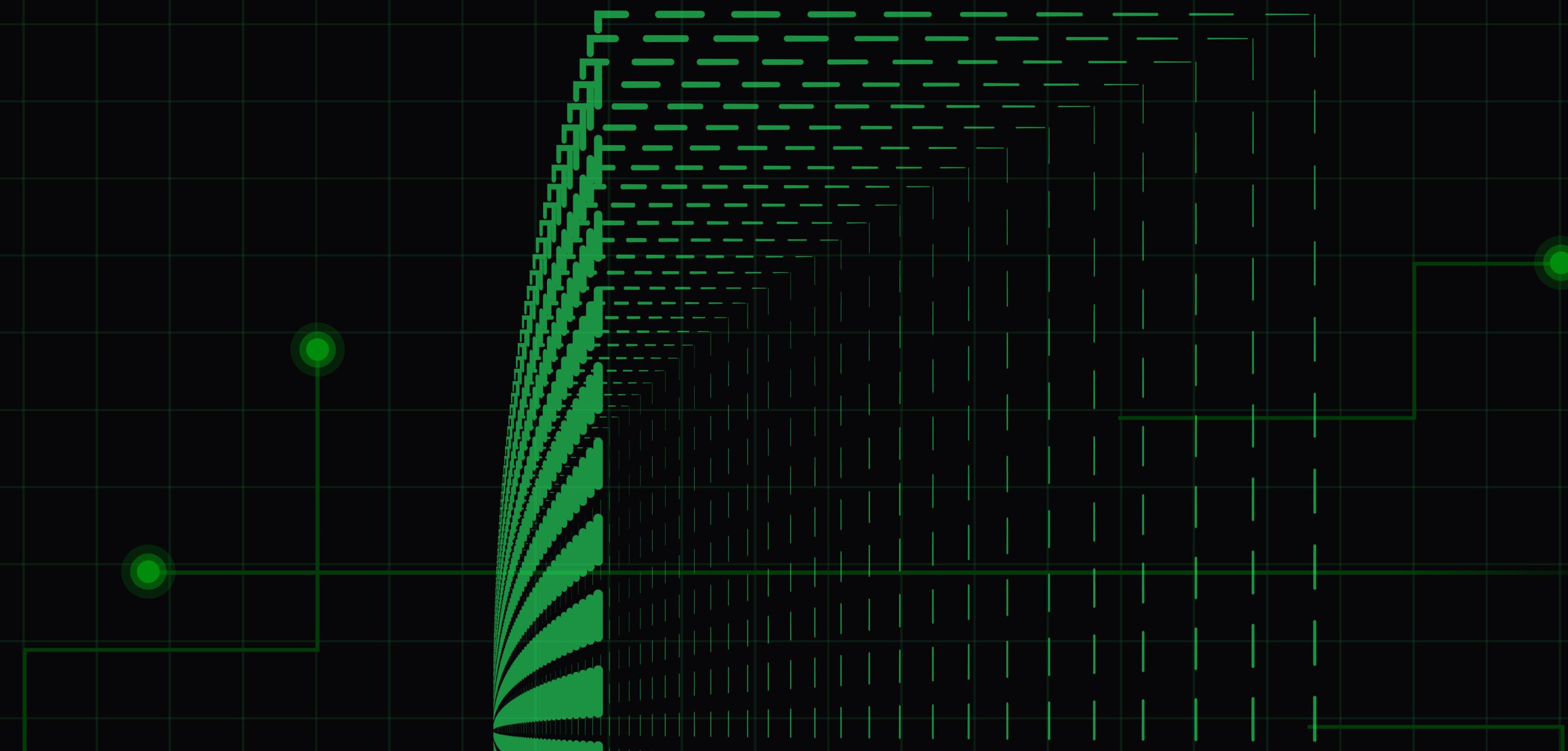
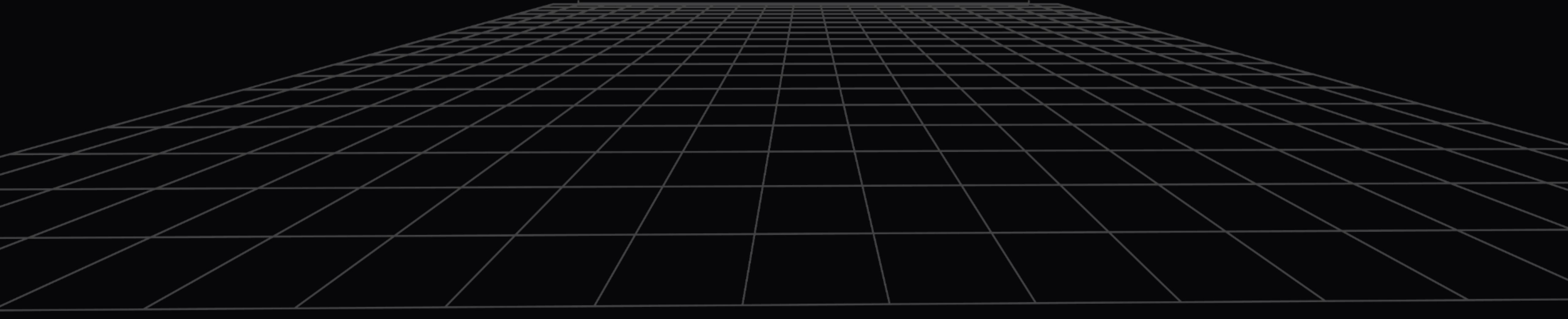


SYSTEM DESIGN

Основы проектирования систем

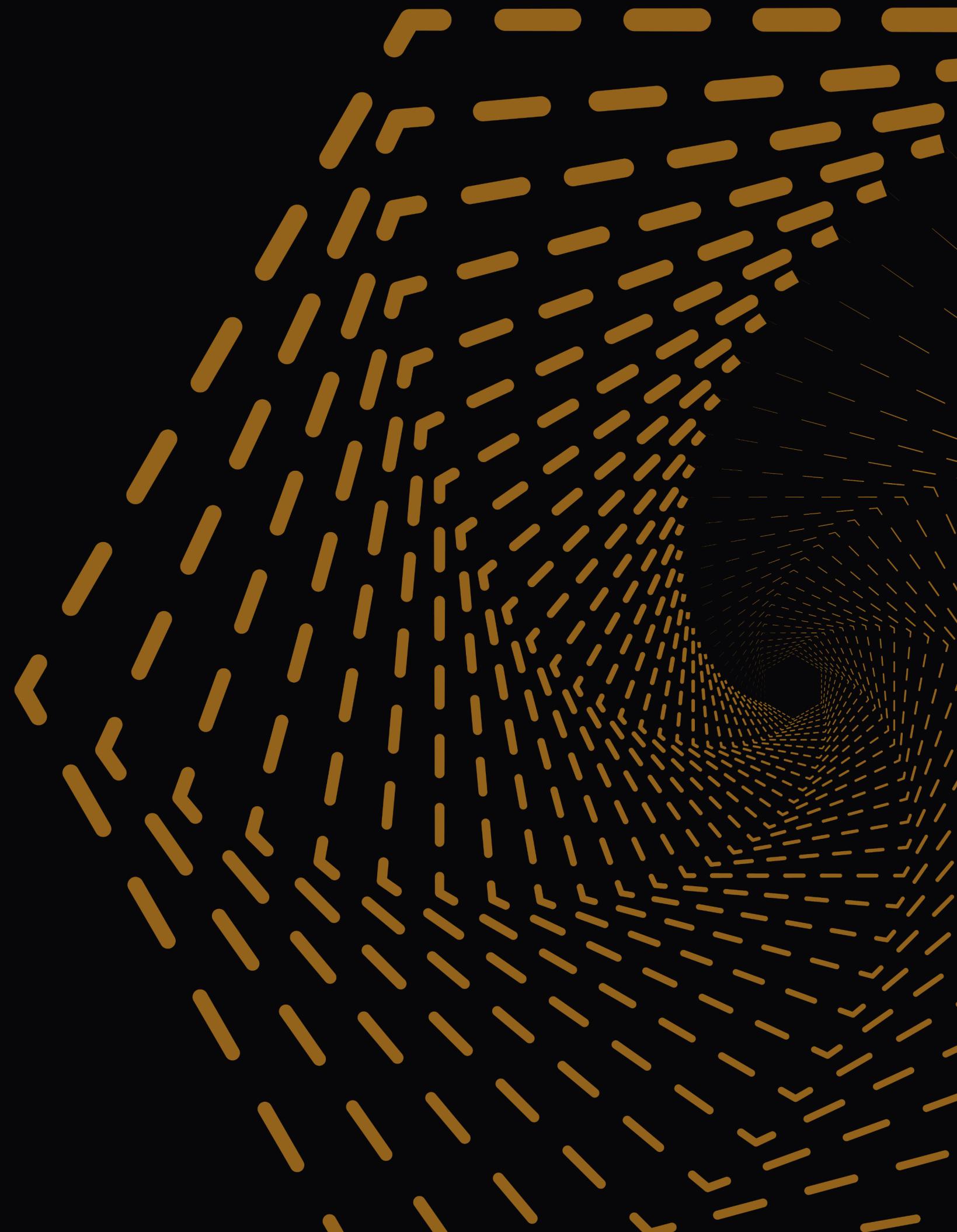


ПРОВЕРЬ ЗАПИСЬ



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. Основные свойства информационных систем
2. Балансировка нагрузки и проксирование
3. Установка требований к системе
4. Расчет нагрузки на систему

ОСНОВНЫЕ СВОЙСТВА ИНФОРМАЦИОННЫХ СИСТЕМ

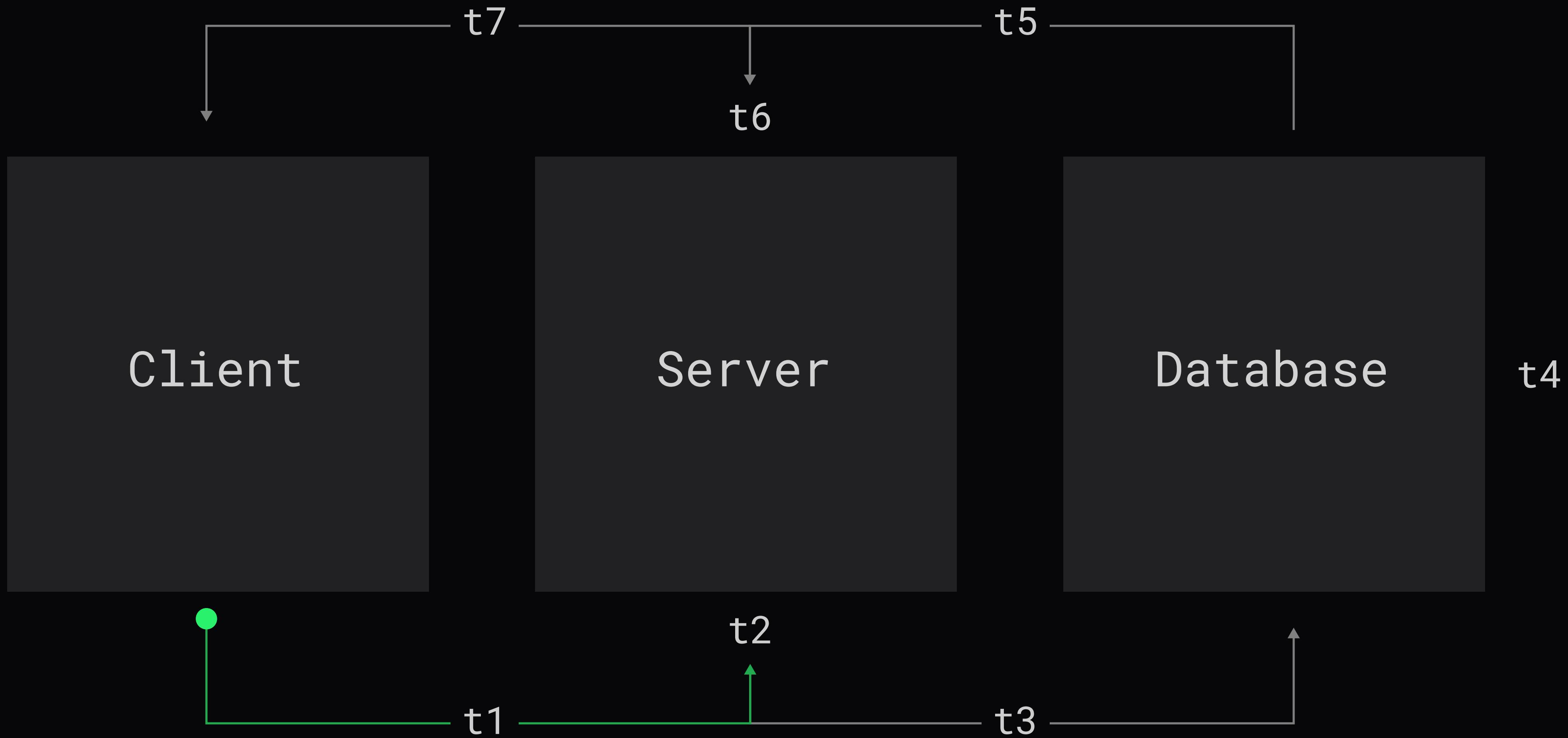
подтема №1

ПРОИЗВОДИТЕЛЬНОСТЬ

ОСНОВНЫЕ ПОКАЗАТЕЛИ

latency	время обслуживания одного клиента
throughput	число обработанных клиентов в единицу времени

