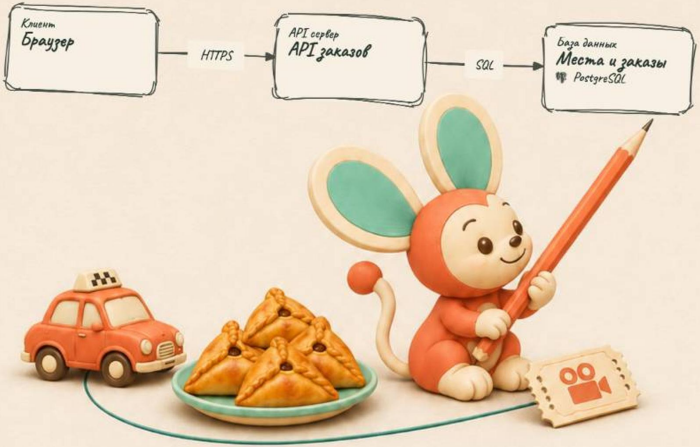
archinario. | Системный дизайн

Миллион запросов спустя

Как пережить субботу, если ты сервер



Как спроектировать сервис
и выяснить, что забыли нарисовать?

Ильдар Гильмутдинов

Ильдар Гильмутдинов

Миллион запросов спустя

https://litres.ru/74968617
SelfPub; 2026

Аннотация

Глеб видит своё такси. Водитель видит Глеба. Они машут друг другу с разных берегов реки. До подачи восемь минут. Как бы вы научили приложение выбирать машину, которая приедет первой?

«Миллион запросов спустя» — книга о системном дизайне для разработчиков и аналитиков, которые впервые проектируют сервис целиком. За субботним обедом друзья делят счёт за эчпочмаки, а вы разбираетесь, как банк отличает повторную попытку от второго перевода на ту же сумму. Перед кинопремьерой посчитаете, сколько запросов в секунду придёт от зрителей, если плеер просит следующий фрагмент фильма раз в четыре секунды.

А потом вы сами спроектируете кинотеатр для друзей. На одном экране нажали паузу — что должно произойти на остальных? Вы нарисуете путь этой команды, выберете, где хранить состояние просмотра, и сверите свою схему с разбором в книге.

Содержание

Часть I. Утром в чате 6
Глава 1. «Я уже выхожу» 8
Четыре галочки на бумаге 8
Запись, номер и ответ 11
Доставка и догоняющее чтение 13
Отказ в середине записи 19
Сколько доставок приходится на одно сообщение 22
Практика. Телефон увидел сообщение 145 раньше 142 23
Телефон лежит в кармане 25
Южные ворота 25
Глава 2. У входа. У какого? 27
Что обещает короткая ссылка 29
Создать короткий код 31
Переход: от кода до адреса 33
Закрытие и горячий код 36
Нагрузка и статистика 38
Сбой, который покажет границу 41
Одна ссылка получает 3 000 переходов в секунду 44
Практика. Вернулась закрытая ссылка 45
Задание для оборота билета 47

Один ID для системы, другой для гостей 47
Глава 3. Молоко уже взяли? 49
Что переживёт закрытие вкладки 51
Как обе копии собирают одну фразу 55
Вкладка вернулась через неделю 57
Право редактирования истекло в дороге 59
Когда тормозит один документ 61
Практика. Закрыли вкладку после обрыва 63
Две корзины 65
Глава 4. Машина в двух минутах. На другом берегу 66
Быстрый поиск и точный маршрут 68
Как закрепить одного водителя за поездкой 72
Горячие районы и приватность 76
Сколько работы в секунду 78
Практика. Кто заберёт пассажира у моста 81
Куда смотреть пассажиру 83
Часть II. До обеда 85
Глава 5. Кофе с собой 86
От меню к индексу 86
Что индексировать 88
Запись и догоняющий индекс 92
Как перейти на новый индекс между страницами 95
Нагрузка и проверка 96
Практика. Какую цену поставить в выдаче 100

Глава 6. Почему вчерашний кофе стоит дороже 104
Как хранить чек, ценник и месячную выручку 105
Один эксперимент до покупки сервера 106
Конец ознакомительного фрагмента. 108

Ильдар Гильмутдинов
Миллион запросов спустя

Часть I. Утром в чате



Как из разговора в чате получаются история сообщений,

адрес на карте и одна назначенная машина.

Глава 1. «Я уже выхожу»

09:40 · Сообщения и подтверждения

В субботу у нас одно правило: кто первый проснулся, тот ничего не пишет. Правило продержалось до 9:40. Глеб прислал голосовое «я уже выхожу». За его спиной отчётливо щёлкнул чайник. «Иди спокойно, — ответила Оля. — Только из розетки его выдерни перед выходом».

Оля ставит встречу у парка. У парка четыре входа. Пока она набирает «южные ворота», Лиза успевает прочесть первое сообщение и положить телефон в карман. Что Оля вправе понять по галочке рядом со своим сообщением? Её телефон знает об отправке. Телефоны друзей обращаются к одной программе, которая принимает сообщения и хранит общую переписку. Это сервер чата. Он может подтвердить запись. А Лиза уже убрала телефон в карман и могла пропустить уточнение про ворота.

