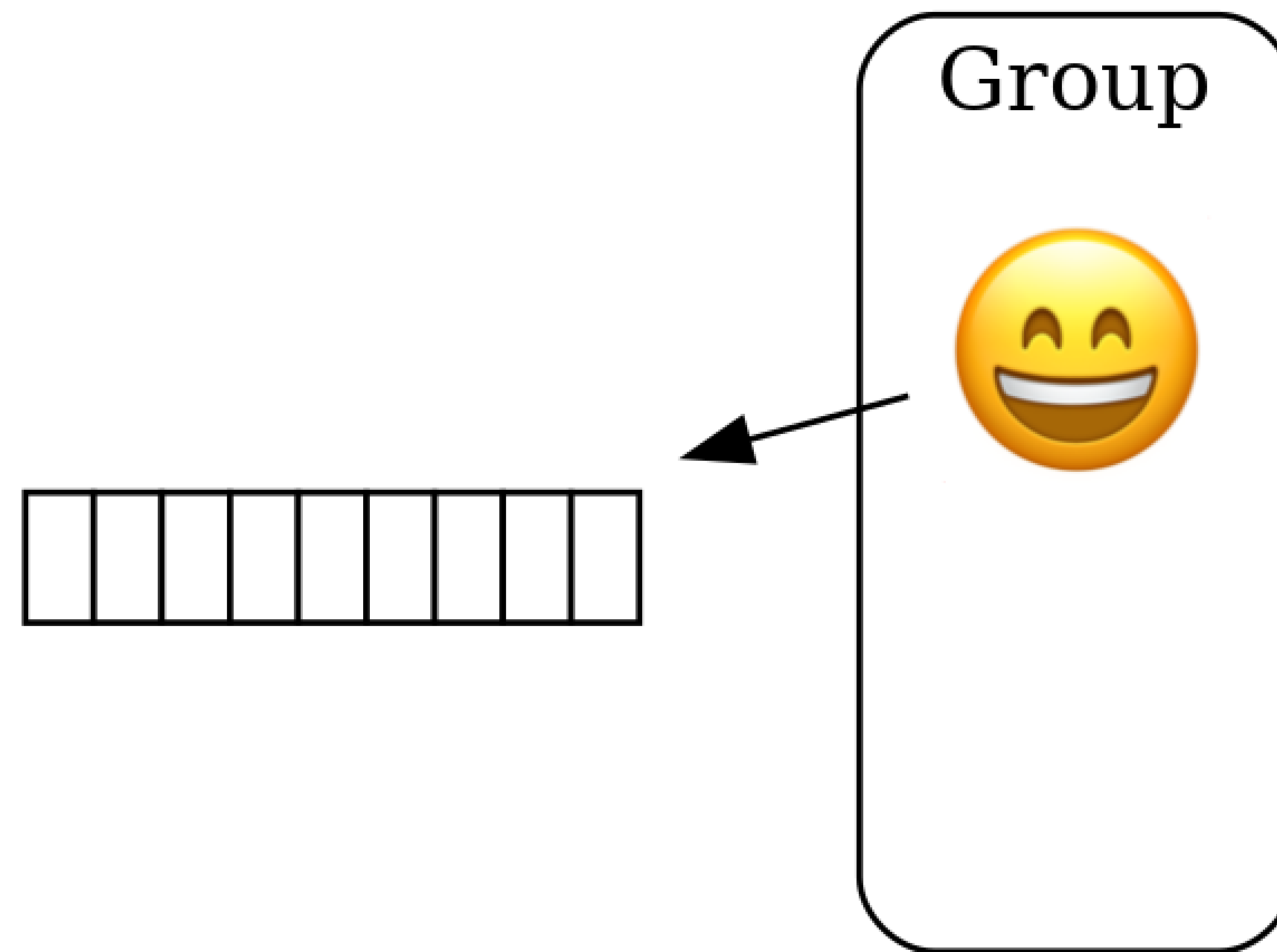
Чтение из Кафки



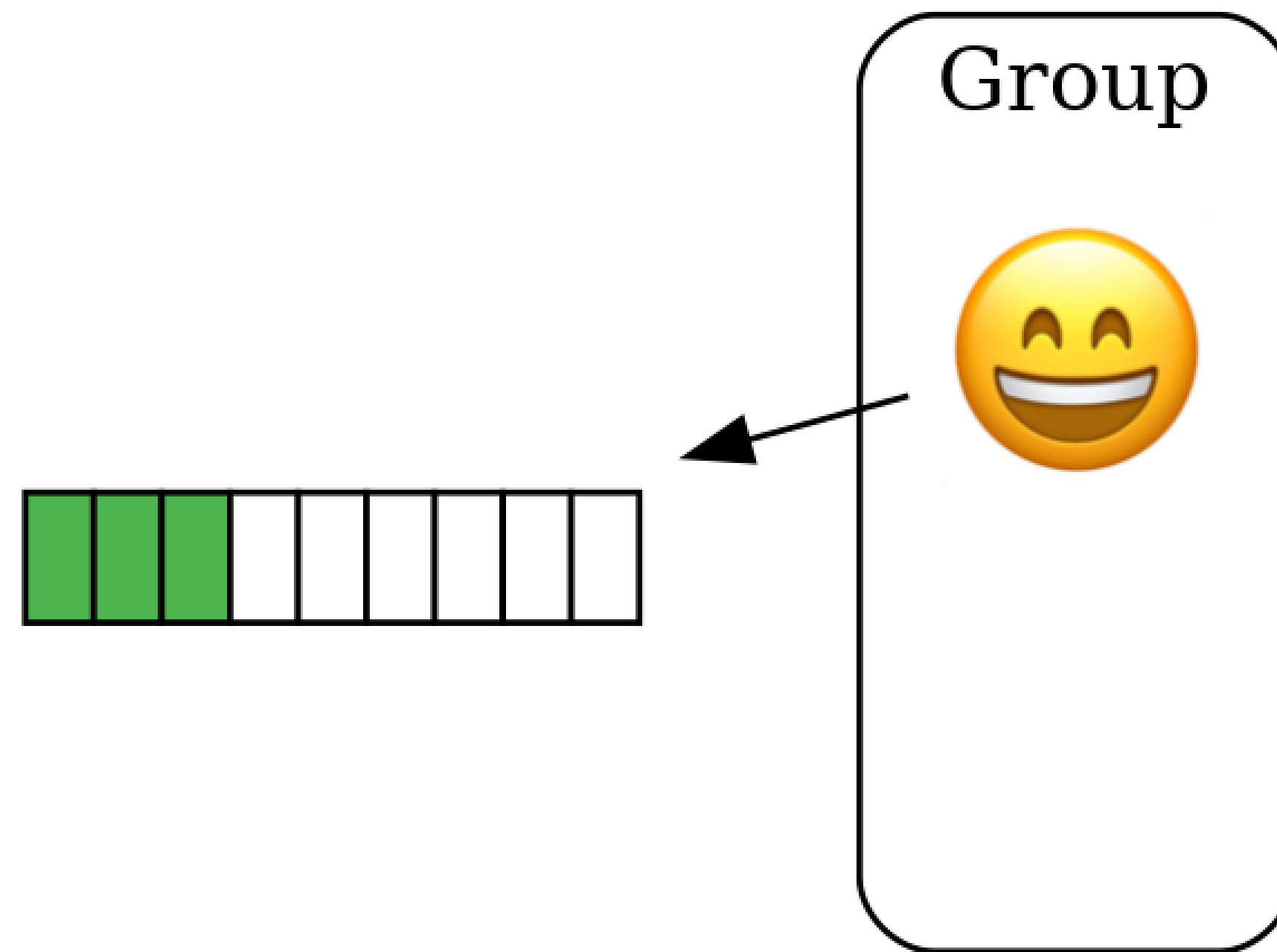
Чтение из Кафки



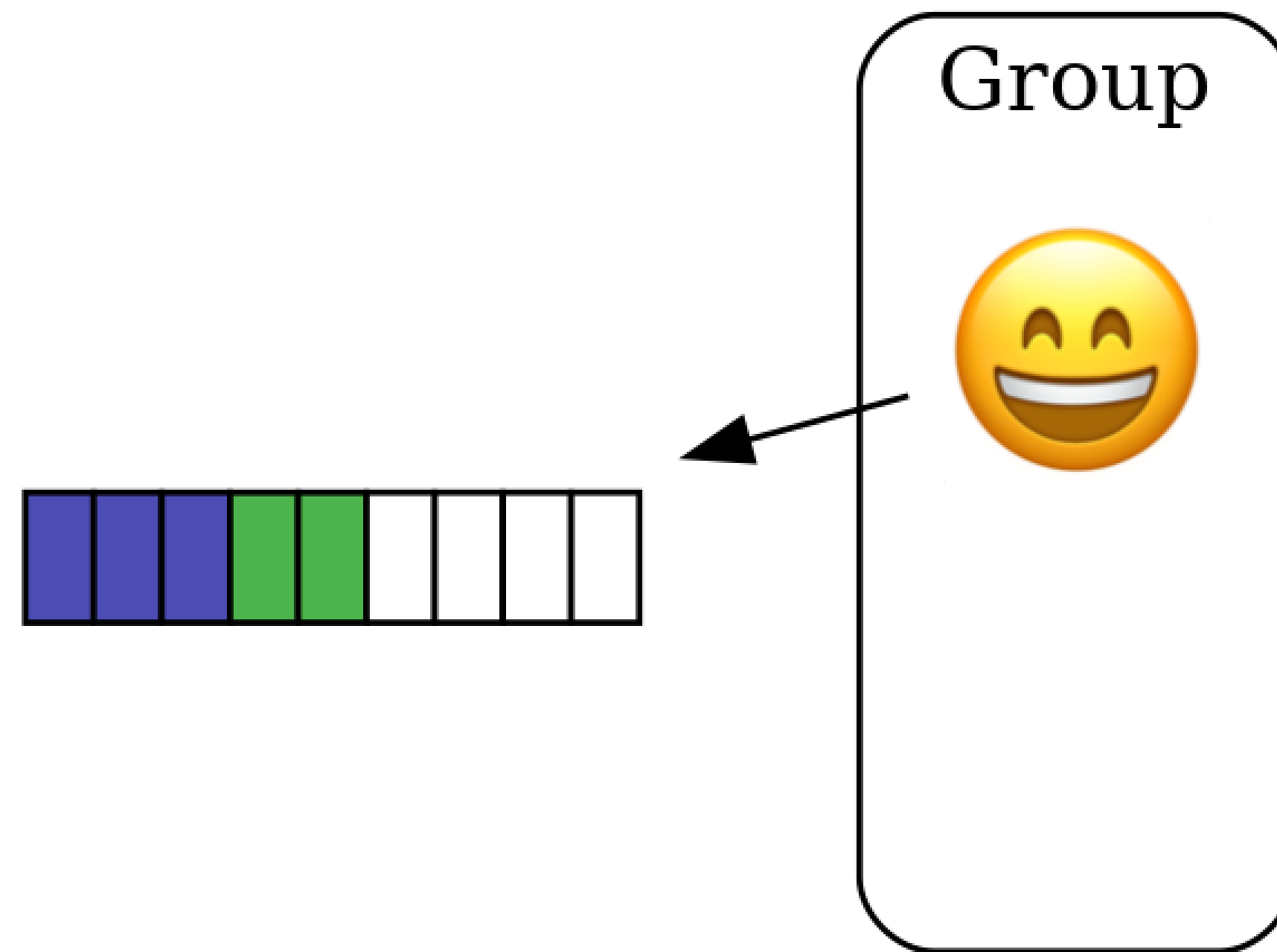
Offset Commit



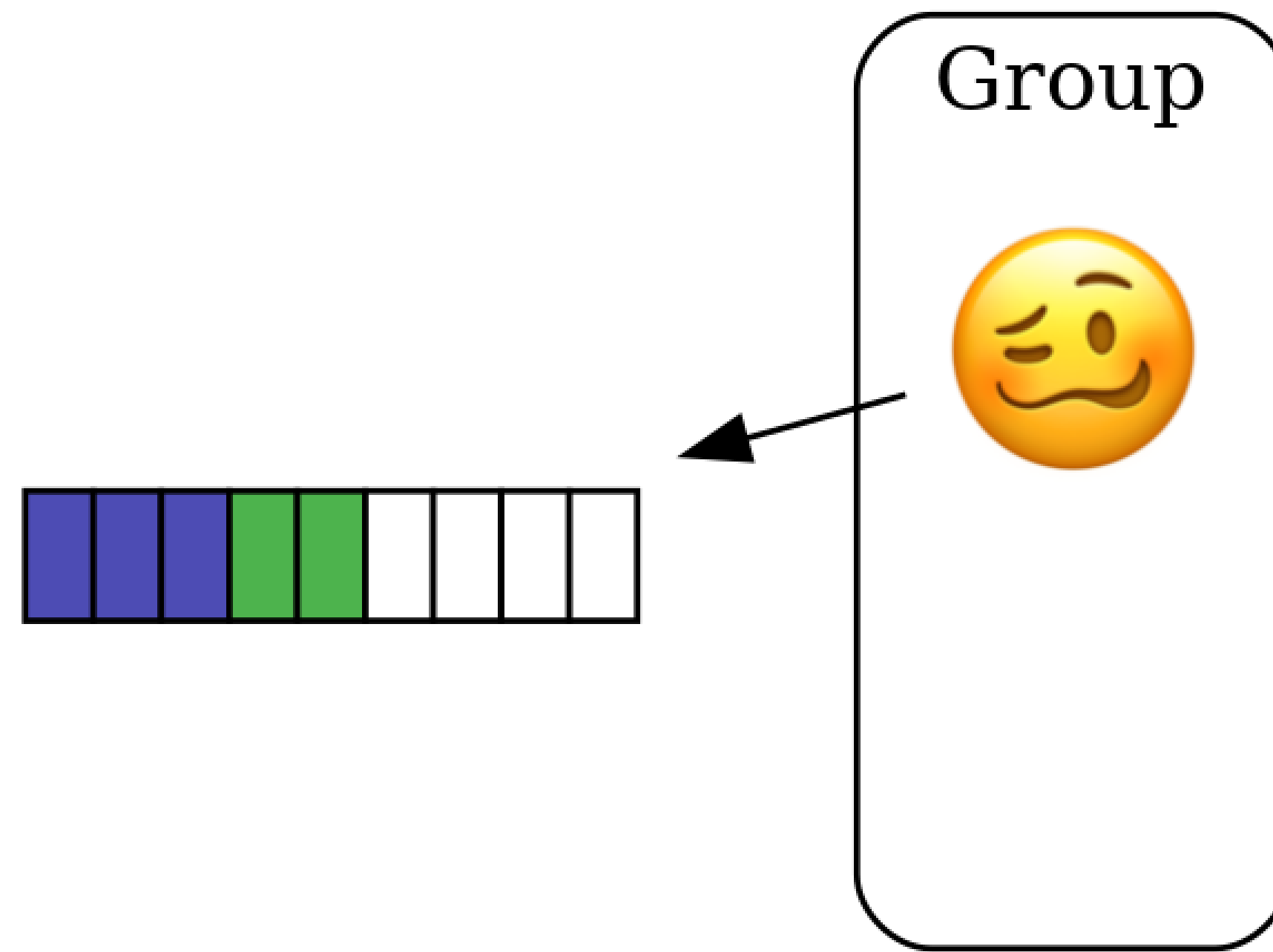
Offset Commit



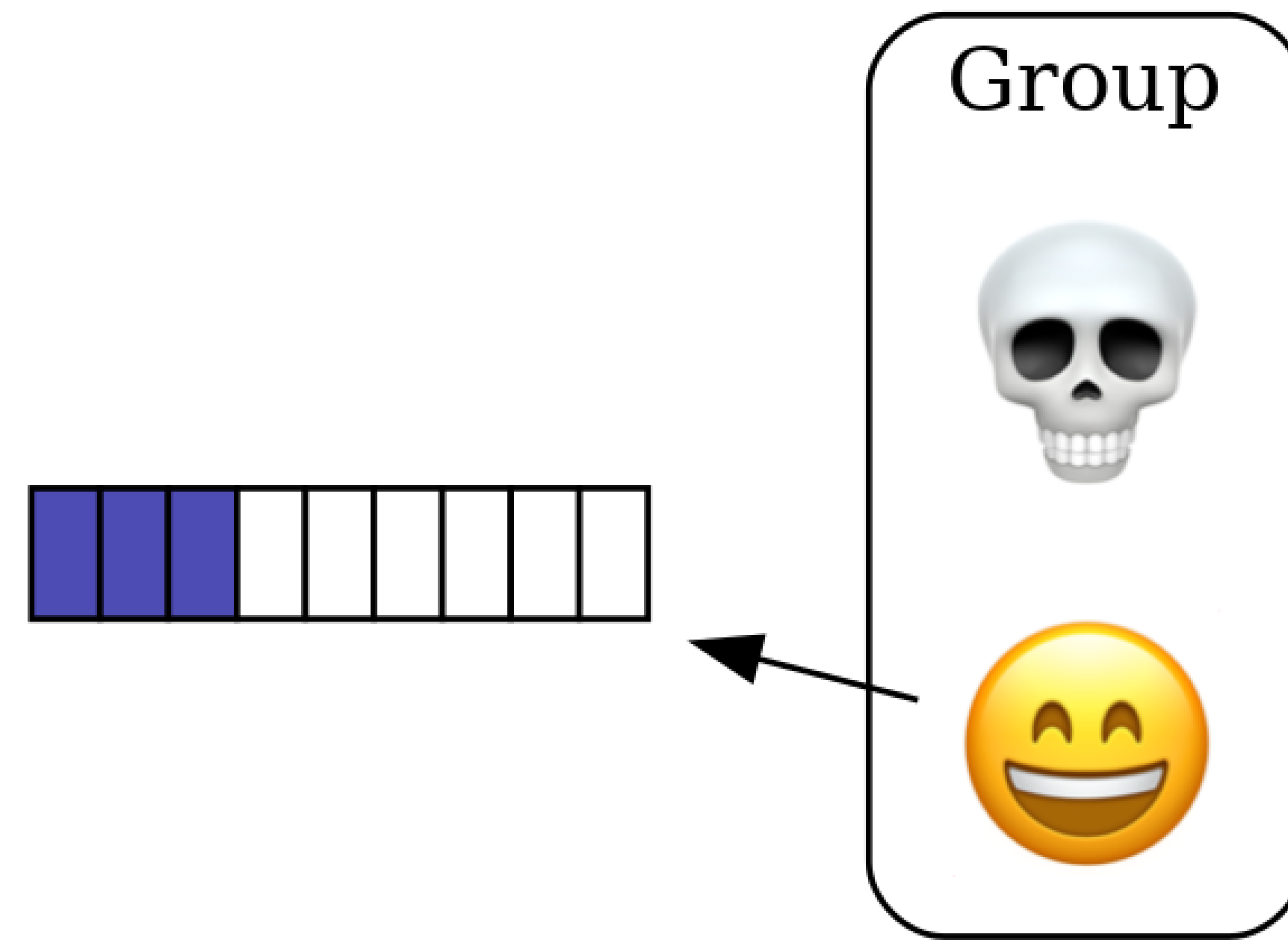
Offset Commit



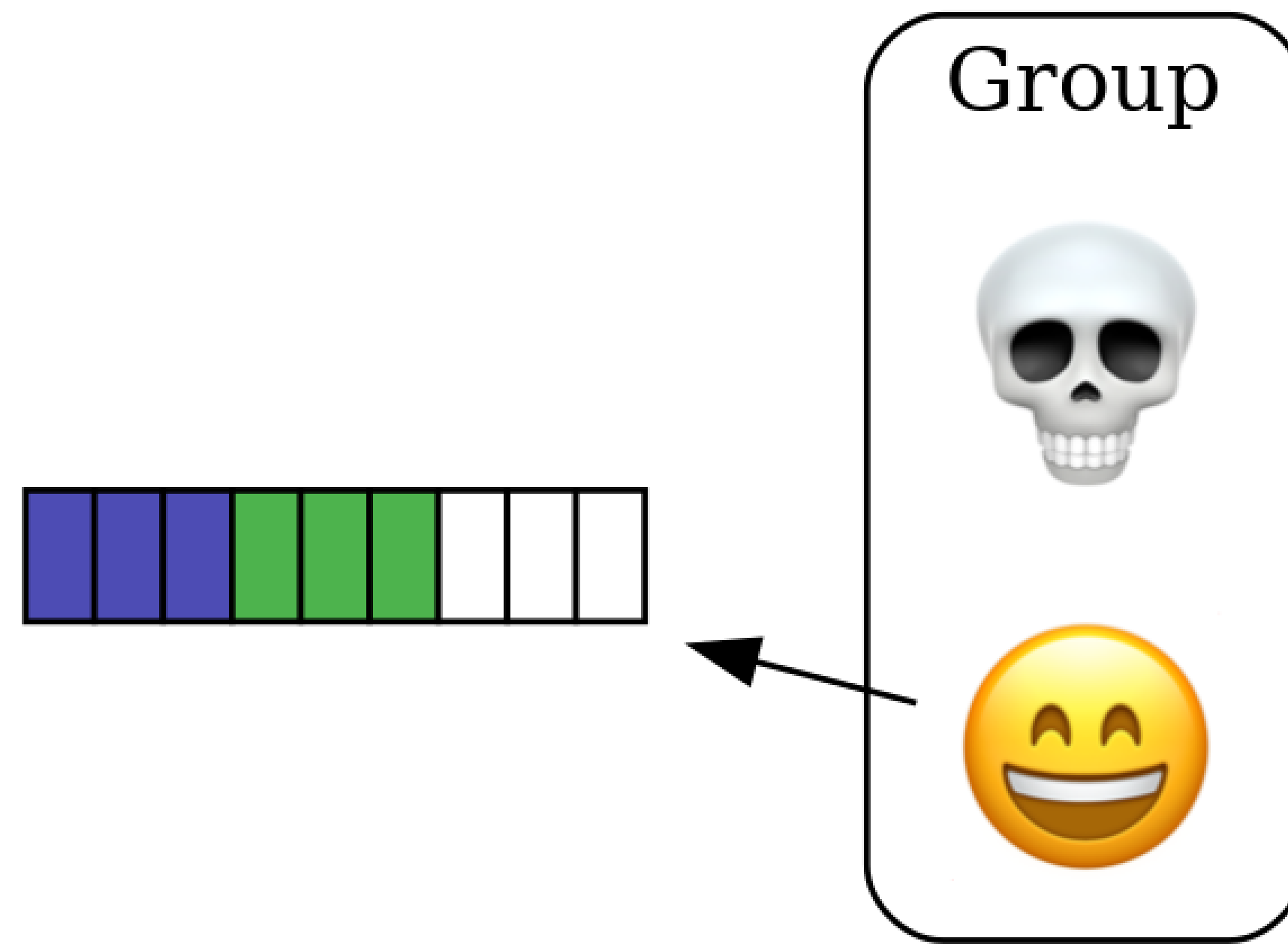
Offset Commit



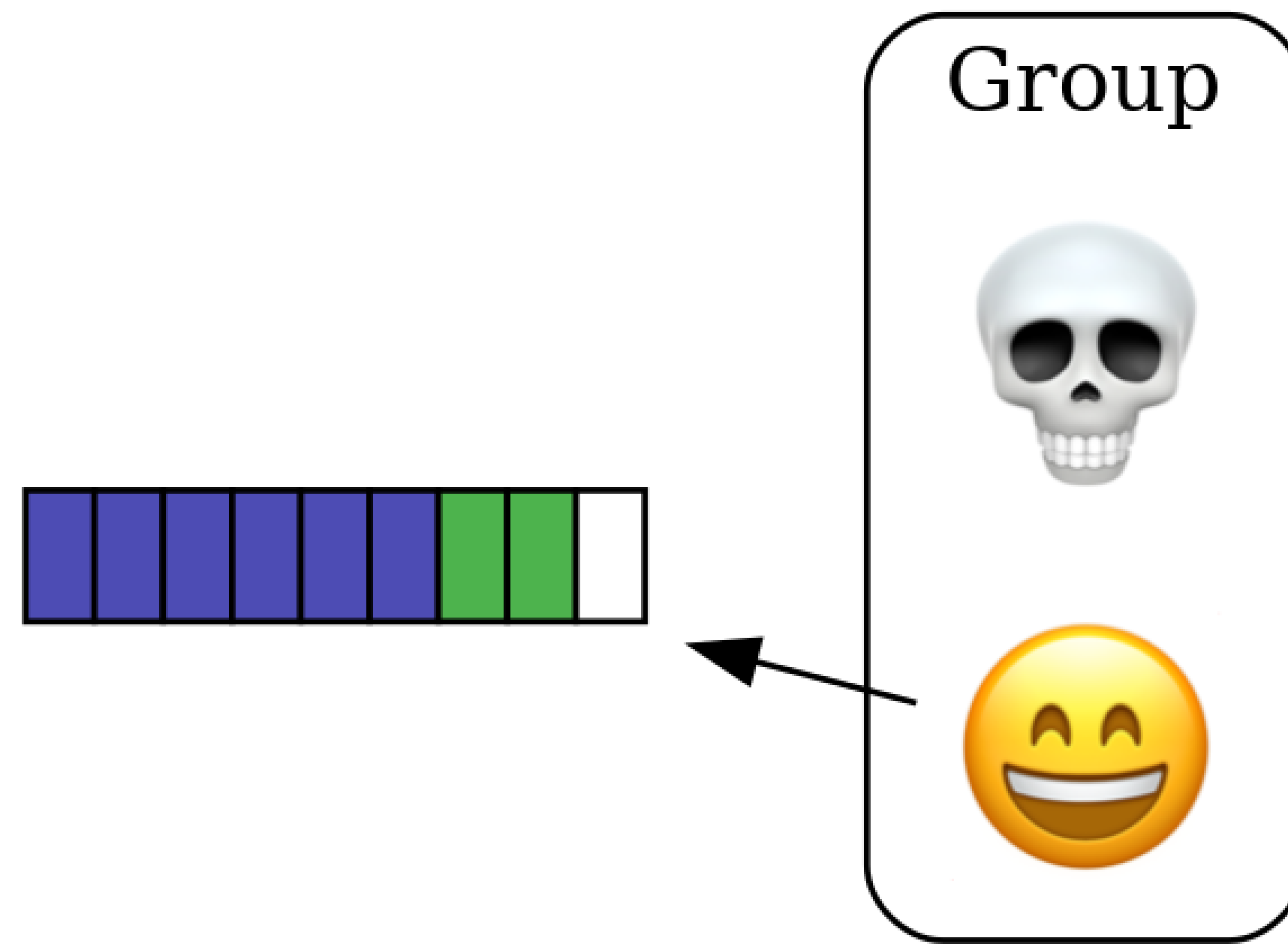
Offset Commit



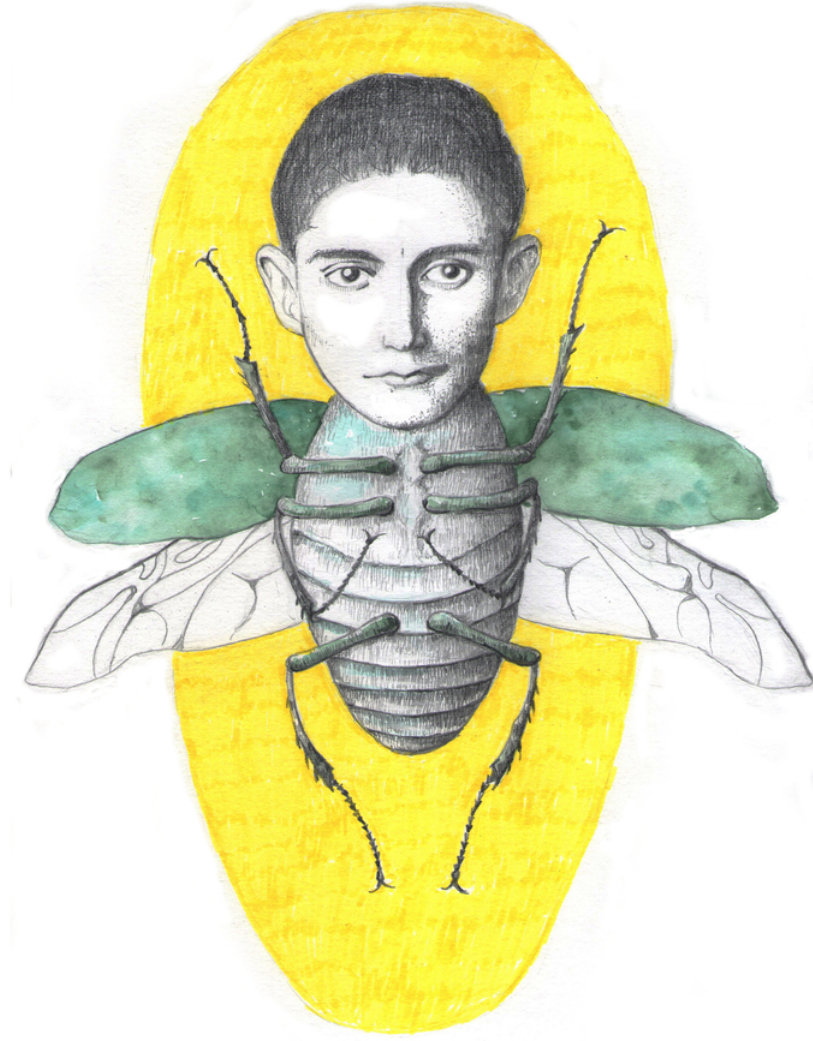
Offset Commit



