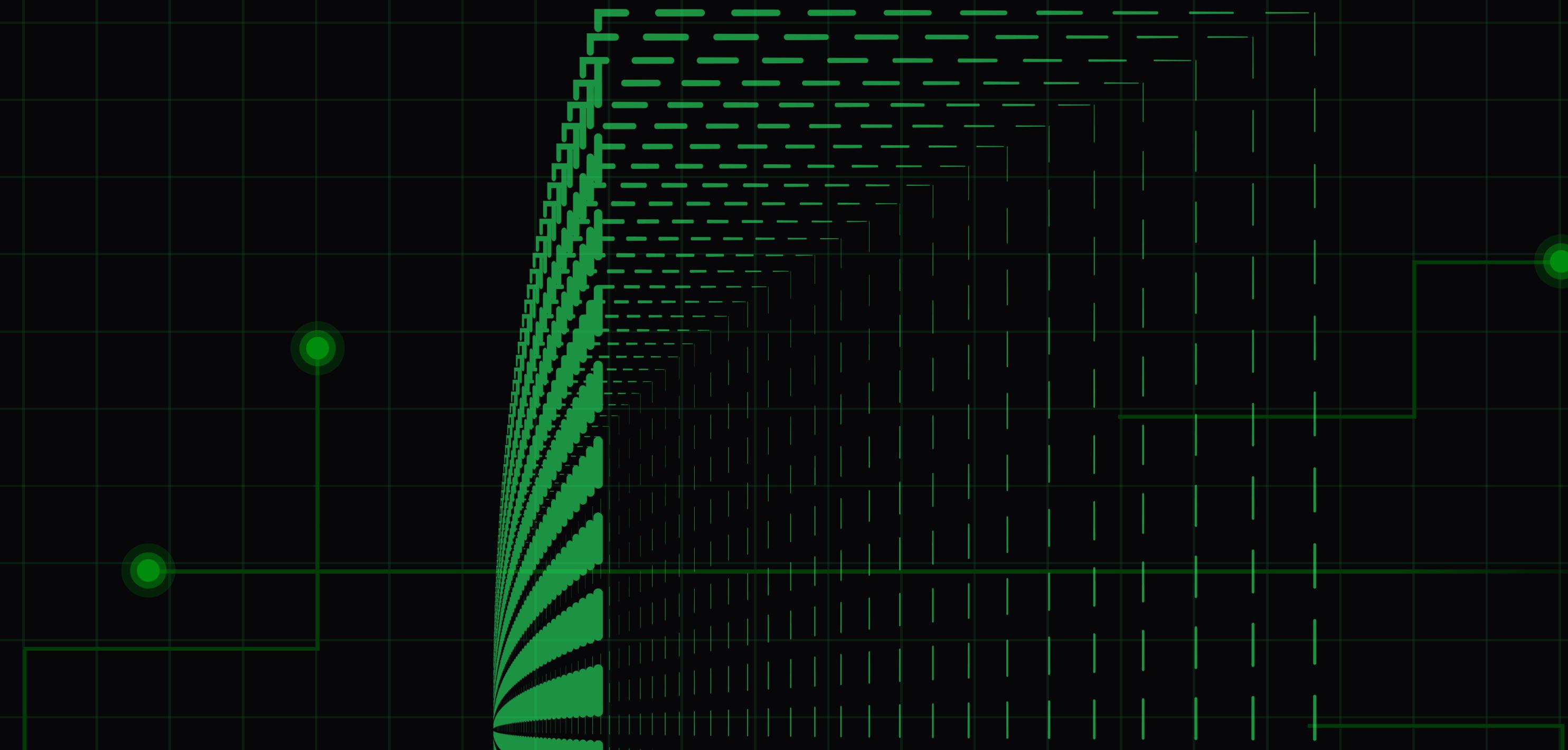
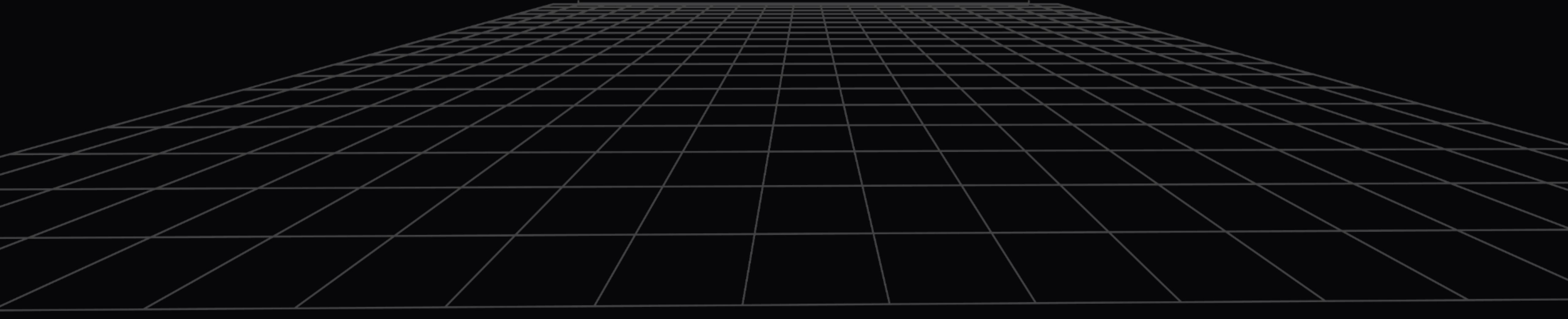


SYSTEM DESIGN

Хранение данных

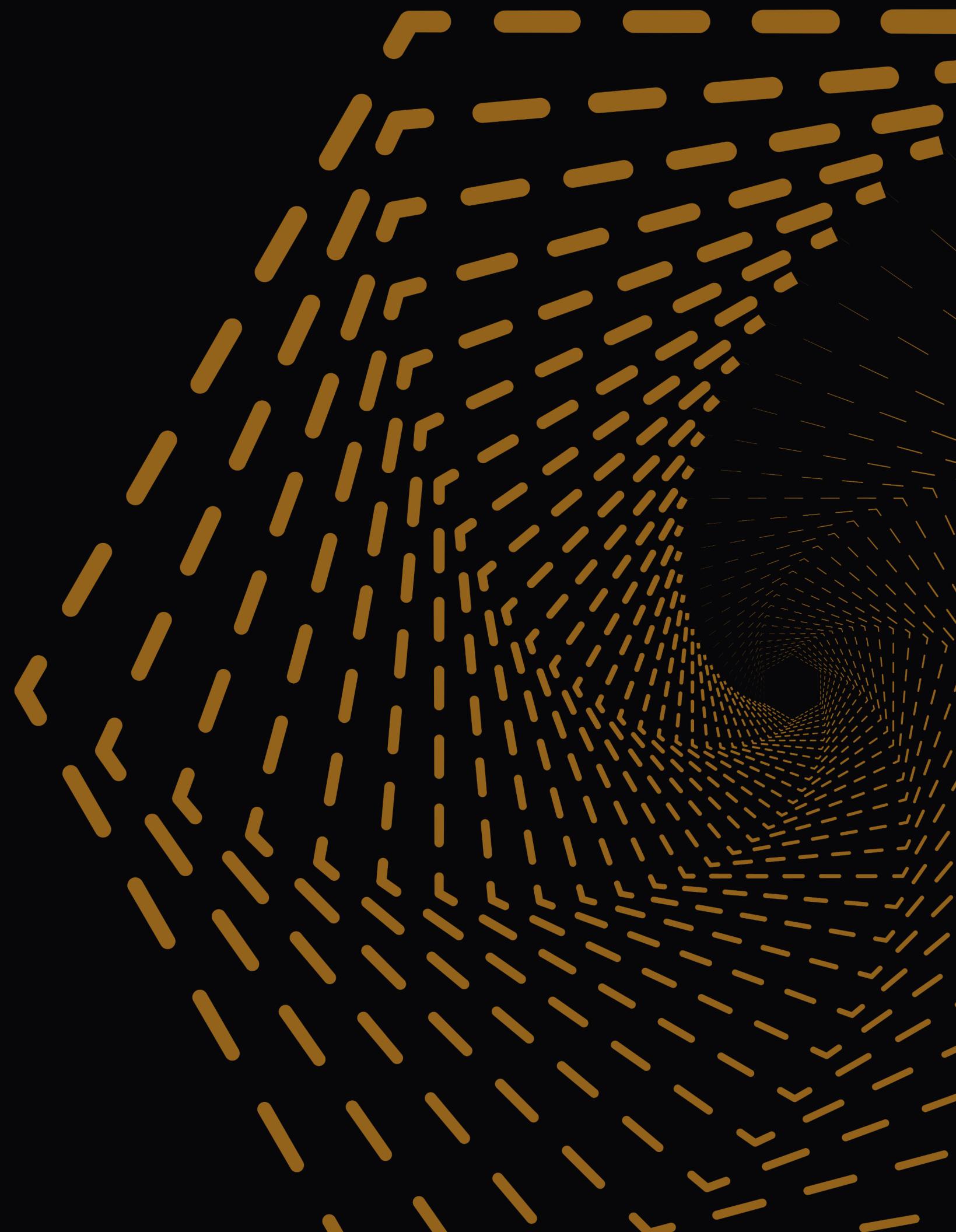


ПРОВЕРЬ ЗАПИСЬ



ПРАВИЛА ЗАНЯТИЯ

1. вопросы в чате можно задавать
в любое время
2. вопросы голосом задаем
по поднятой руке в Zoom
3. ответы на вопросы будут
в запланированных местах



МАРШРУТ ЗАНЯТИЯ

1. Виды баз данных
2. Индексы
3. Транзакции
4. Брокеры сообщений
5. Паттерны хранения и поставки данных

ВИДЫ БАЗ ДАННЫХ

РЕЛЯЦИОННЫЕ



User ID	Name	CityID
23	user_1	1
233	user_2	2
12	user_5	1

CityID	Name
1	Moscow
2	Berlin

```
SELECT id, status  
FROM user_info  
WHERE id IN (20, 30)
```

ID	Name	Status	Description	Photo
10				
20				
30				
50				

КОЛОНОЧНЫЕ

ClickHouse VERTICA

User ID	Name	City
23	user_1	Moscow
233	user_2	Berlin
12	user_5	Moscow

```
SELECT id, status  
FROM user_info  
WHERE id IN (20, 30)
```

ID	Name	Status	Description	Photo
10				
20				
30				
50				

ДОКУМЕНТООРИЕНТИРОВАННЫЕ

```
Terminal: System Design x + ▾
```

```
1 {
2   "name": "Petr",
3   "surname": "Ivanov",
4   "city": "Moscow",
5   "interests": [
6     "football",
7     "hockey"
8   ],
9 }
```

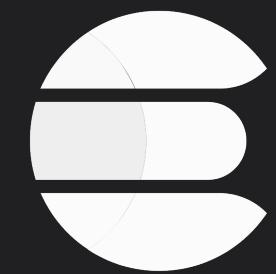


CouchDB
relax



mongoDB®

ПОИСКОВЫЕ ДВИЖКИ

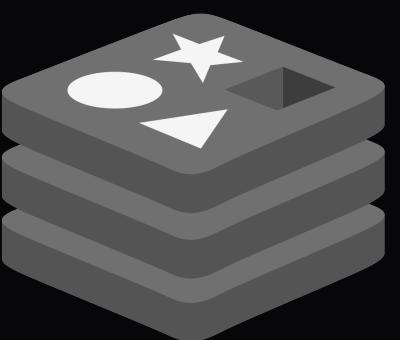


elasticsearch



KEY-VALUE

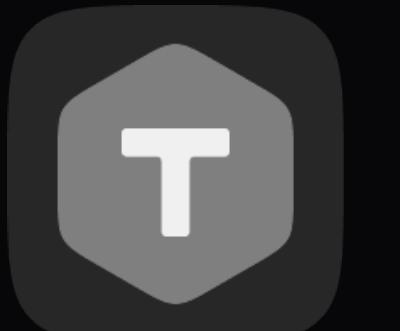
Key	Value
video_1	4234
video_2	23
video_3	2342344
video_4	2323
video_5	434