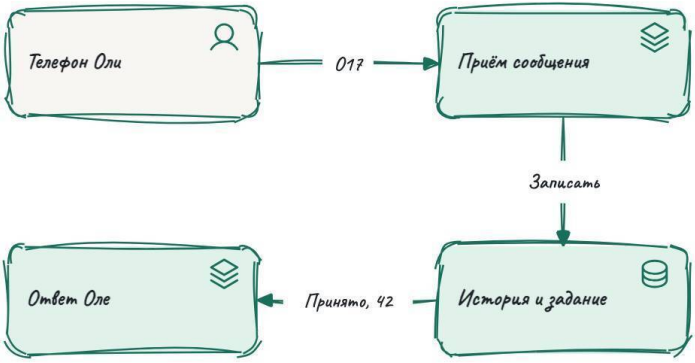
Четыре галочки на бумаге

Нарисуйте три прямоугольника: телефон Оли, сервер, телефон Лизы. Между ними оставьте место для стрелок в обе стороны. Приложение Оли даёт сообщению временный номер О17. Такой опознавательный номер называют идентификатором, сокращённо ID. Сервер сохраняет его под номе-

ром 42 внутри беседы. После подтверждения телефон заменяет временную запись окончательной. Номер 42 не говорит, читала ли что-нибудь Лиза. Это может сообщить только её приложение, да и оно знает лишь то, что текст оказался на открытом экране.

Глеб прикладывает голосовое. Сами байты можно положить в хранилище файлов, а в историю записать ссылку и длительность. Если сначала всем показать сообщение, а отправку звука оставить на потом, друзья долго будут слушать восемь секунд тишины. Согласуем обещание: значок воспроизведения появляется у получателя, когда сервис уже может выдать доступный ему файл. На своём телефоне Глеб увидит запись раньше.




Телефон Оли
017
Приём сообщения
Записать
Ответ Оле
Принято, 42
История и задание

Сервер принимает О17 от Оли. История беседы и задание на рассылку сохраняются одной транзакцией.

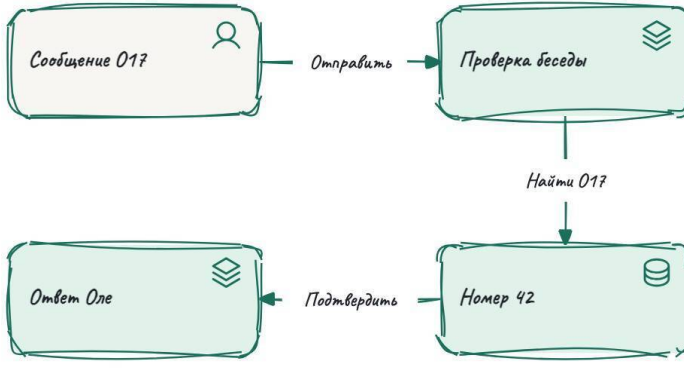
Запись, номер и ответ

Телефон Оли просит сервер: «Сохрани: встречаемся у южных ворот». В запросе он передаёт номер беседы, свой номер сообщения и текст. Сервер отвечает: «Записал под номером 42». Так выглядит обмен по **HTTP** — правилам, по которым одна программа посылает запрос, а другая отвечает. Программу, сделавшую запрос, называют клиентом. В нашем случае это приложение на телефоне.

А как о воротах узнает Лиза? Её телефон может оставить открытый канал с сервером. Такой способ связи называется **WebSocket** («веб-сокет»). По каналу обе стороны могут заговорить первыми. Лиза отправляет «у каких ворот?», когда захочет. Сервер сам присылает уточнение Оли, даже если Лиза ничего не спрашивала. Иначе телефону пришлось бы то и дело повторять «есть новости?». В этом чате по части «ну вы где?» вполне хватает Глеба.

Теперь Лиза заходит в метро и теряет связь. По оборванному каналу уточнение уже не придёт. Когда телефон снова подключится, он спросит, что появилось после последнего сохранённого номера. Поэтому чату нужны две вещи: связь для тех, кто сейчас на линии, и история для тех, кто пропу-

стил поворот к южным воротам.



Телефон повторяет команду О17. Сервис находит уже присвоенный в этой беседе номер 42 и возвращает ту же квитанцию.

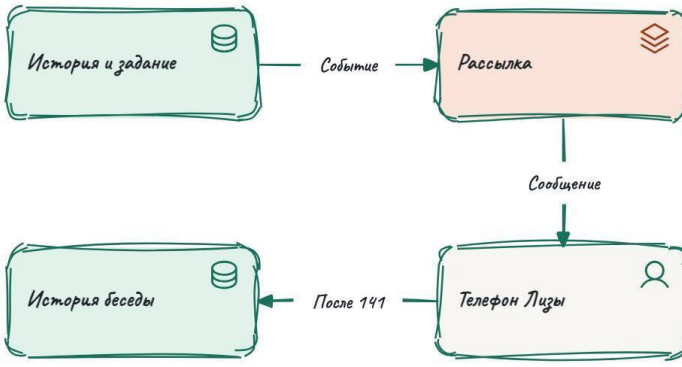
Сервис проверяет доступ. Затем присваивает номер внутри беседы и добавляет сообщение. Учебный вариант держит счётчик и историю в одном разделе базы. Новое сообщение и пометку «надо разослать» сохраняем одной транзакцией: после сбоя в базе останутся обе записи или ни одной. Когда база подтвердила такой результат, говорят, что транзакция зафиксирована. Ответ содержит прежний или новый номер. Подмена номера беседы при повторе не должна открыть чужую историю. Включаем беседу и владельца ключа в провер-

ку.