LATENCY



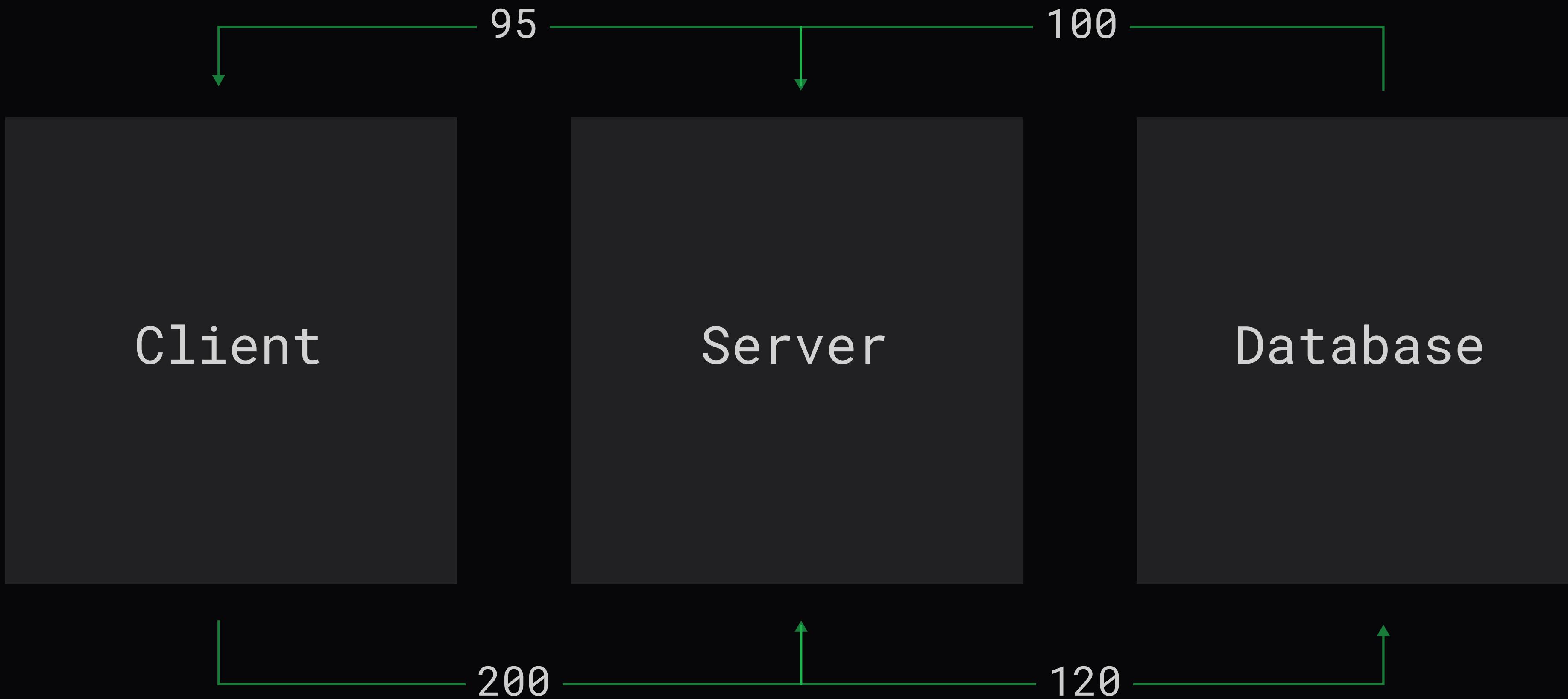
LOW-LATENCY ПРИЛОЖЕНИЯ

Такие приложение надо с самого начала писать с оптимизациями, с прямой ориентацией на low-latency.



Здесь не работает подход, когда пишут сначала просто работающее приложение, а потом оптимизируют его

THROUGHPUT



HIGH-THROUGHPUT ПРИЛОЖЕНИЯ

Приложения, заточенные под максимально возможную пропускную способность



latency в этом случае часто пренебрегают

Terminal: System Design × + ▾



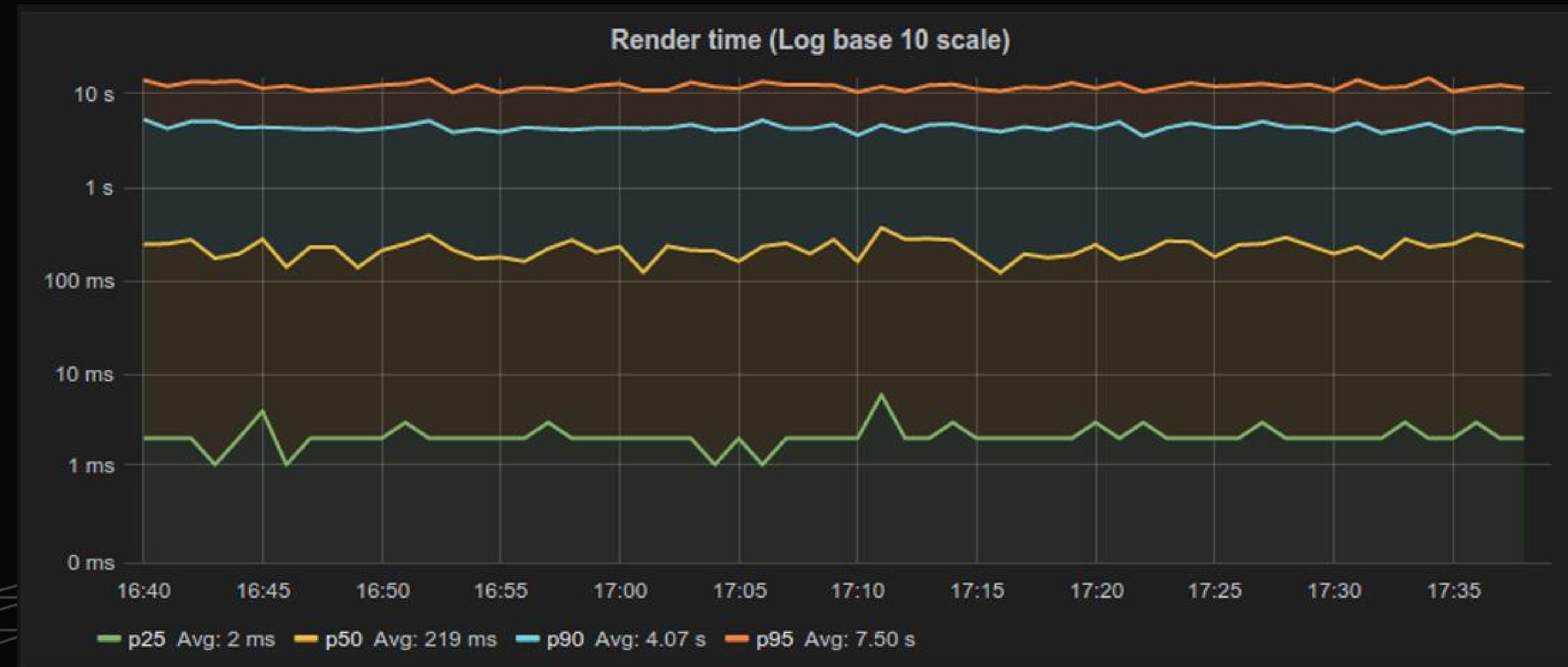
«ТИПИЧНЫЕ» ПРИЛОЖЕНИЯ

Нужен максимально возможный throughput
по какому-то определенному latency



ИЗМЕРЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

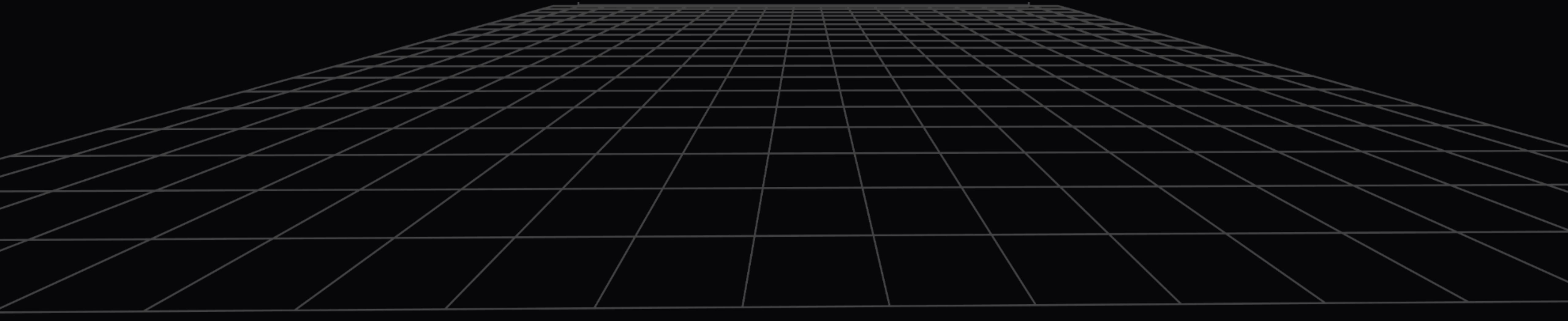
Средних значений не достаточно, важны перцентили



FAQ

Производительность

latency, throughput

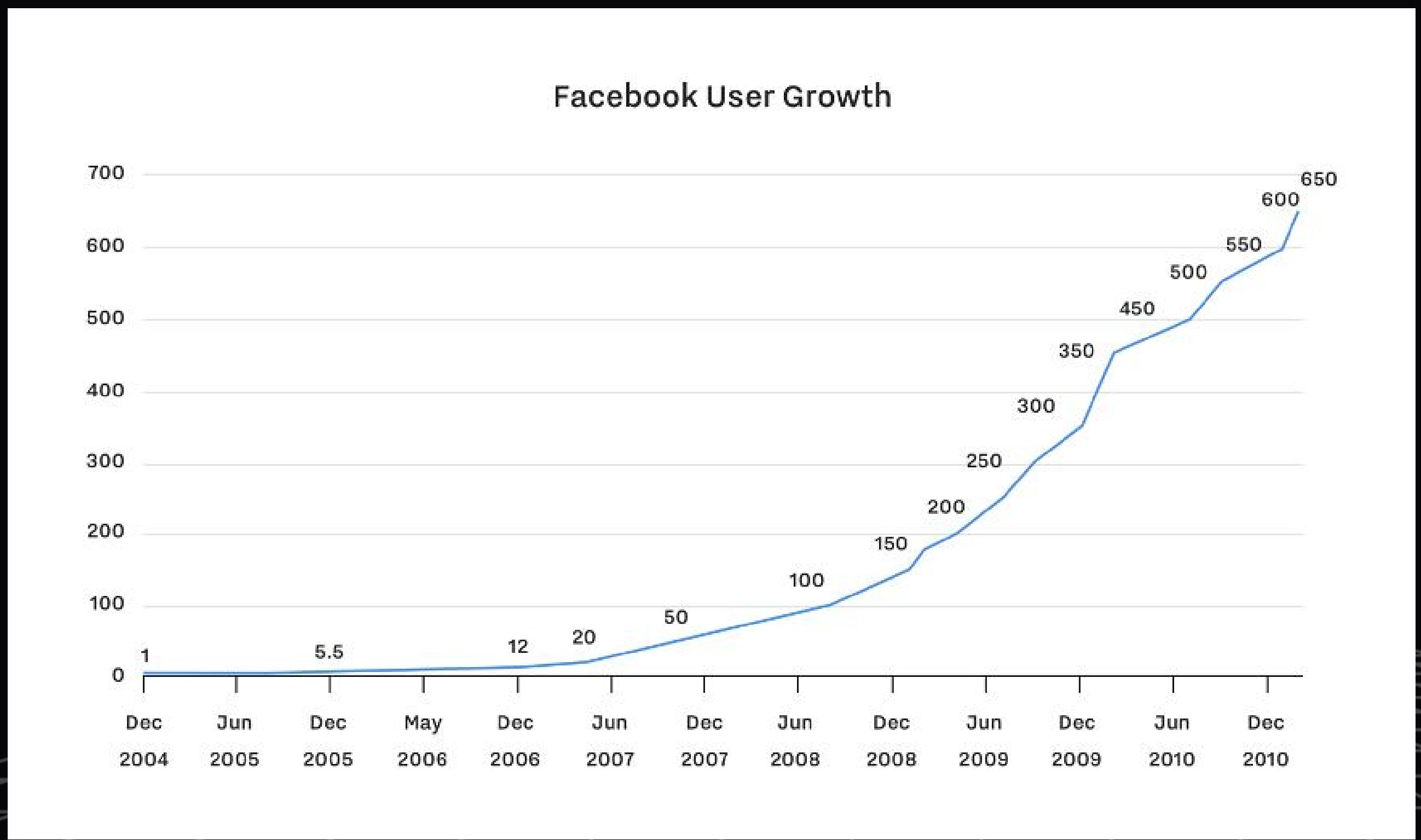


подтема №2

МАСШТАБИРУЕМОСТЬ

МАСШТАБИРУЕМОСТЬ

Способность системы расти без потери производительности и других характеристик, а также без необходимости менять программную реализацию



ПАРАМЕТРЫ НАГРУЗКИ

/1 число запросов в секунду (RPS)

/2 трафик (KB/MB/GB/TB per second)

/3 число одновременных соединений (c10k, c100k)

ЧТО ИЗ ЭТОГО HIGHLOAD?