redis



MEMCACHED



Tarantool

WIDE-COLUMN



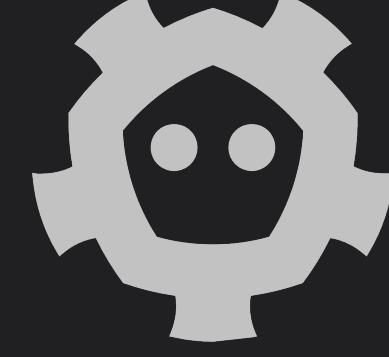
Key Value

video_1	test_1	active	221
video_2	test_2	active	2335
video_3	test_3	inactive	144

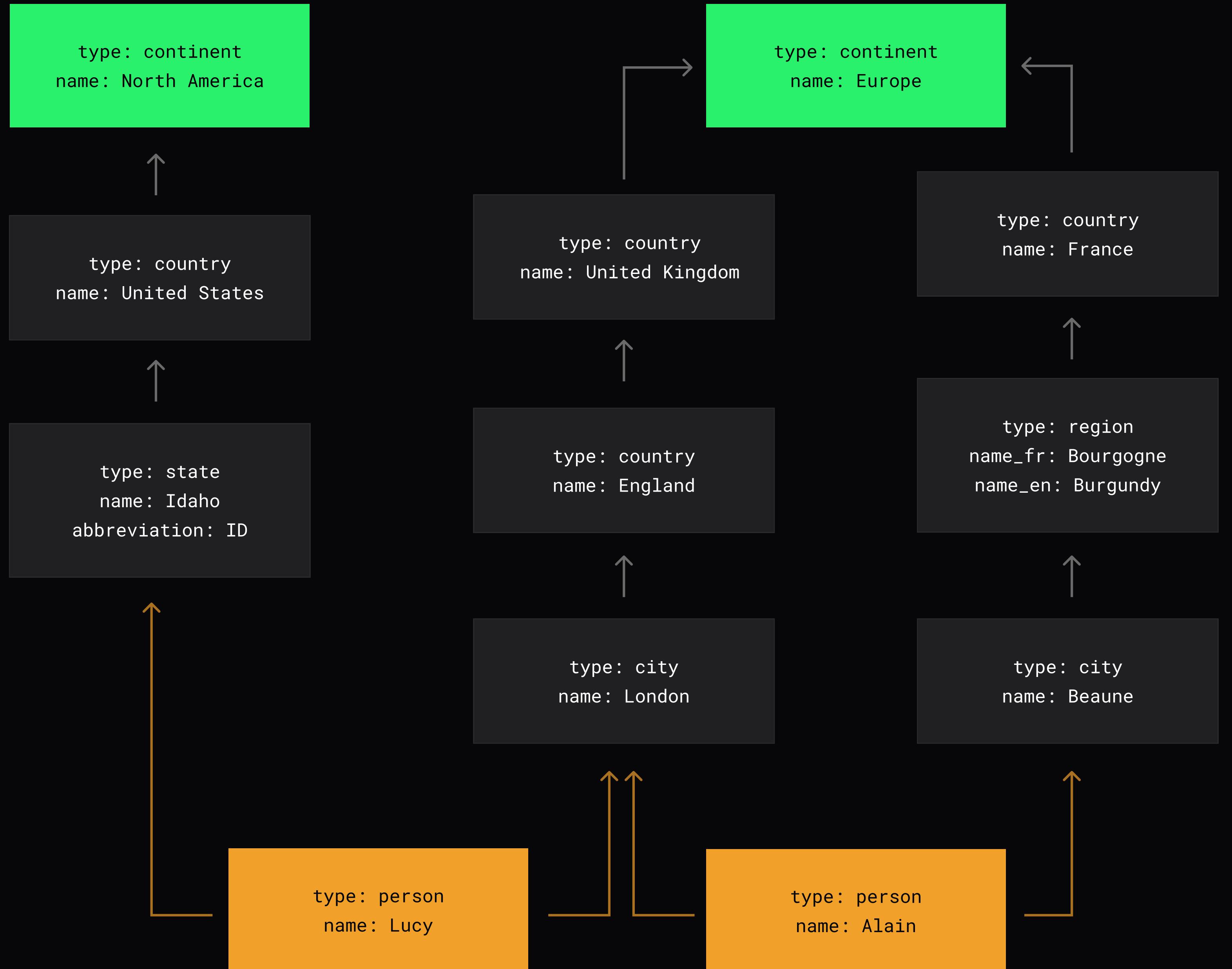
Name Status Views



APACHE
ZooKeeperTM

 etcd

ГРАФОВЫЕ



TIME SERIES



Timestamp Temrepature

10.01.2022 17:00	37.2
10.01.2022 17:01	37.1
10.01.2022 17:02	36.8

BLOB STORE



.mp3

.mp3

.mp3

ВЫБОР БАЗЫ ДАННЫХ

/1 Транзакции

/2 Формат данных

/3 Навык работы с технологией

/4 Характер обращений к данным

/5 Сообщество и зрелость технологии

/6 Частота изменяемости формата данных

ПРИМЕРЫ

- /1 Денежные средства пользователя
- /2 Количество просмотров под видео
- /3 Анкеты пользователей
- /4 Отношения между пользователями
- /5 Исходный код программ
- /6 Показатели температурных датчиков

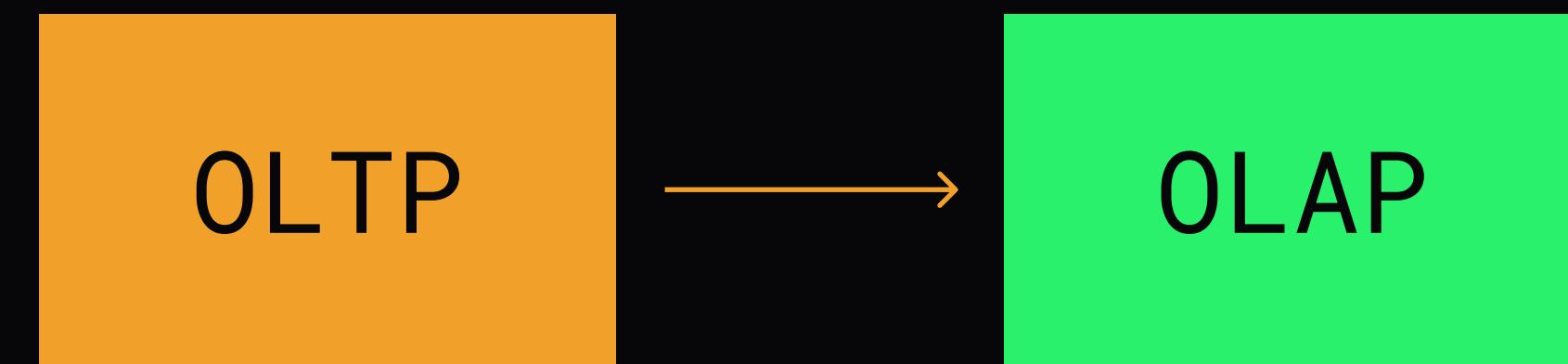
ПРИМЕР

Нужно сохранять и искать посты пользователей
с использованием полнотекстового поиска, хранить
аналитику и кэшировать популярные посты

OLAP VS OLTP

Online Analytical Processing
и Online Transaction Processing

Запросы аналитиков



Запросы пользователей

HTAP

Hybrid transactional / analytical processing

Terminal: System Design × + ▾



IN-MEMORY БАЗЫ ДАННЫХ

Базы данных, хранящие
свои данные в памяти



Terminal: System Design × + ▾



PERSISTENT БАЗЫ ДАННЫХ

Базы данных, хранящие
свои данные на дисках



Terminal: System Design × + ▾



EMBEDDED DATABASE

База данных, которая тесно интегрирована с прикладным программным обеспечением (встроена в приложение)

Application

Business Logic

Storage Logic

Terminal: System Design × + ✓



SINGLE FILE DATABASE

Один файл внутри себя содержит
всю структуру и все данные этой базы данных



Terminal: System Design × + ▾



APPEND-ONLY

Базы данных, в которых данные могут только добавляться, но не могут быть изменены или удалены

FAQ

Виды баз данных

Реляционные, колоночные, документоориентированные,
графовые, wide-column, key-value, timeseries, blobstore

ИНДЕКСЫ

ИНДЕКСЫ

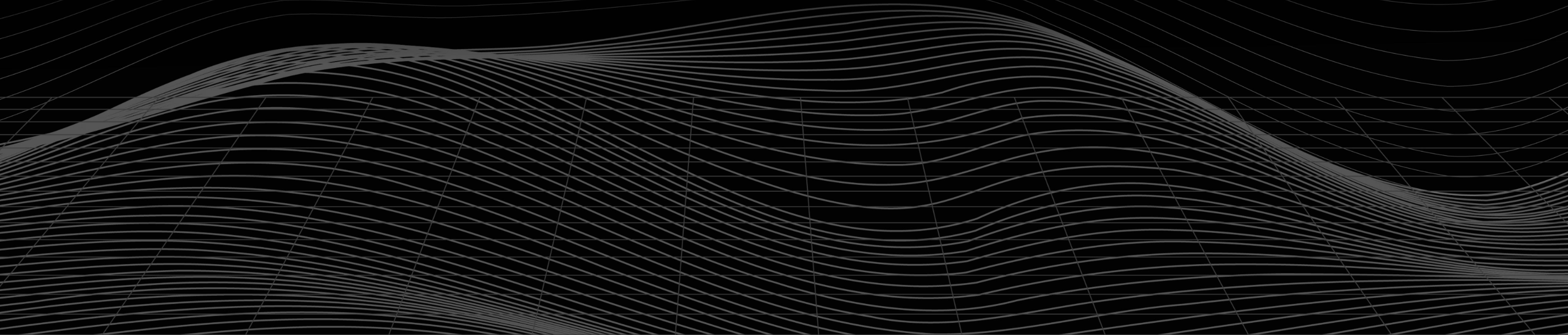
Плюсы:

- 1/ Ускоряют чтение
-

Минусы:

- 1/ Замедляют запись
 - 2/ Используют дополнительную память
 - 3/ Усложняют работу планировщика запросов
-

КОГДА ЗАПРОС МОЖЕТ
ПОЙТИ МИМО ИНДЕКСА?



СЕЛЕКТИВНОСТЬ ИНДЕКСА

ℹ Селективность индексов = 1 / кол-во уникальных
значений в индексированном столбце(-ах)

Например, если это индекс по первичному ключу,
и в таблице 10 000 записей, то его селективность
будет равна $1 / 10000 = 0,0001$

Terminal: System Design × + ▾



```
SELECT * FROM employees WHERE name = "ivan"  
AND department_id = 5
```

Эквивалентно

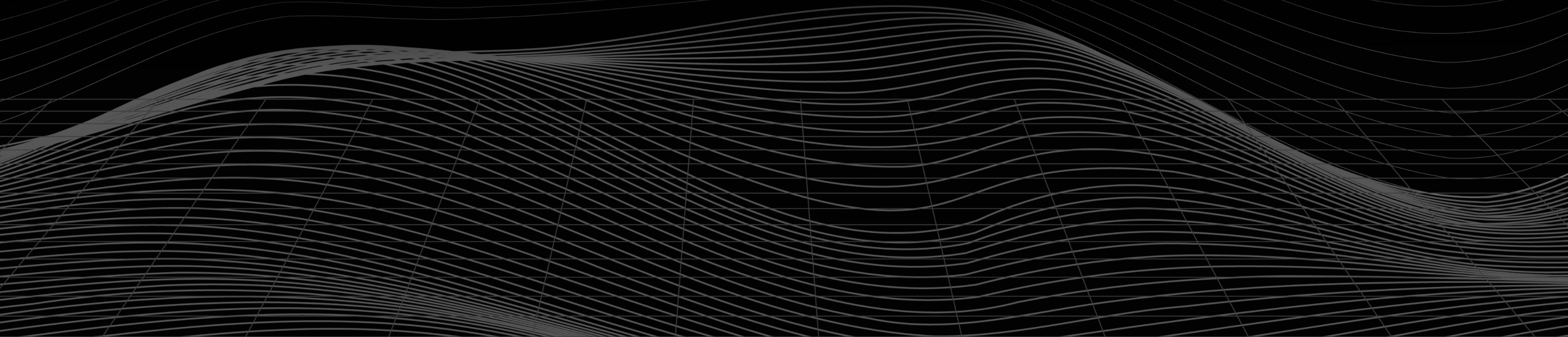
```
SELECT * FROM employees WHERE department_id  
= 5 AND name = "ivan"
```



Но какой индекс лучше?

(department_id, name) или (name, department_id)