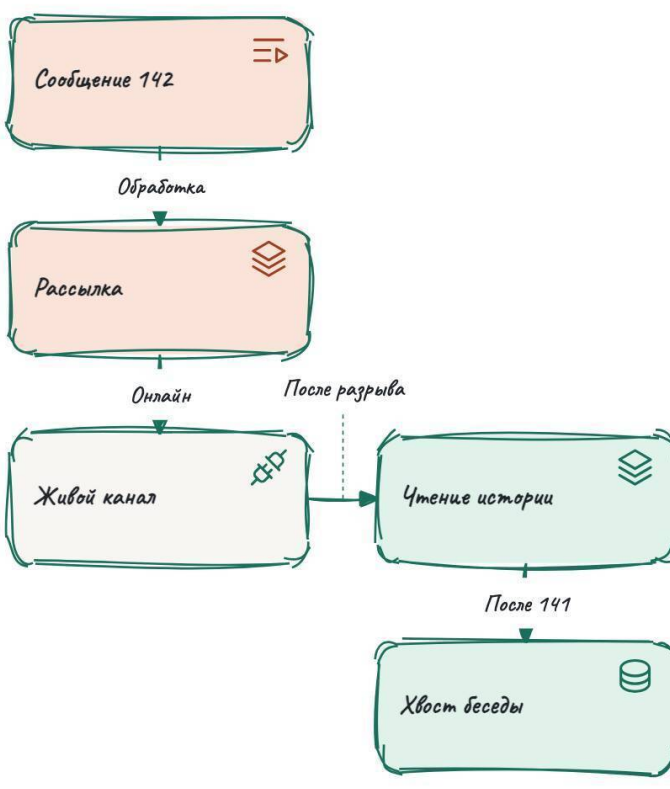
Глобальная нумерация была бы лишней точкой согласования. Последовательность внутри одной беседы проще обосновать: она нужна людям, которые смотрят один разговор. Если за следующим номером все выстраиваются в одну очередь, очень оживлённый чат начнёт задерживать отправку. Тогда отдельно обсуждаем, насколько строга последовательность, какой объём сообщений надо поддержать и нельзя ли выделить такой тип канала в другую модель. Обычные небольшие беседы распределяются по ключу беседы.

Доставка и догоняющее чтение

После записи за дело берутся воркеры. Это фоновые программы: они подбирают задания на рассылку и ищут, у кого сейчас открыт чат. По живому каналу устройство может получить повтор или увидеть более позднее сообщение раньше предыдущего. Повтор распознают по идентификатору. Если пришёл номер после пропуска, телефон временно откладывает его в памяти — в буфере — и запрашивает у сервера пропущенную часть истории.



Узел рассылки находит подключение Лизы. После перерыва телефон запрашивает пропущенную историю по своей закладке.



Отключённые клиенты дочитывают пропущенное.

Представьте, что Лиза уже сохранила сообщение 145 и вернулась в чат. Пока телефон спрашивал «что я пропу-

стил?», Оля отправила 146 и убрала телефон. Если включить живую доставку после чтения истории, 146 может проскочить между ними. И больше никто за весь вечер не напишет.

Поэтому сначала сервер подтверждает, что подписка Лизы уже действует. Телефон пока складывает всё пришедшее в буфер. Затем сервер отмечает, до какого номера доведена история, и отдаёт её до этой границы. Начинает с курсора телефона — закладки «досюда я всё сохранила». Если 146 записали до границы, оно будет в истории. Если позже — придёт по подписке. Даже когда сообщение попало в оба места, Лиза увидит его один раз.

С каждой страницей сервер возвращает позицию, до которой проверил историю, включая удалённые и скрытые сообщения. Телефон сохраняет страницу вместе с этой позицией, убирает из буфера повторы и разбирает оставшиеся сообщения по порядку. Если связь оборвалась посередине или буфер заполнился, телефон заново подписывается и дочитывает от последней сохранённой закладки. Посмотрим, как это выглядело несколькими сообщениями раньше.

Телефон Лизы сохранил беседу по номер 141. Сообщение 145 уже пришло по живому соединению. Из канала истории ещё предстоит узнать, что было между ними.