20 RPS или 20000 RPS

Terminal: System Design × + ▾



READ / WRITE RATIO

Соотношение операций чтения и записи

DATA-INTENSIVE

- /1 Нужно сохранять большие данные
- /2 Нужно запоминать результаты ресурсоемких операций
- /3 Нужно предоставлять пользователям возможность искать в больших объемах данных

ПРИМЕРЫ

/1 Система по хранению и поиску логов

/2 Файловое хранилище

COMPUTE-INTENSIVE

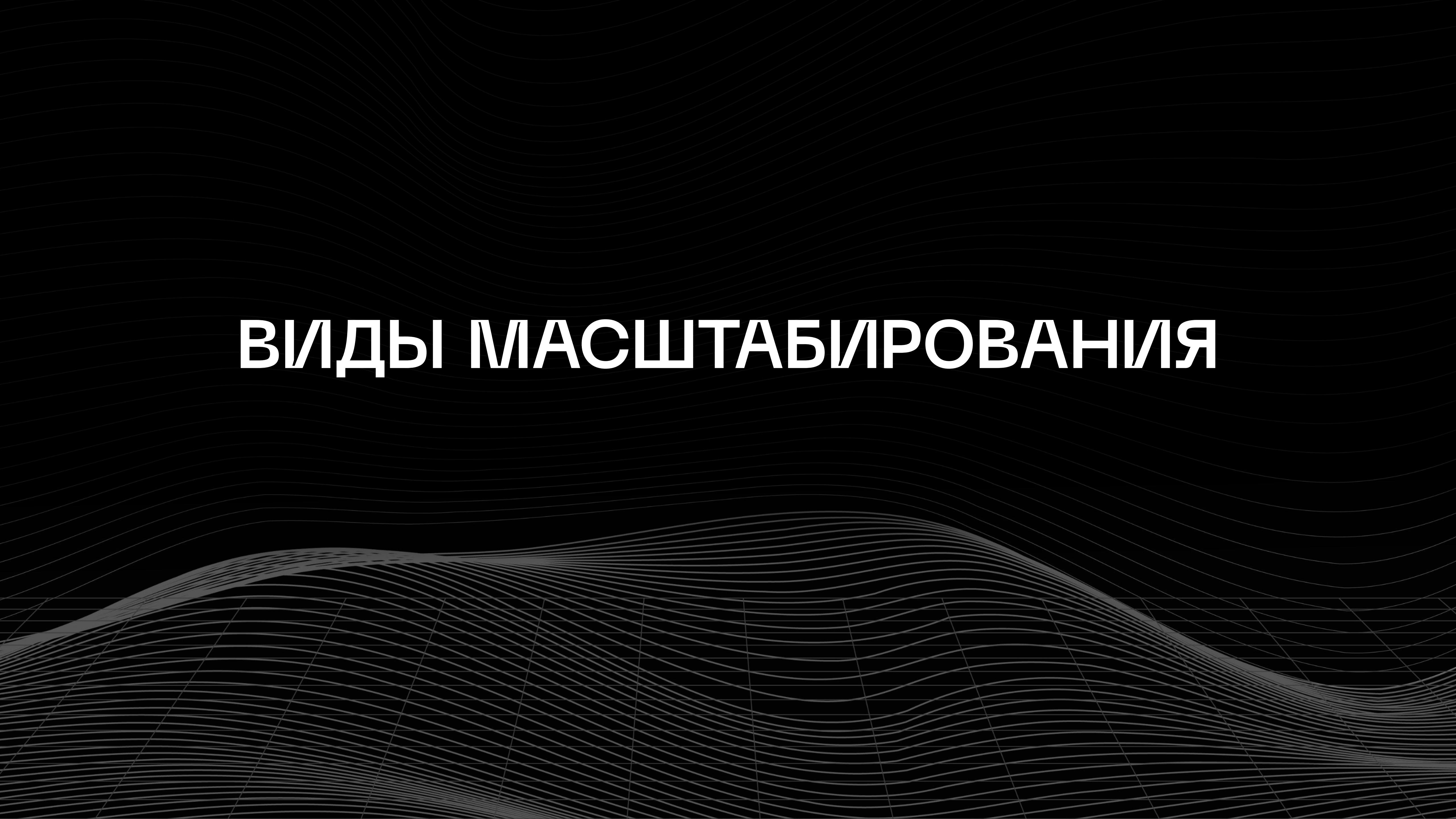
/1 Нужно делать много вычислительных операций

ПРИМЕРЫ

/1 Игровой сервер

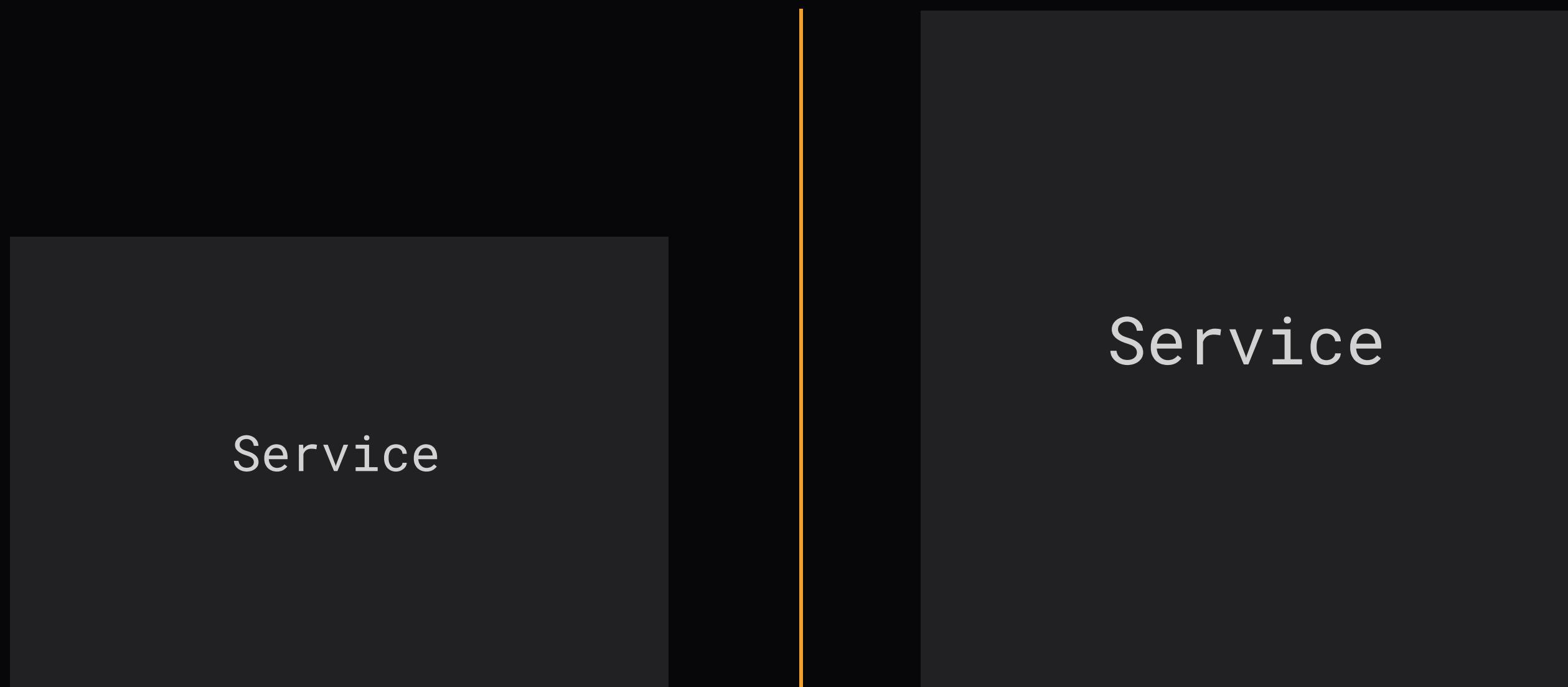
/2 Движок по подбору рекламы

ВИДЫ МАСШТАБИРОВАНИЯ



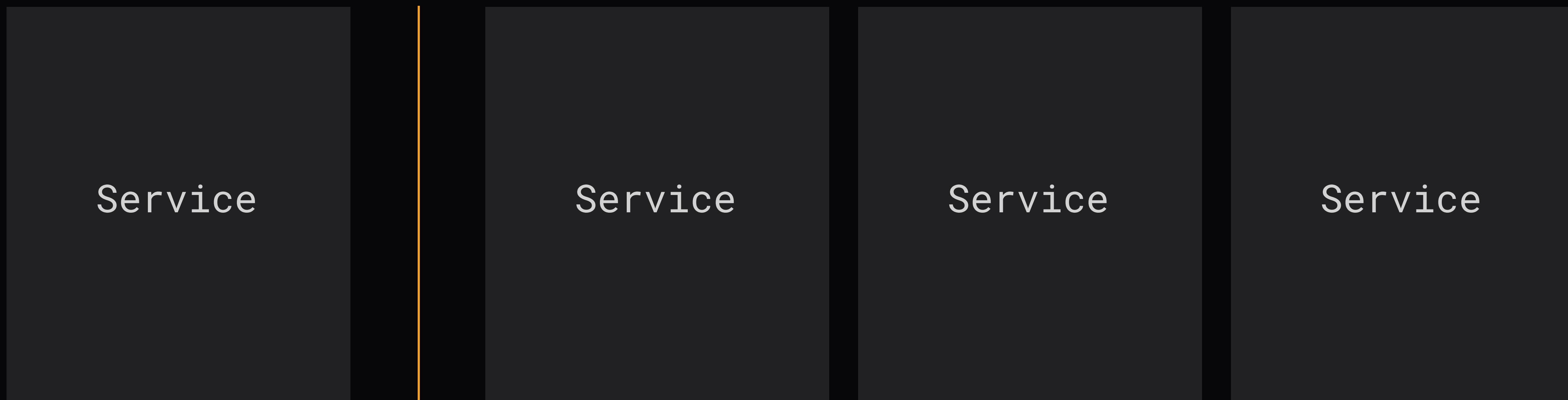
ВЕРТИКАЛЬНОЕ (SCALEUP)

Стоимость увеличения ресурсов растет нелинейно,
относительно их увеличения мощности (возможен downtime),
а также невозможно бесконечно масштабироваться



ГОРИЗОНТАЛЬНОЕ (SCALEOUT)

Чаще всего масштабирование происходит
без каких-либо проблем



Terminal: System Design × + ▾



STATELESS / STATEFULL

Когда ответ сервера зависит или не зависит
от какого-либо состояния внутри сервера

Иногда вертикальное масштабирование дешевле,
чем нанять людей и переписывать архитектуру