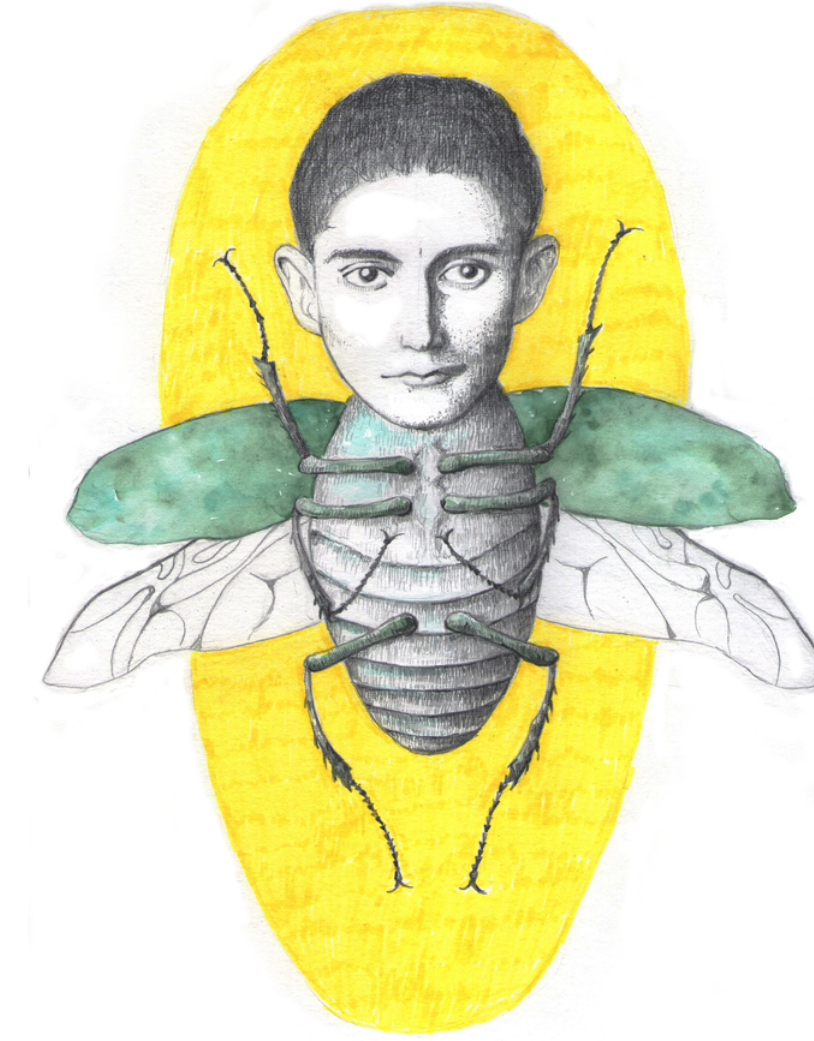
Offset Commit



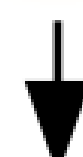
Наш план



- ~~1. Что такое лог и почему это важно~~
2. Что такое Kafka и что она умеет
 - ~~1. Общая информация~~
 - ~~2. Как устроен кластер~~
 - ~~3. Как устроена запись~~
 - ~~4. Как устроено чтение~~
 5. **Retention и компактификация**
3. Что такое потоковая архитектура и на что способны потоковые обработчики
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня

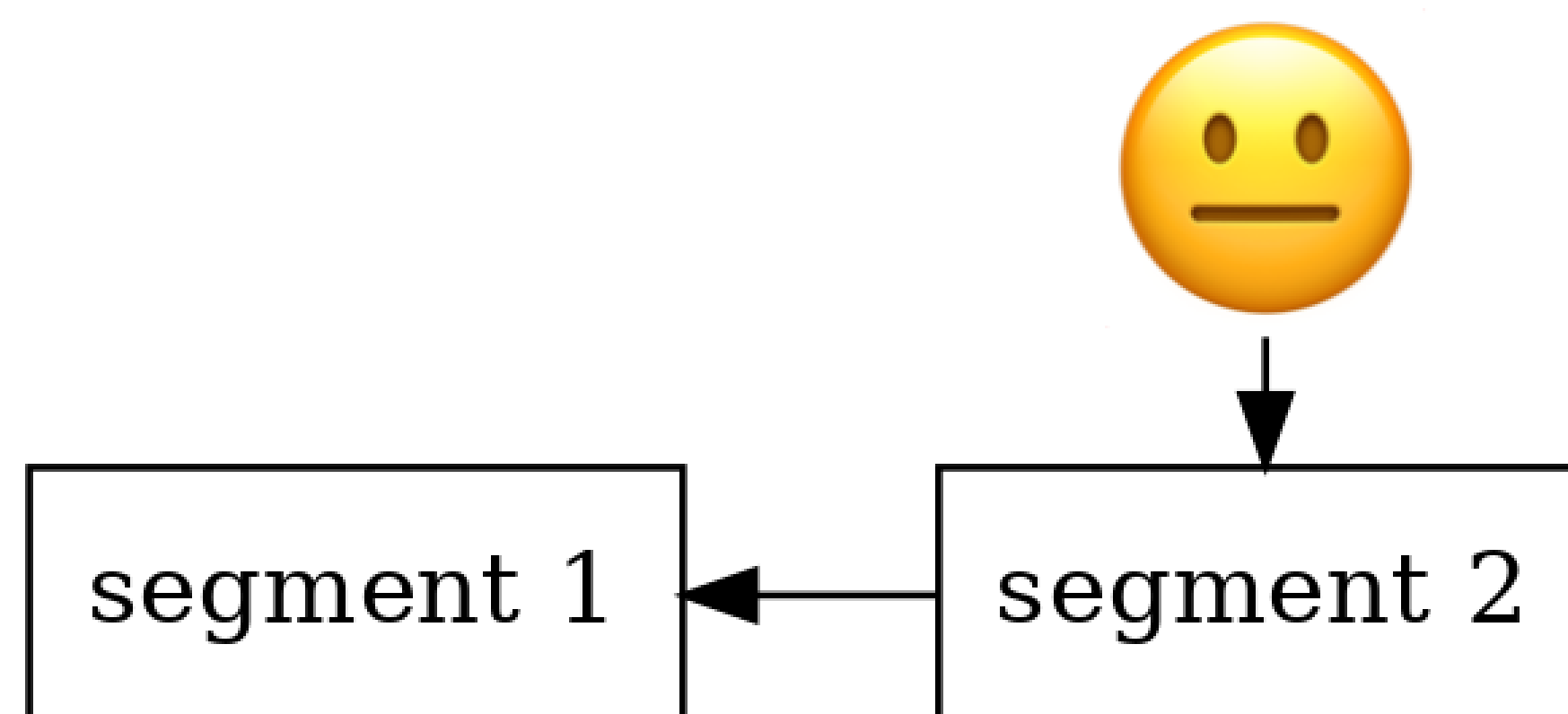


Как работает retention

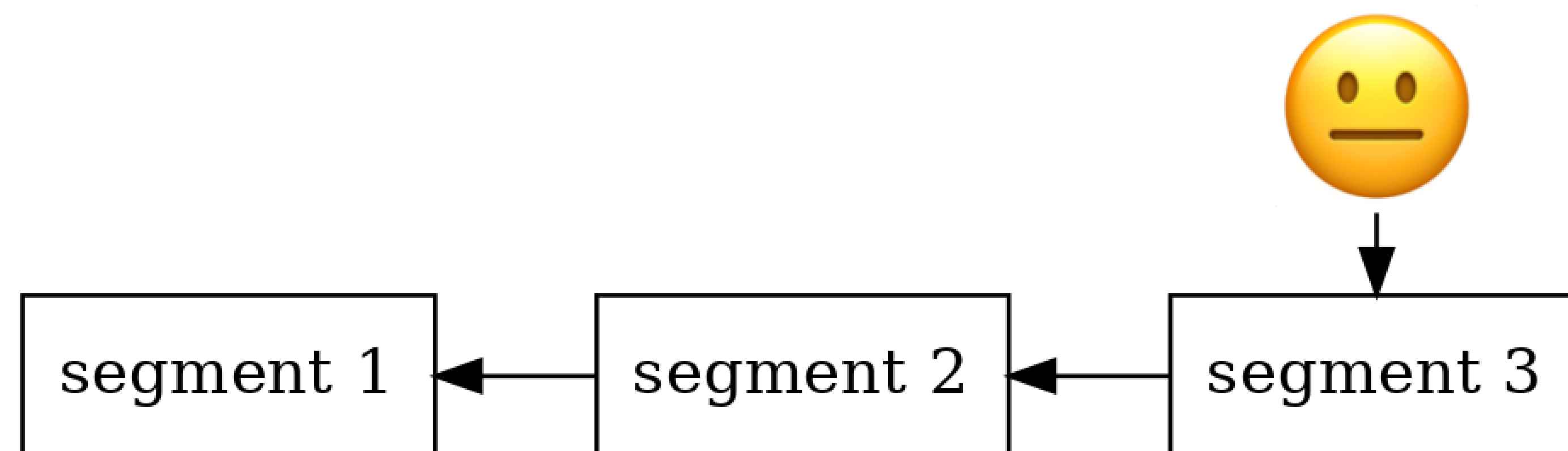


segment 1

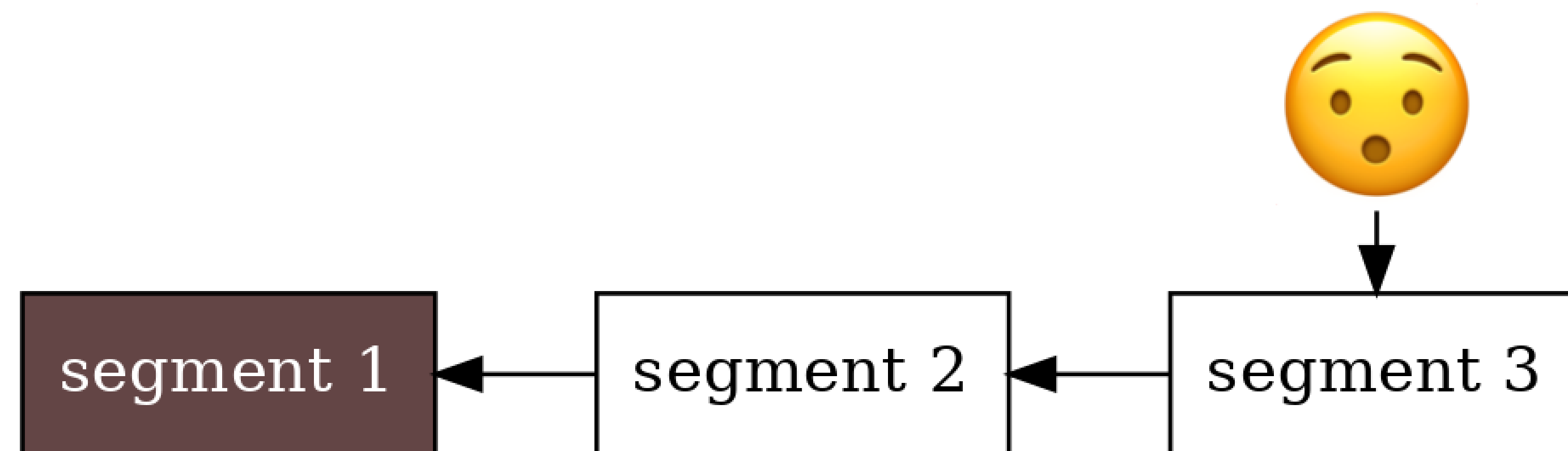
Как работает retention



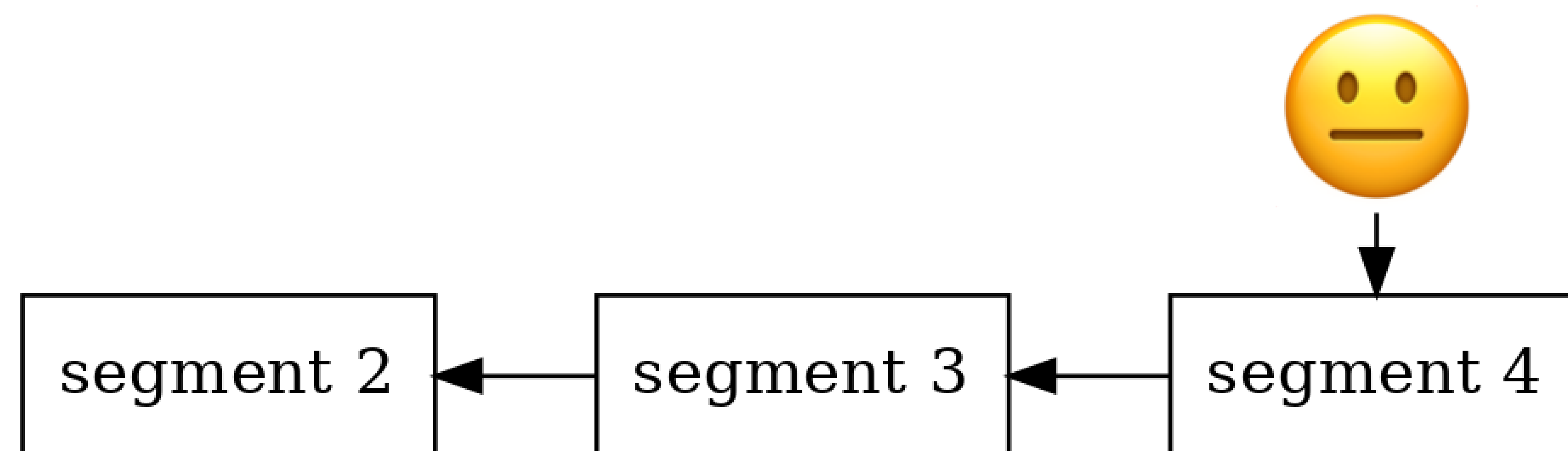
Как работает retention



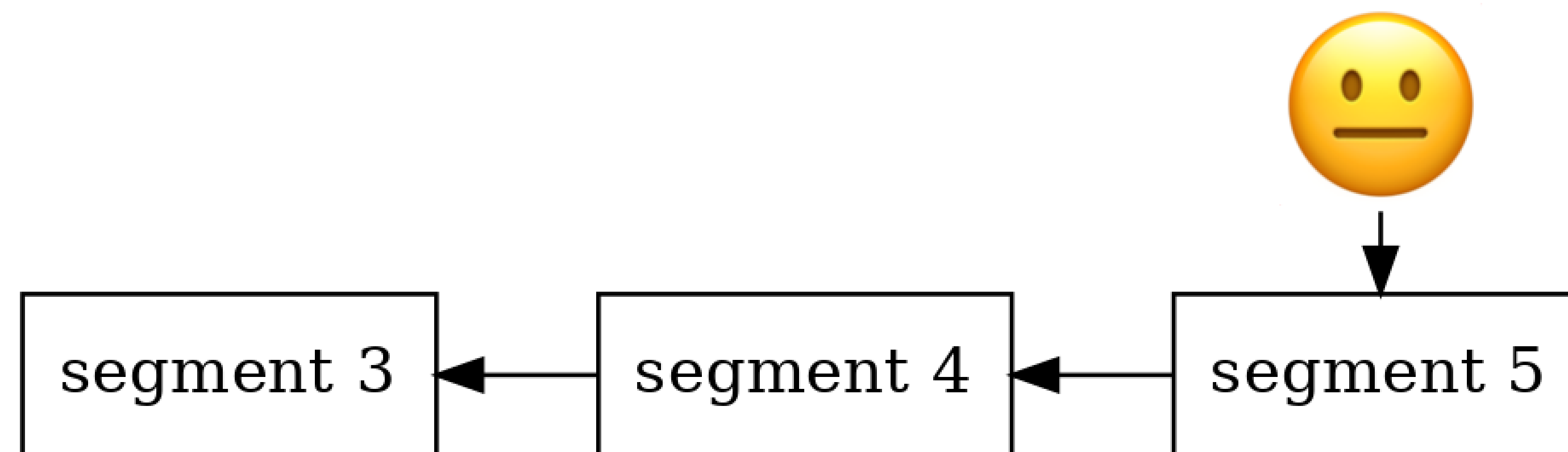
Как работает retention

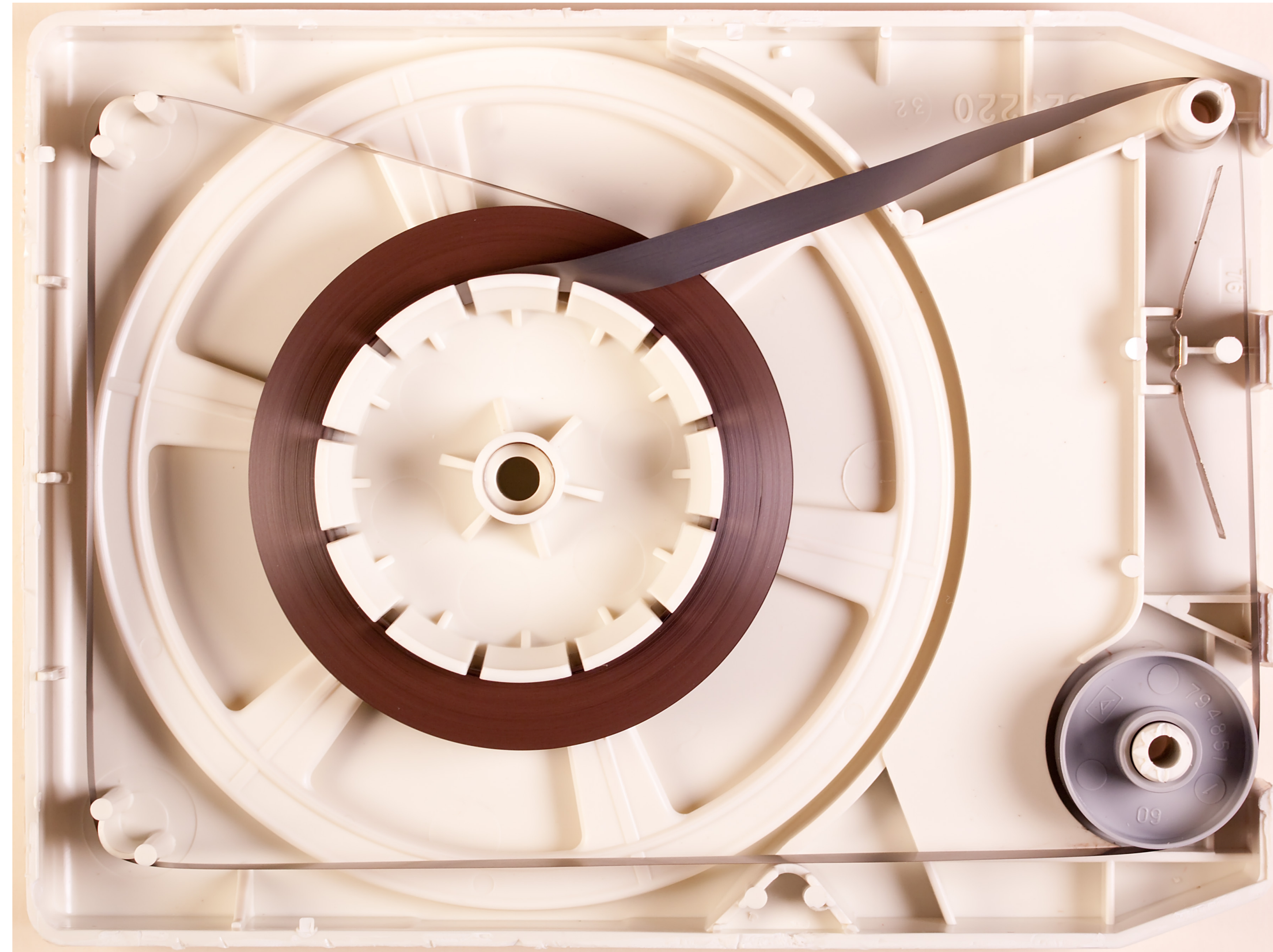


Как работает retention

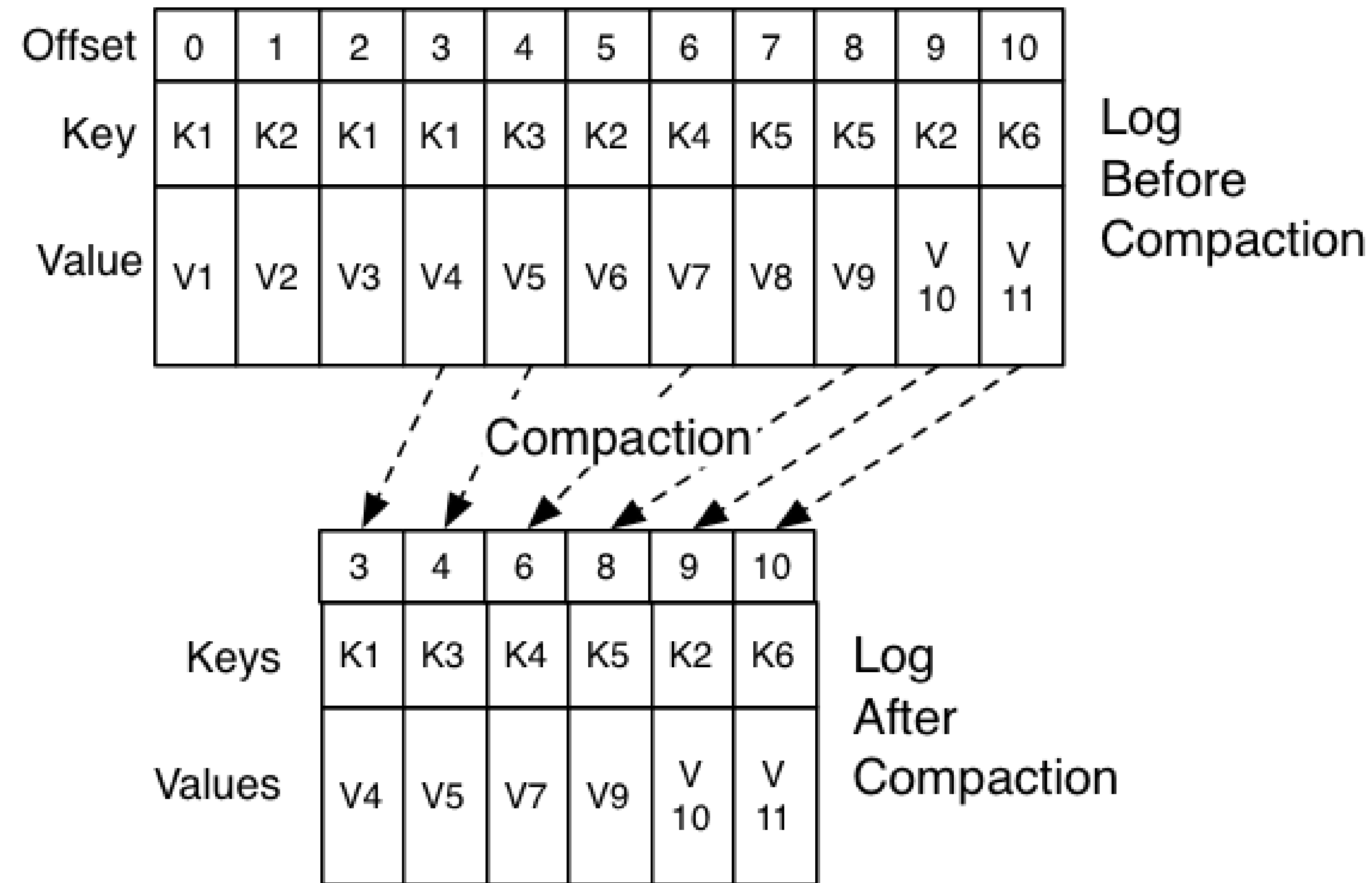


Как работает retention



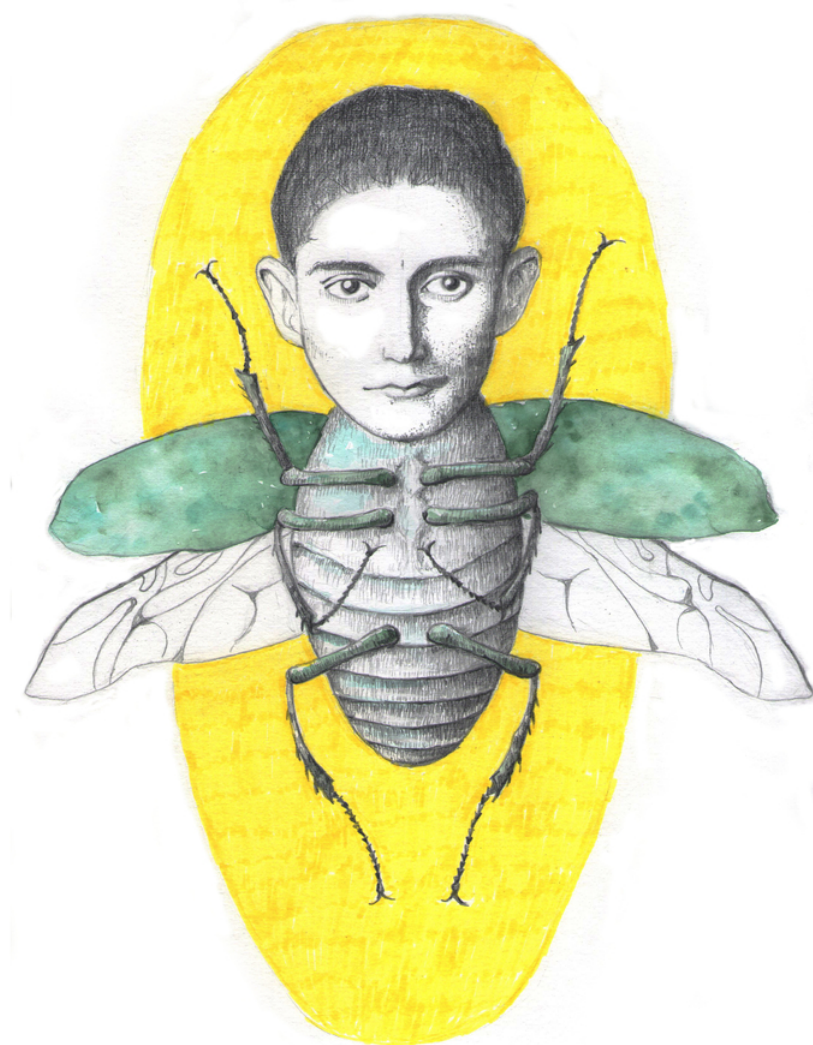


Компактификация топиков

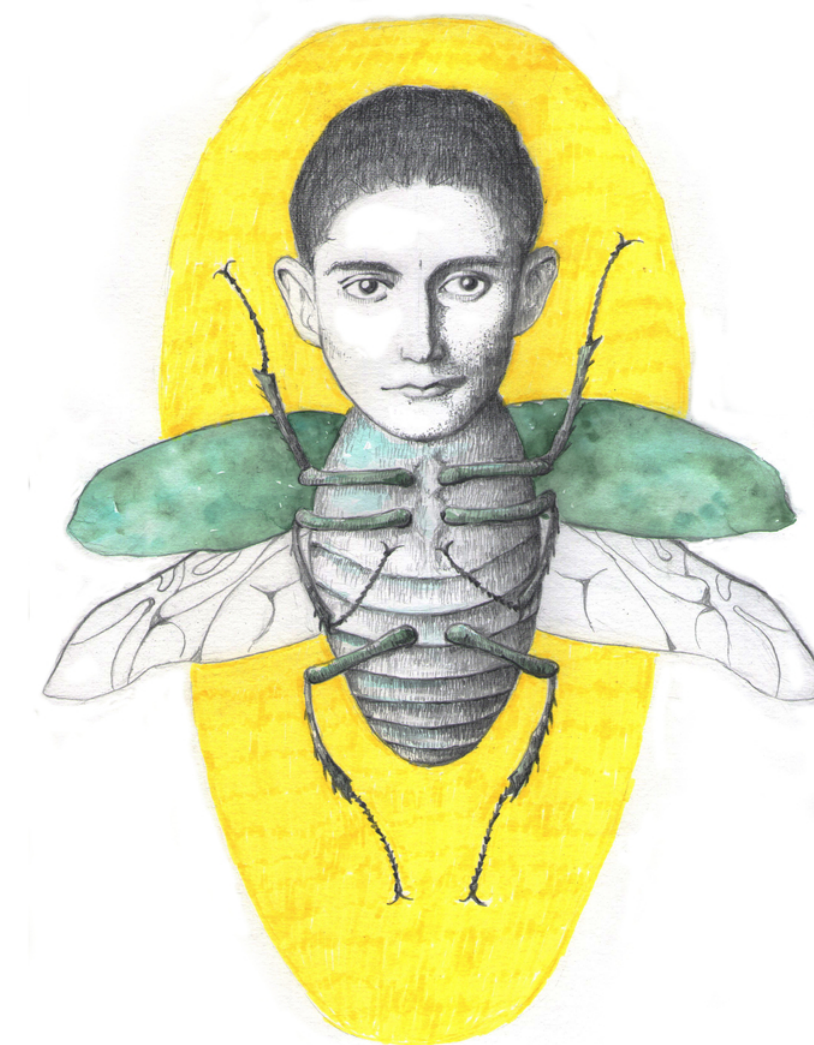


Источник: Kafka Documentation

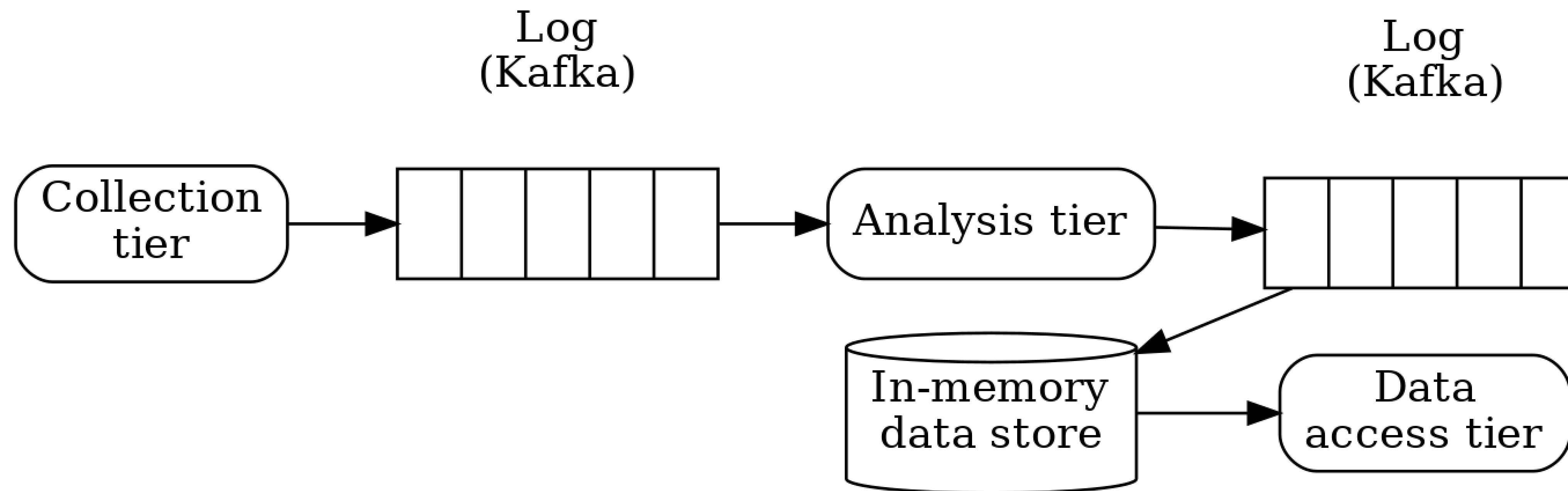
Наш план



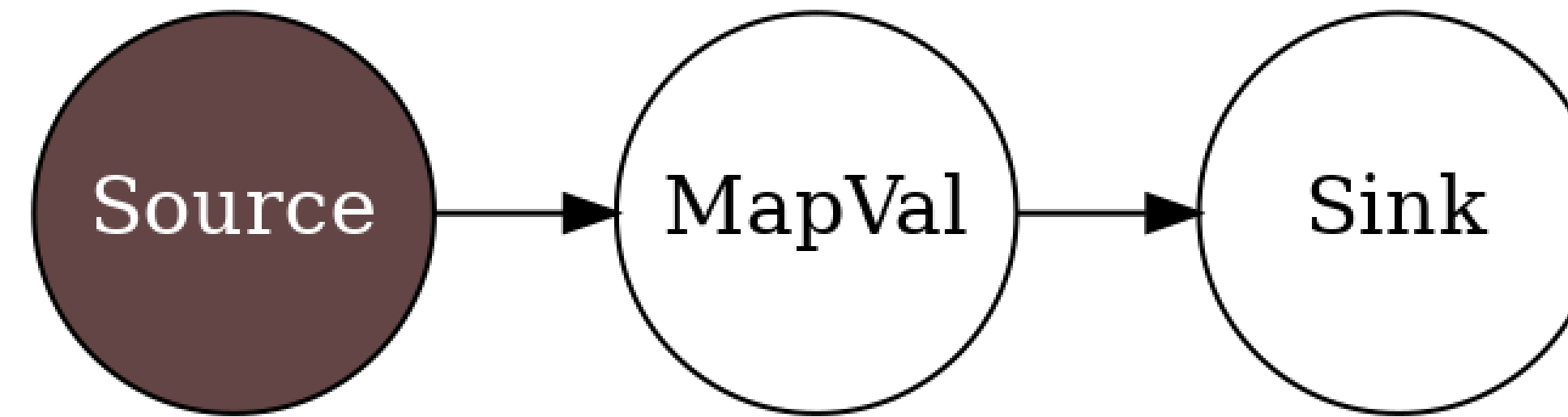
1. ~~Что такое лог и почему это важно~~
2. ~~Что такое Kafka и что она умеет~~
3. **Что такое потоковая архитектура и на что способны потоковые обработчики**
4. Kafka и JavaScript
5. Как попробовать Кафку сегодня