KNOW YOUR DATA



BTREE

UserID

Name

50	user_id
168	user_id
200	user_id



48	50	79
----	----	----

128	140
-----	-----

168	200
-----	-----

270	290
-----	-----

HASH

Hash Index

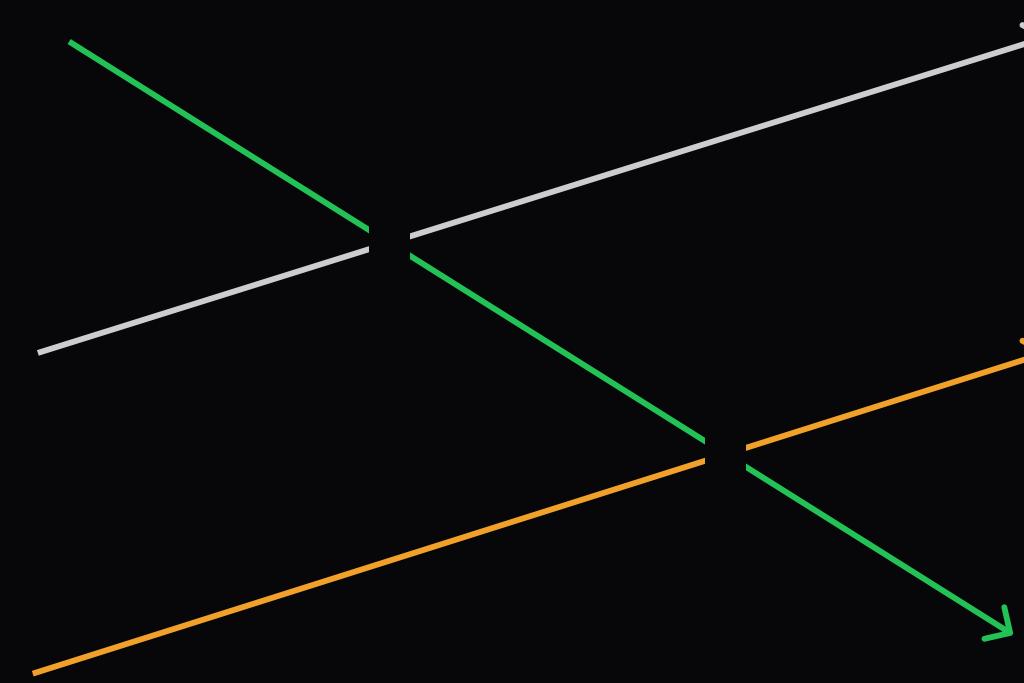
3434
121
324

User ID

121
324
3434

Name

user_id
user_id
user_id

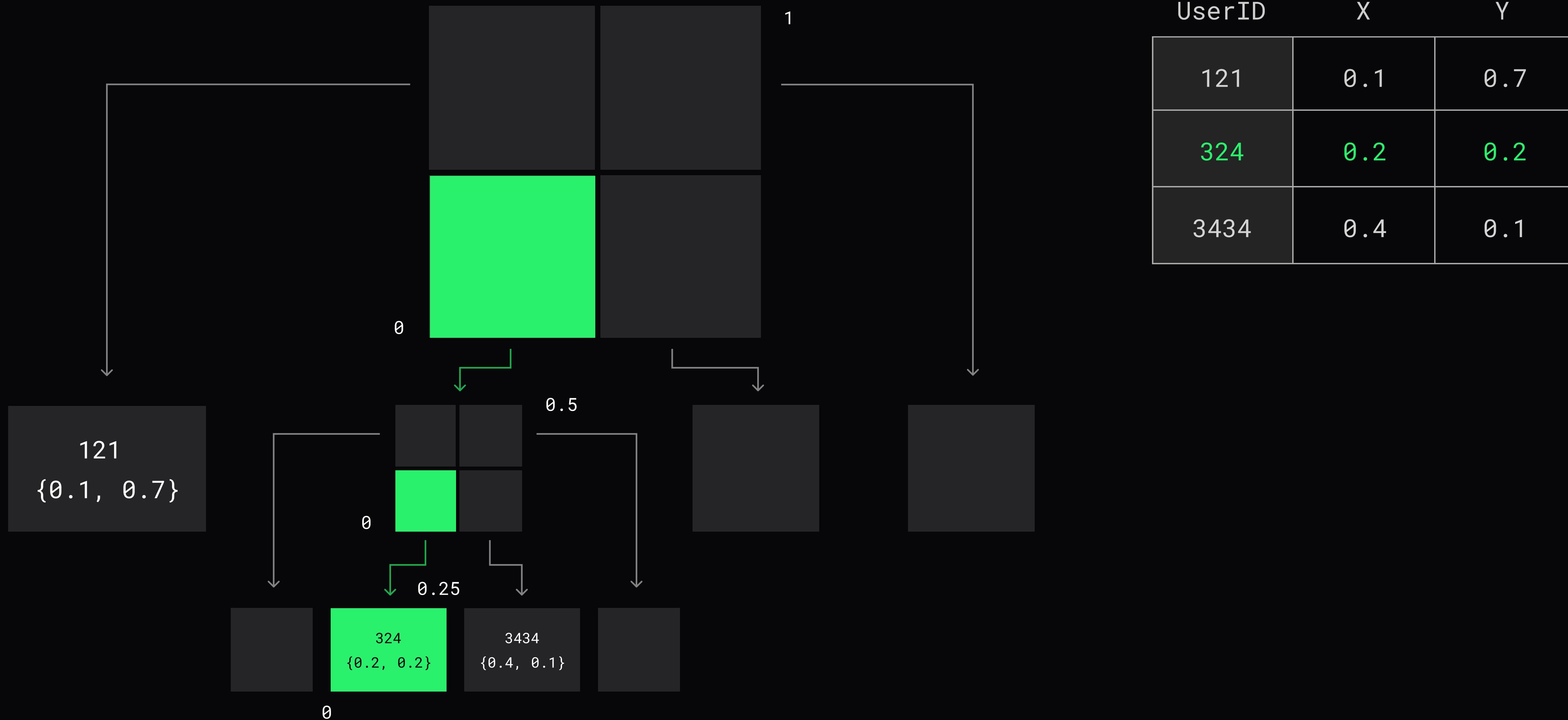


BITMAP

Bitmap Index		
3434	1	1
121	1	0
324	0	1

UserID	Smoking	Drinking
121	true	false
324	false	true
3434	true	true

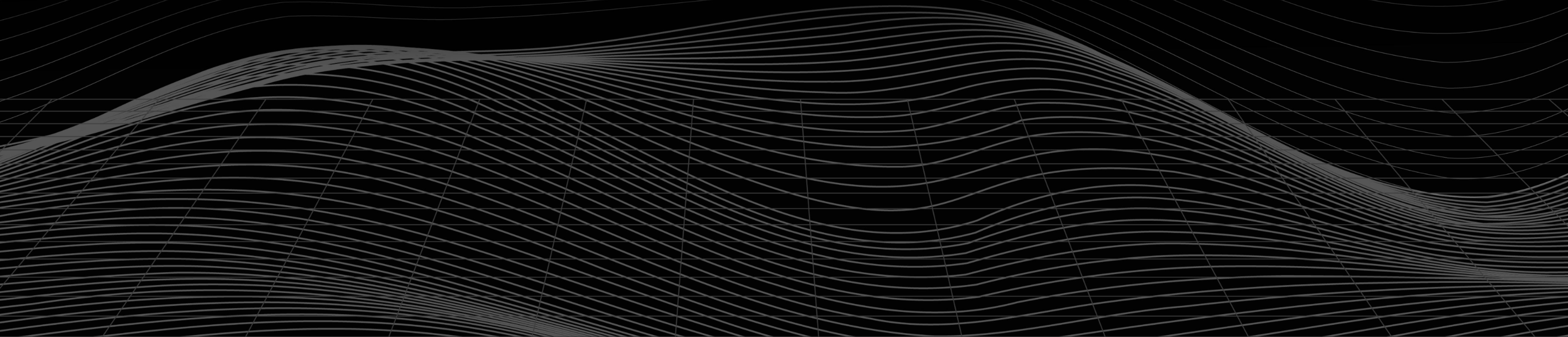
SPATIAL



REVERSED

Word	TextIDs	ID	Text
a	[2]	0	it is what it is
banana	[2]	1	what it is
is	[0, 1, 2]	2	it is a banana
it	[0, 1, 2]		
what	[0, 1]		

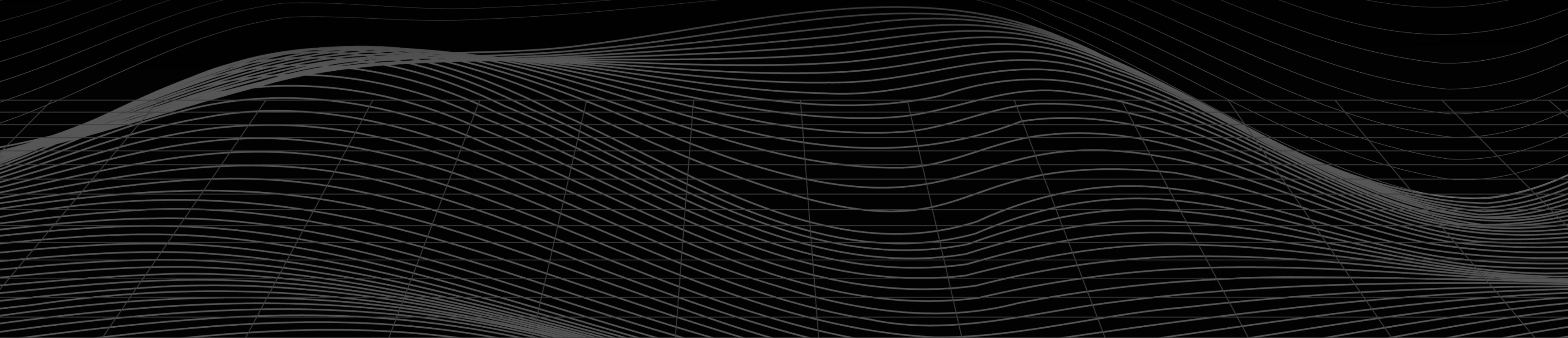
КАКОЙ ВЫБРАТЬ ИНДЕКС ДЛЯ БЫСТРОГО ПОИСКА ПАРТНЕРА НА САЙТЕ ЗНАКОМСТВ ПО ДИСКРЕТНЫМ ПАРАМЕТРАМ?



КАКОЙ ВЫБРАТЬ ИНДЕКС ДЛЯ БЫСТРОГО ПОИСКА ПАРТНЕРА НА САЙТЕ ЗНАКОМСТВ ПО ДИСКРЕТНЫМ ПАРАМЕТРАМ?

– BitMap

КАКОЙ ВЫБРАТЬ ИНДЕКС ДЛЯ ПОИСКА ДОКУМЕНТОВ ПО КЛЮЧЕВЫМ СЛОВАМ?



КАКОЙ ВЫБРАТЬ ИНДЕКС ДЛЯ ПОИСКА ДОКУМЕНТОВ ПО КЛЮЧЕВЫМ СЛОВАМ?

– Reversed

ФУНКЦИОНАЛЬНЫЙ

Индекс, ключи которых хранят результат пользовательских функций

 Например: CREATE INDEX idx_total_price
ON orders (quantity * price);

В этом примере индекс создается в результате умножения столбцов “`quantity`” и “`price`”, что позволяет быстро находить заказы по их общей стоимости

РАЗРЯЖЕННЫЙ ИНДЕКС

Характеризуется тем, что каждый ключ ассоциируется с определённым указателем на блок в сортированном файле данных, а не с какой-то определенной записью



Terminal: System Design × + ▾



ПОКРЫВАЮЩИЙ ИНДЕКС

Это некластеризованные индексы, которые разрешают один или несколько схожих результатов запроса напрямую, без доступа к базовой таблице и без уточняющих запросов

КЛАСТЕРНЫЕ И НЕКЛАСТЕРНЫЕ

При наличии кластерного индекса строки таблицы упорядочены по значению ключа этого индекса

- ⓘ Некластерный индекс, созданный для такой таблицы, содержит только указатели на записи таблицы, либо уникальные идентификаторы записей

FAQ

Индексы

Btree, Hash, Bitmap, Reversed, Spatial,
Кластерные и некластерные, Покрывающие

дополнительные возможности

ХРАНИМЫЕ ПРОЦЕДУРЫ

```
1 CREATE PROCEDURE us_customers ()  
2 LANGUAGE SQL  
3 AS $$  
4   SELECT customer_id, first_name  
5   FROM Customers  
6   WHERE Country = 'USA';  
7 $$;
```

```
1 CALL us_customers();
```

ТРИГГЕРЫ

```
1 CREATE OR REPLACE FUNCTION log_last_name_changes()
2   RETURNS TRIGGER
3   LANGUAGE PLPGSQL
4   AS
5 $$
6 BEGIN
7   IF NEW.last_name <> OLD.last_name THEN
8     INSERT INTO
9       employee_audits(employee_id, last_name, changed_on)
10      VALUES(OLD.id, OLD.last_name, now());
11
12   RETURN NEW;
13 END;
14 $$
```

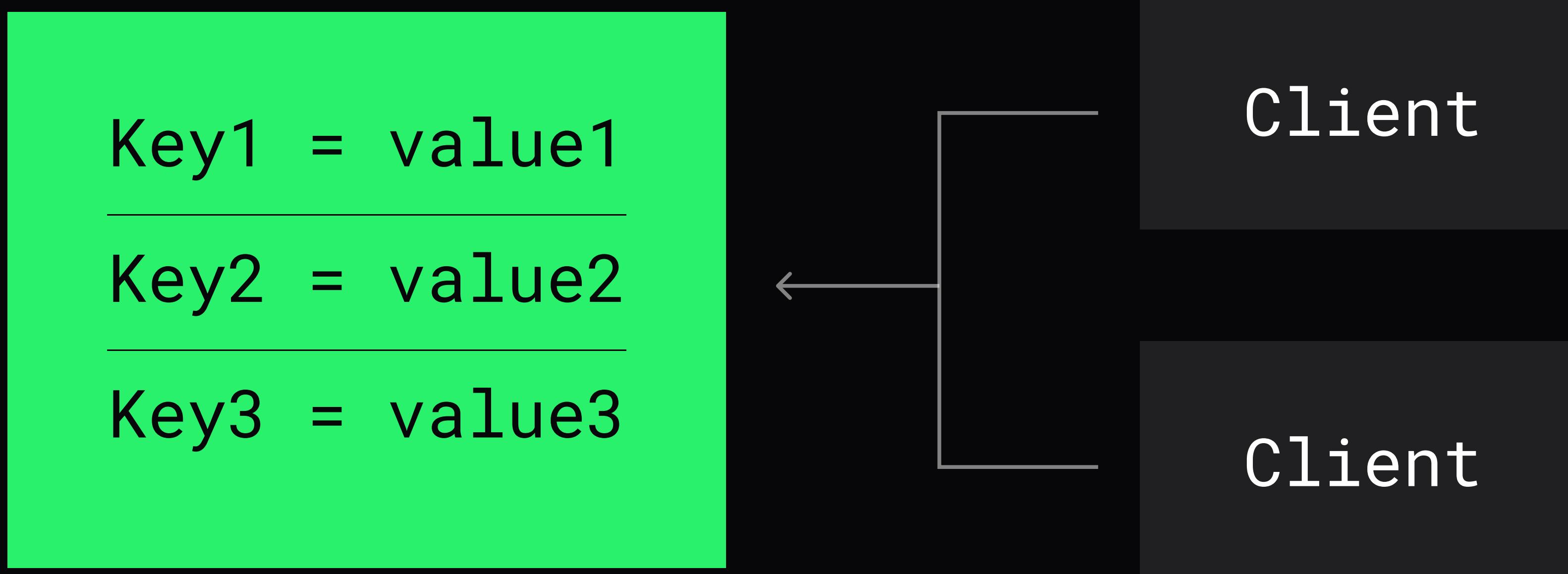
```
1 CREATE TRIGGER last_name_changes
2   BEFORE UPDATE
3   ON employees
4   FOR EACH ROW
5   EXECUTE PROCEDURE log_last_name_changes();
```

MATERIALIZED VIEW

```
1 CREATE MATERIALIZED VIEW dept_stats  
2 REFRESH COMPLETE  
3 AS  
4   SELECT  
5     d.dept_id,  
6     d.dept_name,  
7     COUNT(e.*)  
8       AS emp_count,  
9     SUM(e.salary)  
10    AS total_salary  
11   FROM department d  
12   INNER JOIN employee e  
13     ON d.dept_id = e.dept_id  
14   GROUP BY d.dept_id, d.dept_name;
```

```
1 SELECT dept_id, dept_name, emp_count, total_salary  
2 FROM dept_stats;
```

WATCH API



FAQ

Дополнительные возможности

Триггеры, Хранимые процедуры,
Materialized View, WatchAPI

ПЕРЕРЫВ 5 МИНУТ

ТРАНЗАКЦИИ

Terminal: System Design × + ✓



ACID

Популярные SQL базы данных, появились
как раз на почве ACID

ACID – это стандарт того, какие гарантии должна
давать база данных, чтобы поддерживать транзакции
(он не указывает деталей реализации)

Terminal: System Design × + ▾



АТОМАРНОСТЬ

Каждая транзакция базы данных является единым блоком, который использует подход «**все или ничего**» к выполнению



Если какой-либо оператор в транзакции терпит неудачу, вся транзакция откатывается

OK

Error

user_id (111) = 1500
user_id (222) = 1500

UPDATE account
SET balance = balance - 500
WHERE user_id = 111

UPDATE account
SET balance = balance + 500
WHERE user_id = 222

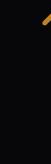
user_id (111) = 1000
user_id (222) = 1500