Terminal: System Design × + ▾



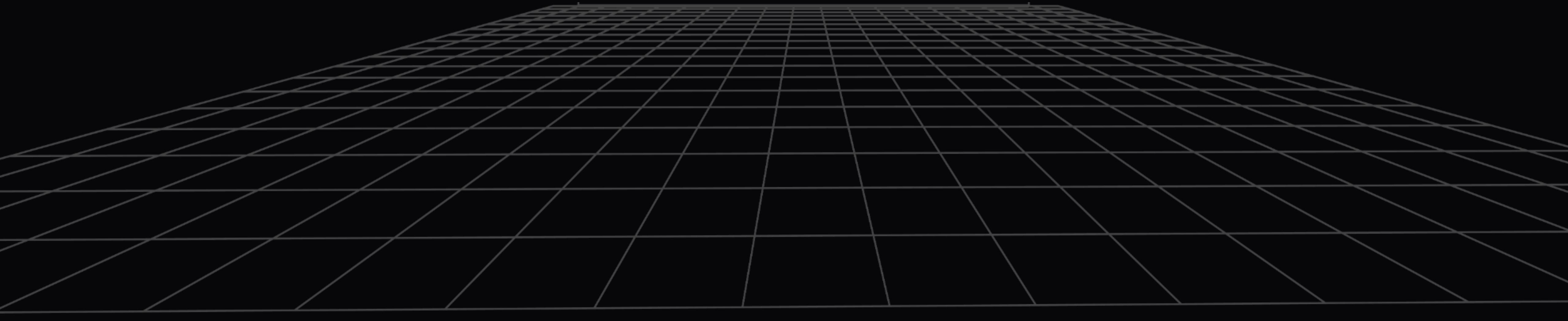
«ВО ВРЕМЕНИ»

Различные данные имеют различные требования
к обновлению – это позволяет отложить часть
обработки данных до более удобного случая

FAQ

Масштабируемость

вертикальное, горизонтальное, во времени



подтема №3

ОТКАЗОУСТОЙЧИВОСТЬ

ОТКАЗОУСТОЙЧИВОСТЬ

Способность системы продолжать функционировать корректно в присутствии отказов компонентов



ПРИМЕР

Приложение А отправило запрос приложению Б,
но не получило ответ. Что это может значить:

- Запрос потерялся и не дошел до Б
- Запрос дошел до Б, но Б не успел его обработать
- Запрос дошел до Б, но Б упал, не успев его обработать
- Запрос дошел до Б, но Б его просто проигнорировал
- Запрос дошел до Б и был обработан, но ответ пока не дошел до А
- Запрос дошел до Б и был обработан, но ответ потерялся и не дошел до А

**НЕЛЬЗЯ ОТЛИЧИТЬ ОТКАЗ СЕТИ
ОТ ОТКАЗА УЗЛА ИЛИ ПРИЛОЖЕНИЯ**

~70% ОТКАЗОВ

В ПРИЛОЖЕНИЯХ ПРОИСХОДЯТ ИЗ-ЗА ИЗМЕНЕНИЙ:

1. Развёртывания нового кода
2. Примененных миграций в базе данных
3. Изменения конфигурационных файлов
4. И т.д.

ОСТАЛЬНЫЕ 30% СБОЕВ ПРОИСХОДЯТ

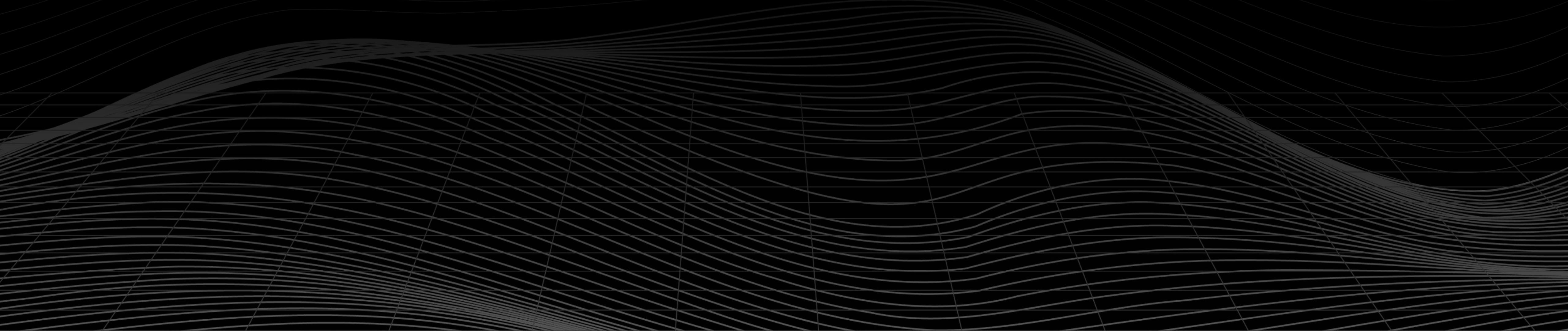
В ходе работы приложения без прямого вмешательства разработчиков и системных администраторов из-за:

1. проблем с сетью или дисками
2. возросшей нагрузки от пользователей
3. аварии в дата-центре

JEAF DEAN – ОТКАЗЫ ЗА ГОД

- ~0.5 overheating (power down most machines in <5 mins, ~1-2 days to recover)
- ~1 PDU failure (~500-1000 machines suddenly disappear, ~6 hours to come back)
- ~1 rack-move (plenty of warning, ~500-1000 machines powered down, ~6 hours)
- ~1 network rewiring (rolling ~5% of machines down over 2-day span)
- ~20 rack failures (40-80 machines instantly disappear, 1-6 hours to get back)
- ~5 racks go wonky (40-80 machines see 50% packetloss)
- ~8 network maintenances (4 might cause ~30-minute random connectivity losses)
- ~12 router reloads (takes out DNS and external vips for a couple minutes)
- ~3 router failures (have to immediately pull traffic for an hour)
- ~dozen sof minor 30-second blips for dns
- ~1000 individual machine failures
- ~thousandsof hard drive failures

КОГДА МАСШТАБИРУЕМ СИСТЕМУ,
ЧИСЛО ОТКАЗОВ РЕЗКО РАСТЕТ



ОБЕСПЕЧЕНИЕ ОТКАЗОУСТОЙЧИВОСТИ

-
- /1 избыточность на аппаратном уровне
 - /2 обнаружение и обработка отказа
на программном уровне
 - /3 прогнозирование и предотвращение отказа
-

Terminal: System Design × + ▾



ДОСТУПНОСТЬ

Система доступна, когда пользователи могут взаимодействовать с системой, получая корректные ответы за приемлемое время
(измеряется в кол-ве девяток)

NUMBER OF NINES

Screenshot

Availability Level		Average Yearly Downtime
Conventional Server	99%	87 hours, 40 minutes
Public Cloud Service	99.5%	43 hours, 50 minutes
	99.9%	8 hours, 46 minutes
High-Availability Cluster	99.95%	4 hours, 23 minutes
Virtual Fault Tolerance	99.995%	26 minutes, 18 seconds
Continuous Availability	99.999%	5 minutes, 16 seconds
The Stratus Zone	99.9999%	31.6 seconds

Terminal: System Design × + ▾



НАДЕЖНОСТЬ – ЭТО...

Характеристика системы, означающая
ее способность работать без сбоев или ошибок
в течение длительного времени



MTBF – ЕДИНИЦА ИЗМЕРЕНИЯ НАДЕЖНОСТИ

ⓘ MTBF (Mean Time Between Failure) – среднее время наработки на отказ. **Иначе говоря, среднее время безотказной работы**

ПРЕДПОЛОЖИМ...

Мы оцениваем 24-часовой период. За этот период мы потеряли 2 часа из-за двух отдельных инцидентов.

Наше общее время безотказной работы = 22 часа.

Разделим на два и получаем 11 часов. Итак, наша наработка на отказ составляет 11 часов



MTBF = 22 / 2 = 11 часов

ТАКИМ ОБРАЗОМ:

Доступность – возможность использования системы в любое время

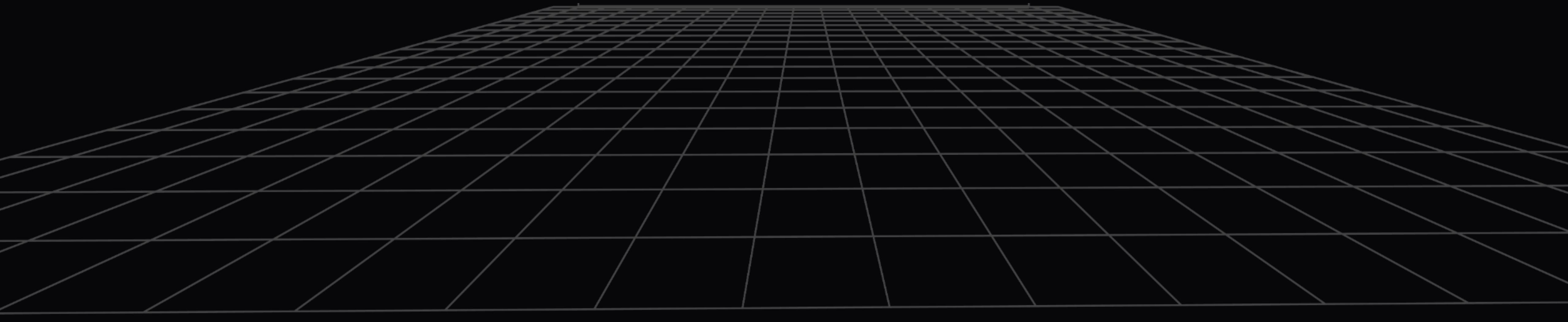
Отказоустойчивость – способность системы сохранять работоспособность при сбоях

Надежность – способность системы работать без ошибок и сбоев в течение длительного времени

FAQ

Отказоустойчивость

доступность, count of nines, надежность, MTBF



УДОБСТВО СОПРОВОЖДЕНИЯ

/1 Можно ли выполнять обновление системы без downtime?

/2 Можно ли отключить часть машин и продолжать работать?

/3 Есть ли удобный мониторинг системы
и подробное логирование?

/4 Насколько быстро можно диагностировать
и устранить проблему?

/5 Насколько легко можно проводить расширение системы?

БЕЗОПАСНОСТЬ

/1 Предотвращение возможных угроз (утечка, фальсификация, вандализм, ...)

/2 Защита от атак (DDoS, ...)



Удовлетворять все, получается крайне редко,
поэтому часто – это компромисс

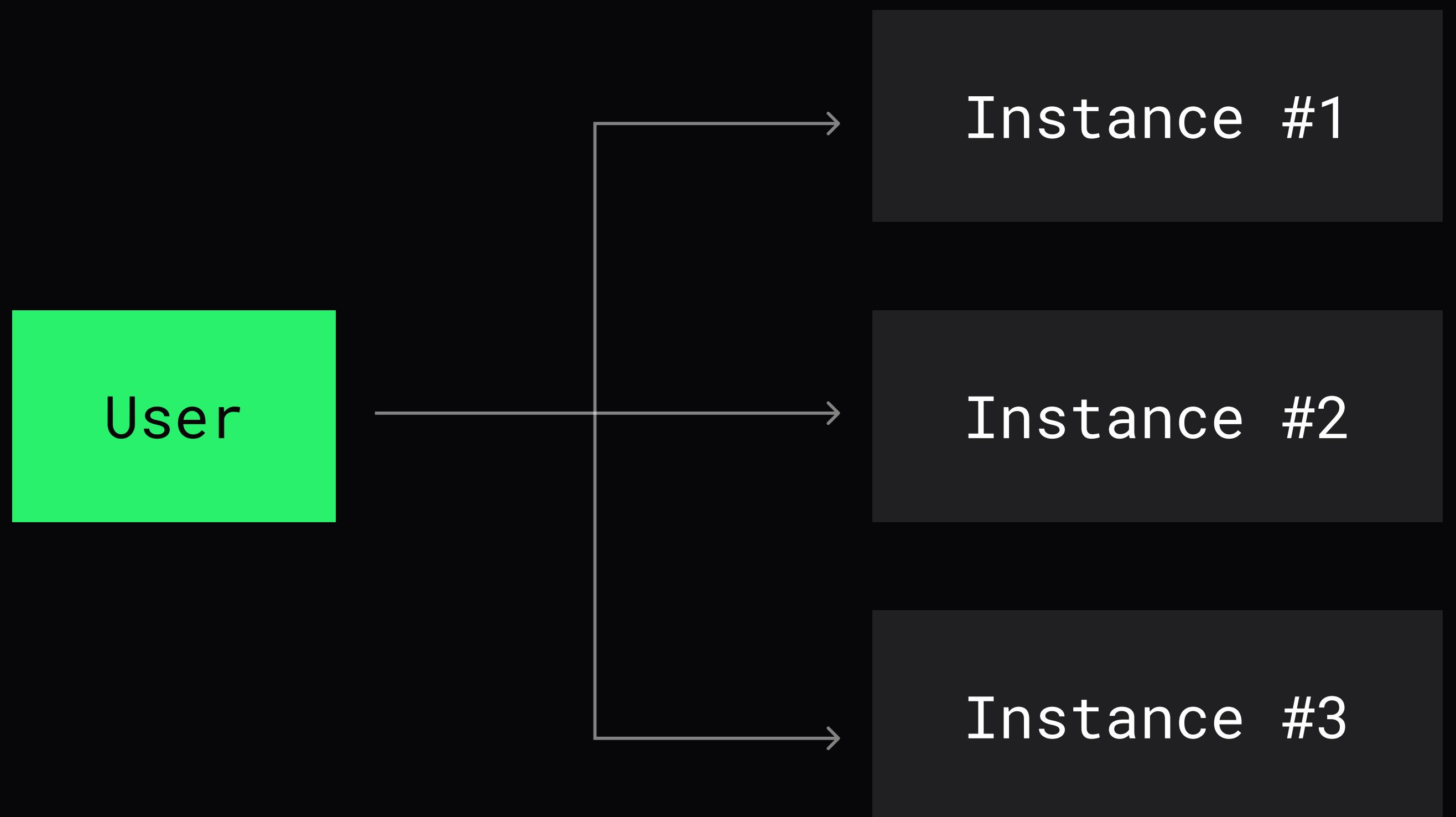
FAQ

Основные свойства

Производительность, отказоустойчивость,
масштабируемость, удобство сопровождения, безопасность