Потоковая обработка данных: архитектура

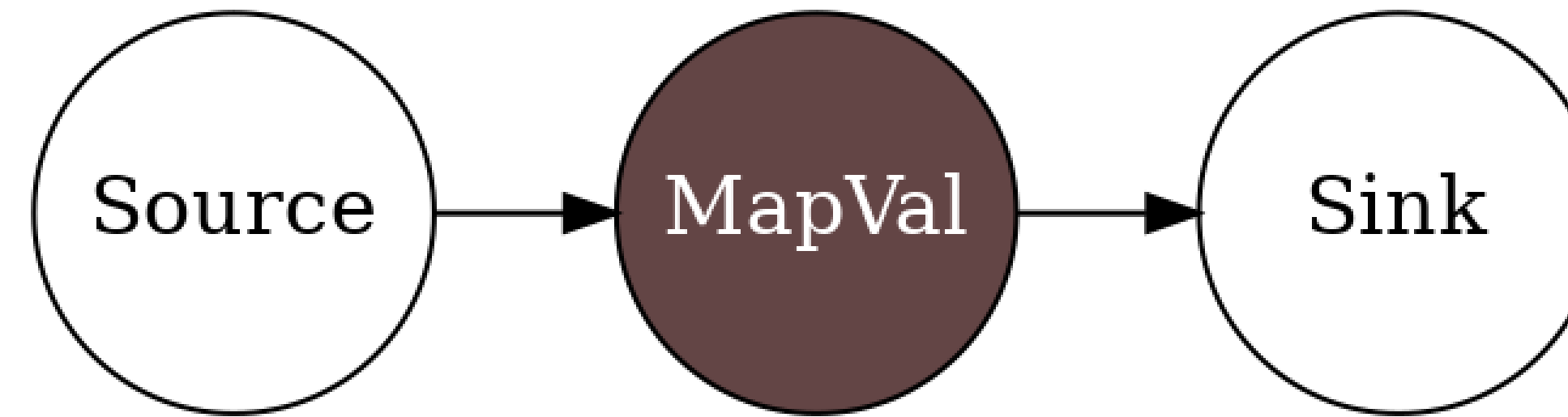


Stateless Transformation



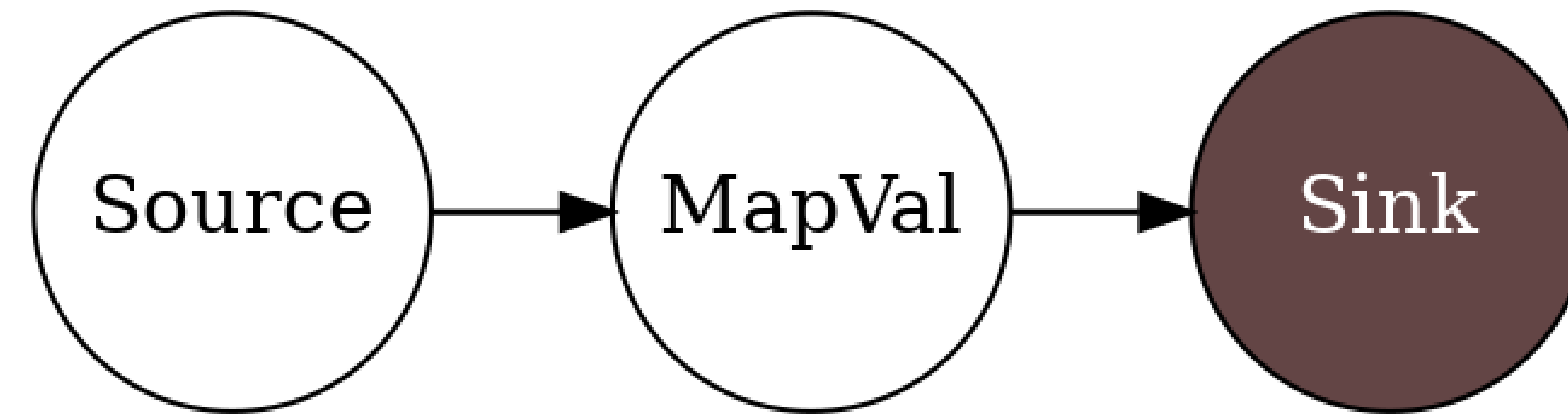
```
KStream<String, String> stream = streamsBuilder.stream(  
    SRC_TOPIC, Consumed.with(Serdes.String(), Serdes.String()));
```

Stateless Transformation



```
KStream<String, String> upperCasedStream =  
    stream.mapValues(String::toUpperCase);
```

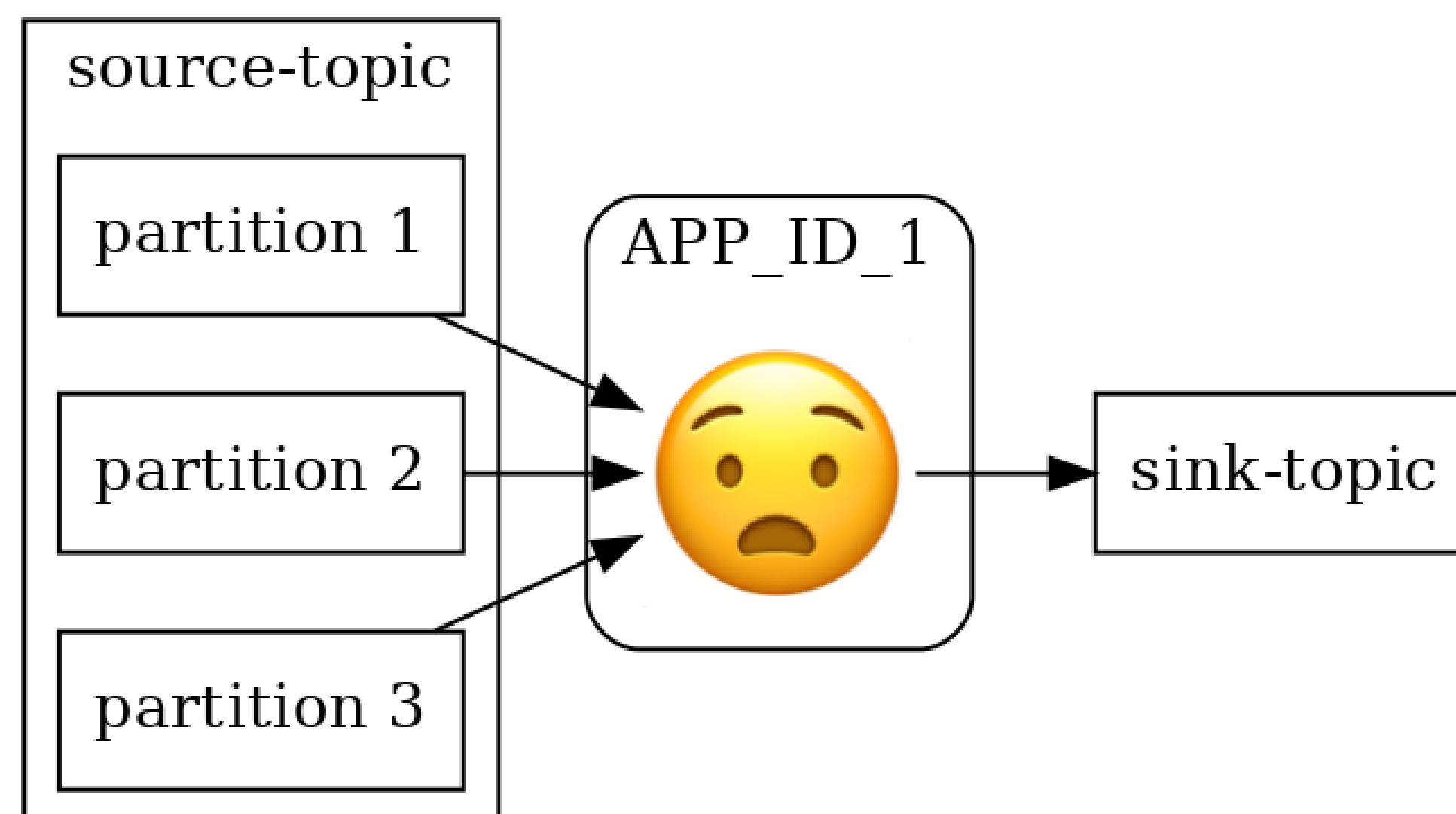

Stateless Transformation



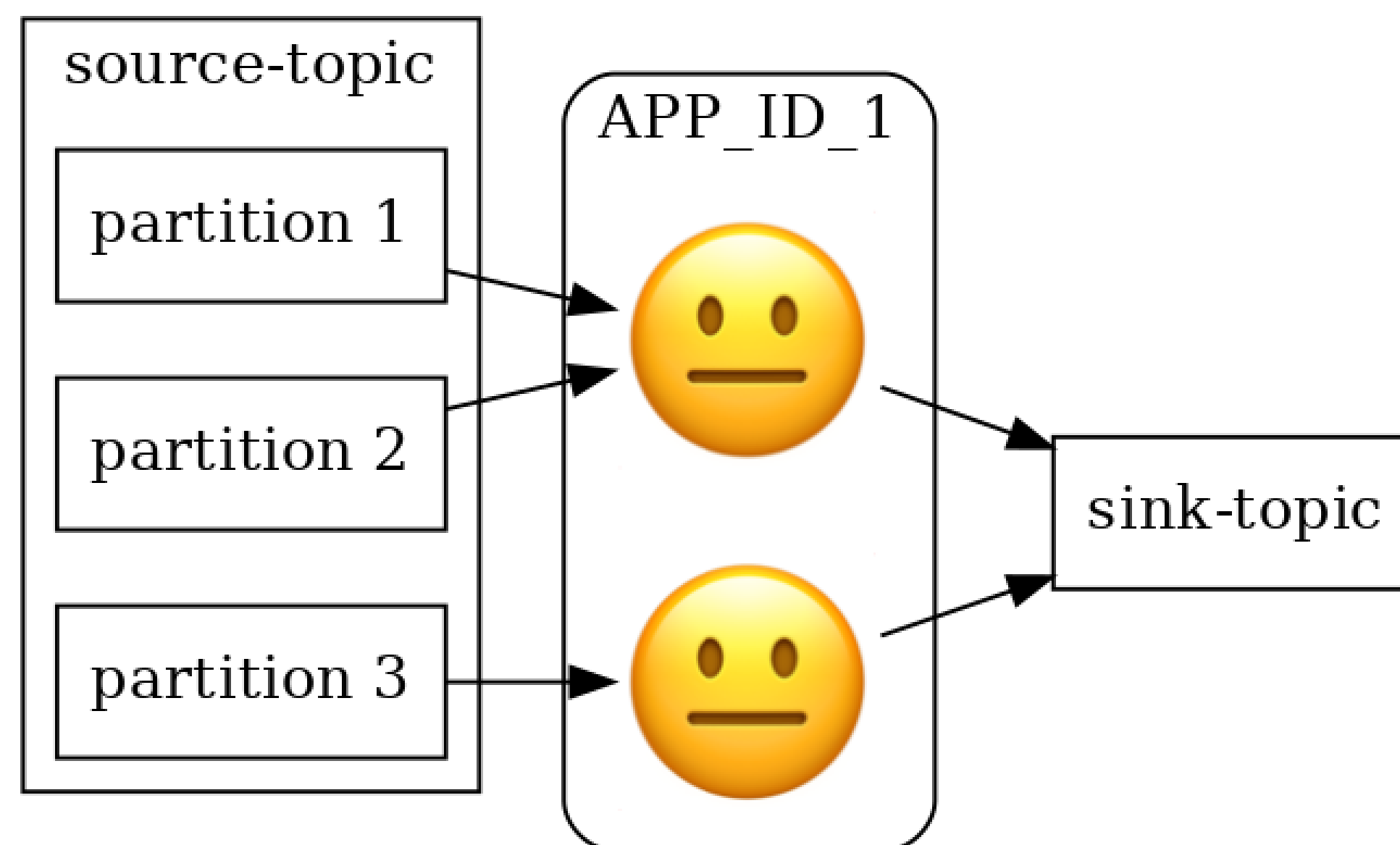
```
upperCasedStream.to(SINK_TOPIC,  
    Produced.with(Serdes.String(), Serdes.String()));
```

Три строчки кода, ну и что тут такого?

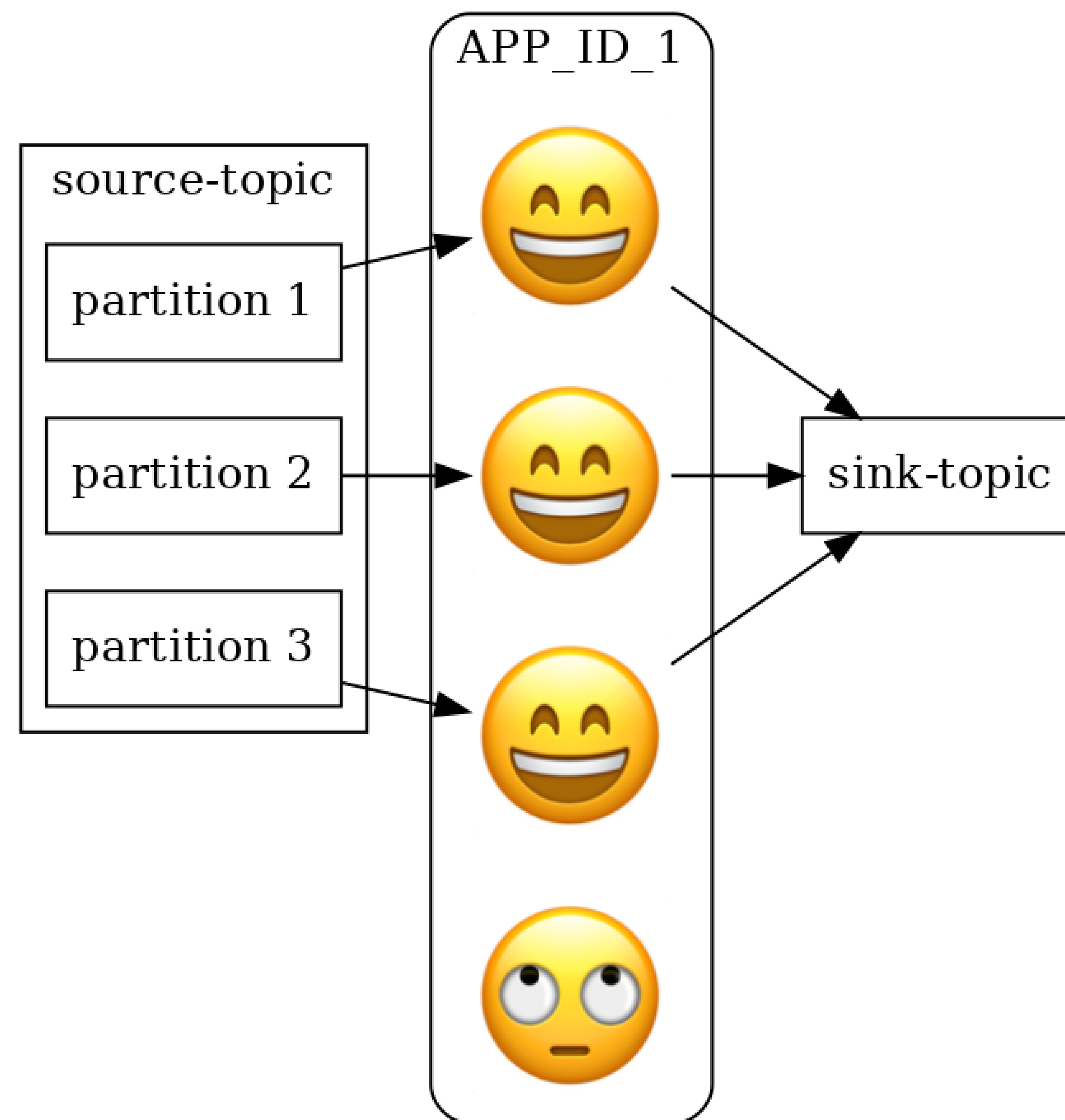
- ★ Больше сообщений в секунду? — больше машин с одинаковым `application.id`!