ROLLBACK

OK



Error



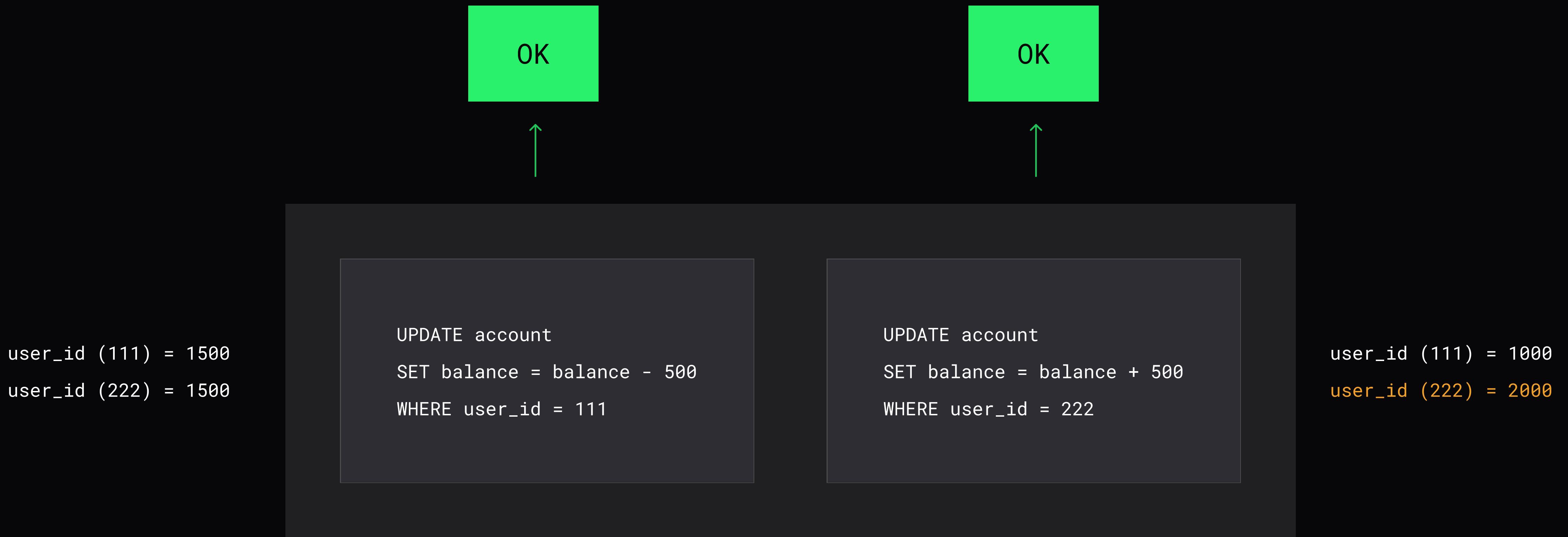
```
user_id (111) = 1500  
user_id (222) = 1500
```

```
UPDATE account  
SET balance = balance - 500  
WHERE user_id = 111
```

```
UPDATE account  
SET balance = balance + 500  
WHERE user_id = 222
```

```
user_id (111) = 1500  
user_id (222) = 1500
```

COMMIT



Terminal: System Design × + ▾



СОГЛАСОВАННОСТЬ

Различные утверждения относительно
данных (инварианты) должны всегда
оставаться справедливыми



Constraint	Description
NOT NULL	values cannot be null
UNIQUE	values cannot match any older value
PRIMARY KEY	used to uniquely identify a row
FOREIGN KEY	references a row in another table
CHECK	validates condition for new value
DEFAULT	set default value if not passed
CREATE INDEX	used to speedup the read process

DEFERRABLE TRANSACTIONS



```
1 CREATE TABLE husbands (
2     id int PRIMARY KEY,
3     wife_id int NOT NULL
4 );
5
6 CREATE TABLE wives (
7     id int PRIMARY KEY,
8     husband_id int NOT NULL
9 );
10
11 ALTER TABLE husbands ADD CONSTRAINT h_w_fk
12     FOREIGN KEY (wife_id) REFERENCES Wives;
13
14 ALTER TABLE wives ADD CONSTRAINT w_h_f
15     FOREIGN KEY (husband_id) REFERENCES husbands;
```

Terminal: System Design × + ▾



ИЗОЛЯЦИЯ ТРАНЗАКЦИЙ

Каждая транзакция происходит до или после каждой другой транзакции, и представление базы данных, которое транзакция видит в своем начале, изменяется только самой транзакцией до ее завершения



Ни одна транзакция не должна видеть промежуточный продукт другой транзакции

ПОТЕРЯННОЕ ОБНОВЛЕНИЕ

video_id (10) = 100

```
UPDATE view  
SET count = count + 1  
WHERE video_id = 10;
```

video_id (10) = 101

video_id (10) = 100

```
UPDATE view  
SET count = count + 1  
WHERE video_id = 10;
```

video_id (10) = 101

ГРЯЗНОЕ ЧТЕНИЕ

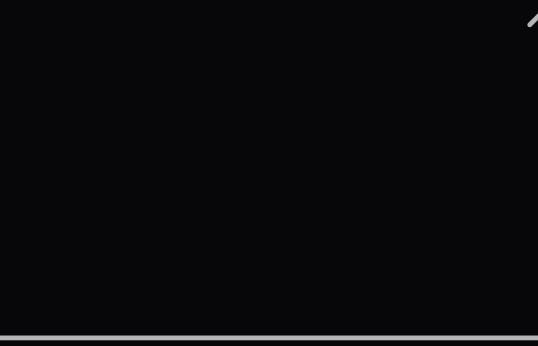
```
user_id (111) = 1500
```

```
START TRANSACTION;  
  
UPDATE account  
SET balance = balance - 500  
WHERE user_id = 111;  
  
ROLLBACK;
```

```
user_id (111) = 1500
```

```
SELECT balance  
FROM account  
WHERE user_id = 111;
```

```
usser_id(111) = 1000
```



```
user_id (111) = 1500
```

```
user_id (111) = 1500
```

НЕПОВТОРЯЮЩЕЕСЯ ЧТЕНИЕ

user_id (111) = 1500

```
START TRANSACTION;

UPDATE account
SET balance = balance + 500
WHERE user_id = 111;

COMMIT;
```

user_id (111) = 1500

```
START TRANSACTION;
```

```
SELECT balance
FROM account
WHERE user_id = 111;

SELECT balance
FROM account
WHERE user_id = 111;
```

balance = 1500

balance = 2000

user_id (111) = 2000

user_id (111) = 2000

ЧТЕНИЯ «ФАНТОМОВ»

```
user_id (111) = 1500  
user_id (222) = 2500
```

```
START TRANSACTION;  
  
INSERT INTO  
    account (user_id, balance)  
VALUES (333, 1000);  
  
COMMIT;
```

```
user_id (111) = 2000  
user_id (222) = 2500  
user_id (333) = 1000
```

```
user_id (111) = 1500  
user_id (222) = 2500
```

```
START TRANSACTION;  
  
SELECT SUM (balance)  
FROM account;
```

```
SELECT SUM (balance)  
FROM account;
```

sum = 4000

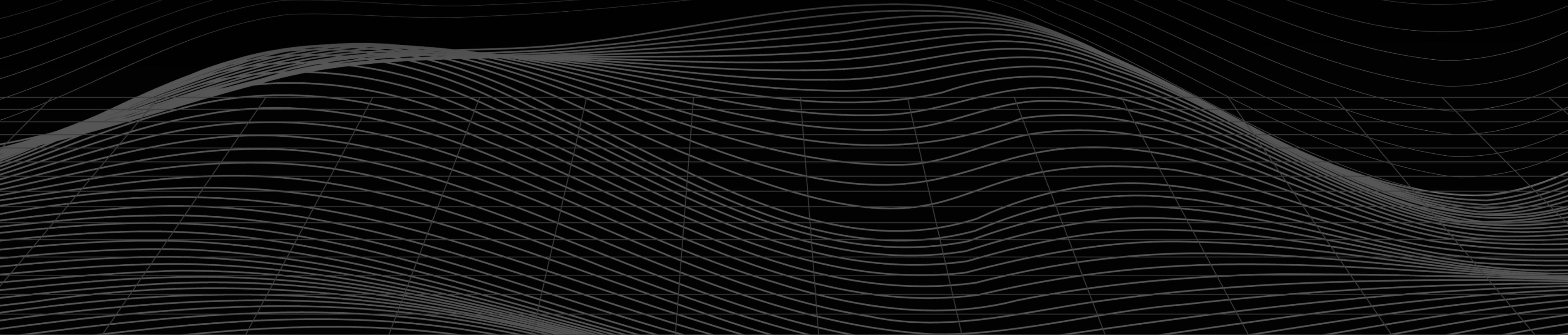
sum = 5000

```
user_id (111) = 2000  
user_id (222) = 2500  
user_id (333) = 1000
```

УРОВНИ ИЗОЛЯЦИИ

Уровень изоляции	Фантомное чтение	Неповторяющееся чтение	Грязное чтение	Потерянное обновление
SERIALIZABLE	+	+	+	+
REPEATABLE READ	-	+	+	+
READ COMMITTED	-	-	+	+
READ UNCOMMITTED	-	-	-	+

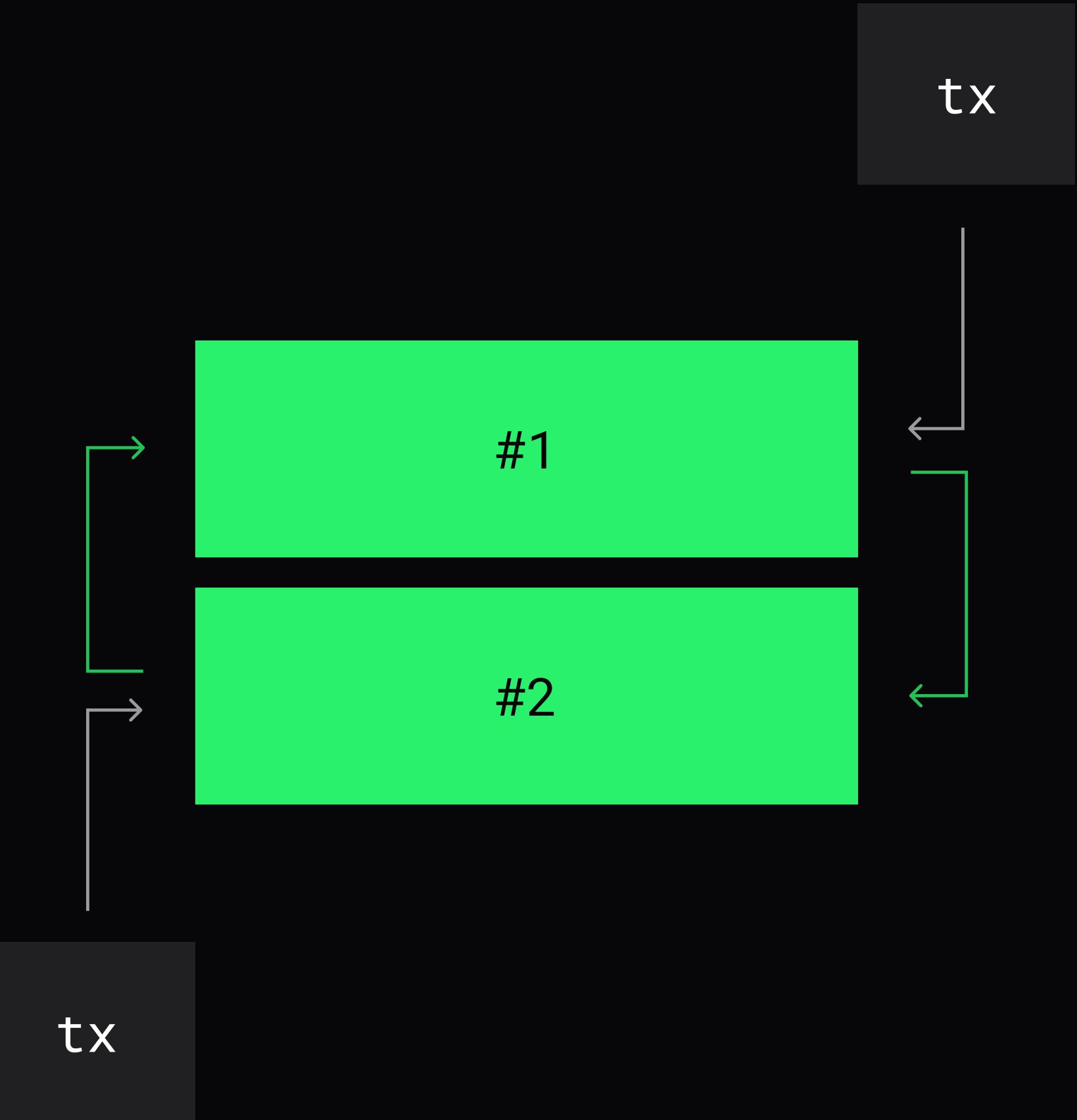
ЧЕМ ВЫШЕ УРОВЕНЬ ИЗОЛЯЦИИ
ТРАНЗАКЦИЙ, ТЕМ НИЖЕ ПРОПУСКНАЯ
СПОСОБНОСТЬ БАЗЫ ДАННЫХ



2PL

Фаза расширения – запрашиваются все необходимые для определённой транзакции блокировки и никакие блокировки не высвобождаются

Фаза сжатия – все полученные на фазе роста блокировки высвобождаются



MVCC

t_xmin	t_xmax	...	user_data
--------	--------	-----	-----------

t_xmin – идентификатор транзакции, создавшей запись

t_xmax – идентификатор транзакции, удалившей запись

MVCC

1		...	user_data_1
2		...	user_data_2
3	5	...	user_data_3
4		...	user_data_4
5		...	user_data_5
6		...	user_data_6

Terminal: System Design × + ▾



УСТОЙЧИВОСТЬ

Гарантирует, что после фиксации транзакции в базе данных она постоянно сохраняется с помощью резервных копий и журналов транзакций



В случае сбоя эти механизмы могут использоваться для восстановления зафиксированных транзакций

Terminal: System Design × + ▾



WAL (WRITE-AHEAD LOG)

Изменения в файлах с данными должны записываться только после того, как эти изменения были занесены в журнал



Записывать страницы данных на диск после подтверждения каждой транзакции нет необходимости

1. Пишем в журнал

WAL

2. Отпускаем клиента

3. В фоне доносим изменения
в движок базы данных

Engine

SET key1 value1

key1 = value1

SET key2 value2

key2 = value2

DEL key1

key3 = value3

SET key3 value3

SET key1 value1

Terminal:

System Design ×



BASE

(Basically Available, Soft state, Eventually consistent

– это альтернативная концепция, которая фокусируется на гибкости и доступности в ущерб строгой согласованности



Базы данных, построенные на принципах BASE, могут быть более гибкими и могут позволять временные несогласованности данных, которые в конечном итоге будут согласованы

FAQ

Транзакции

ACID / BASE, Уровни изоляции,
2PL / MVCC, WAL



LET'S PRACTISE

БРОКЕРЫ СООБЩЕНИЙ

БРОКЕРЫ СООБЩЕНИЙ

Плюсы:

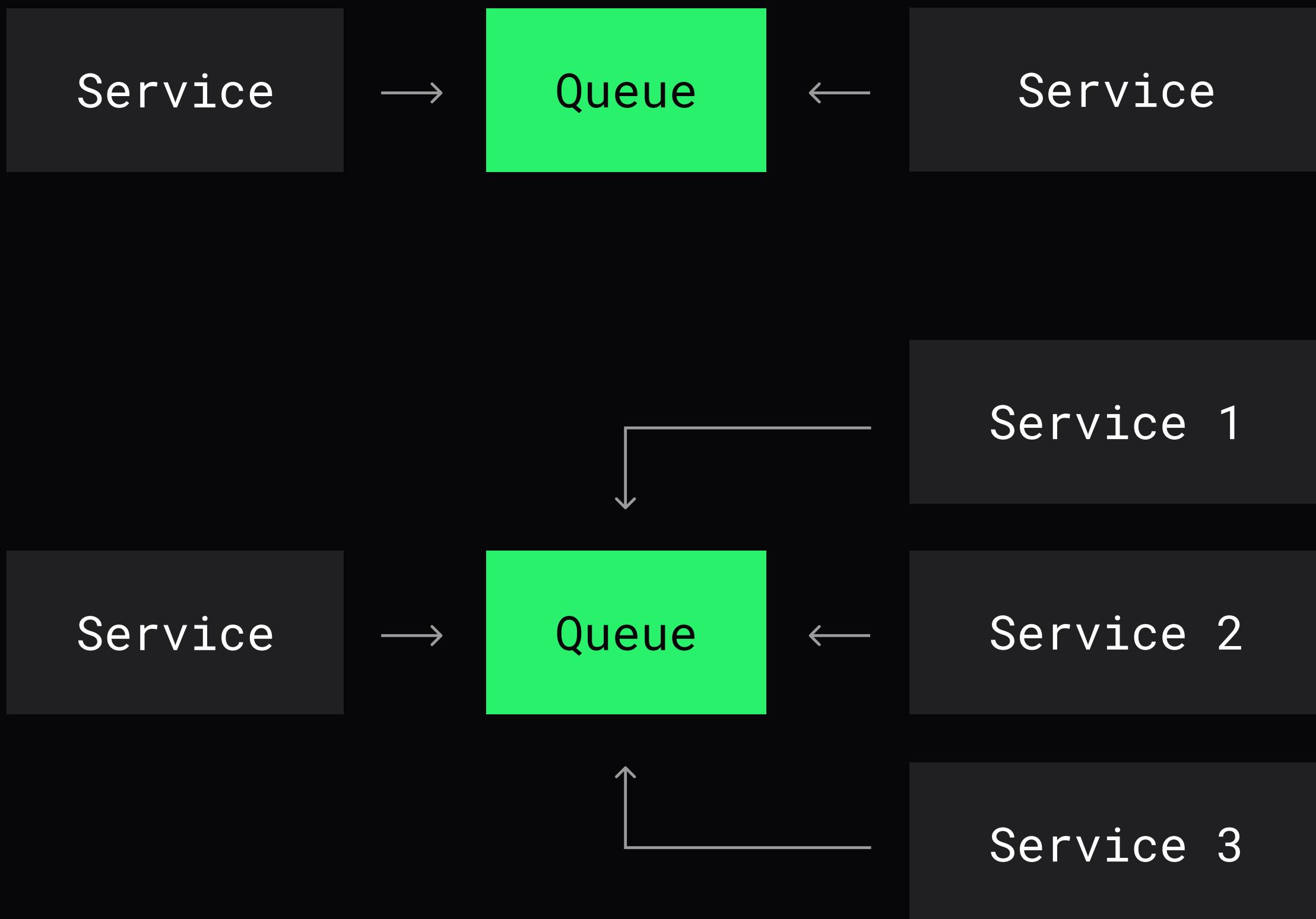
1/ Асинхронная связь

2/ Слабое связывание

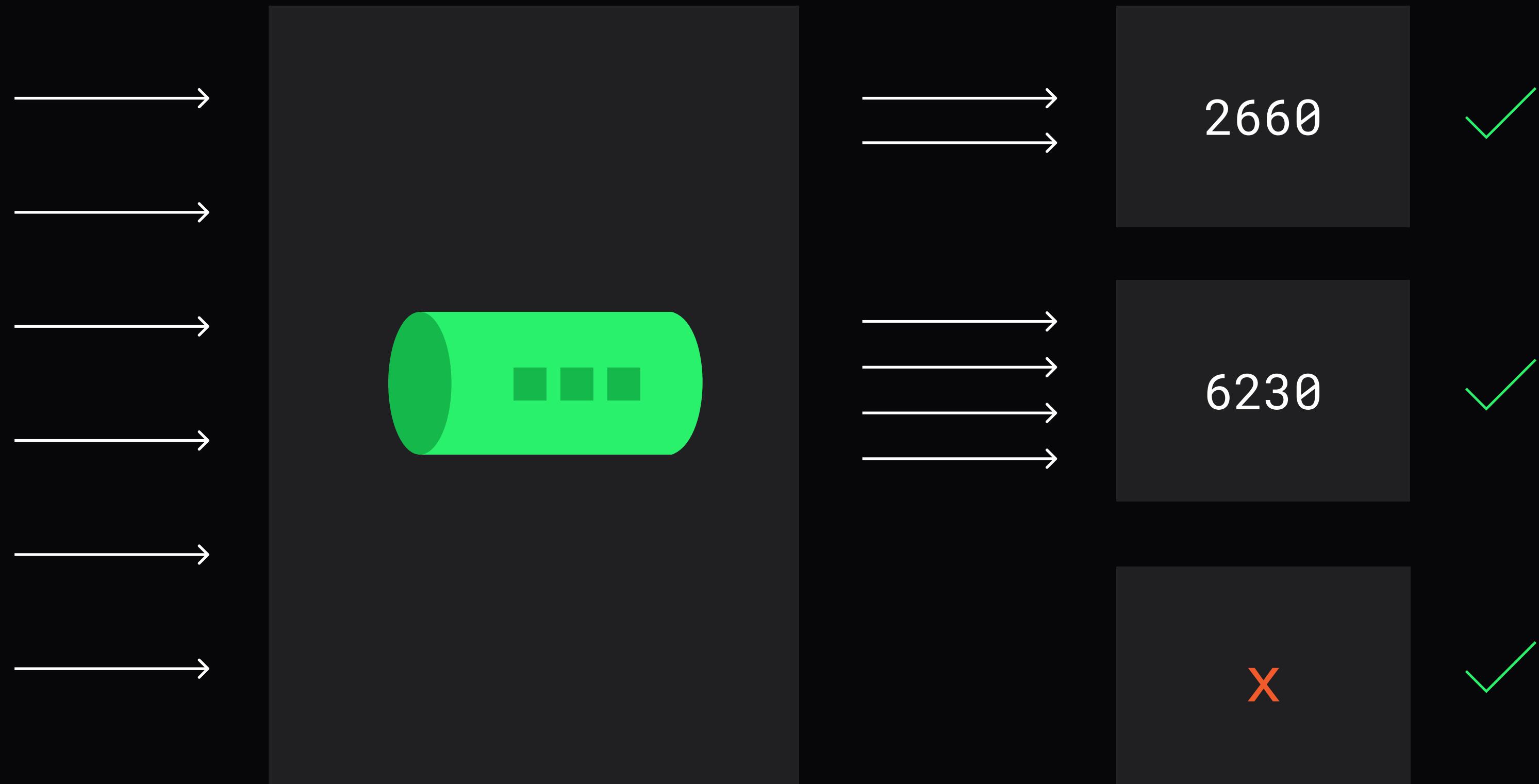
3/ Масштабируемость

4/ Отказоустойчивость

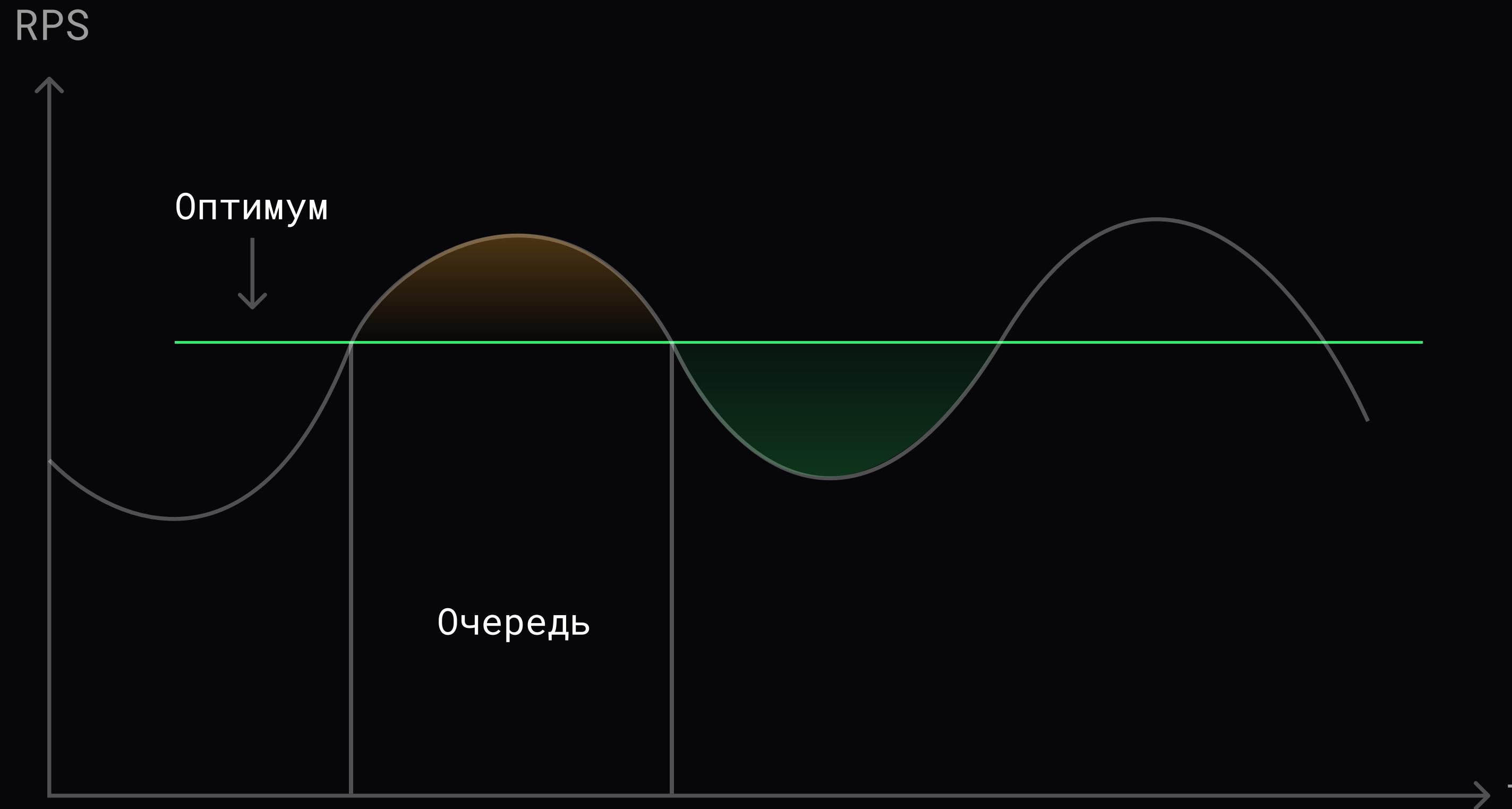
5/ Понимание потоков данных



РАСПРЕДЕЛЕНИЕ ЗАДАЧ



ПЛАНИРОВАНИЕ ИСПОЛНЕНИЯ



KAFKA

Producer

писатель

Consumer

читатель

Broker

один узел Kafka

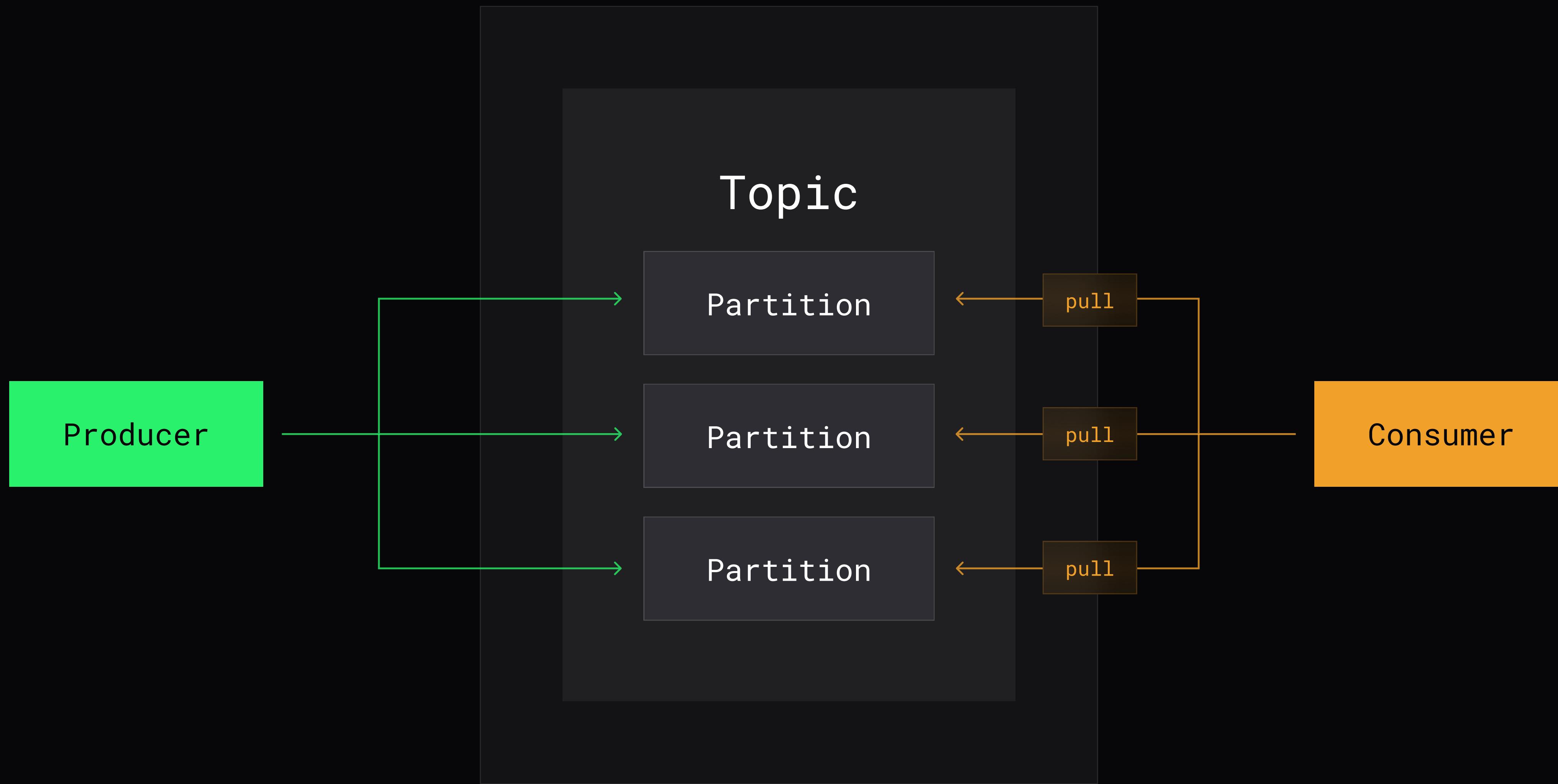
Topic

логическая очередь

Partition

физическая часть очереди

KAFKA





Terminal: System Design × + ▾

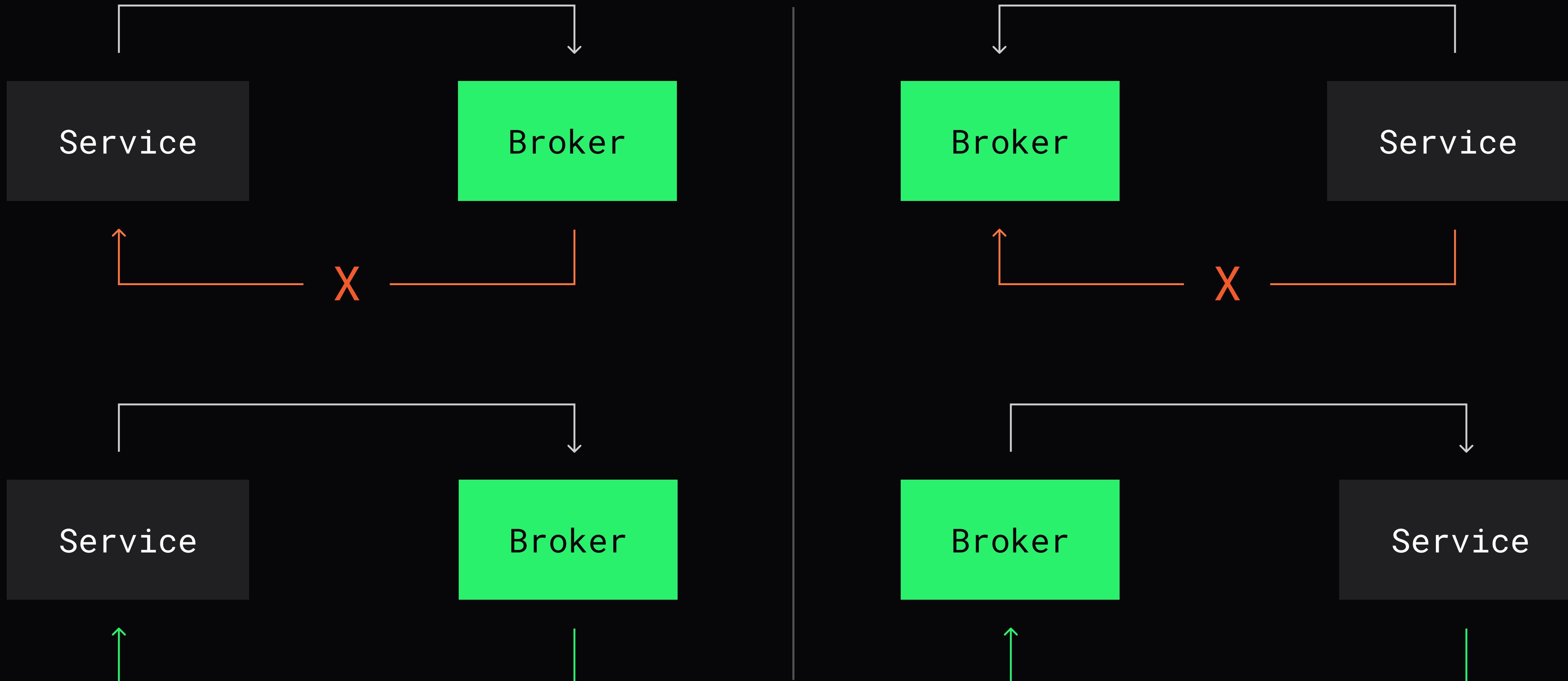


DATA RETENTION

Относится к стратегии и политикам, определяющим, как долго данные должны храниться в базе данных.