Позиция в беседе	142	143	144	145	Курсор
Живой канал	○	○	○	◇	141
Первая страница	●	—	●	◇	144
Вторая страница	●	—	●	●	145

● Сохранено ◇ В буфере — Скрыто ○ Неизвестно

Сообщение 145 пришло раньше истории

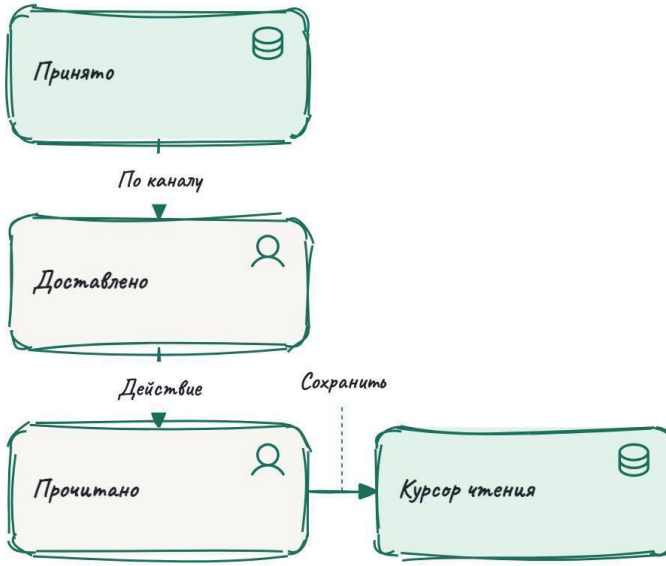
После первой страницы телефон одной транзакцией сохраняет сообщения 142, 144 и курсор 144. Сервер разрешил пройти скрытую позицию 143. На второй странице приходит знакомое сообщение 145. На экране оно появляется один раз.

Доказывает ли сохранённый курсор 145, что Лиза прочитала сообщение?

Он подтверждает, что устройство догнало историю. Отметку прочтения приложение передаёт отдельно, когда выполнено принятое в продукте условие.

Оля видит «отправлено», хотя телефон Лизы пока без сети. Это корректно, если «отправлено» означает подтверждение сервера. «Доставлено» появляется после определённого

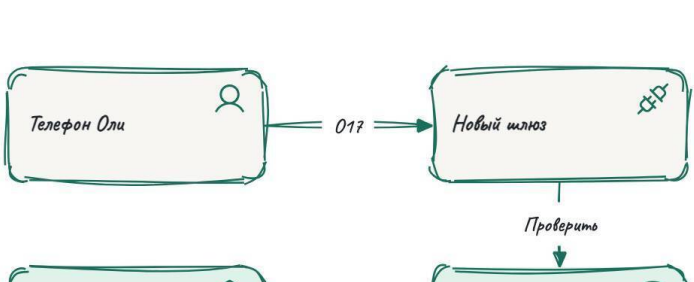
подтверждения устройства. «Прочитано» формирует сам получатель. Для нескольких устройств уточняем, означает ли «доставлено» любое устройство или каждое. В проекте статус относится хотя бы к одному активному устройству, а история остаётся доступной остальным.



Запись, доставка и чтение относятся к разным событиям. Для отметки прочтения можно хранить монотонный кур-

сор: последнее прочитанное значение у пользователя не уменьшается при позднем сообщении от старого устройства. Если пользователь вручную пометил беседу непрочитанной, лучше считать это другой настройкой, а не пытаться переписать историю прочтения для других участников.

Отказ в середине записи

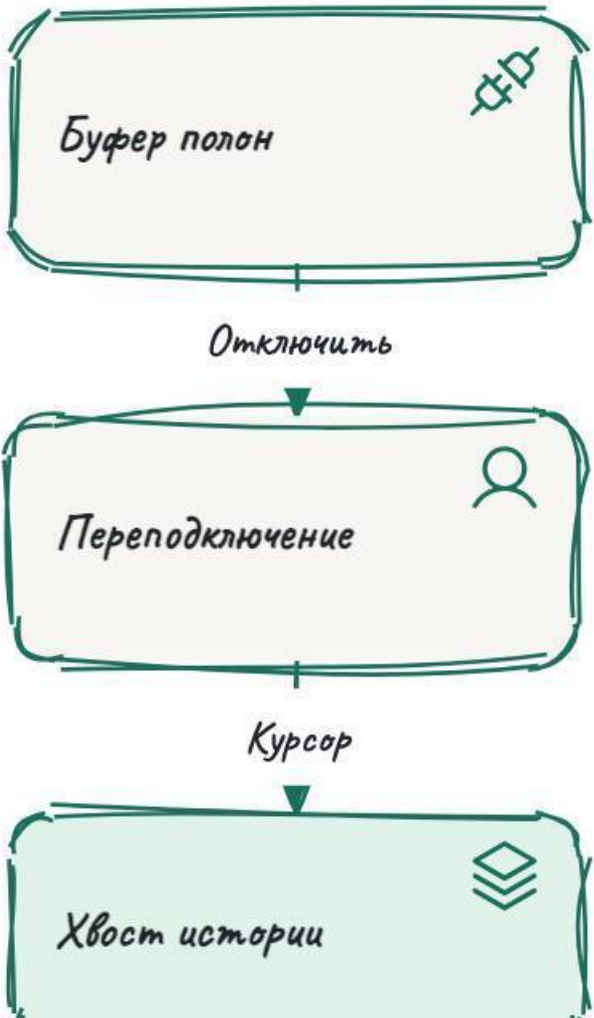


Оля посылает О17 через другой шлюз. По ключу Оли в этой беседе сервер находит прежнюю запись и возвращает номер 42.