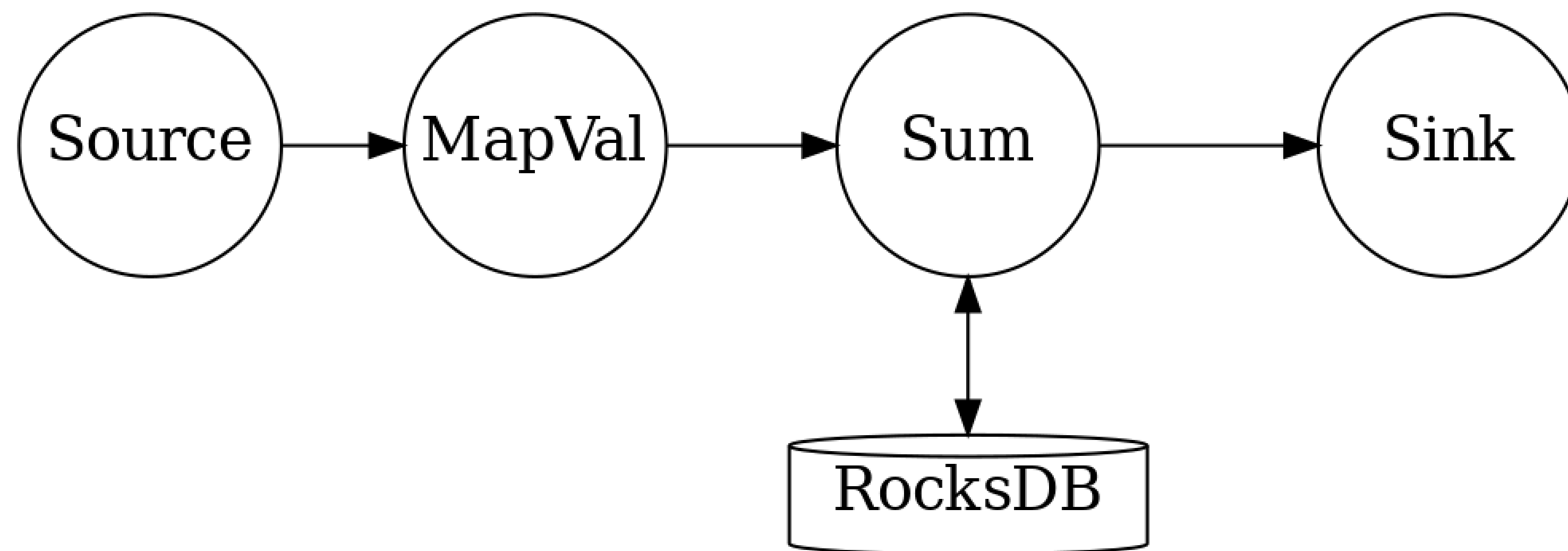
Добавляем ноды



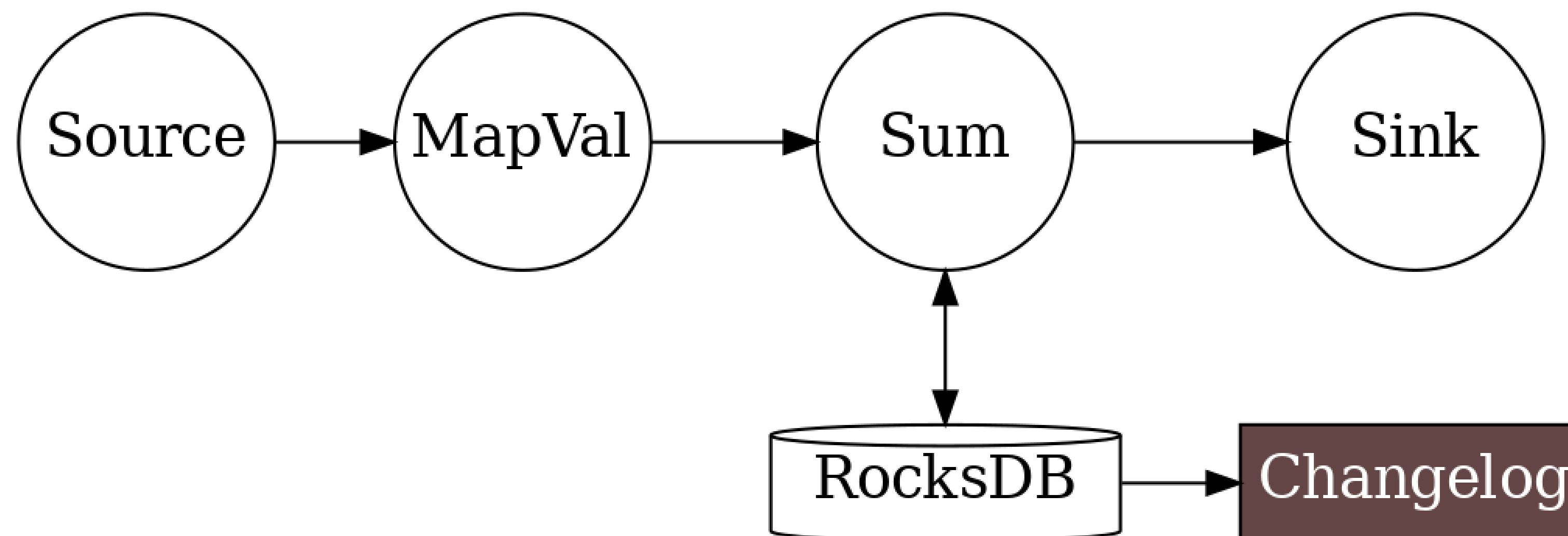
Ограничены только числом партиций



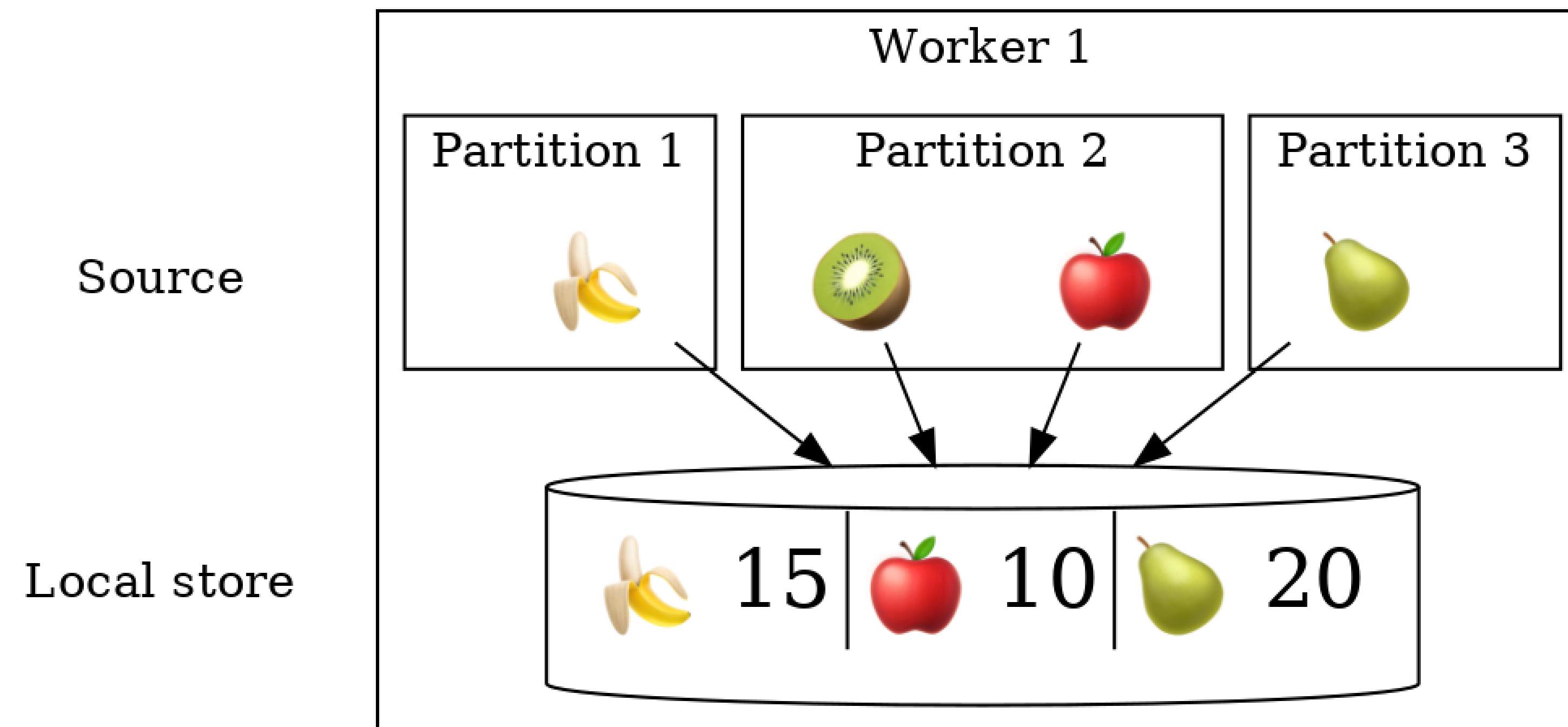
Магия Stateful Transformation



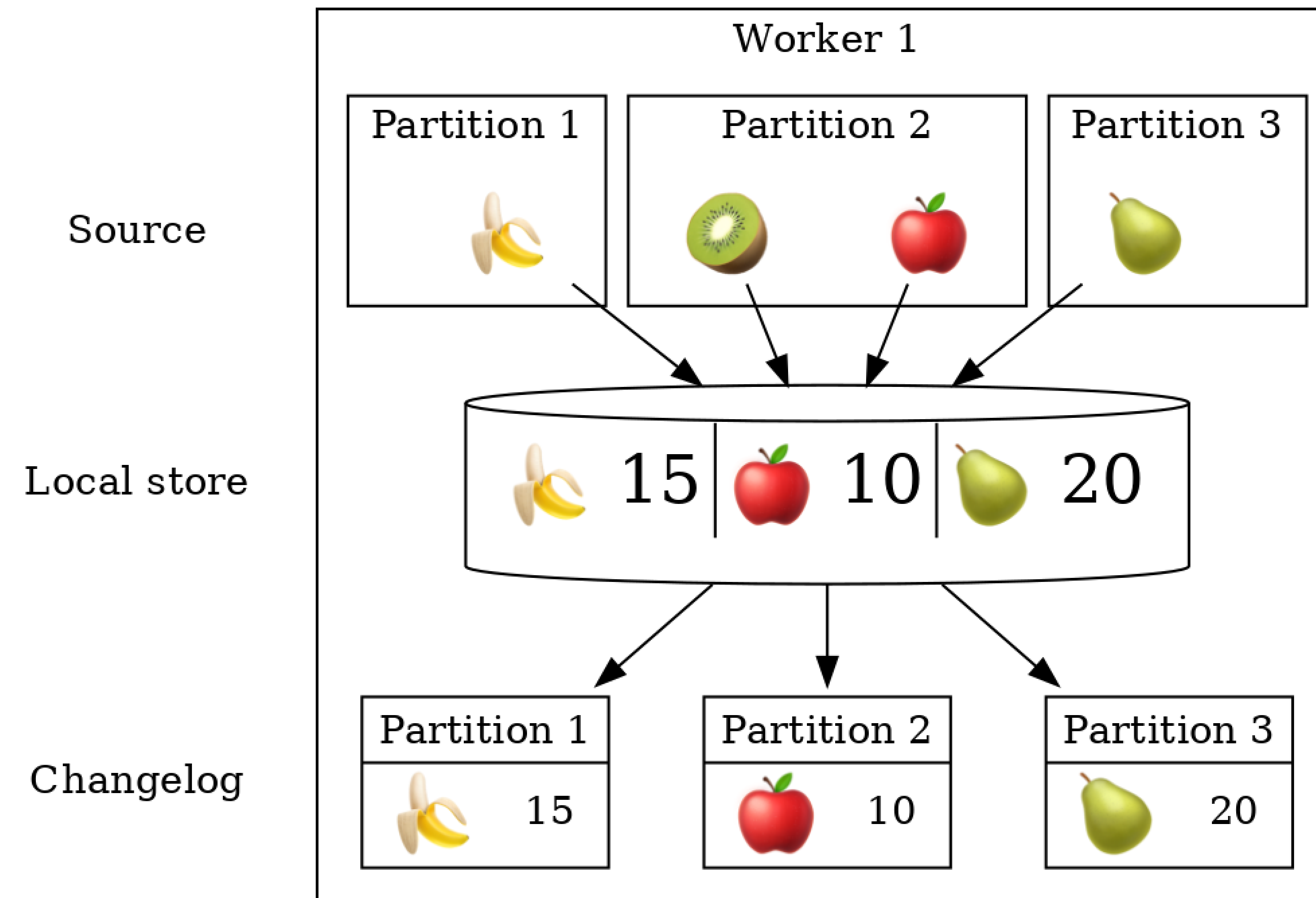
Изменения реплицируются в топик!