Это включает в себя определение периода хранения данных, после которого они могут быть удалены, модифицированы или архивированы

КАК БЫТЬ С ДУБЛИКАТАМИ?



ГАРАНТИИ ДОСТАВКИ

At least once

сообщение будет доставлено
хотя бы один раз

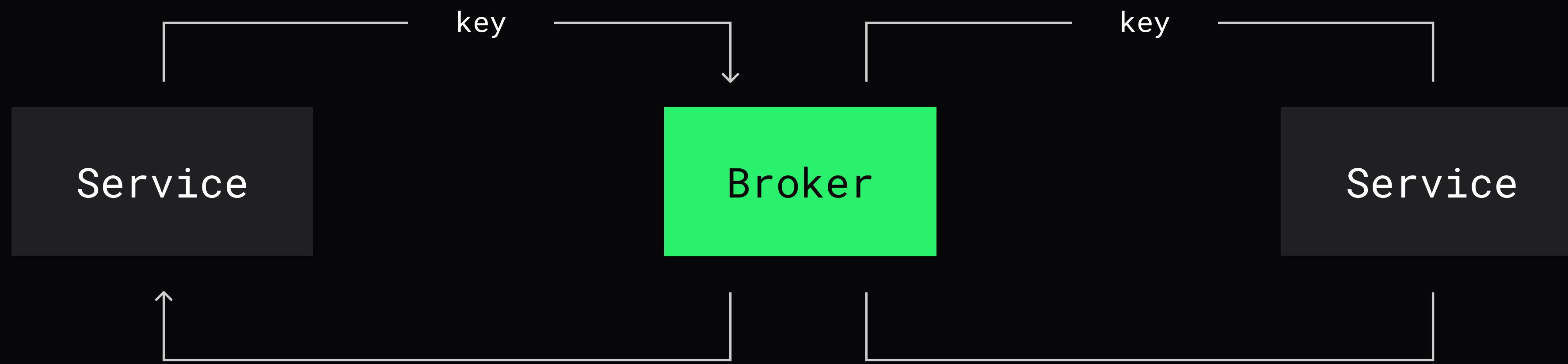
At most once

сообщение будет доставлено
не более одного раза

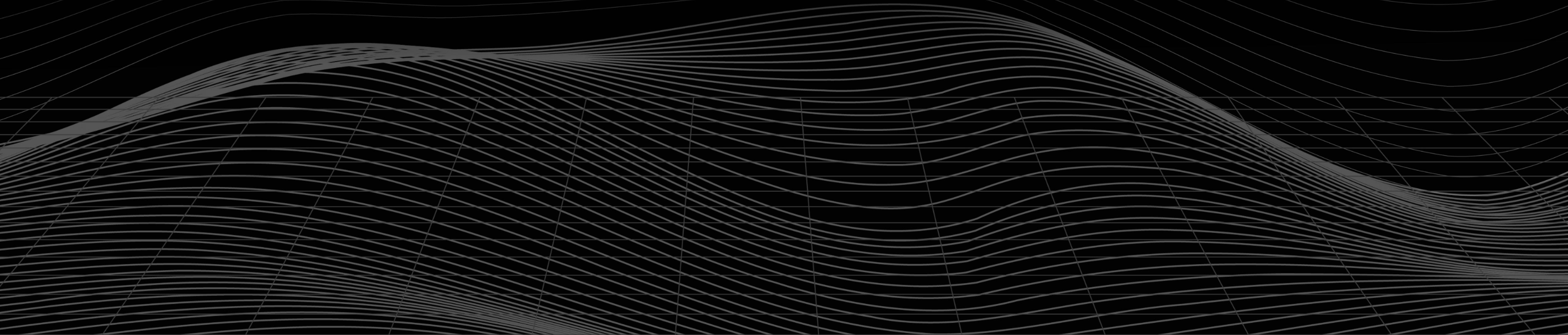
Exactly once

сообщение будет доставлено
ровно один раз

КЛЮЧ ИДЕМПОТЕНТНОСТИ



МОЖНО ЕЩЕ ДЕЛАТЬ
ДЕДУПЛИКАЦИЮ ПРИ ЧТЕНИИ,
А НЕ ПРИ ЗАПИСИ



FAQ

Брокеры сообщений

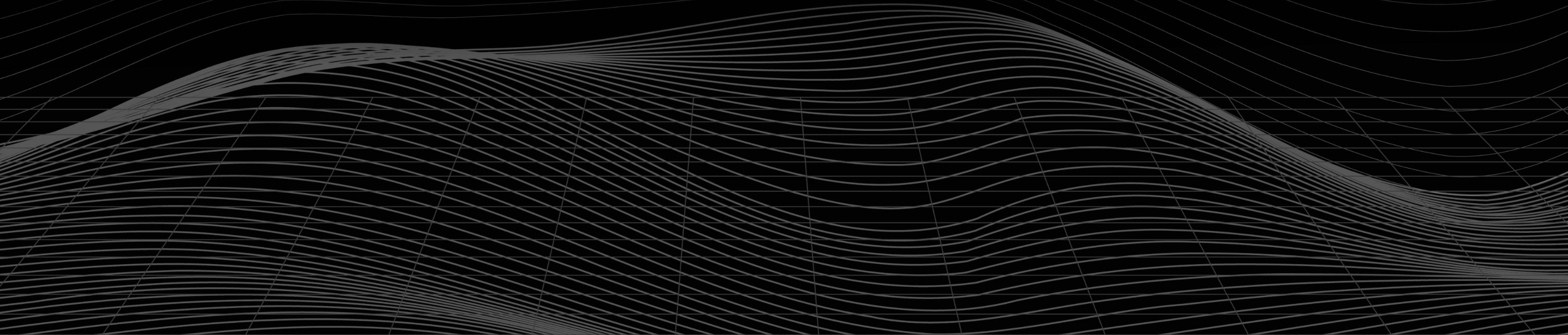
Kafka, Data retention,

Гарантии доставки

ПАТТЕРНЫ ХРАНЕНИЯ ДАННЫХ

ДЕНОРМАЛИЗАЦИЯ

КАК ДЕШЕВЛЕ
ХРАНИТЬ ДАННЫЕ?