Оля вернулась в сеть. Теперь её телефон подключён к другому шлюзу — так называют сервер на входе в чат. Телефон снова посылает О17. Шлюз проверяет, что Оля вправе пи-

сать в эту беседу, и находит запись 42 с тем же ключом и текстом. Оля получает потерянную квитанцию. В истории одна реплика. Если под O17 пришёл другой текст, шлюз сообщит о конфликте. Когда Оля сама пишет «я у ворот» второй раз, телефон выдаёт новый ключ. Такое правило повтора называют идемпотентностью.

Шлюз не должен бесконечно хранить данные для медленного устройства. У его очереди ограниченный размер, а при переполнении есть способ закрыть соединение или перейти в режим догоняющего чтения. Иначе один пользователь может занять память общего процесса. Уведомление для офлайн-режима может указать, что появились новые сообщения. Оно не выдаёт содержимое беседы без согласованных правил приватности.


Буфер полон
Отключить
Переподключение
Курсор
Хвост истории

Ограниченный буфер оставляет способ восстановиться.

Сколько доставок приходится на одно сообщение

При 200 000 подключений и проверенном пределе 10 000 на шлюз нужны минимум 20 шлюзов. Это оценка для равномерной нагрузки. При отказе одного из двадцати останется место для 190 000 подключений. Для 10 000 телефонов места не хватит. Для потери одного узла нужен как минимум 21 шлюз. Даже тогда стоит проверить, успеют ли телефоны заново подключиться все разом.

60 млн сообщений в сутки дают в среднем 694 записи/с. При пике 5 000 записей/с и трёх активных устройствах получателей на сообщение нужны 15 000 доставок/с до повторов. Крупный канал считайте отдельно: при длинном списке подписчиков выгоднее дочитывать историю по запросу.

Сценарий — Сервер зафиксировал, ответ потерялся

Проверяемый результат — Ключ возвращает тот же номер

Сценарий — Получатель пропустил два сообщения

Проверяемый результат — После подключения дочитал историю

Сценарий — Один шлюз упал

Проверяемый результат — Телефоны подключились к другим шлюзам. Каждое сообщение появилось на экране один раз

Сценарий — Клиент читает чужую беседу

Проверяемый результат — Сервис выдаёт контролируемый отказ

Сценарий — Сеть очень медленная

Проверяемый результат — Буфер ограничен, сохранённые записи доступны

Практика. Телефон увидел сообщение 145 раньше 142

В метро Лиза открывает беседу. Телефон сохранил историю по номер 141 и курсор 141. Сервер объяснил все позиции до курсора, включая скрытые. Лиза по-прежнему участвует в беседе.

Что получил телефон — Живое сообщение 145

Что известно в этот момент — Шлюз ещё не прислал пропущенную историю

Что получил телефон — Первая страница после 141

Что известно в этот момент — Сообщения 142 и 144. Сервер разрешает продвинуть курсор до 144. Позиция 143 скрыта

Что получил телефон — Вторая страница после 144

Что известно в этот момент — В истории есть сообщение 145. Сервер подтверждает курсор 145

Заполните три графы для каждого шага: история, буфер, курсор. Где сохранить курсор на случай, если телефон выключится между страницами?

Ход проверки. При живой доставке положите 145 в буфер. Курсор остаётся на 141. После первой страницы добавьте 142 и 144. Историю и курсор 144 сохраните вместе. Сервер уже разрешил пройти скрытую позицию 143.

После перезапуска телефон запросит историю после 144 и получит 145. Если 145 ещё в буфере, телефон узнает его по номеру в беседе. На экране появятся 142, 144 и 145, каждое по одному разу.

Прервите запись первой страницы на диск. Телефон должен возобновить чтение с 141 или сохранить и сообщения, и курсор 144. Курсор 144 без сообщения 142 выдаёт ошибку. Отметку о прочтении Лиза меняет отдельно.

Телефон лежит в кармане

Пока приложение на экране, сервер может передать сообщение по открытому соединению. Когда система телефона приостанавливает приложение, о новом сообщении может напомнить push — уведомление через службу операционной системы. Для доставки она выдаёт приложению адресный жетон, токен. Сервер связывает его с текущим сеансом пользователя. В уведомление для закрытого чата можно положить только сигнал о новой активности. После открытия приложение проверяет права и дочитывает историю со своей позиции.

Пусть телефон пробыл в самолётном режиме весь обед, а срок жизни сигнала — десять минут. Что пропало? Напоминание может так и не появиться. Сообщение по-прежнему в истории. Подтверждение приёма от службы push ещё не означает, что человек увидел экран. При выходе из аккаунта привязку устройства надо разорвать: следующим владельцем того же телефона может оказаться весьма любопытный младший брат.

Южные ворота

Выберите, что будет делать клиент, который получил сообщение 44, а последним сохранил 42. Можно показать 44

сразу с временным пропуском. Можно подождать 43 и придержать 44 в буфере. При любом варианте отметку «сохранено по 44 включительно» разрешено поставить лишь после обработки пропуска. Иначе Лиза никогда не получит уточнение про южные ворота. До встречи осталось двадцать минут. Оля копирует из карты точку и отправляет её следом.