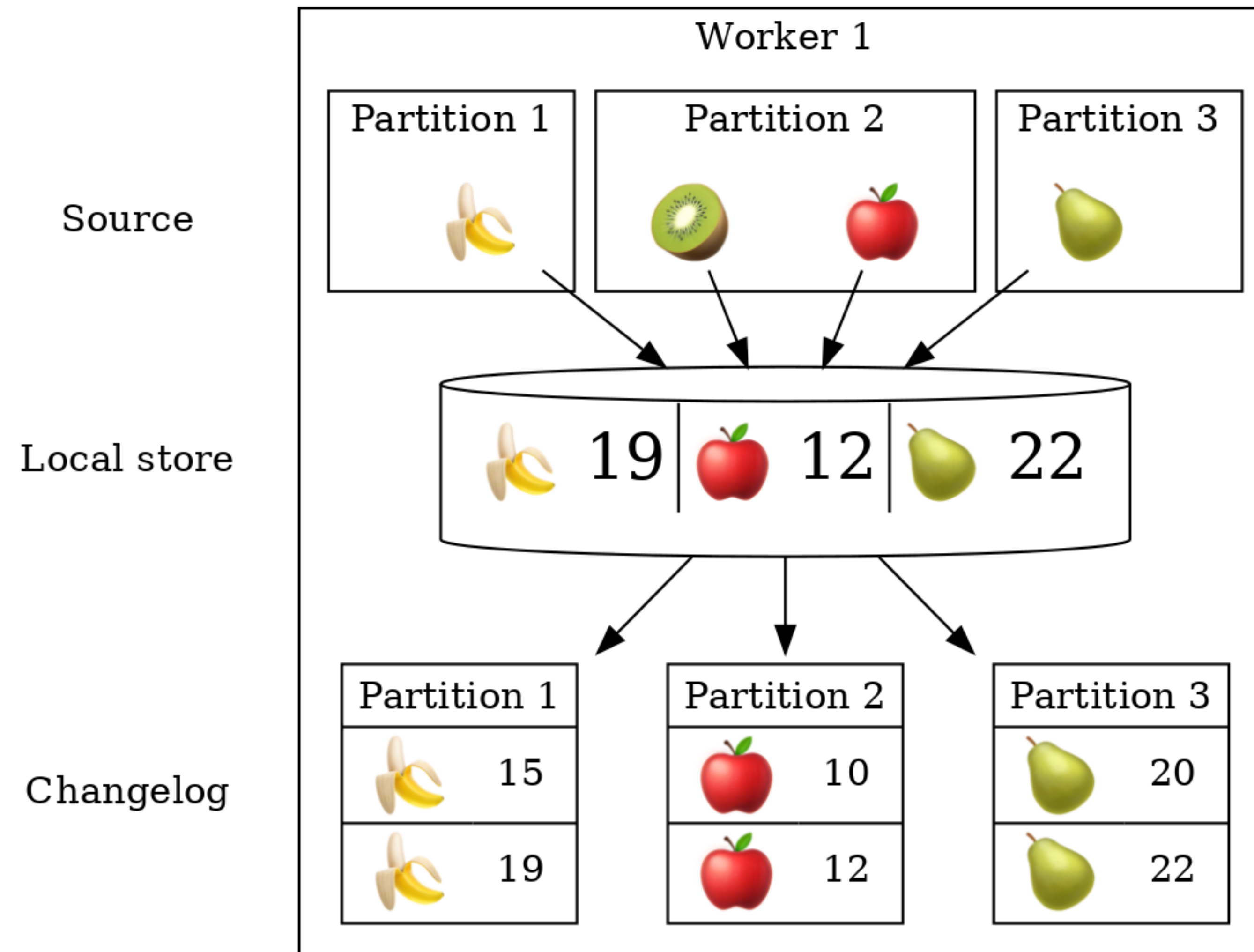
Партиционирование и local state



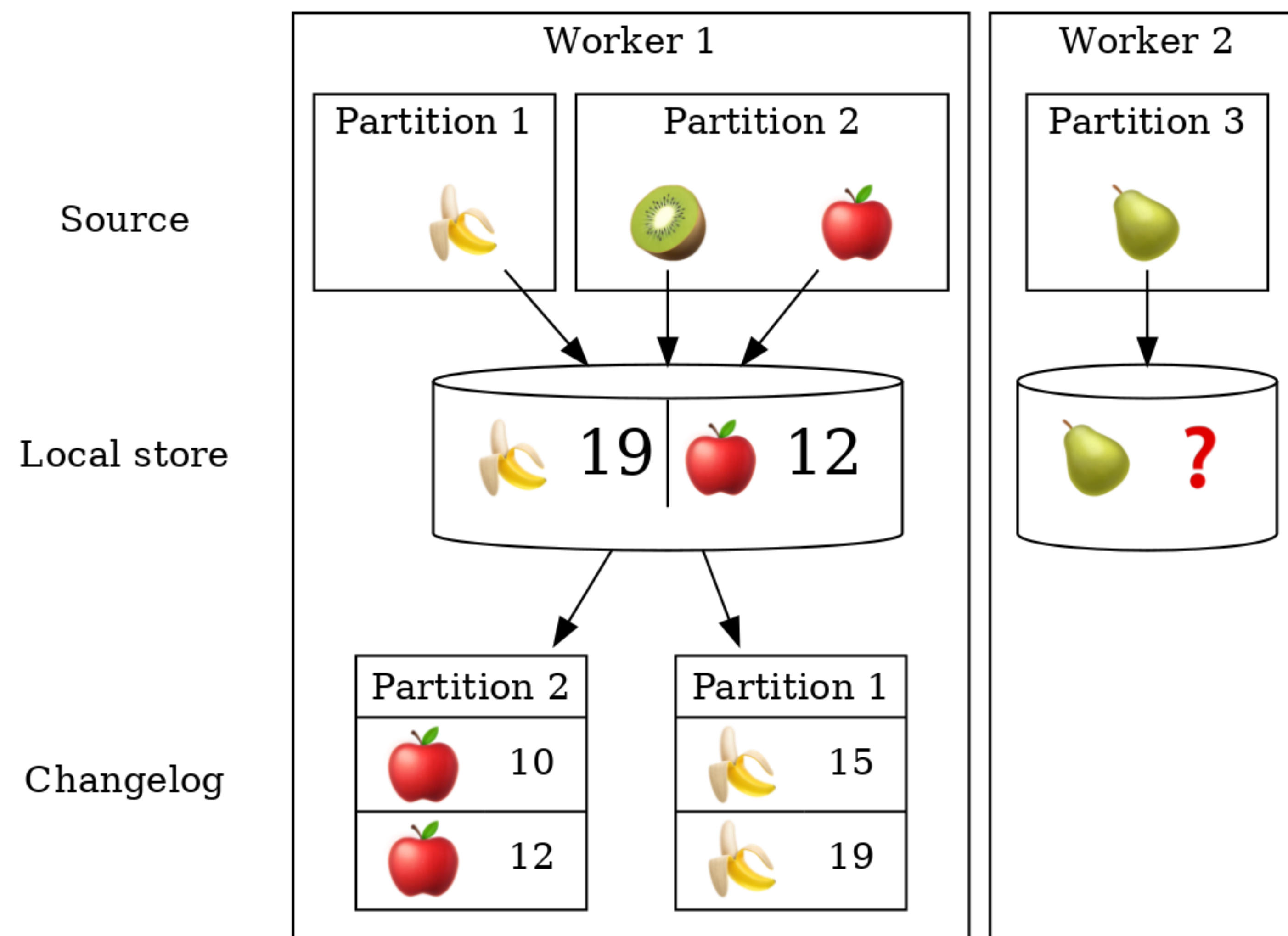
Партиционирование и local state



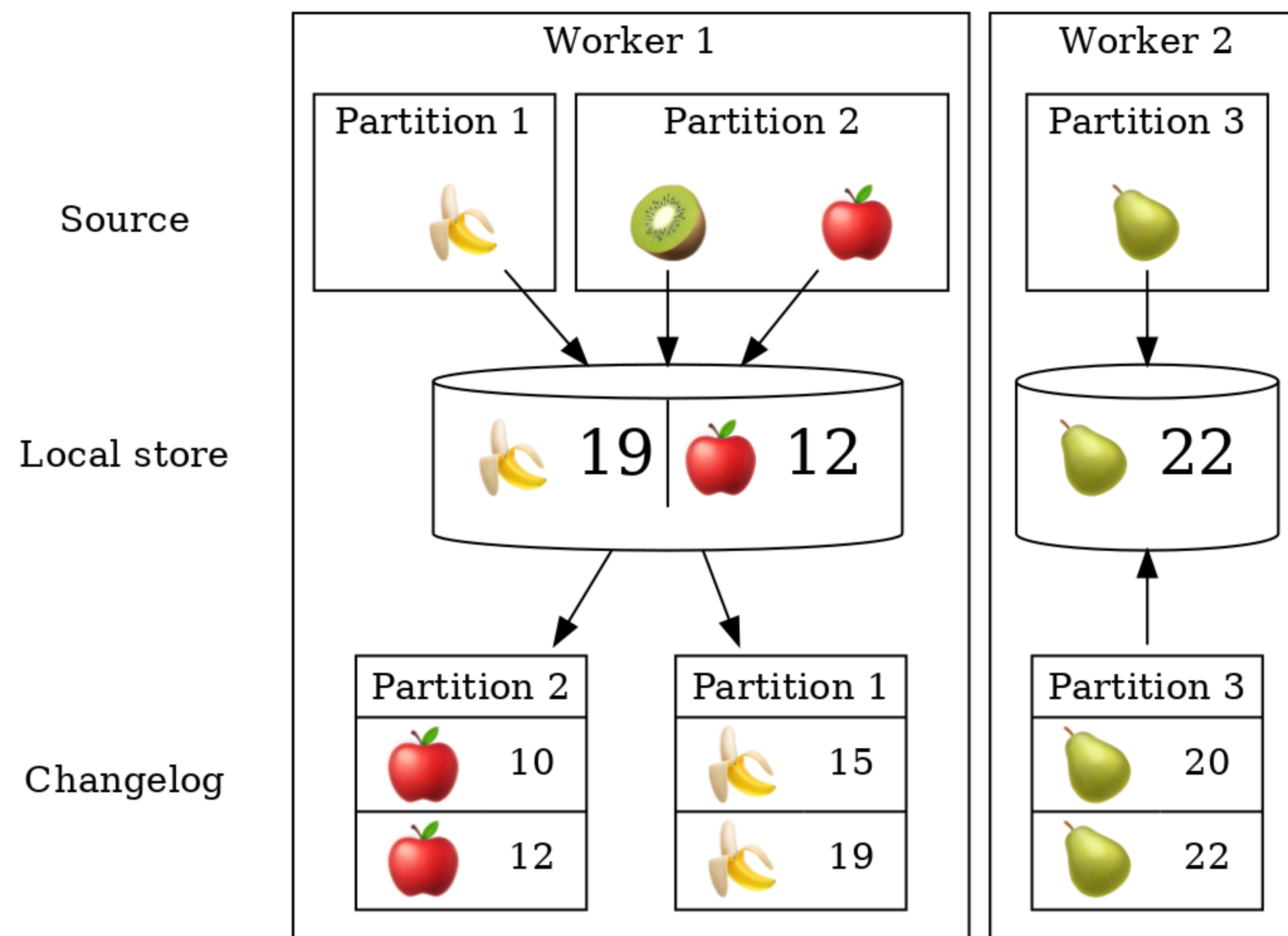
Партиционирование и local state



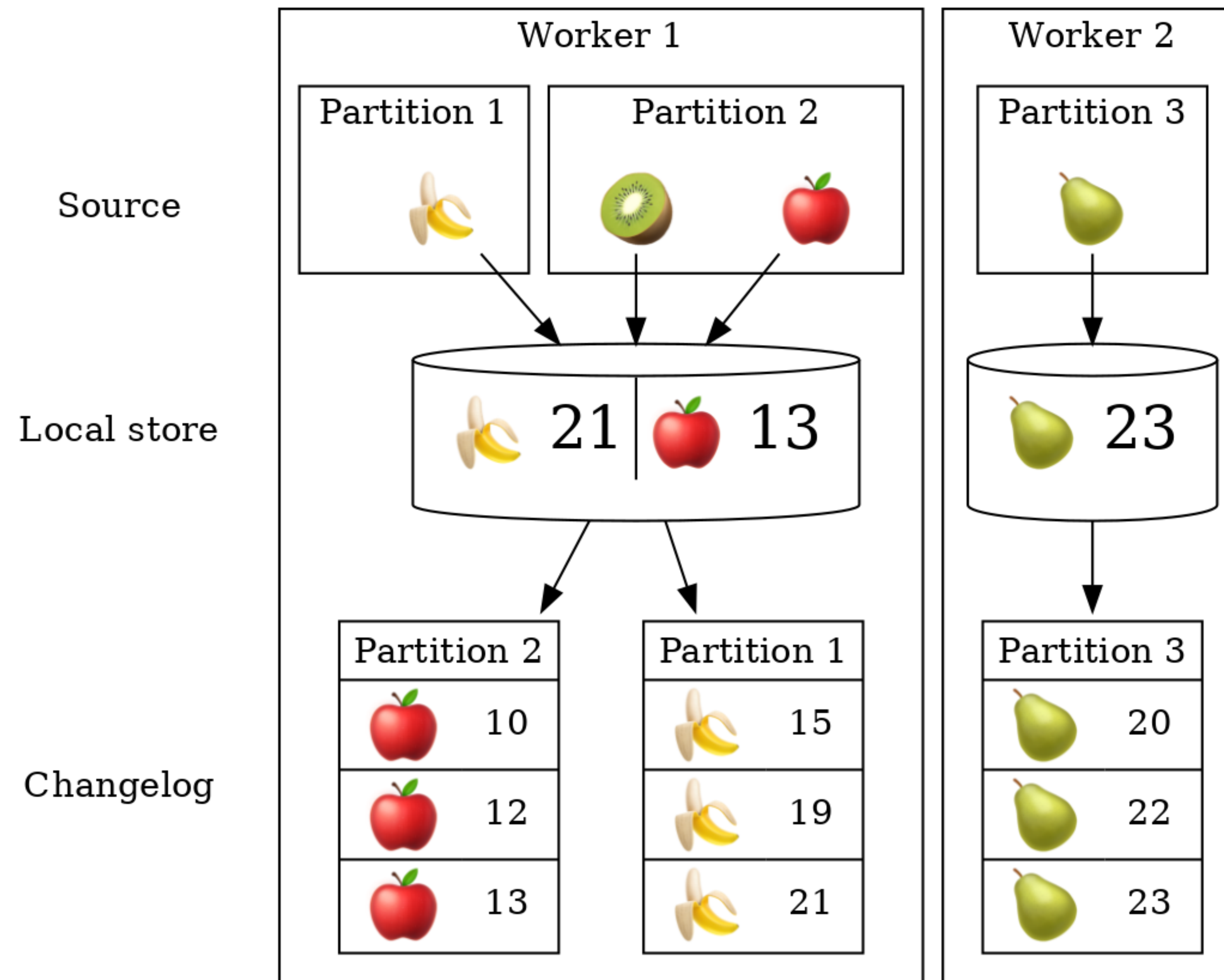
Партиционирование и local state



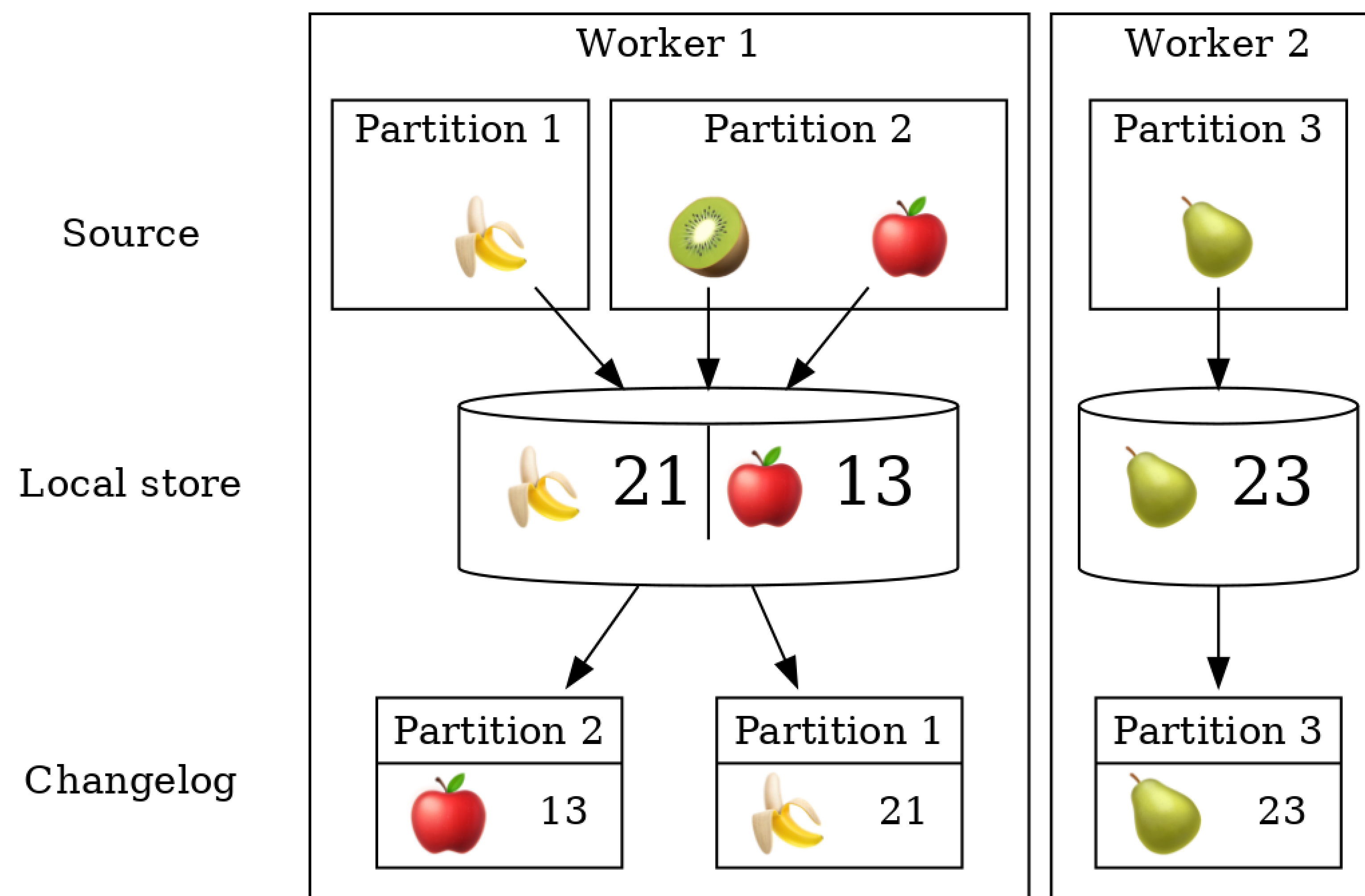
Партиционирование и local state



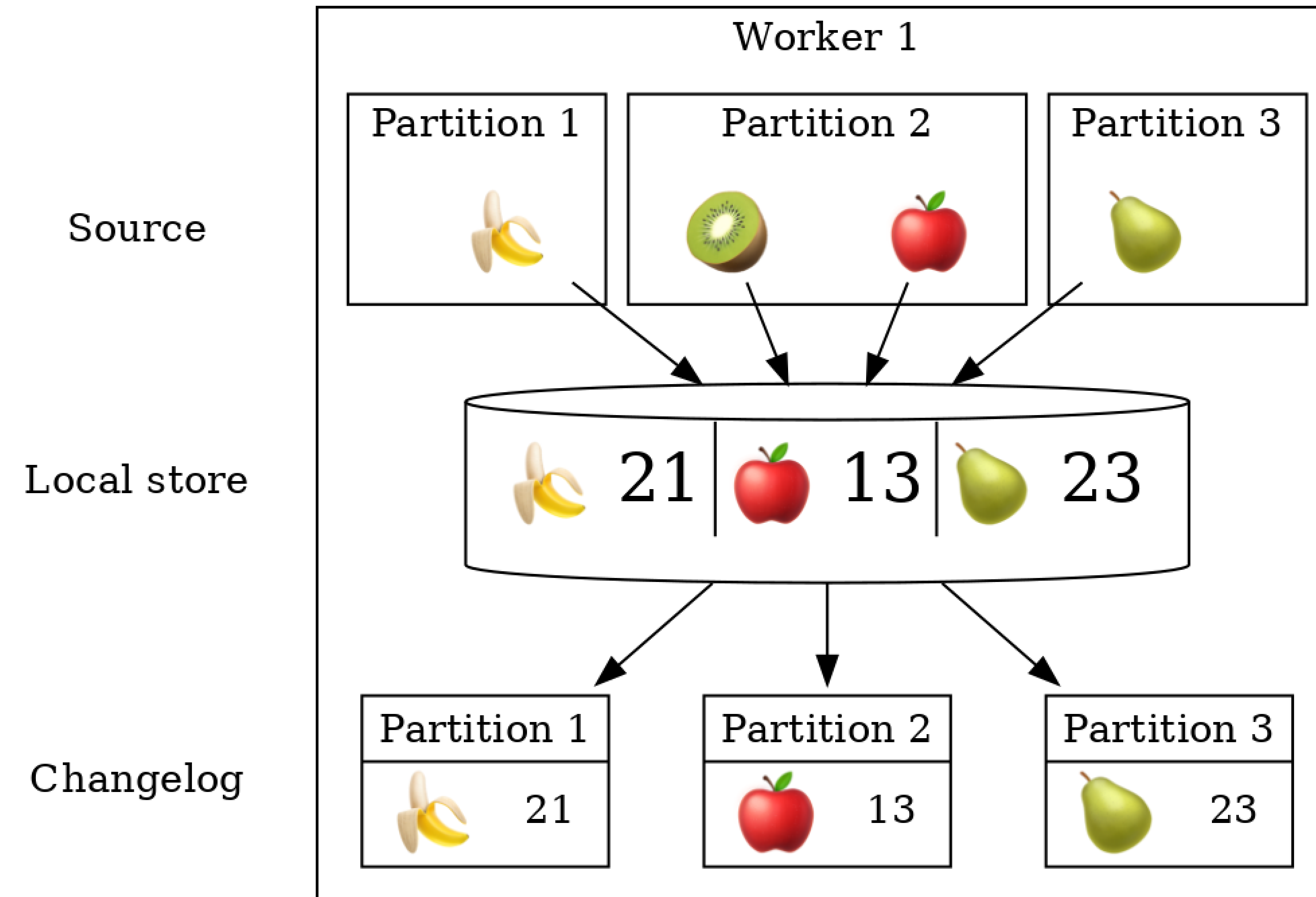
Партиционирование и local state



Партиционирование и local state

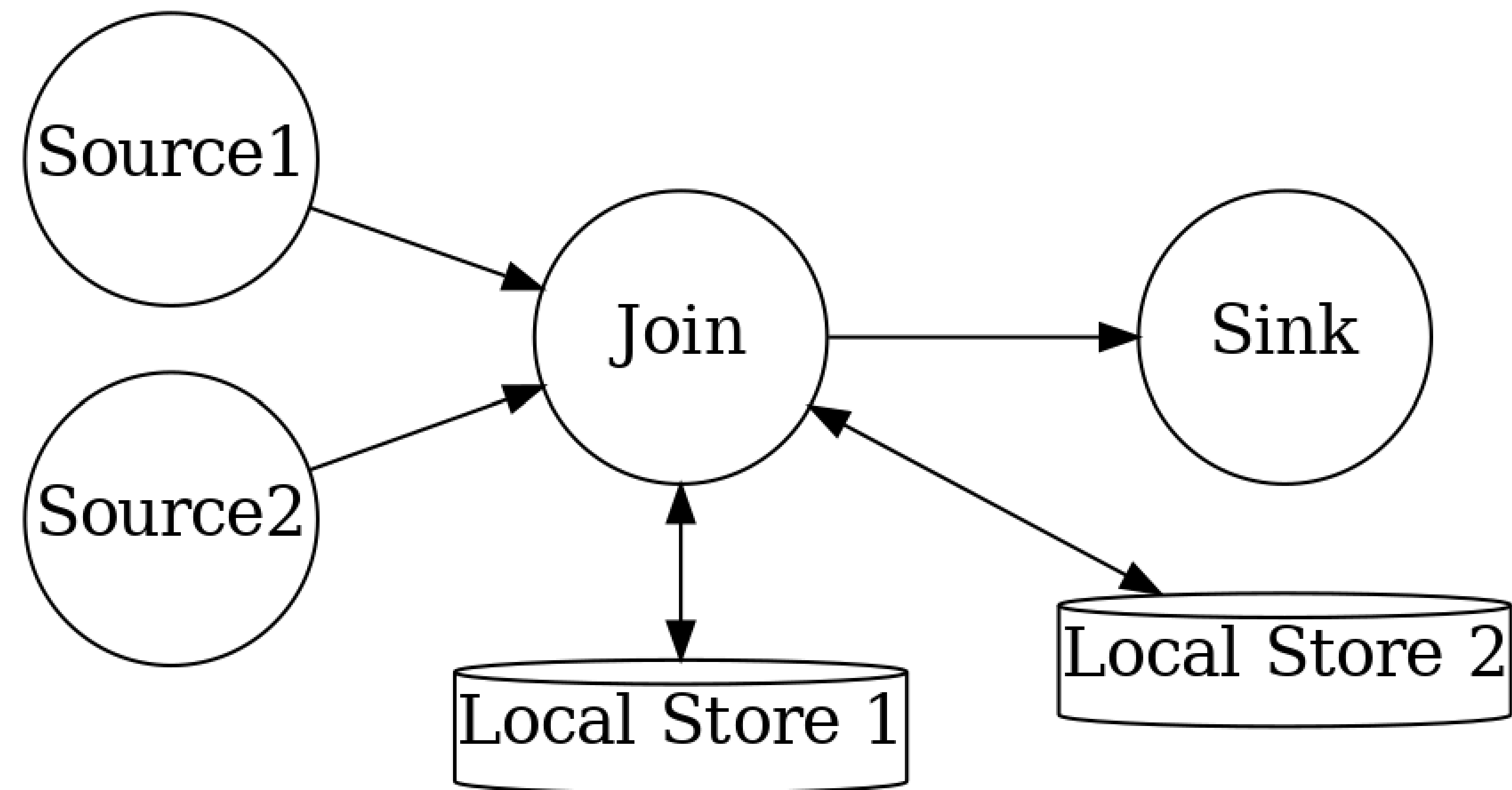


Партиционирование и local state

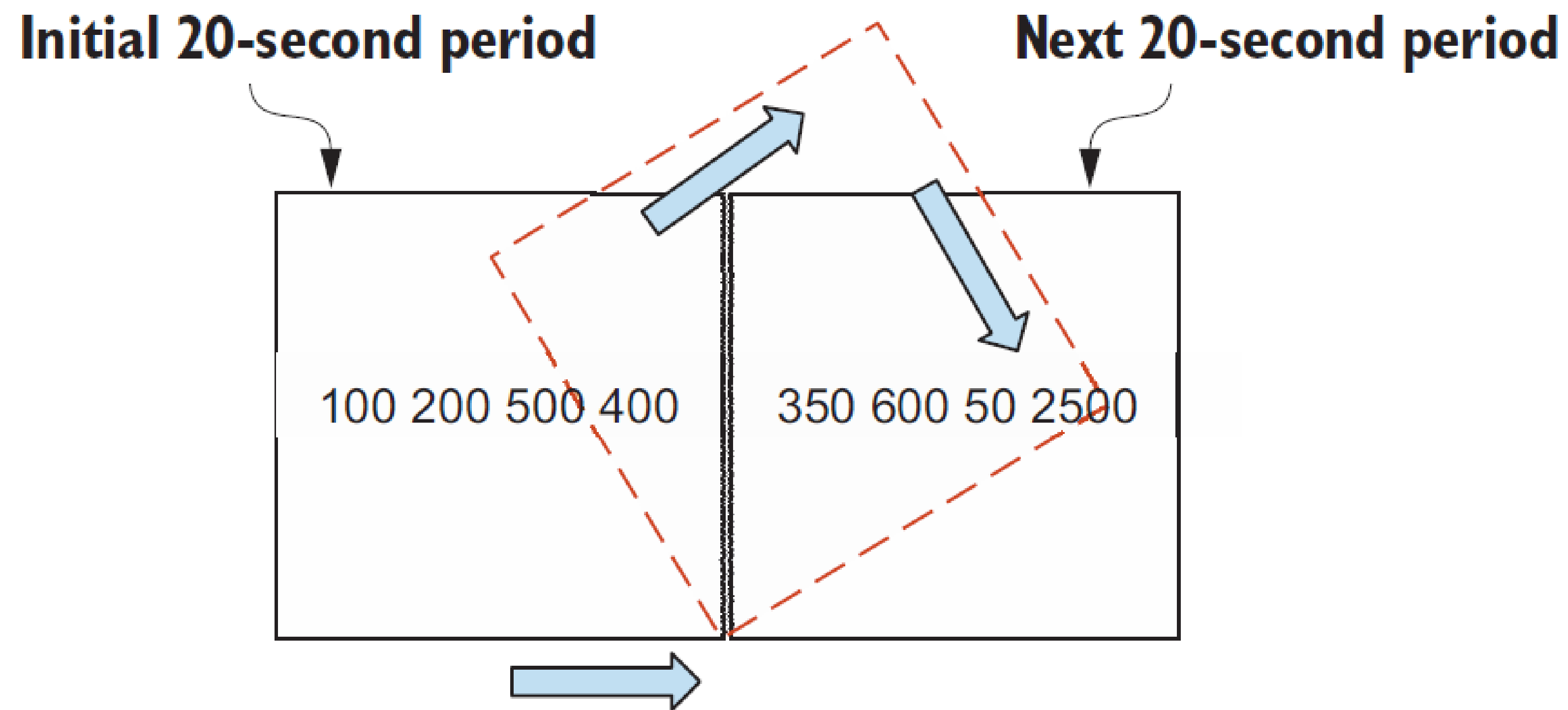


Что ещё умеют стримы?

Соединять источники!