БАЛАНСИРОВКА НАГРУЗКИ

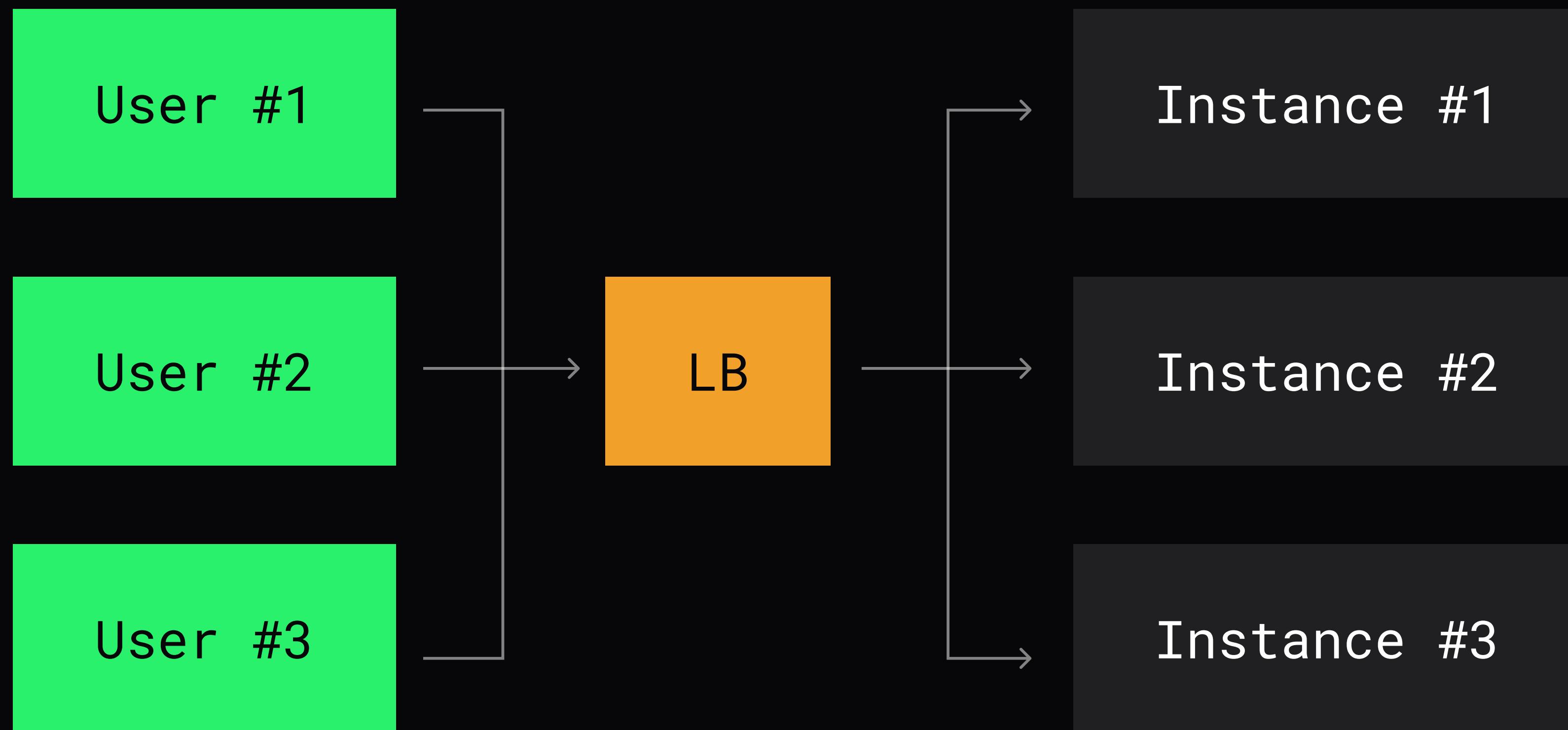
КЛИЕНТСКАЯ БАЛАНСИРОВКА



СЕРВЕРНАЯ БАЛАНСИРОВКА

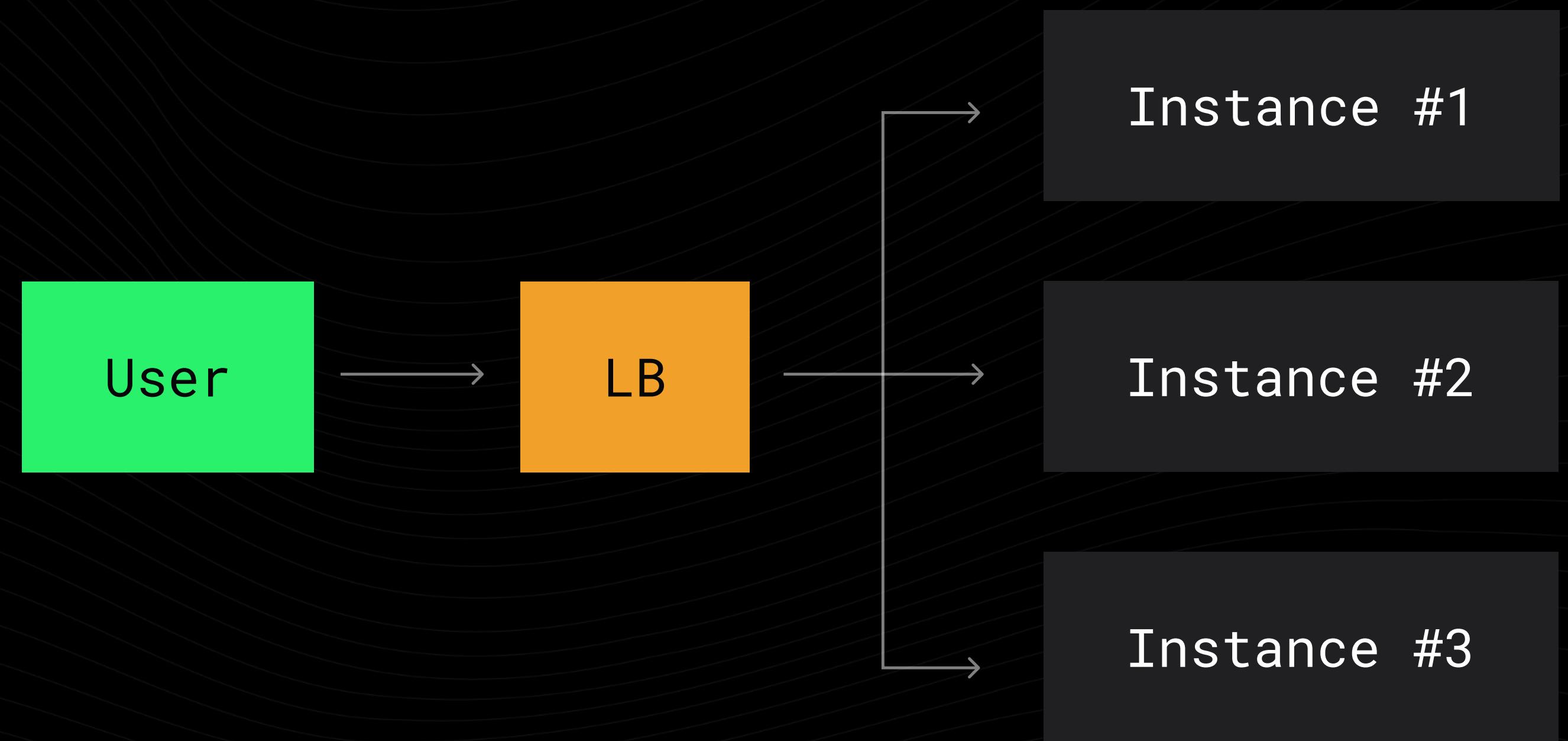


RANDOM



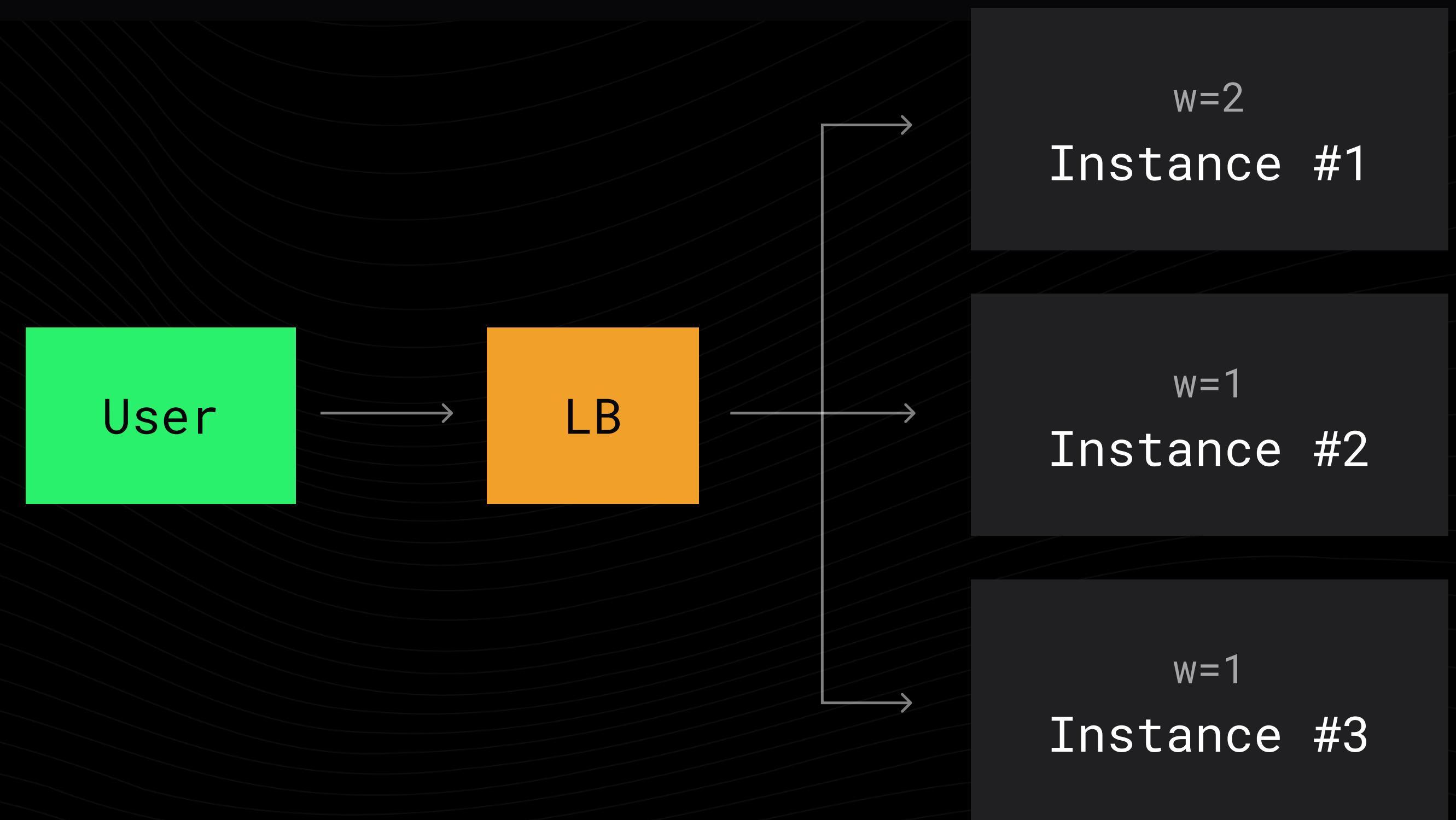
ROUND ROBIN

- 1: User -> Instance #1
- 2: User -> Instance #2
- 3: User -> Instance #3
- 4: User -> Instance #1
- 5: User -> Instance #3
- 6: User -> Instance #2