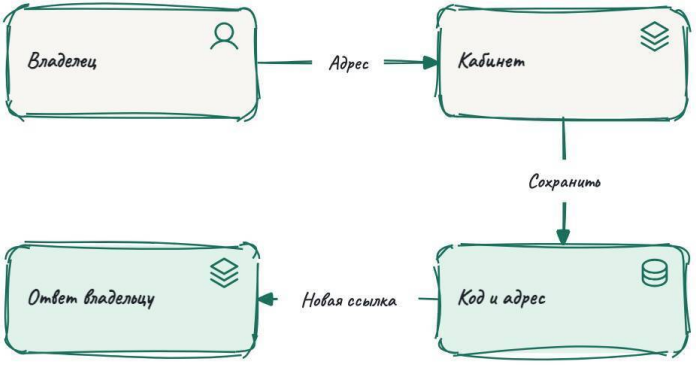
Глава 2. У входа. У какого?

10:05 · Адреса, права и кэш

На этот раз вместо описания «за ларьком, который раньше был киоском» в чате появилась короткая ссылка. Пока Глеб допивает чай, остальные уже открыли карту. От сервиса требуется совсем немного: завтра эта же строка должна вести к тем же воротам. Отдельно решим, имеет ли автор право адрес изменить.

Если ссылка говорит «здесь», за ней достаточно хранить координаты. Если говорит «я здесь», понадобится источник обновлений, срок и согласие человека. Ошибиться одним словом на экране — и из справочника адресов получился сервис слежения. Для нашей прогулки хватит неподвижной точки. Теперь можно выбирать между длинным адресом, который сам содержит координаты, и коротким кодом, за которым сервер хранит запись.




Владелец
Адрес
Кабинет
Сохранить
Код и адрес
Новая ссылка
Ответ владельцу

Владелец сохраняет адрес назначения. Теперь код позволяет гостям найти точку встречи.

Что обещает короткая ссылка

Короткой ссылкой могут воспользоваться сто друзей подряд. Чтобы сто раз не искать один и тот же адрес, узел выдачи может ненадолго оставить его у себя. Такая быстрая копия называется кэшем. У неё есть срок: Оля может сменить точку встречи раньше, чем Глеб наконец выйдет из дома.

Кабинет проверяет владельца и принимает *POST /links*. У запроса есть метод — указание, что нужно сделать. Здесь POST просит создать короткую ссылку, а GET чуть ниже просит открыть уже созданную. Эту договорённость о командах и ответах называют API: ей следует и сайт, и мобильное приложение. В запросе длинный адрес, возможный срок и ключ попытки. Ответ содержит код, короткую ссылку и время создания. Посетитель открывает *GET /r/{code}*. Сервис проверяет состояние и возвращает перенаправление. Знание кода само по себе не даёт доступа к приватным данным. Закрыть ссылку может её владелец отдельной командой по идентификатору записи.

Клики собираются отдельно. Редактор может несколько минут видеть приблизительное число переходов, пока посетители по-прежнему открывают активную ссылку. Для пер-

вого выпуска достаточно базы кодов и сервиса перенаправления. Узлы выдачи и очередь статистики можно добавить позже.

Что храним — Соответствие кода

Владелец — Запись ссылок

Поля, нужные для решения — Код, адрес, владелец, срок, статус, версия

Что храним — Результат создания

Владелец — Запись ссылок

Поля, нужные для решения — Владелец, ключ, проверенные параметры, код

Что храним — Копия перехода

Владелец — Кэш и узел выдачи

Поля, нужные для решения — Код, адрес или отказ, версия, срок

Что храним — Клик и итог

Владелец — Контур статистики

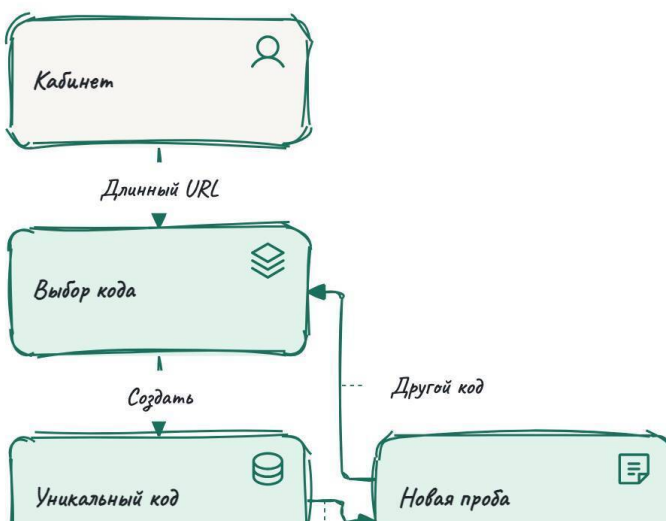
Поля, нужные для решения — Идентификатор собы-

тия, код, время, итог по окну

Авторизация на создание нужна, чтобы управлять ссылками и разбирать жалобы. Владелец присылает URL — полный веб-адрес, куда приведёт ссылка. Сервис проверяет, допустимы ли такой адрес и способ перехода по правилам площадки. Переход не должен давать браузеру произвольную команду или раскрывать внутренние данные. Даже проверенный сегодня адрес может измениться завтра. Нужны способ закрыть ссылку, ограничение массового создания и журнал решений по жалобам.