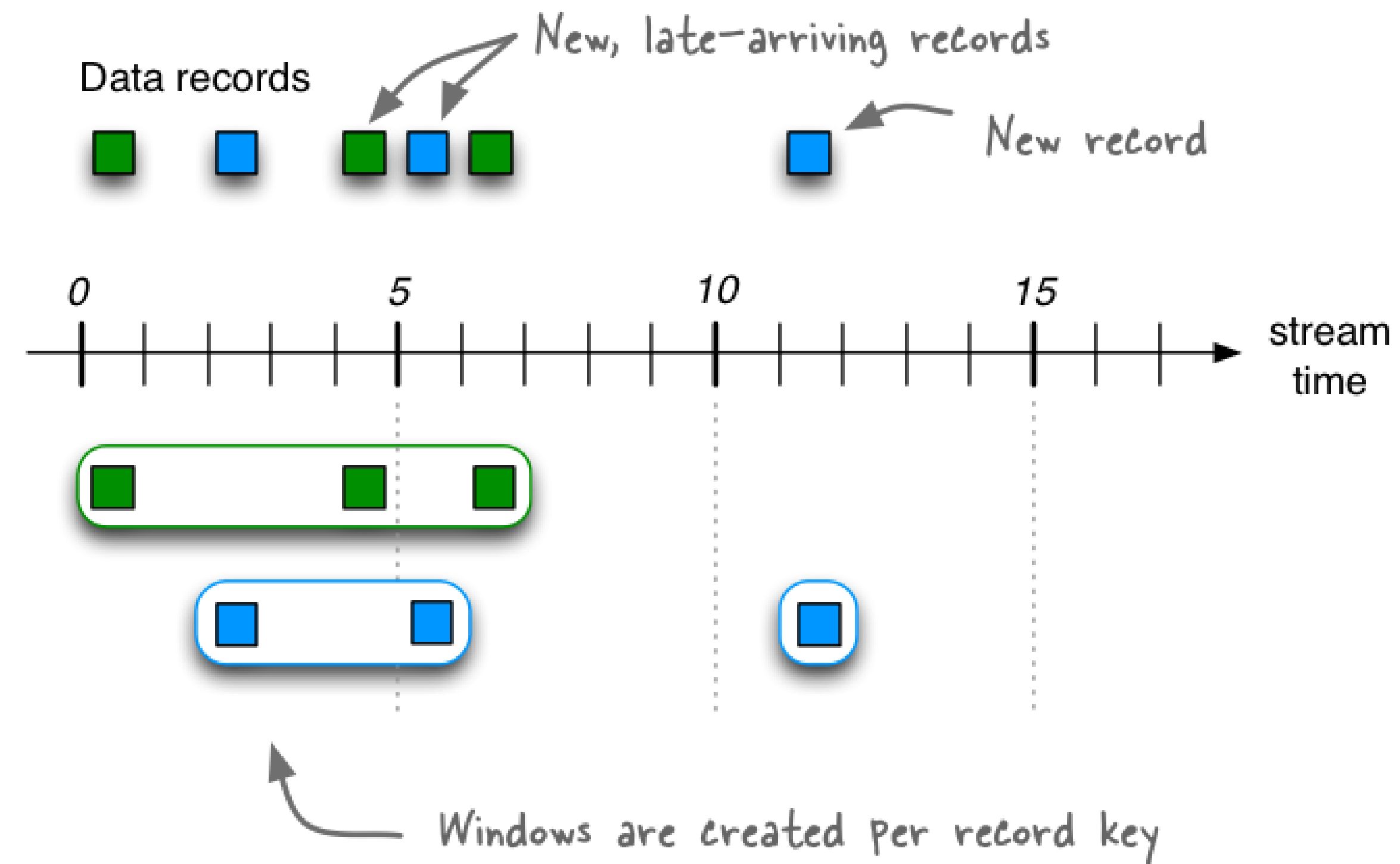
Агрегировать данные во временных окнах



Источник: Kafka Streams in Action

«Склеивать» данные, относящиеся к рабочей сессии

A Session Window with a 5-min inactivity gap



KSQL

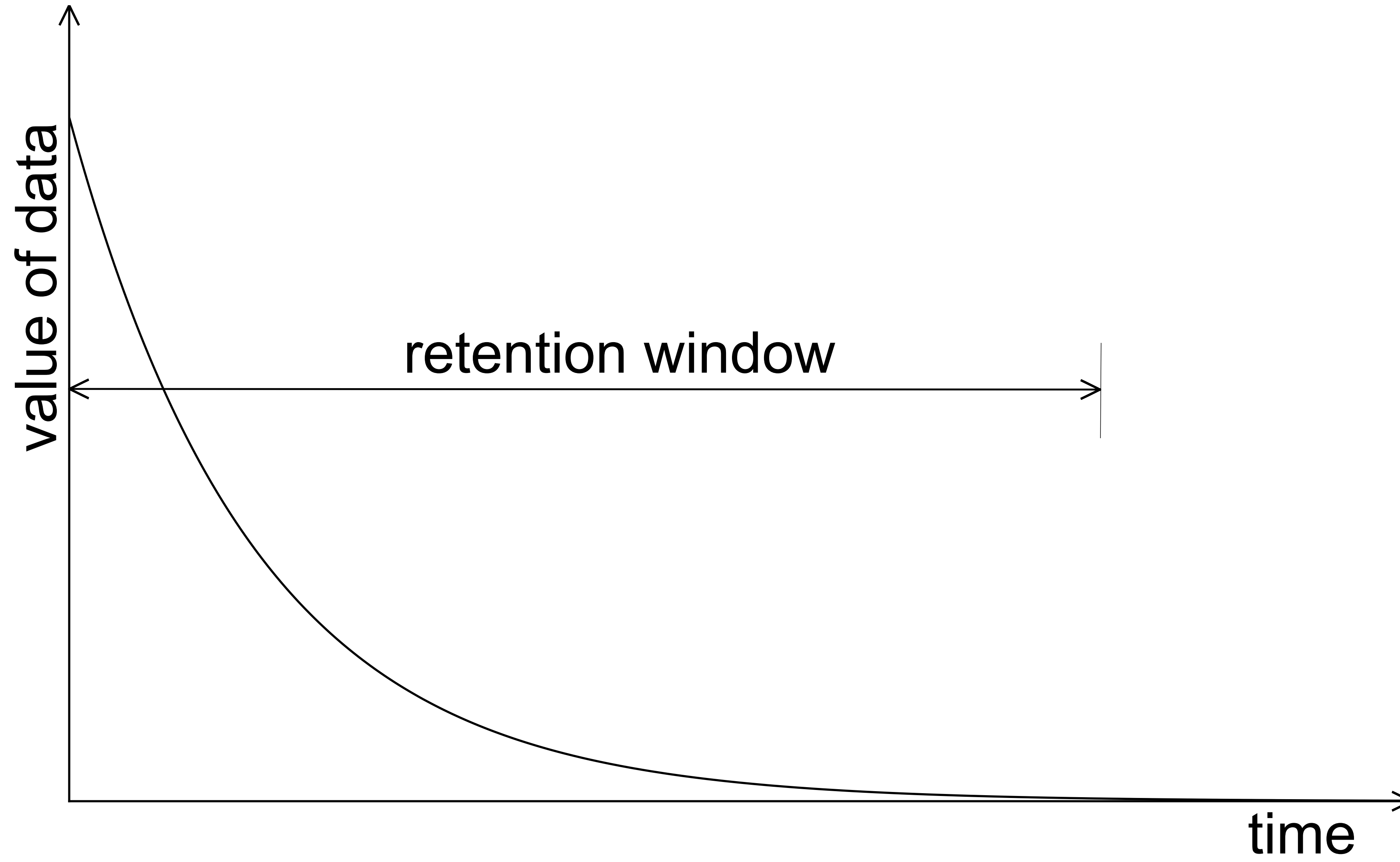
```
CREATE STREAM pageviews_enriched AS  
  SELECT pv.viewtime,  
          pv.userid AS userid,  
          pv.pageid,  
          pv.timestring,  
          u.gender,  
          u.regionid,  
          u.interests,  
          u.contactinfo  
FROM pageviews_transformed pv  
LEFT JOIN users_5part u ON pv.userid = u.userid  
EMIT CHANGES;
```

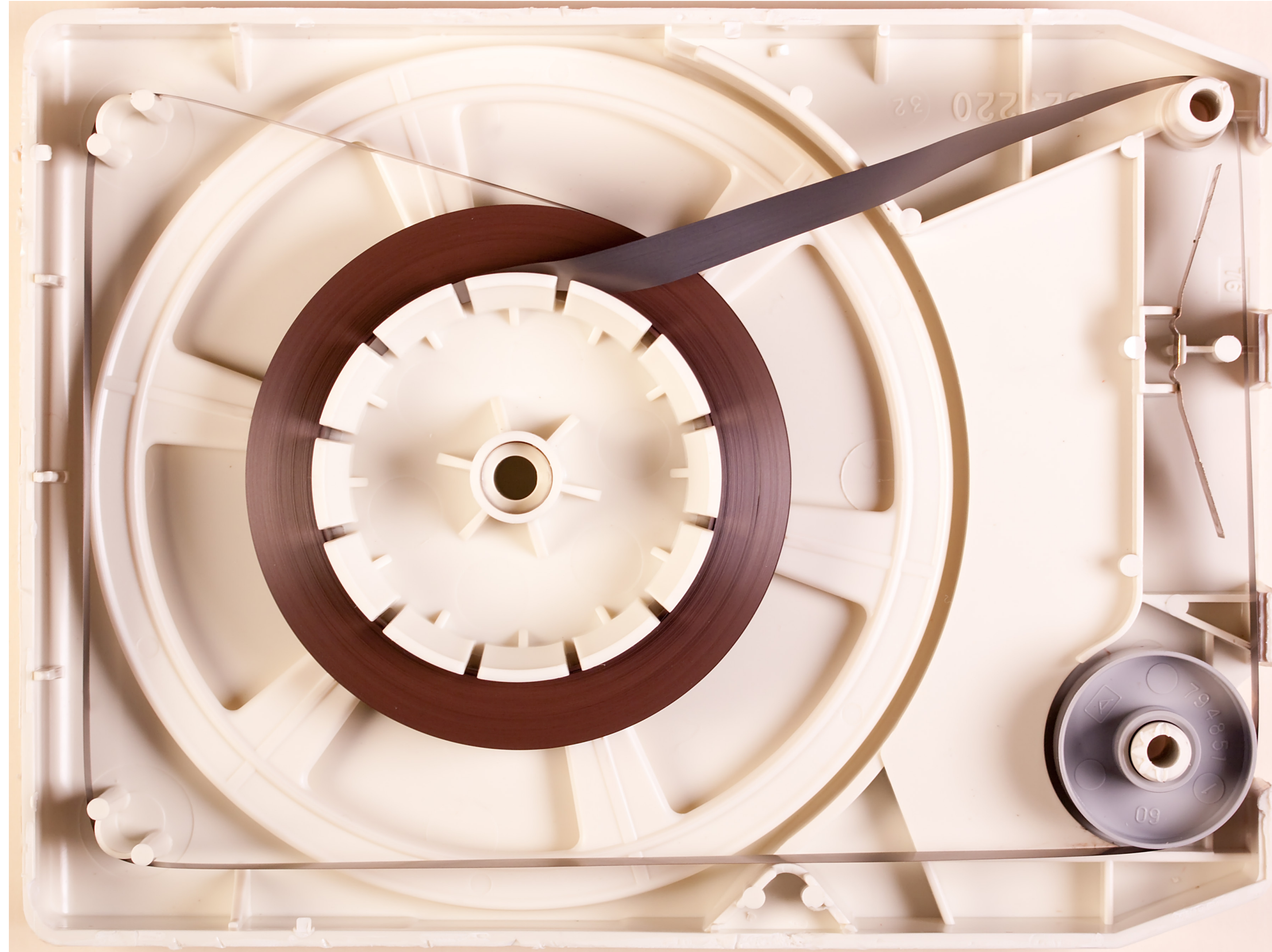
KSQL

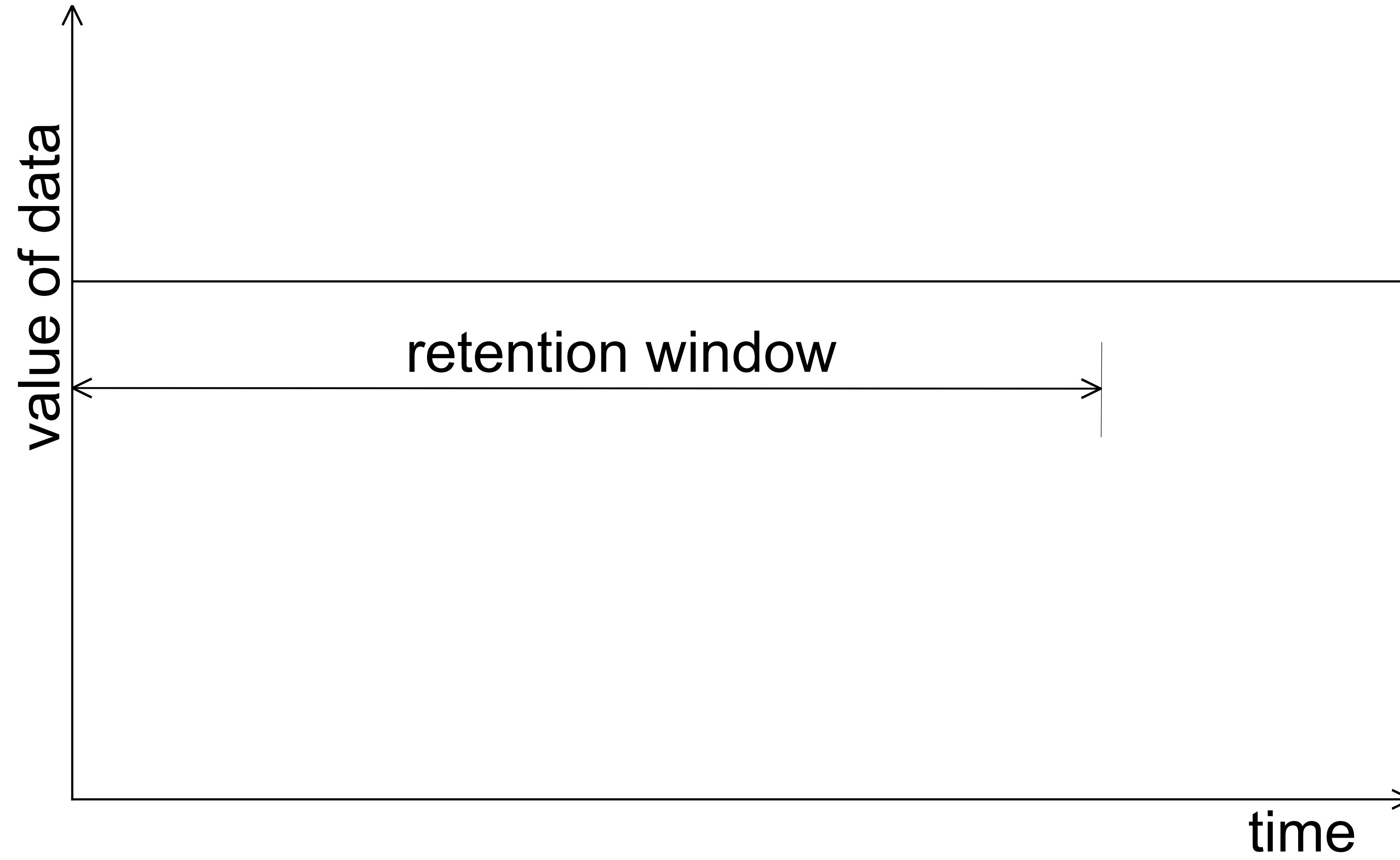
```
CREATE TABLE pageviews_per_region_per_30secs AS  
  SELECT regionid,  
          count(*)  
FROM pageviews_enriched  
WINDOW TUMBLING (SIZE 30 SECONDS)  
WHERE UCASE (gender) = 'FEMALE' AND LCASE (regionid) = 'region_6'  
GROUP BY regionid  
EMIT CHANGES;
```

Кейсы для потоковых систем

- ★ Мониторинг! Логи!
- ★ Отслеживание действий пользователей
- ★ Выявление аномалий (в т. ч. попыток мошенничества)



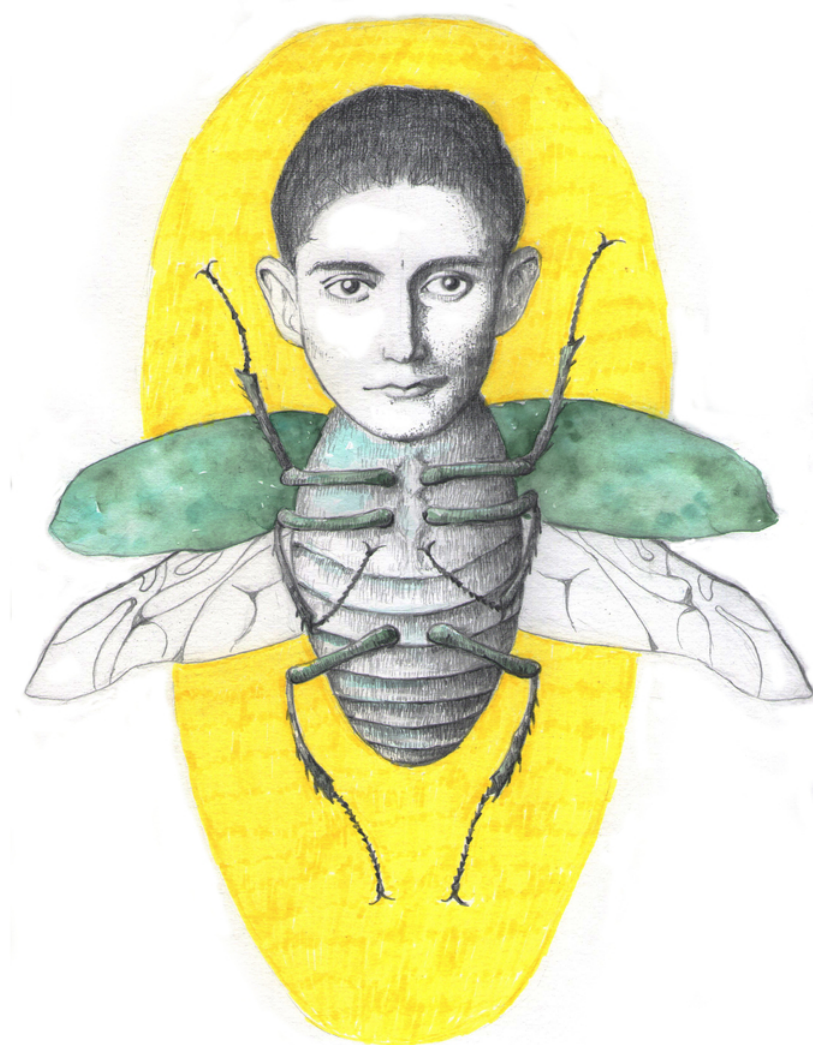




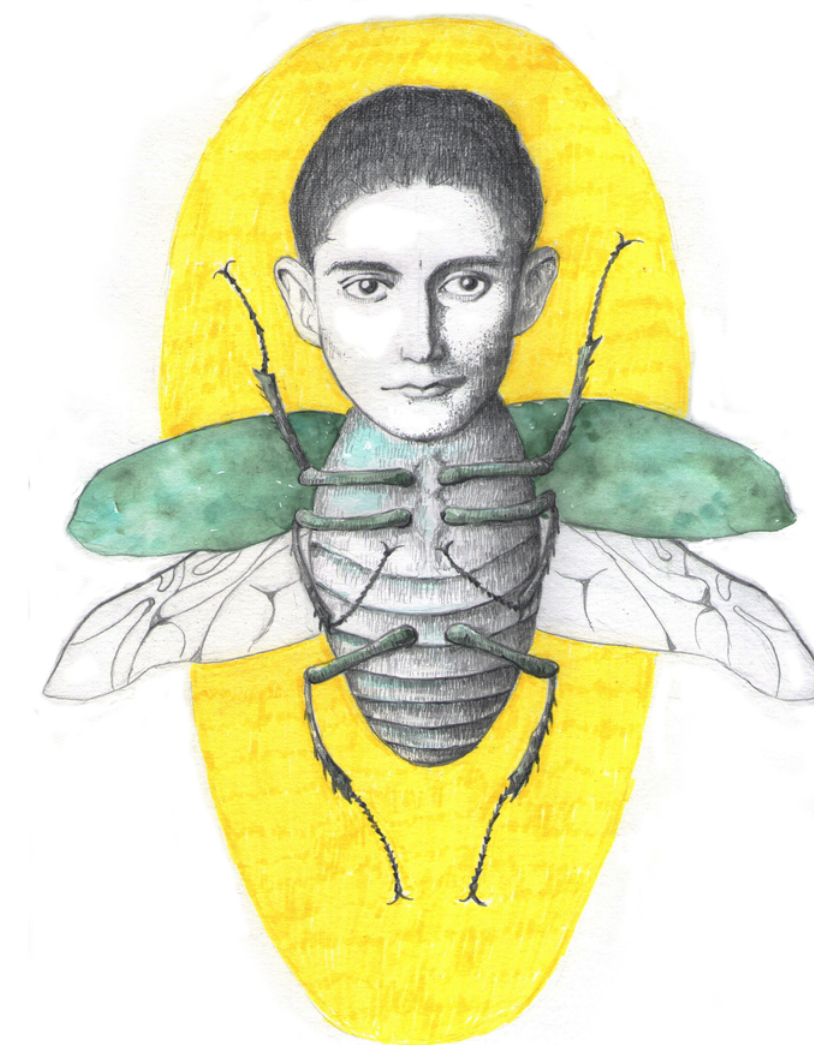
О чём следует помнить

- ★ При изменении схемы данных миграция не похожа на RDBMS.
- ★ Once-only delivery:
 - ★ В обычном режиме, сбой в KStreams приводит к повторному чтению данных.
 - ★ В режиме once-only delivery — чтение и запись между топиками Kafka.