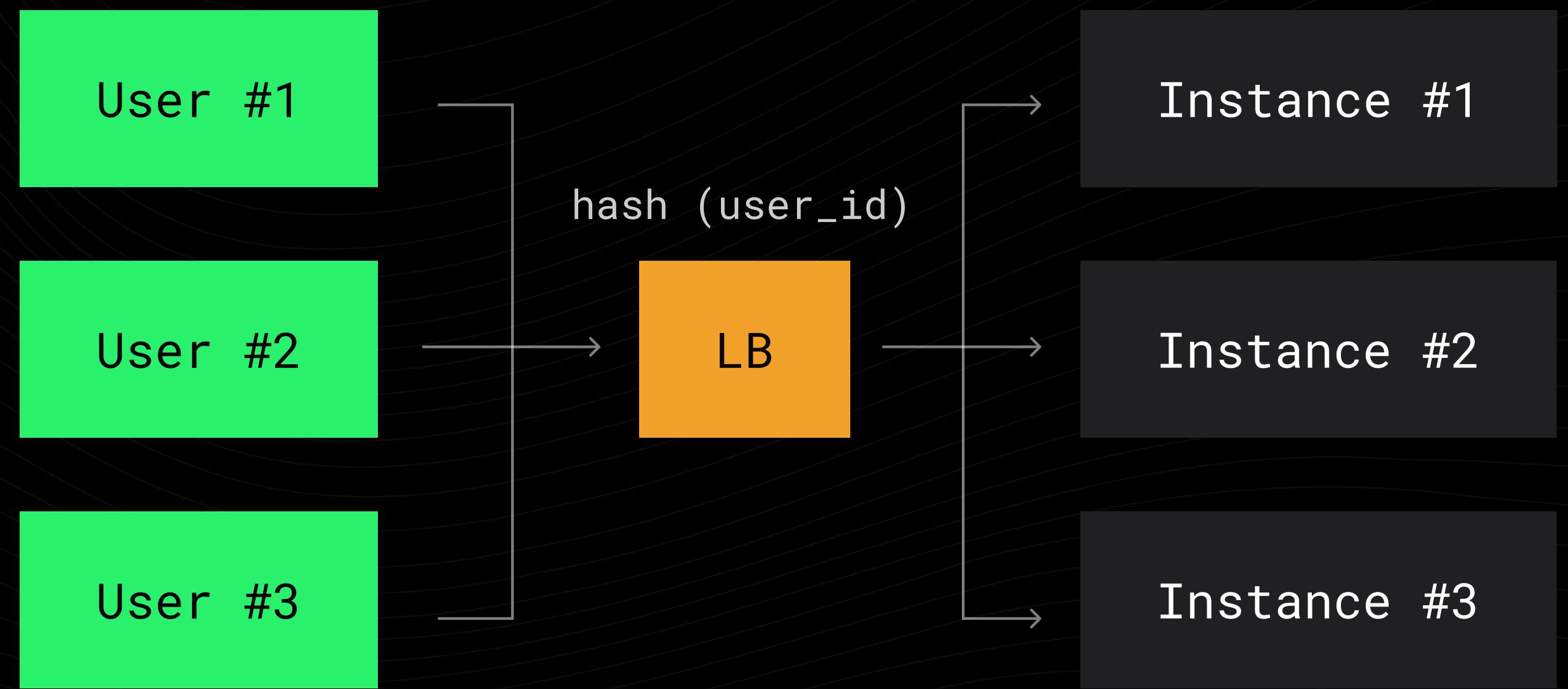
WEIGHTED RR

- 1: User -> Instance #1
- 2: User -> Instance #1
- 3: User -> Instance #2
- 4: User -> Instance #3
- 5: User -> Instance #1
- 6: User -> Instance #3

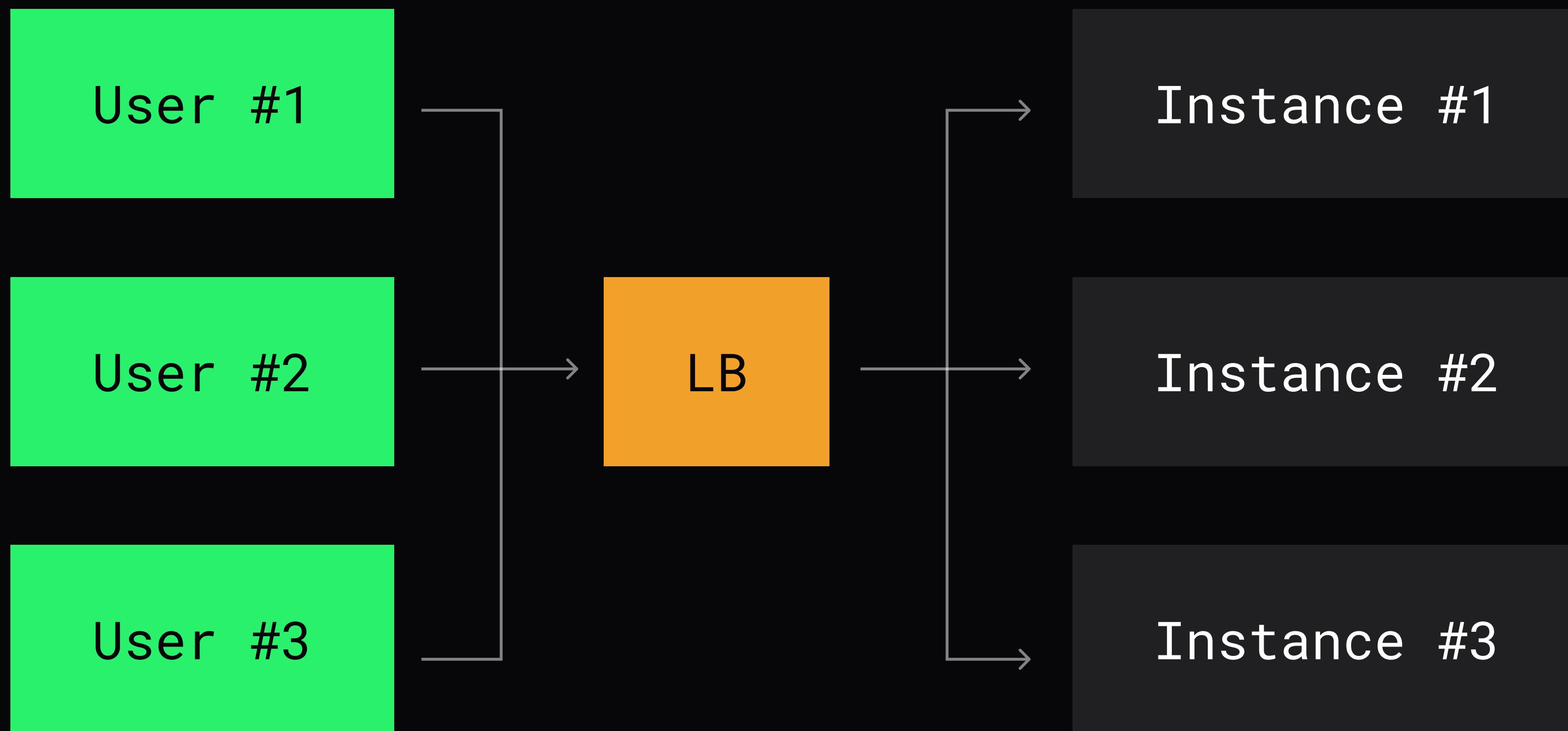


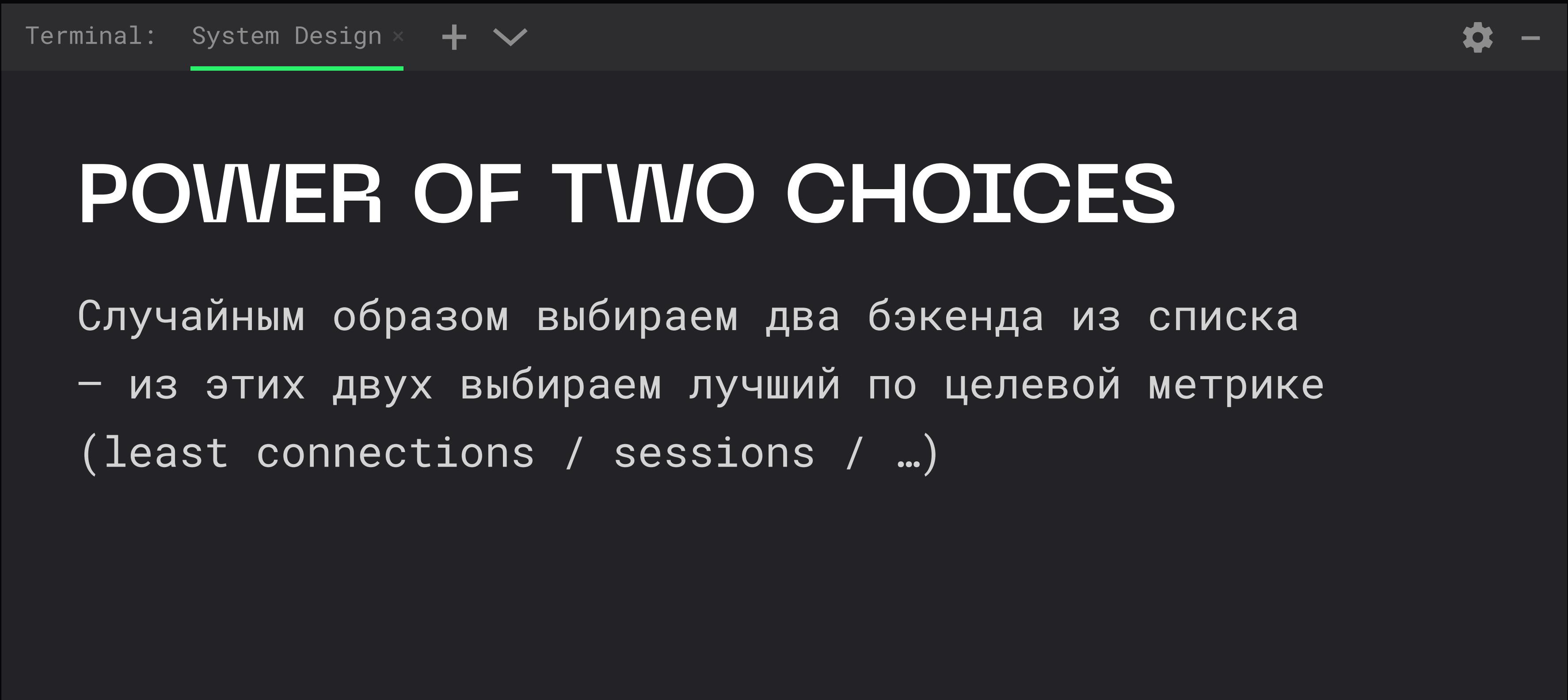
STICKY SESSIONS

- 1: User #1 -> Instance #2
- 2: User #2 -> Instance #1
- 3: User #3 -> Instance #3
- 4: User #2 -> Instance #1
- 5: User #3 -> Instance #3
- 6: User #1 -> Instance #2



LEAST CONNECTIONS / RT / THROUHPUT





КАКОЙ ИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ ВЫБРАТЬ

для сервисов по созданию
пользовательских анкет?