Создать короткий код

В простой версии выбираем семь случайных символов из 62: цифры, строчные и прописные латинские буквы. Пространство содержит 62^7, примерно 3,52 трлн вариантов. База использует одно представление кода с учётом регистра. Если инфраструктура приводит путь к одному регистру, смешивать его в алфавите нельзя. Ошибка здесь проявится не в формуле, а при переходе.



Уникальность кода проверяется при записи.

В случае коллизии база отклоняет второй одинаковый код. Приложение пробует новое значение в ограниченном числе попыток. При 50 млн занятых кодов вероятность попадания одного равномерного выбора в занятый код около 50 млн ÷ 3,52 трлн, то есть примерно 14 на миллион попыток. Это доля для одного запроса при такой заполненности. Ве-

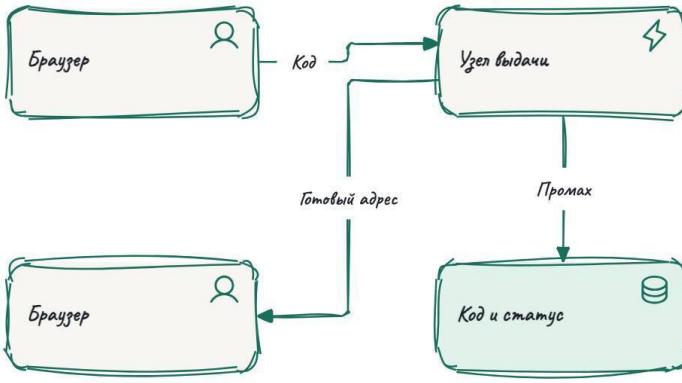
роятность хотя бы одной коллизии за длительный срок значительно выше, поэтому ограничение нужно с первого дня.

Последовательный номер в системе счисления с большим алфавитом проще выдавать без коллизий. Он делает соседние коды предсказуемыми. Случайный код затрудняет перебор, но не заменяет права доступа. Короткое пространство всё равно можно пробовать по частям. Для нашей публичной ссылки достаточно начать со случайного алфавита, а защиту приватных ресурсов оставить у ресурса.

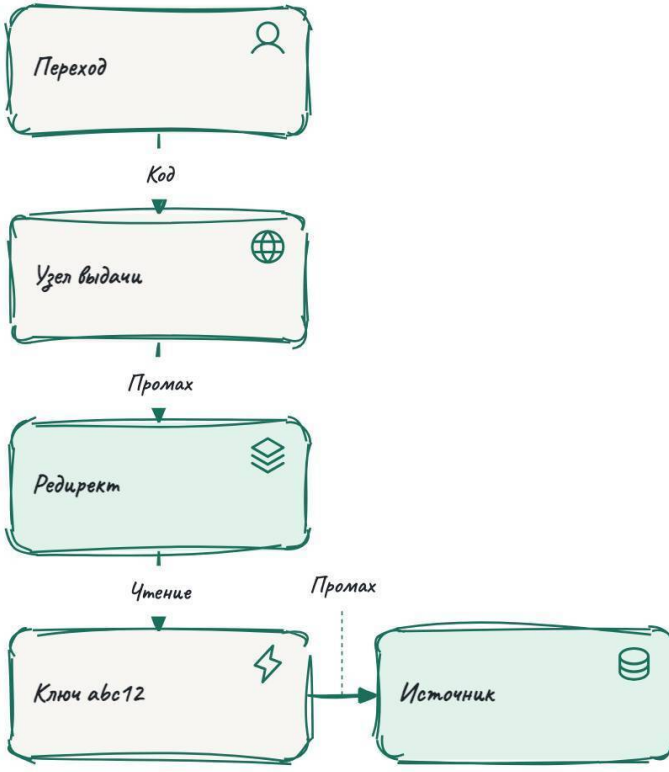
Два одинаковых запроса кабинета должны возвращать одну ссылку, если относятся к той же попытке. Уникальность кода защищает от другой ошибки, совпадения двух независимых записей. Оба правила могут жить в одной транзакции. Если сервер записал ссылку, но ответ потерялся, кабинет повторяет свой ключ и получает тот же код.

Переход: от кода до адреса

Обработчик ищет код. Если запись активна и срок не истёк, отвечает кодом 302 и заголовком *Location* с адресом назначения. Браузер делает следующий запрос к этому адресу. Короткий сервис не загружает страницу назначения за пользователя. Время открытия конечной страницы зависит уже и от другого сайта.



При промахе узел выдачи читает актуальный адрес и статус. Для действующей ссылки браузер получает перенаправление.



Кэши уменьшают чтения общего источника.