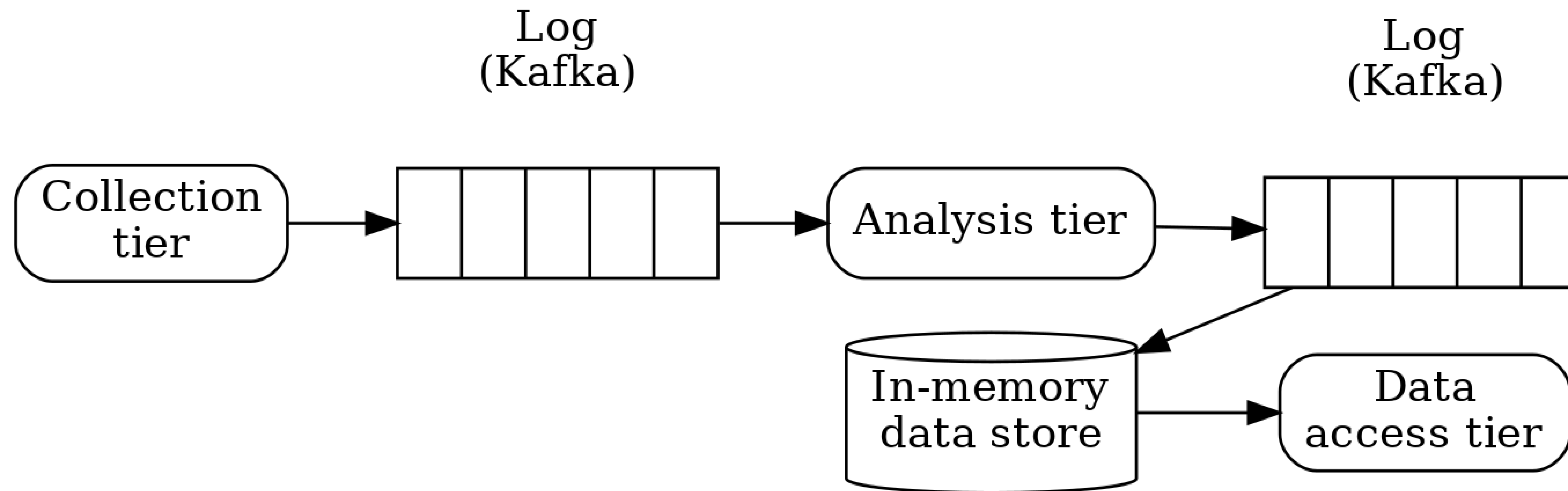
Наш план



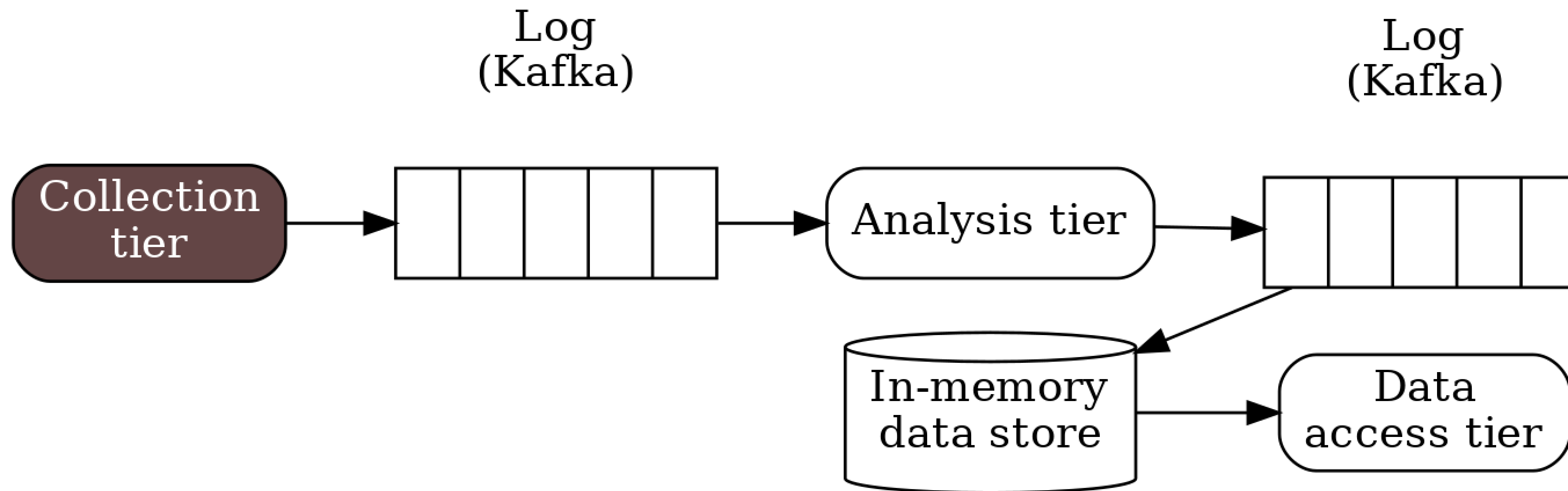
- ~~1. Что такое лог и почему это важно~~
- ~~2. Что такое Kafka и что она умеет~~
- ~~3. Что такое потоковая архитектура и на что способны потоковые обработчики~~
- 4. Kafka и JavaScript**
5. Как попробовать Кафку сегодня



Kafka и Javascript



Kafka и Javascript




kafka-node

 SOHU-Co / kafka-node

 Used by ▾ 4.4k

 Watch ▾ 97

 Star 2.2k

 Fork 594

 Code

 Issues 341

 Pull requests 43

 Actions

 Projects 0

 Wiki

 Security

 Insights

Node.js client for Apache Kafka 0.8 and later.

 827 commits

 6 branches

 0 packages

 53 releases

 90 contributors

 MIT

● JavaScript 98.2%

● Other 1.8%

- ★ Лидер по звёздам и количеству использований
- ★ Чистая JavaScript-реализация
- ★ Умеет не всё.

node-rdkafka

Blizzard / node-rdkafka

Used by 646

Watch 63

Star 1.4k

Fork 270

Code

Issues 43

Pull requests 7

Actions

Projects 0

Wiki

Security

Insights

Node.js bindings for librdkafka

kafka

librdkafka

nodejs

391 commits

6 branches

0 packages

38 releases

60 contributors

MIT

JavaScript 51.7%

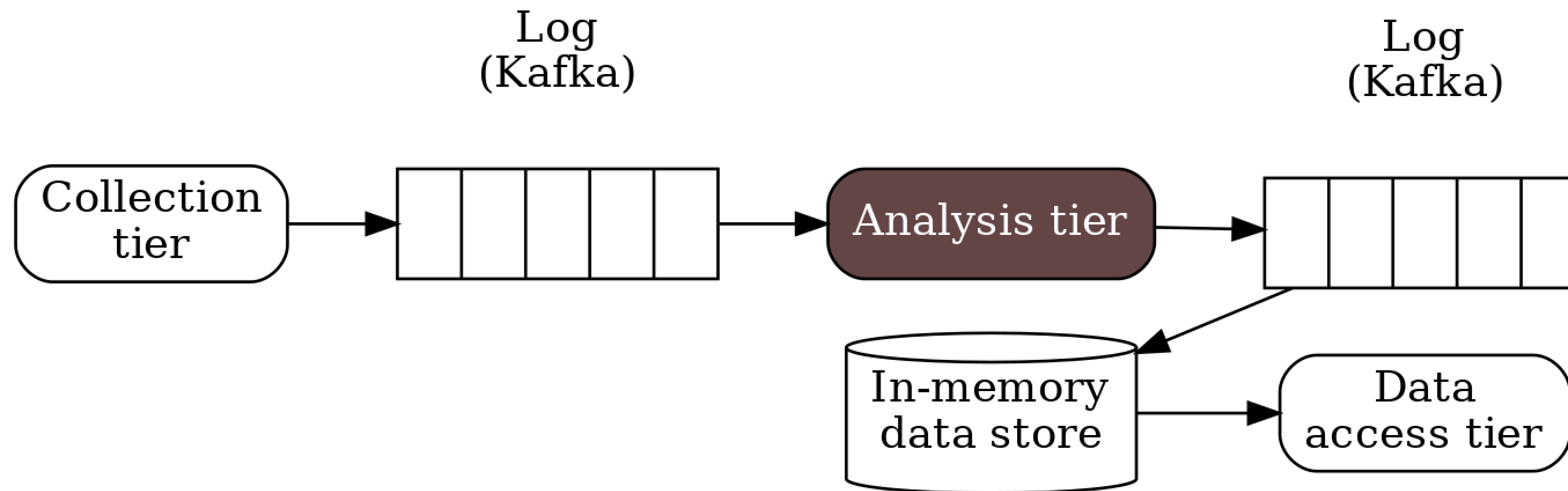
C++ 45.7%

Python 1.1%

Other 1.5%

- ★ Github-метрики проигрывают kafka-node
- ★ JavaScript/C++
- ★ Обёртка вокруг librdkafka — очень зрелого проекта
- ★ Умеет работать с Confluent Cloud

Преобразующий/выходной каскад на node.js?



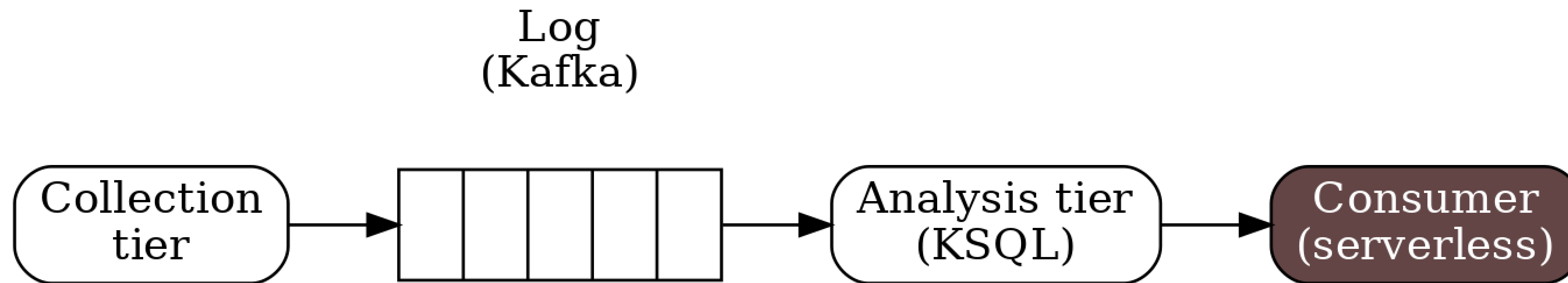
Преобразующий/выходной каскад на node.js?



Nodefluent

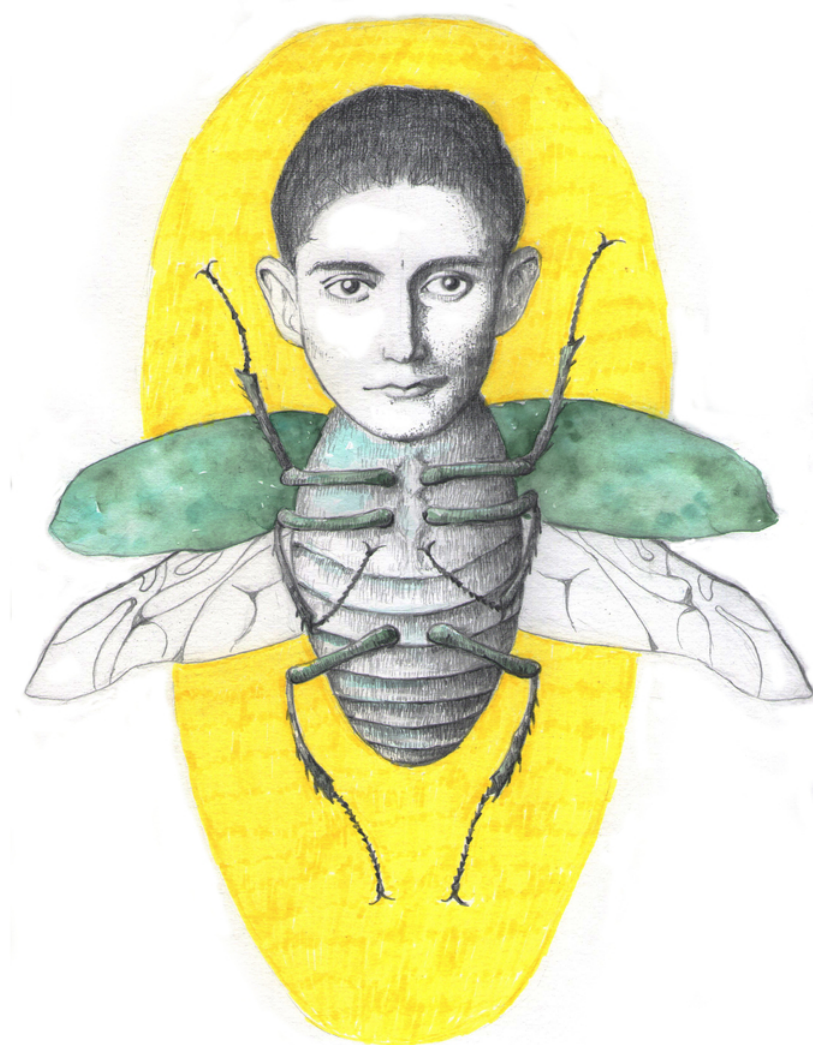
- ★ node-sinek — ещё один клиент
- ★ kafka-streams — реимплементация KStreams для node.js
- ★ kafka-connect — реимплементация kafka-connect

KSQL + Serverless

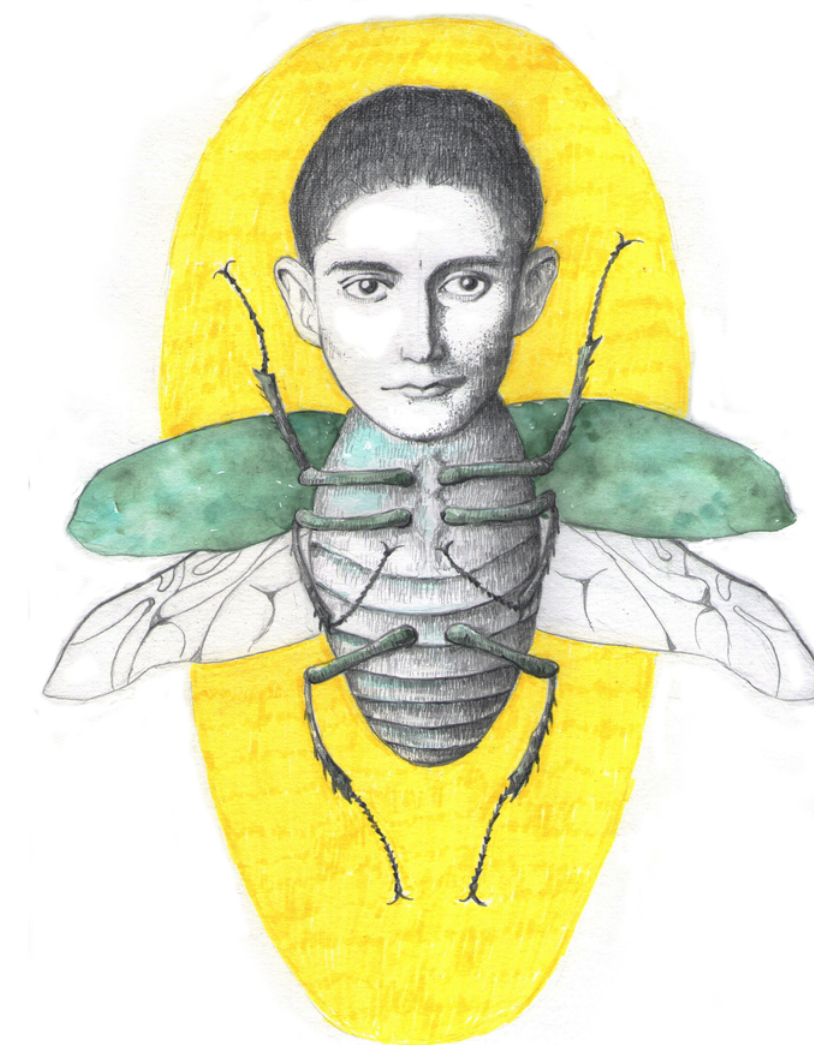


Benjamin Stopford: Event Sourcing, Stream Processing and Serverless

Наш план



- ~~1. Что такое лог и почему это важно~~
- ~~2. Что такое Kafka и что она умеет~~
- ~~3. Что такое потоковая архитектура и на что способны потоковые обработчики~~
- ~~4. Kafka и JavaScript~~
- 5. Как попробовать Кафку сегодня**



Вы решили попробовать Кафку. С чего начать?

<https://kafka.apache.org/downloads>

`kafkacat` — лучший CLI инструмент

CONDUKTOR

Conduktor — лучший GUI инструмент

Conduktor

CONDUKTOR CONSUMER Read-only ON Options

Confluent pkc-43m...

- Overview
- Brokers 8
- Topics 11
- Consumers 2
- Schema Registry ?
- Kafka Connect ?
- Security 0
- Performance (soon)

« Collapse

BROKERS

8 Brokers	Yes Controller	+2.3.x Version	Yes Similar Config
---------------------	--------------------------	--------------------------	------------------------------

TOPICS

11 Topics	132 Partitions	0 URP	0 No Leader	0 < Min ISR
---------------------	--------------------------	-----------------	-----------------------	-----------------------

CONSUMERS

1 Active	1 Empty	0 Rebalancing	0 Dead
--------------------	-------------------	-------------------------	------------------

SECURITY

Yes Enabled	0 Acls
-----------------------	------------------

<https://www.conduktor.io/> → What do you need?

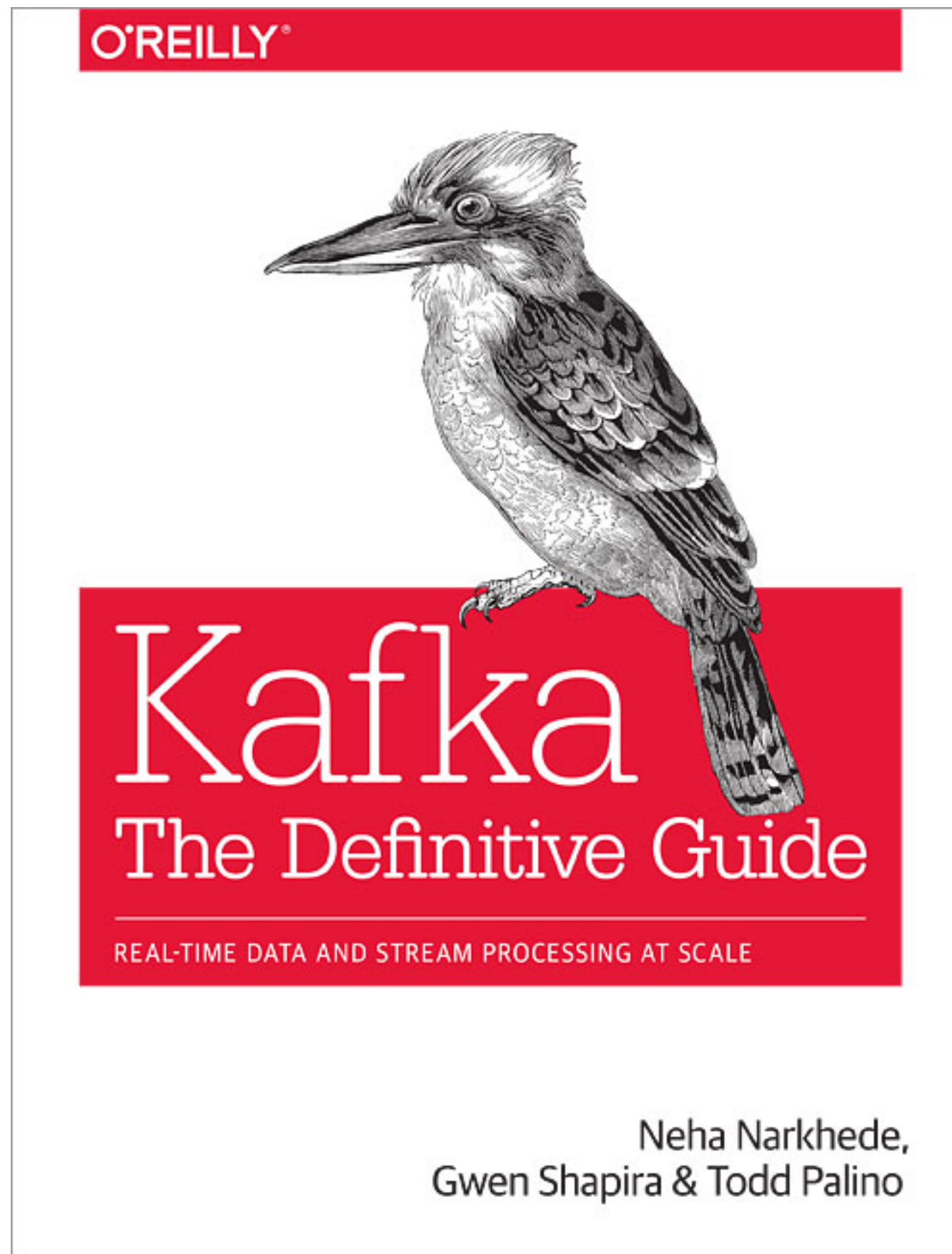
Эксплуатировать Кафку в production тяжело



Григорий Кошелев

- ★ Когда всё пошло по Кафке
- ★ Как готовить Кафку, чтобы не пригорало

Kafka: The Definitive Guide



- ★ Gwen Shapira, Neha Narkhede, Todd Palino
- ★ September 2017

Сообщества, конференции

★ Kafka Summit Conference: <https://kafka-summit.org/>

K A F K A 2
S U M M I T 0
2
0

Телеграм

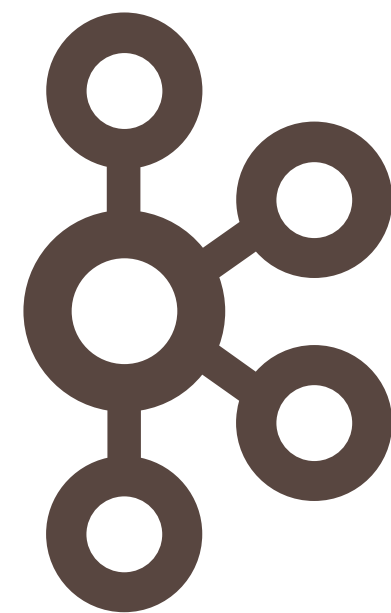
Грефневая Кафка

- ★ https://t.me/AwesomeKafka_ru
- ★ <https://t.me/proKafka>

Митап в Москве

Moscow Apache Kafka[®] Meetup by Confluent — раз в квартал

На этом всё!



CONDUKTOR

 [@inponomarev](#)

ponomarev@corchestra.ru

Спасибо!