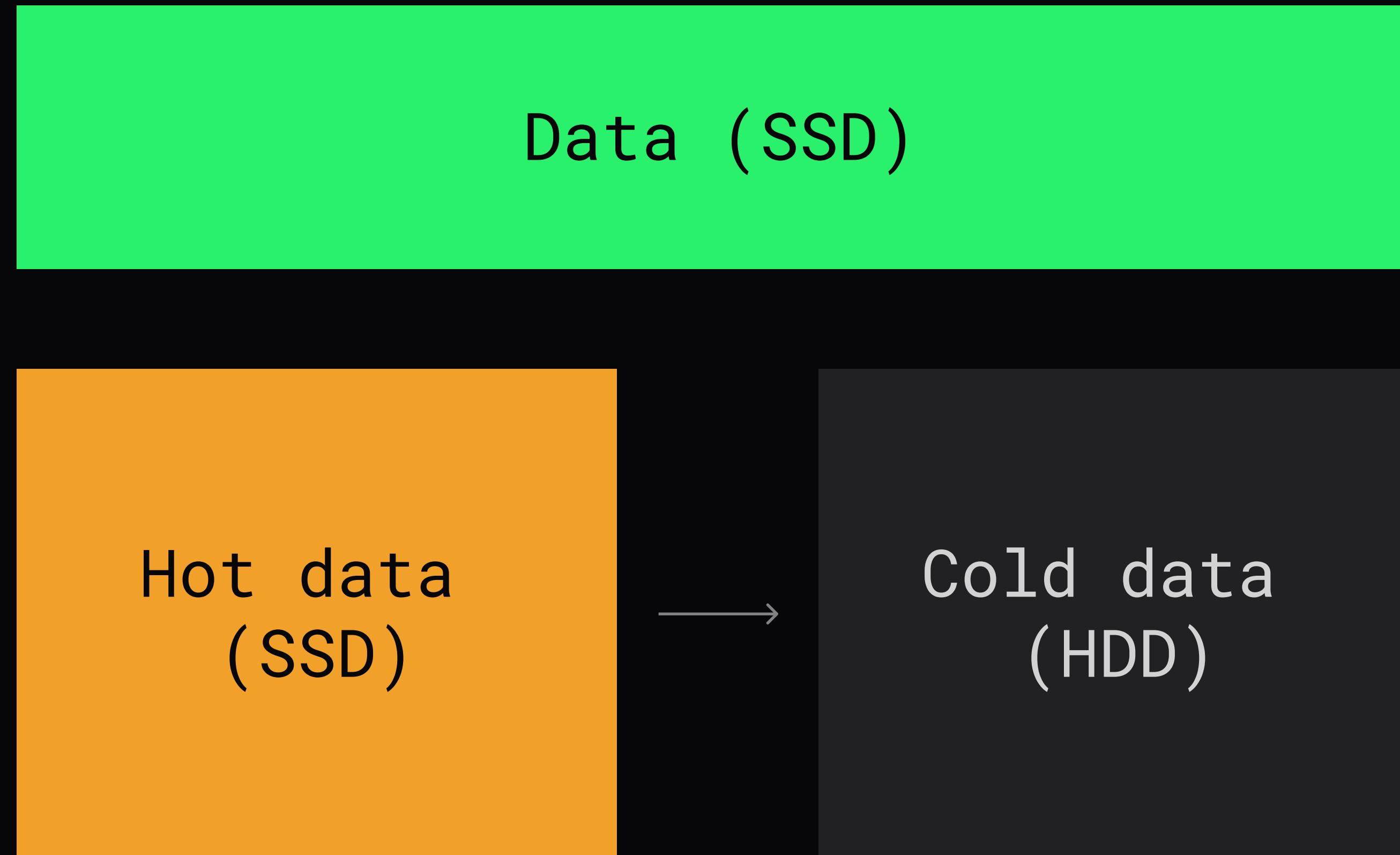


СЖАТИЕ ДАННЫХ

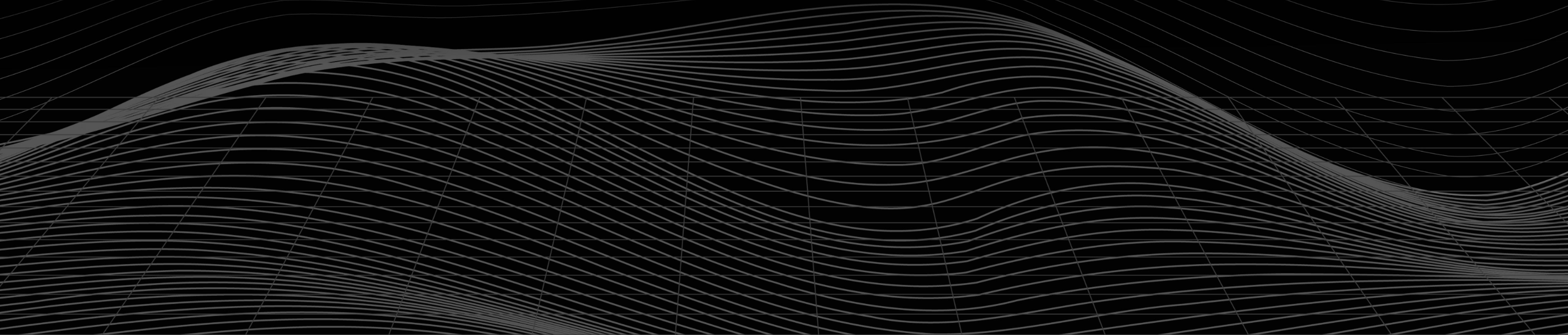
Source
data

Compressed
data

ОХЛАЖДЕНИЕ ДАННЫХ

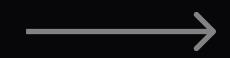


КАК БЫСТРО СТРОИТЬ ЕЖЕМЕСЯЧНЫЕ ОТЧЕТЫ?



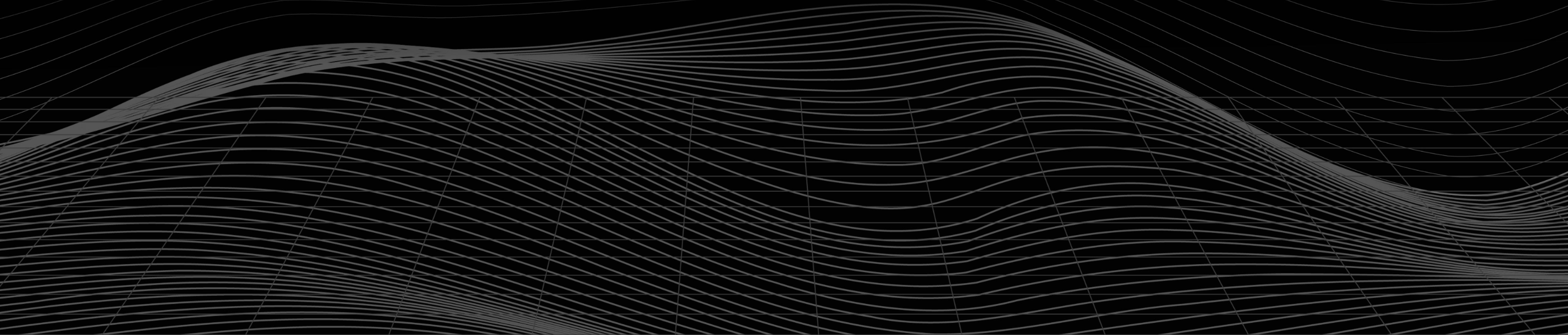
ДЕЛАЕМ ДОПОЛНИТЕЛЬНУЮ РАБОТУ ПРИ ЗАПИСИ

Aggregate table



Data table

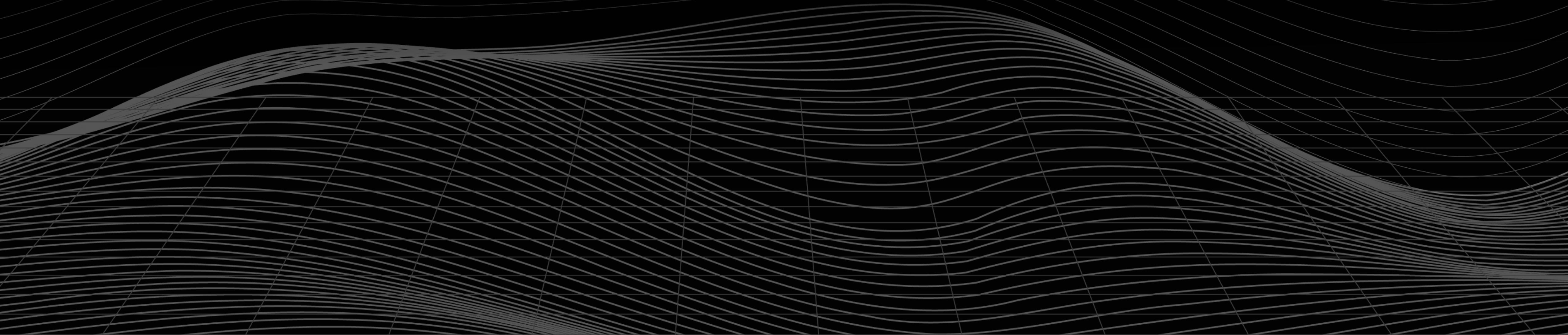
КАК НАДЕЖНО ЗАПИСЫВАТЬ ДАННЫЕ В БАЗУ ДАННЫХ?



ПИШЕМ ЧЕРЕЗ ОЧЕРЕДЬ



КАК ЗАПИСЫВАТЬ БОЛЬШЕ
ДАННЫХ В БАЗУ ДАННЫХ?

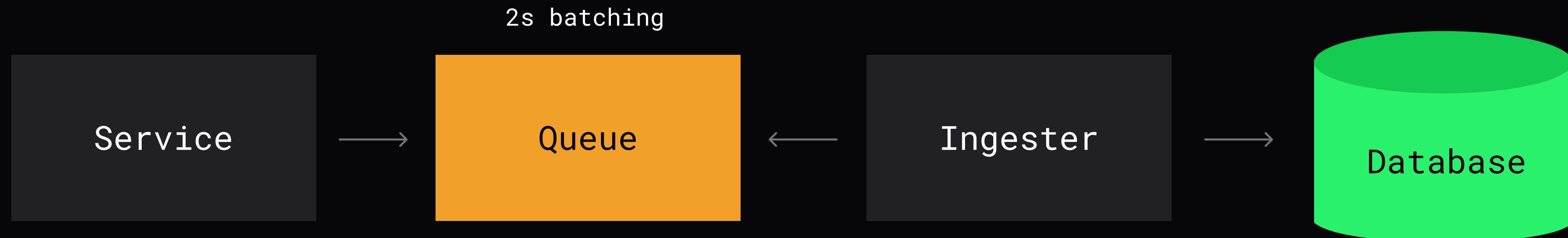


ПИШЕМ БАТЧАМИ

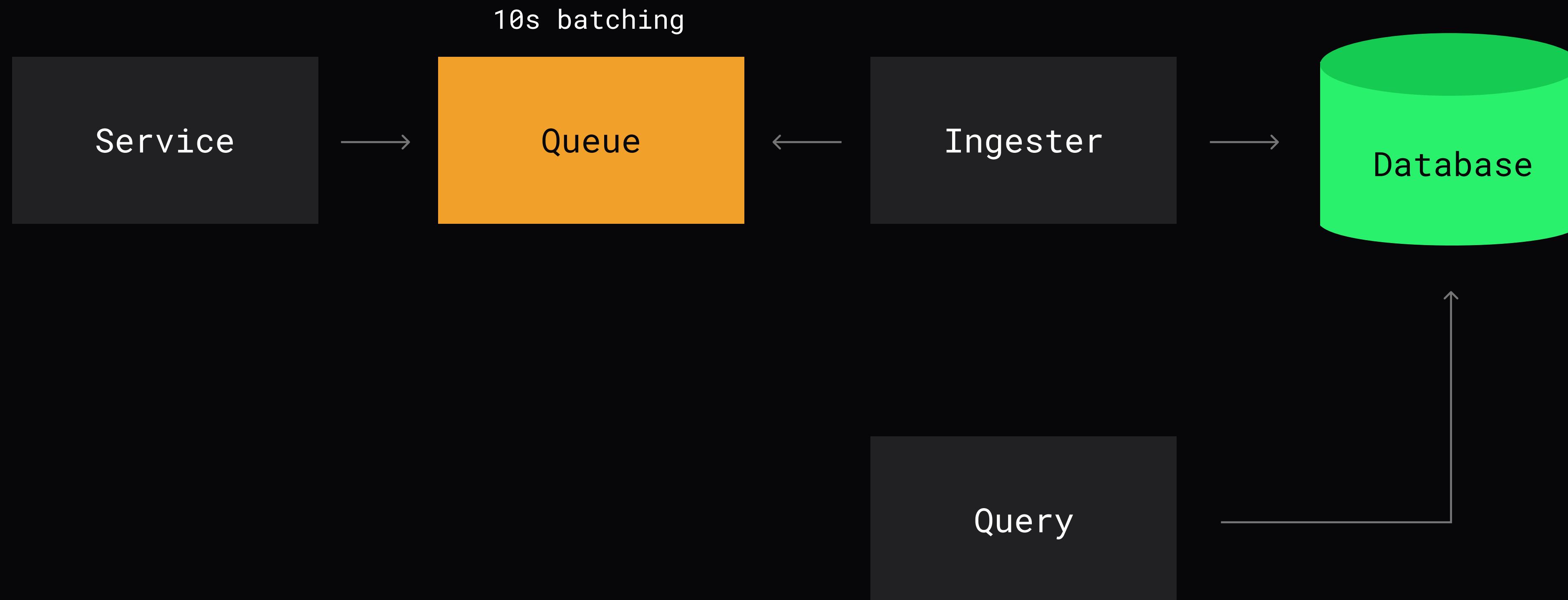
#1



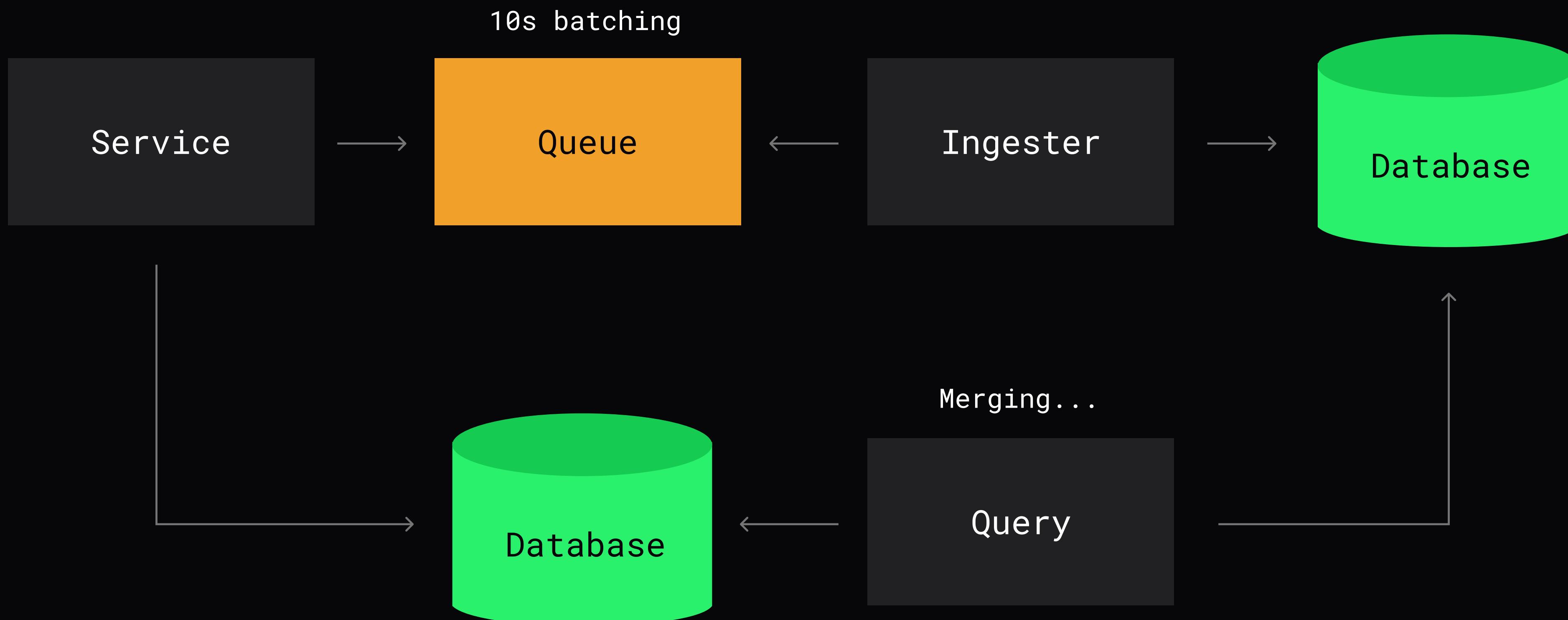
#2



КАК СДЕЛАТЬ, ЧТОБЫ ПОЛЬЗОВАТЕЛЬ МОГ ЧЕРЕЗ СЕКУНДУ НАЙТИ СВОИ ДАННЫЕ?