Ответ:

ROUND ROBIN

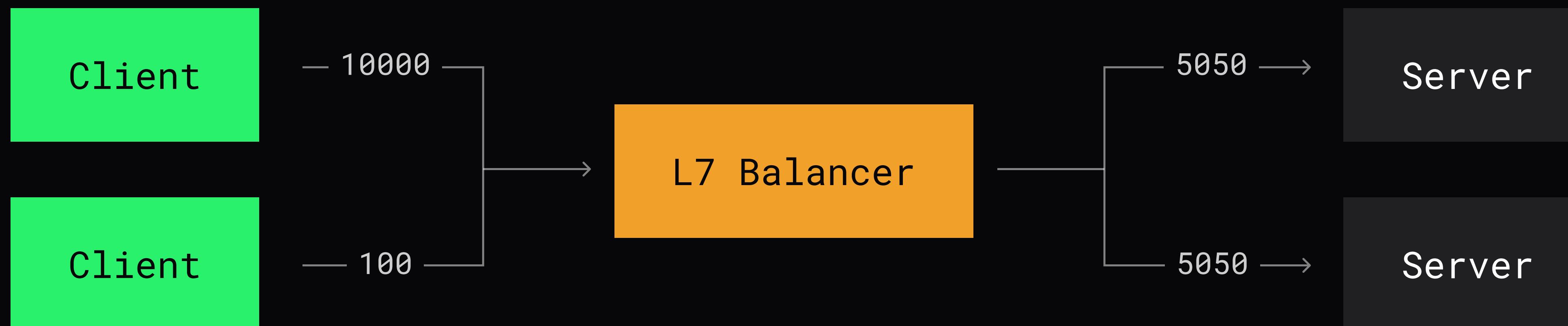
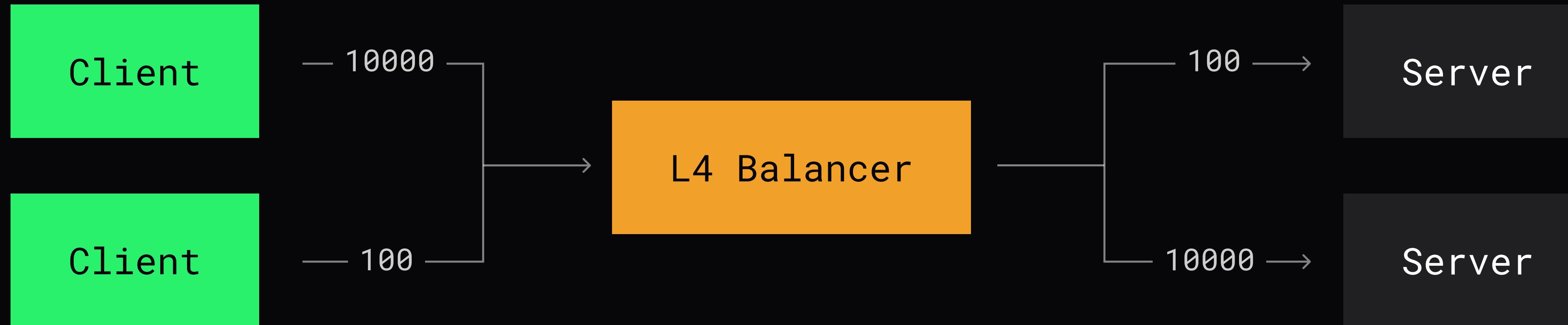
КАКОЙ ИЗ АЛГОРИТМОВ БАЛАНСИРОВКИ ВЫБРАТЬ

для сервисов, на которых хранятся
пользовательские сессии?

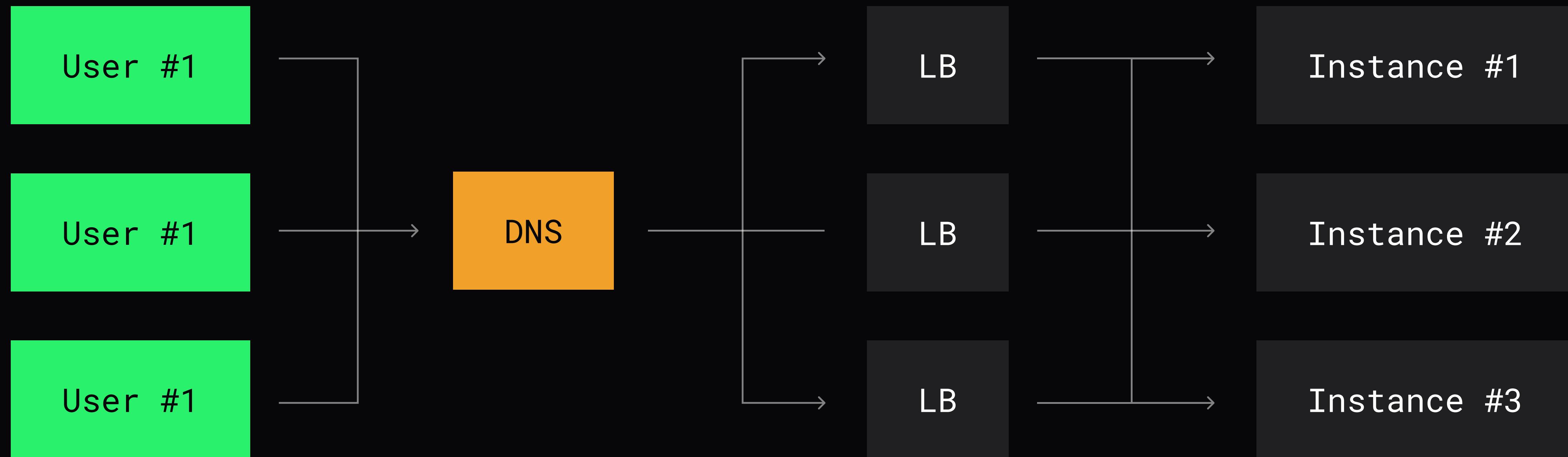
Ответ:

STICKY SESSIONS

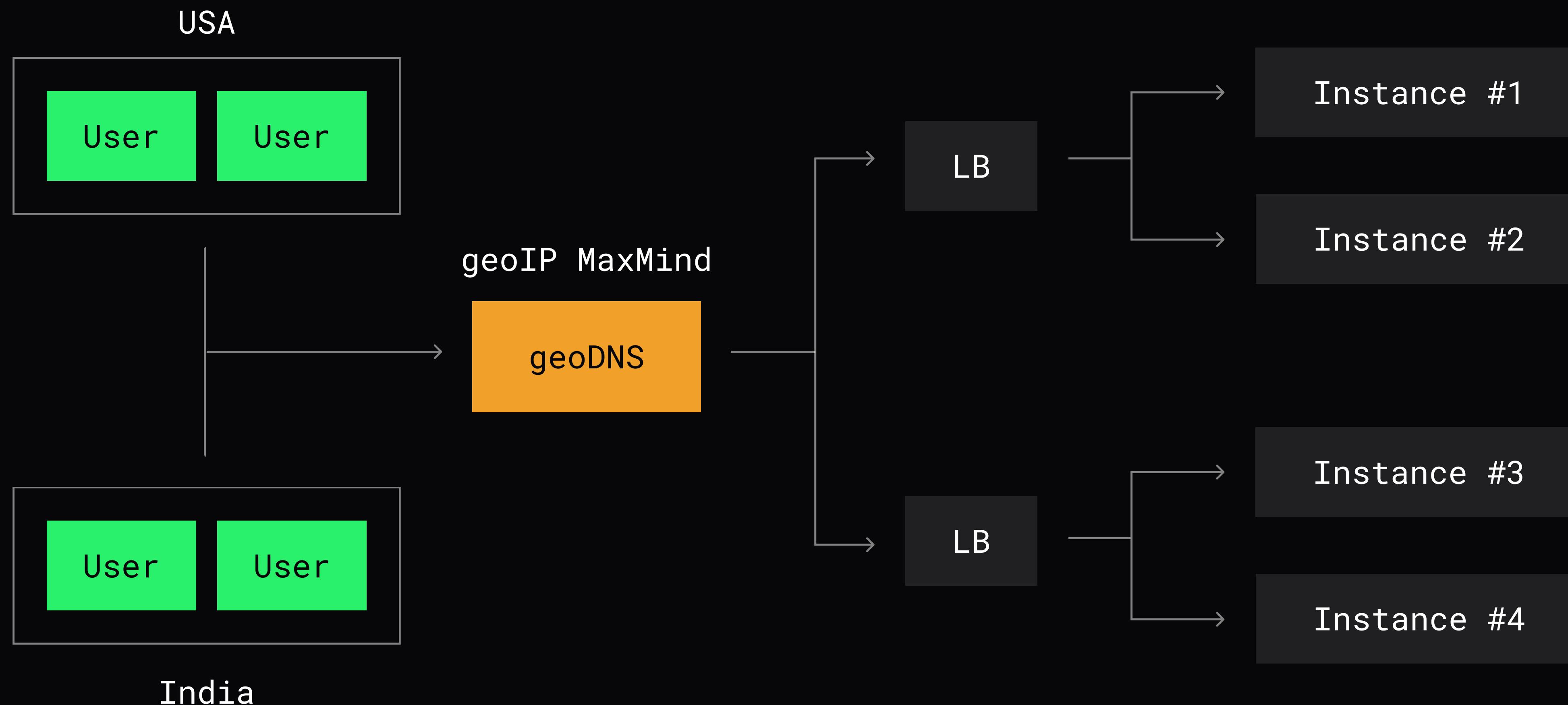
L4 / L7 БАЛАНСИРОВКА



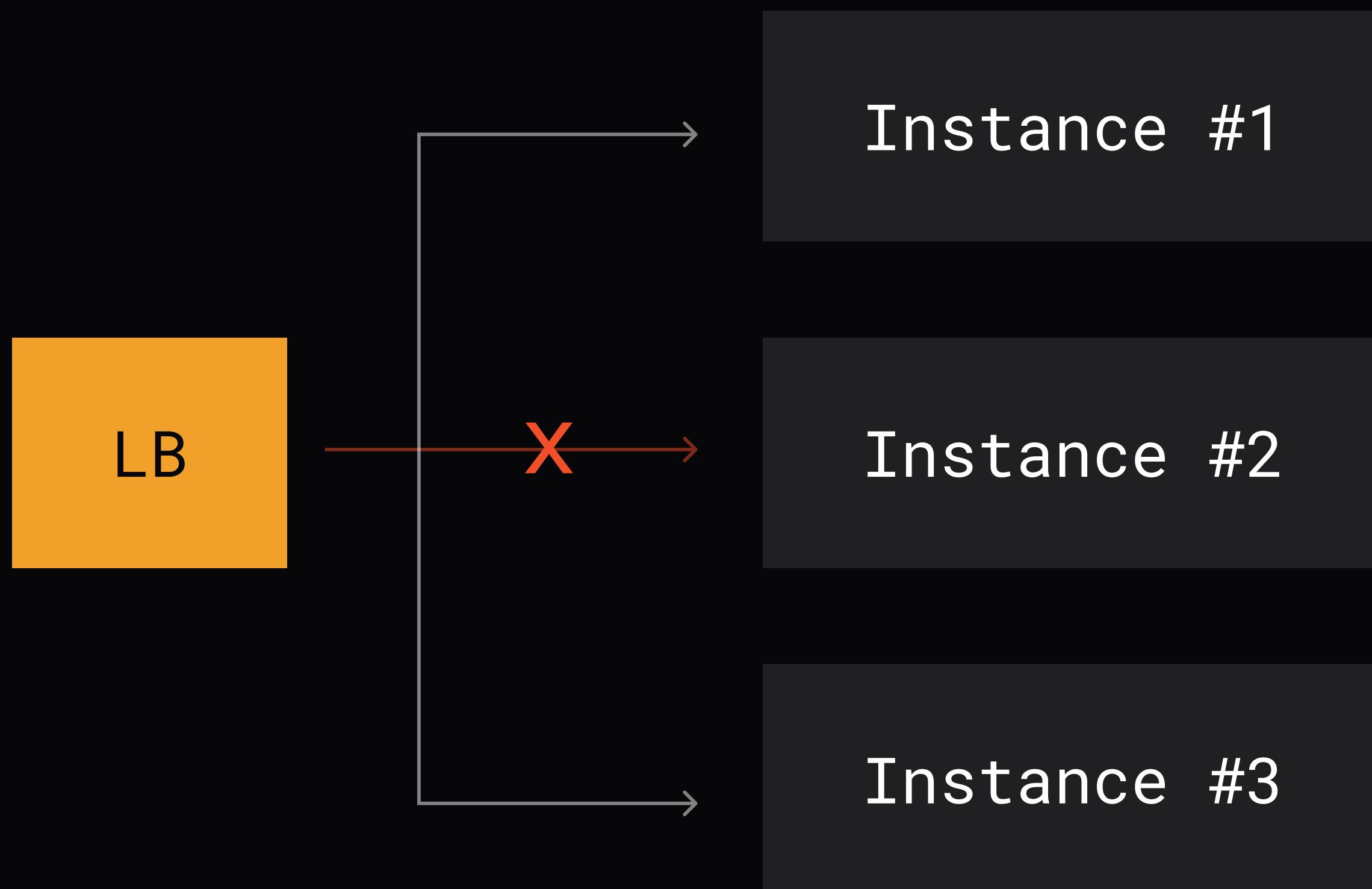
DNS БАЛАНСИРОВКА



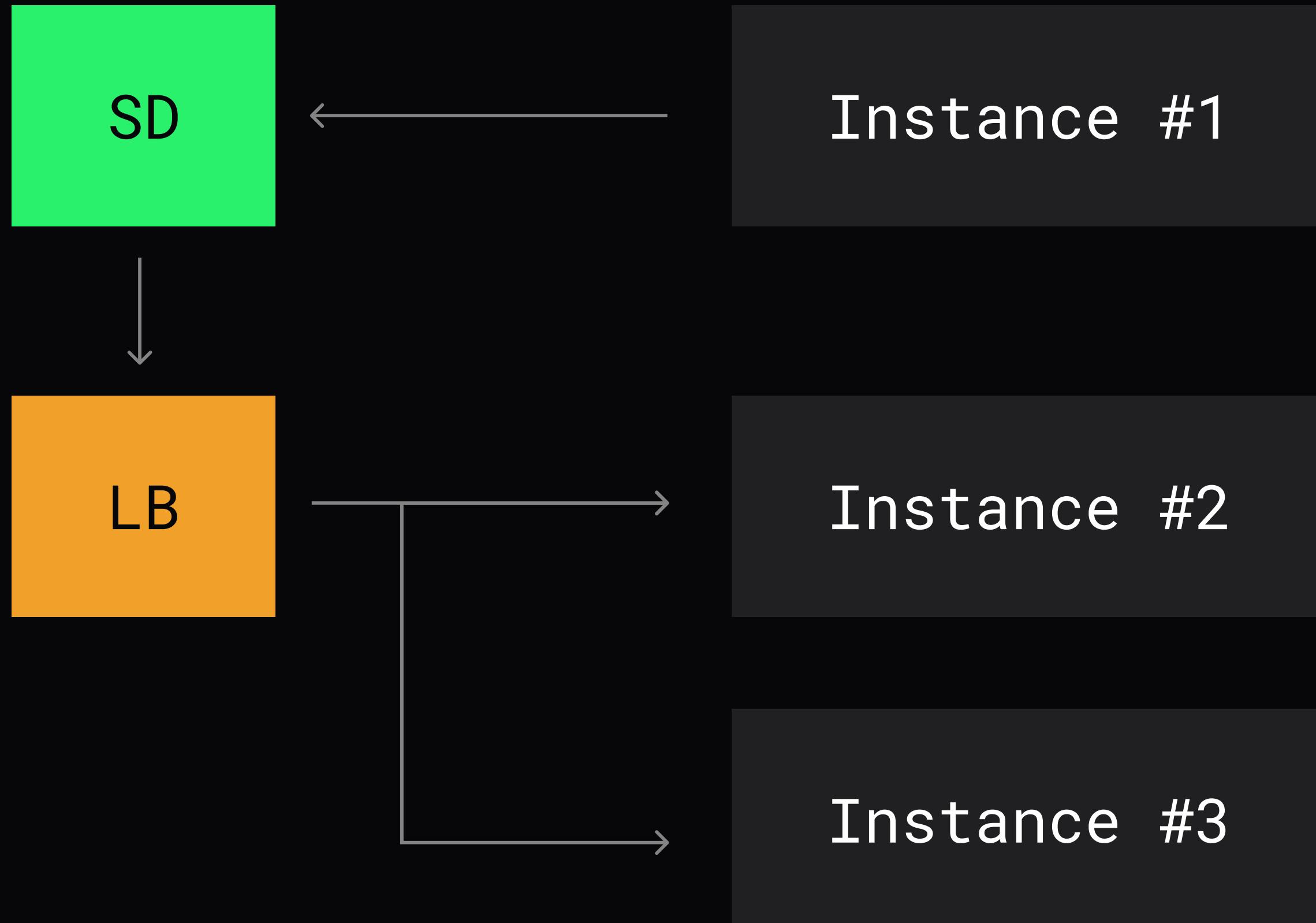
geoDNS БАЛАНСИРОВКА



HEALTH CHECKS (HEARTBIT)



SERVICE DISCOVERY



FAQ

Балансировка нагрузки

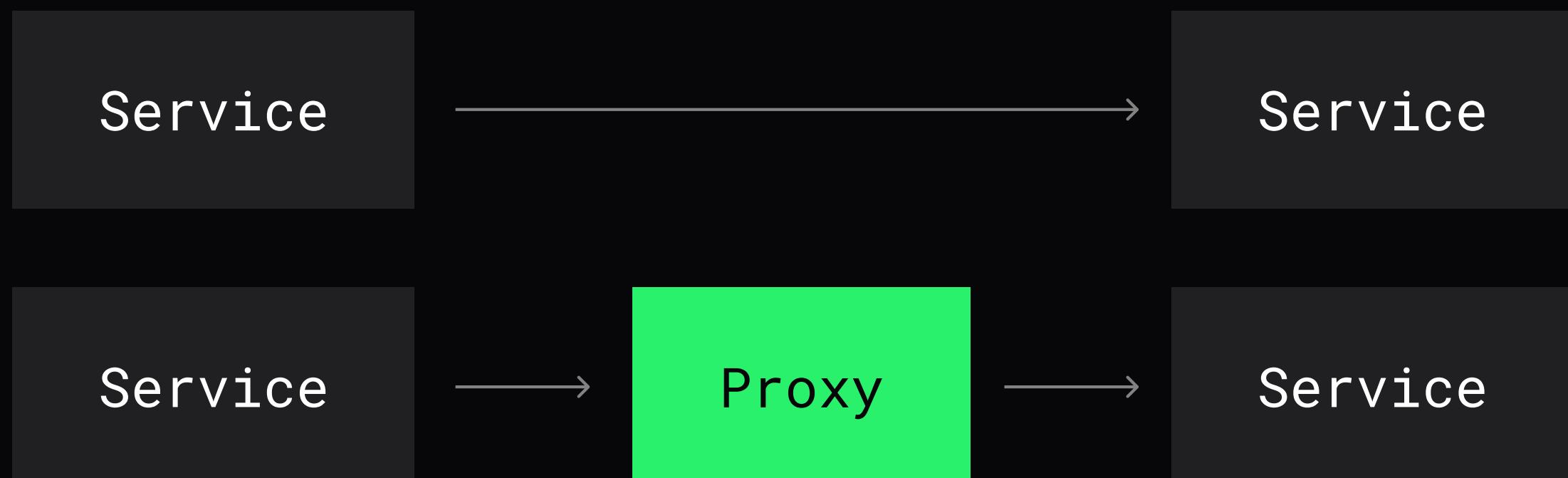
Round Robin, Weighted RR, Sticky Sessions,
Least Connections, L4/L7 балансировка

ПЕРЕРЫВ 5 МИНУТ

ПРОКСИРОВАНИЕ

ПРОКСИРОВАНИЕ

- Взлом / защита
- Кэширование данных
- Ограничения трафика
- Обход ограничений доступа
- Анонимность пользователей
- Сжатие и модификация данных



FORWARD PROXY



Screenshot × + ⚙ -

14 марта 2022, 09:56 / Медиа

Роскомнадзор заблокировал Instagram в России

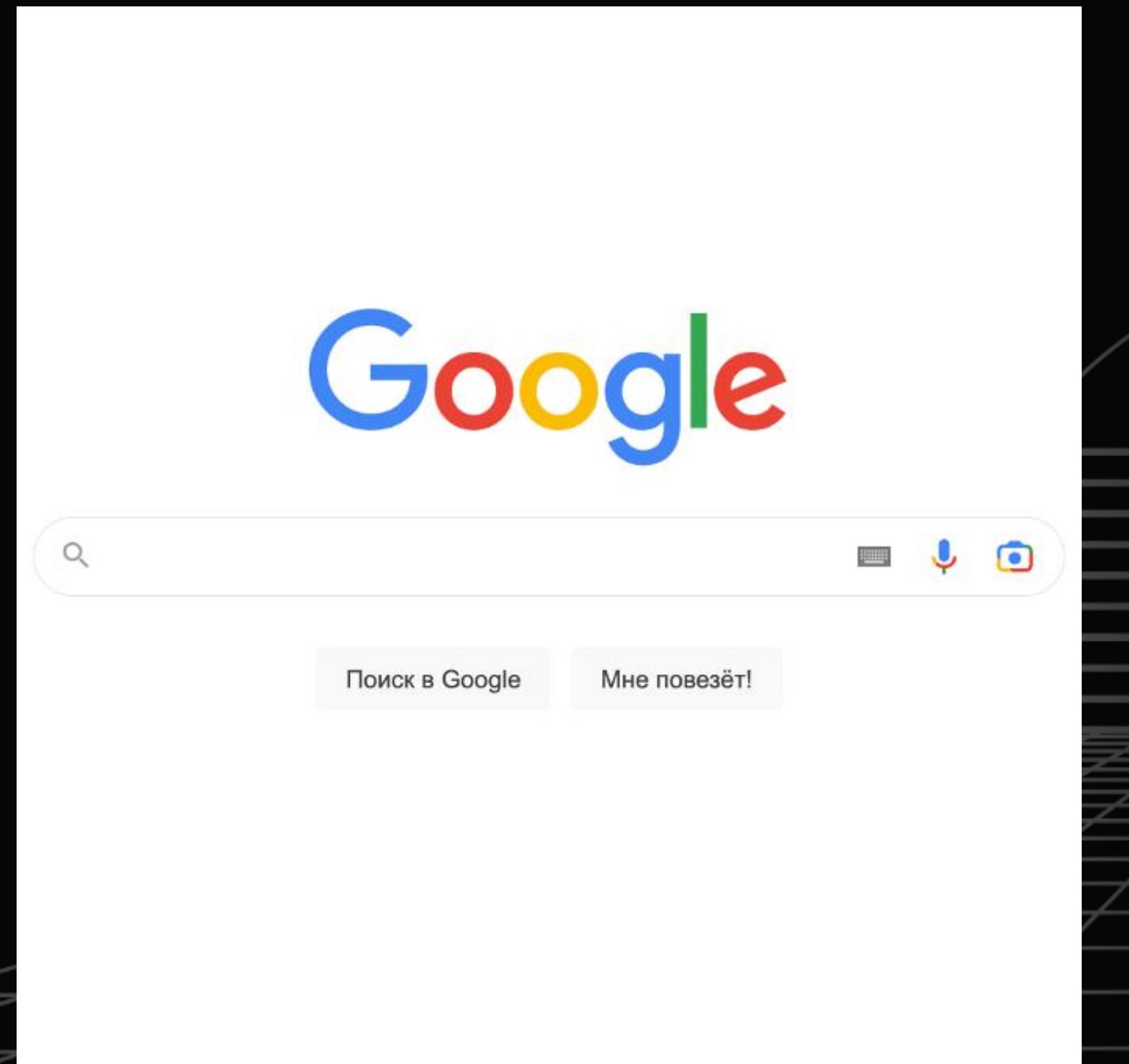
Ведомости

↗ Прочту позже

Роскомнадзор внес социальную сеть Instagram в реестр запрещенных сайтов. Согласно [сервису](#) ведомства по проверке блокировок страниц и сайтов, доступ к Instagram.com ограничен по требованию Генпрокуратуры от 11 марта. По [данным](#) сервиса по мониторингу блокировок Globalcheck, доступность соцсети в России составляет 0%.

REVERSE PROXY





```
1 vladimirbalun@Vladimirs-MacBook-Pro ~ % nslookup google.com
2 Server:      192.168.1.254
3 Address:     192.168.1.254#53
4
5 Non-authoritative answer:
6 Name:        google.com
7 Address:    173.194.222.100
8 Name:        google.com
9 Address:    173.194.222.101
10 Name:       google.com
11 Address:   173.194.222.113
12 Name:       google.com
13 Address:   173.194.222.139
14 Name:       google.com
15 Address:   173.194.222.138
16 Name:       google.com
17 Address:   173.194.222.102
```

FAQ

Проксируемое

Reverse, Forward

УСТАНОВКА ТРЕБОВАНИЙ К СИСТЕМЕ

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Заявление о том, как должна вести себя система.
Оно определяет, что система должна делать, чтобы
удовлетворить потребности или ожидания пользователя



Функциональные требования можно рассматривать как функции,
которые обнаруживает пользователь

функциональные требования

ПРИМЕРЫ

/1 оповещения о пропущенных звонках

/2 отправка сообщений с медиа файлами

/3 поиск товаров по названию

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Требования, определяющие свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, не относящиеся к поведению системы

нефункциональные требования

ПРИМЕРЫ

/1 Производительность

/2 Отказоустойчивость

/3 Безопасность

ПРАВИЛО «1–10–100»

Чем раньше вы обнаружите ошибку
– тем дешевле сможете исправить

1\$

10\$

100\$



LET'S PRACTISE

FAQ

Установка требований

Функциональные требования,
нефункциональные требования

ОЦЕНКА НАГРУЗКИ



LET'S PRACTISE

ДОМАШНЕЕ ЗАДАНИЕ

ДО ВСТРЕЧИ
НА СЛЕДУЮЩЕМ
ЗАНЯТИИ!