ПИШЕМ В ДВА МЕСТА



FAQ

Паттерны хранения и поставки данных

РАСЧЕТ РЕСУРСОВ ДЛЯ СИСТЕМЫ

подтема №1

СЕТЬ

Terminal: System Design × + ▾



ПРОПУСКНАЯ СПОСОБНОСТЬ

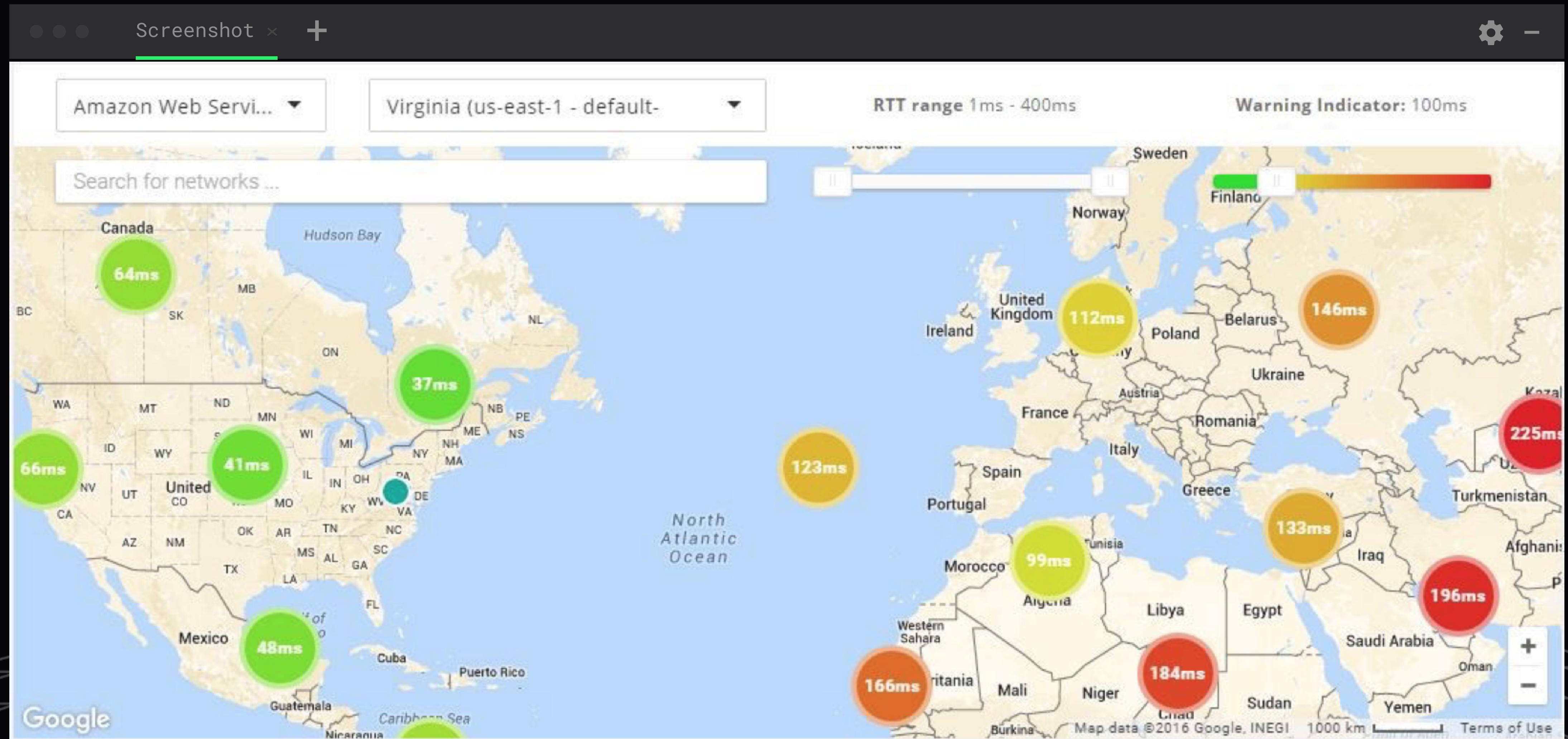
Показывает, какой максимальный объем может быть передан в единицу времени (чаще всего это одна секунда)

ПИНГ (RTT – ROUND-TRIP TIME)

Время, за которое отправленный в сеть запрос достигает адресата и возвращается обратно



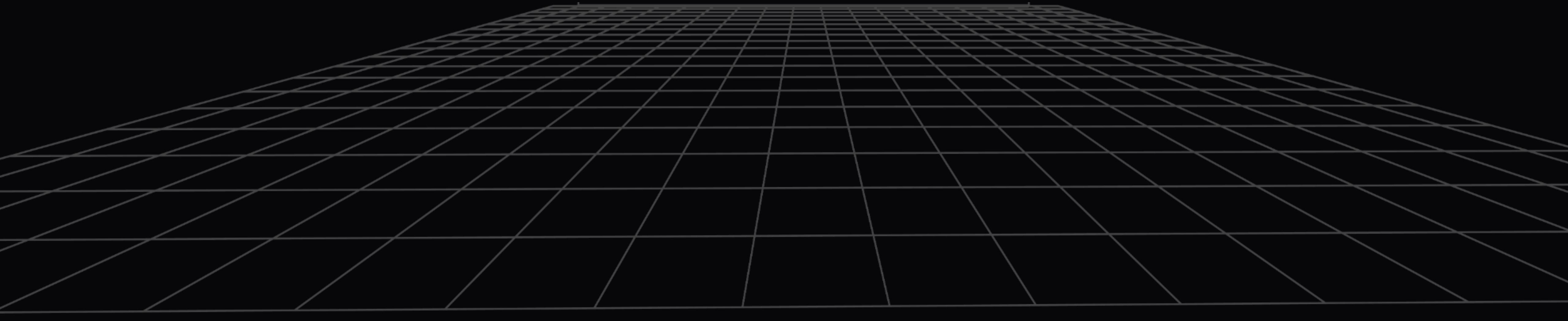
Считается, что хороший пинг – до 100 мс. Пинг, равный 100–200мс означает, что производительность может понизиться, но ваши пользователи по-прежнему смогут пользоваться сервисом



FAQ

Сеть

Пропускная способность,
пинг(RTT)



подтема №2

RAM

ТАКТОВАЯ ЧАСТОТА

Означает скорость передачи данных между процессором и памятью

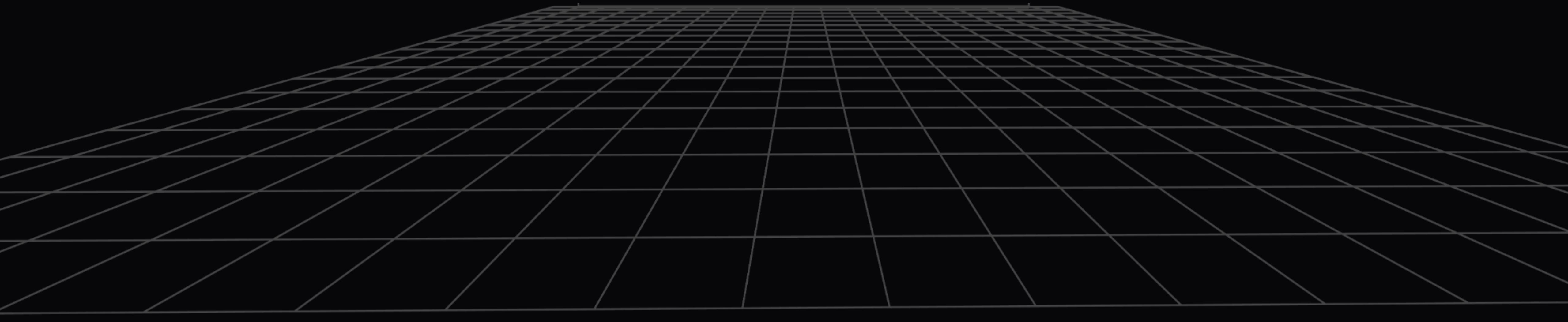
- ⓘ По частоте можно посчитать пропускную способность конкретной планки в мегабайтах - просто умножайте её частоту на 8

	DDR3	DDR4	DDR5
Размер	до 16 ГБ	до 128 ГБ	до 256 ГБ
Частота	1066-1866 МГц	2133-4266 МГц	4800-8400 МГц
Пропускная способность	от 8 528 МБ/с до 14 928 МБ/с	от 17 064 МБ/с до 34 128 МБ/с	от 38 400 МБ/с до 67 200 МБ/с
Задержка	13мкс	14мкс	16мкс
Стоимость	60\$ (8ГБ x 4)	80\$ (16ГБ x 2)	110\$ (16ГБ x 2)

FAQ

RAM

Частота, тайминги



подтема №3

CPU

ТАКТОВАЯ ЧАСТОТА

Говорит, сколько циклов обработки происходит в секунду, а IPC (Instructions Per Cycle) определяет среднее количество инструкций, выполняемых процессором за один такт

Terminal: System Design × + ▾



ФОРМУЛА ДЛЯ ОЦЕНКИ КОЛ-ВА ОПЕРАЦИЙ

Количество операций = Тактовая частота процессора * IPC

3 ГГц * 2 IPC = 6 000 000 000 оп/с



Приближенное значение без учитывания многих факторов

КОЛИЧЕСТВО ЯДЕР

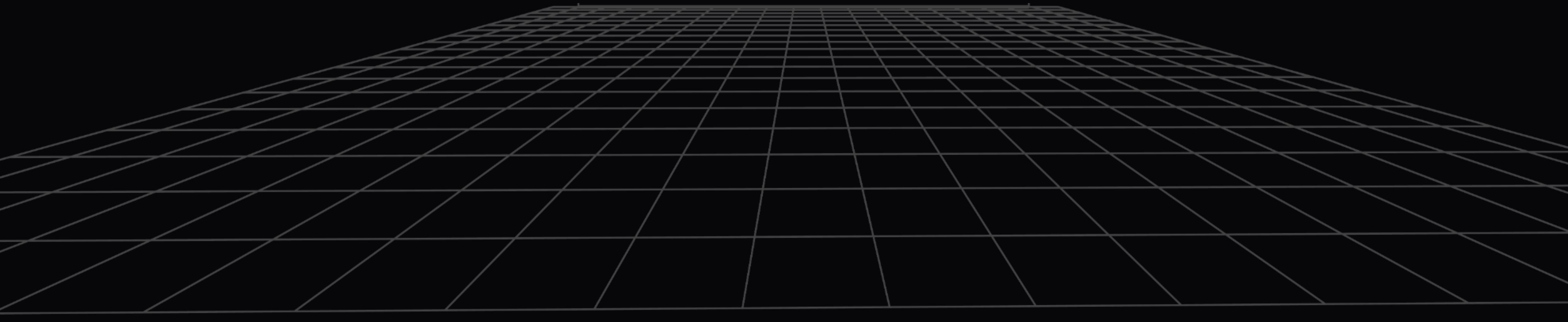
Ядро – это независимый процессор, способный выполнять вычисления, что позволяет ему выполнять несколько задач одновременно

- ⓘ Если процессор с четырьмя ядрами – значит он может выполнять четыре задачи одновременно

FAQ

CPU

Тактовая частота, кол-во ядер



подтема №4

диски

РАЗМЕР

В отличие от принятой в информатике системы приставок, обозначающих кратную 1024 величину, производителями при обозначении ёмкости жёстких дисков используются величины, кратные 1000



Например, ёмкость жёсткого диска, маркированного как «200 ГБ», составляет 186.2ГБ, а также часть пространства диска может быть занята системными файлами и служебной информацией

Terminal: System Design × + ▾



ВРЕМЯ ПРОИЗВОЛЬНОГО ДОСТУПА

Время, за которое диск гарантированно выполнит
операцию чтения или записи на любом участке диска



КОЛИЧЕСТВО ОПЕРАЦИЙ ВВОДА-ВЫВОДА(IOPS)

IOPS показывает, какое количество блоков успевает записаться/считаться за секунду



максимальная скорость, на практике диск будет функционировать с меньшей скоростью

	HDD	SSD (SATA)	SSD (nVME)
Размер	до 32 ТБ	до 100 ТБ	до 30 ТБ
Операций ввода-вывода	100	1000	10000
Пропускная способность	100 МБ/с	500 МБ/с	3 ГБ/с
Задержка	2мс	100-200мкс	10-20мкс
Стоимость	100\$ (6ТБ)	200\$ (6ТБ)	500\$ (6ТБ)



Дополнительно еще можно посмотреть на размер дискового кэша

ЧТО ДЕЛАТЬ

если не хватает пропускной
способности или размера диска?

RAID-МАССИВ

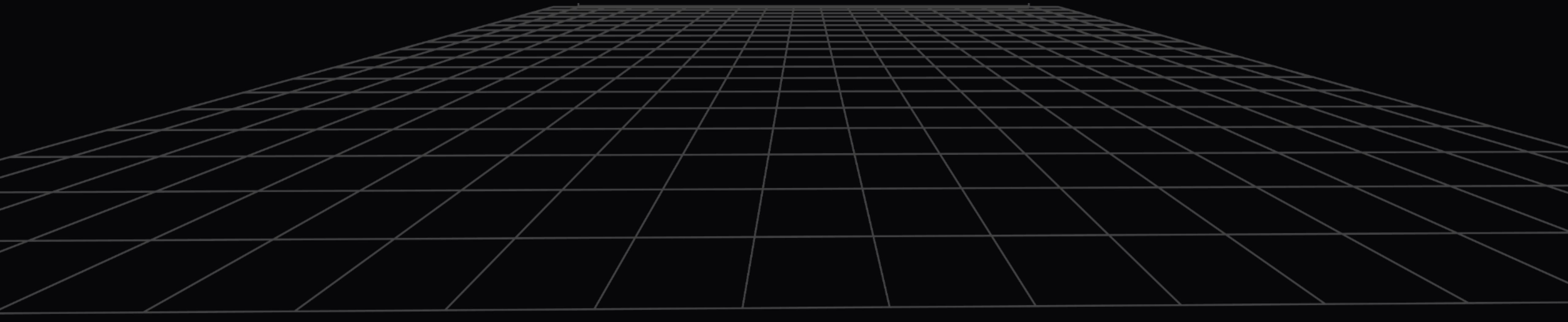
RAID – (Redundant Arrays of Independent Discs) избыточный массив независимых дисков, используемый для увеличения объема, пропускной способности и отказоустойчивости дисков



FAQ:

Диски

Размер, время произвольного доступа,
количество операций ввода-вывода, RAID



НУЖНО НЕ ЗАБЫТЬ
ОБЕСПЕЧИТЬ ЗАПАС РЕСУРСОВ

НА 15–20%

Когда вы хотите обеспечить отказоустойчивость на
уровне двух дата-центров, вам нужно обеспечить
запас ресурсов, как минимум на 50%



LET'S PRACTISE

ДОМАШНЕЕ ЗАДАНИЕ

ДО ВСТРЕЧИ
НА СЛЕДУЮЩЕМ
ЗАНЯТИИ!