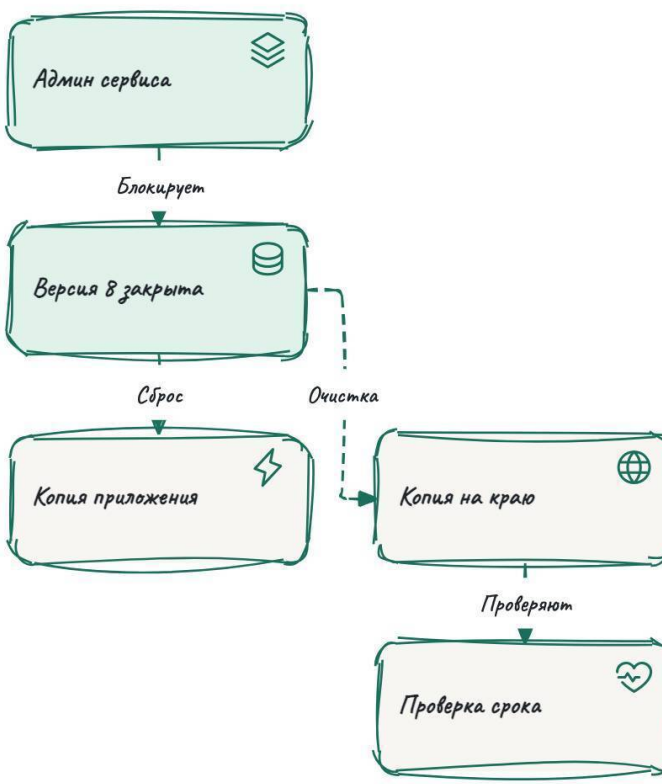
Узел выдачи рядом с посетителем сокращает сетевой путь и снимает часть запросов с приложения. Кэш около прило-

жения разгружает базу. Если браузер надолго сохранил перенаправление, очистка серверных копий на него не подействует. В этой модели время хранения ответа задаём явно. Чтобы проверить отзыв ссылки, повторяем переход и со свежего клиента, и с клиента, уже видевшего ответ.

Промах, истёкший срок и недоступность хранилища — три разных результата. Доказанное отсутствие кода можно ненадолго отрицательно кэшировать, чтобы повторы не нагружали базу. Ошибка чтения не доказывает отсутствия. Если возвращать 404 при каждом сбое, посетители и операторы не смогут отличить неизвестный код от недоступной системы. Срок отрицательной копии согласуйте с правилами создания. Для редких ручных кодов длинный срок может скрыть недавно созданную ссылку.

Закрытие и горячий код

Владелец закрывает ссылку в 14:00. У приложения ещё есть копия. В другом городе узел выдачи держит старый ответ. Сначала фиксируем новый статус и версию. Затем удаляем копию приложения и отправляем запрос на очистку узлов. Проверяем ответ известной точки после очистки. Если до закрытия уже началось чтение старого значения, оно не должно заново заполнить копию более старой версией.



Запись меняется раньше удаления копий.

Условие бизнеса формулируем наблюдаемо, например «новый переход через наши узлы прекращается в пределах

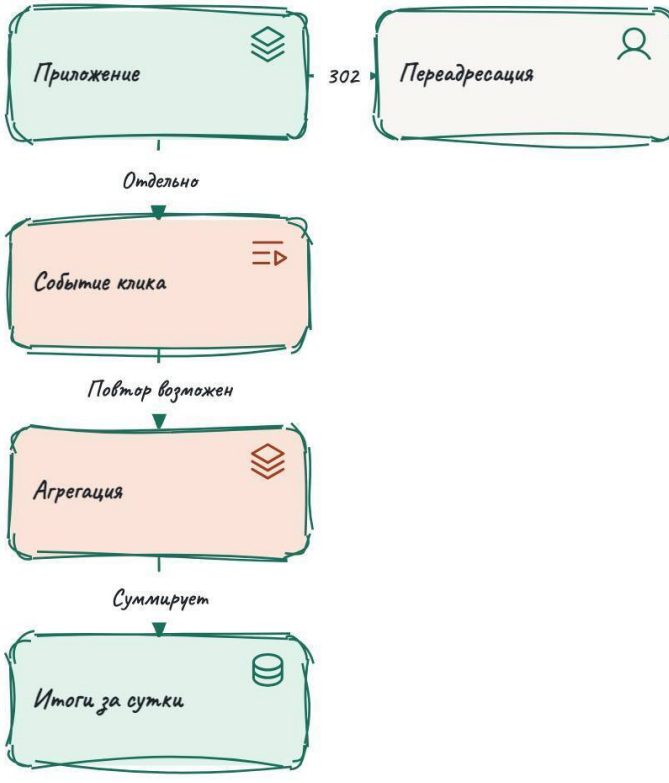
30 секунд после закрытия». Его нельзя проверить одним запросом к базе. Замеряем путь через узел выдачи и стандартный клиент. Техническое обещание ограничено контролируемыми копиями. Пользователь мог скопировать исходный адрес или уже открыть его непосредственно, и короткий сервис не может отозвать такую страницу.

Один популярный код может получить десятую часть всего трафика. Можно вычислить по коду число — хеш — и по нему выбрать раздел базы. Разные коды разойдутся по разделам, а за одной популярной ссылкой все продолжат ходить в одно место. Сначала используем короткие копии в разных узлах и ограниченное объединение промахов. На холодном старте требуется защита базы. Если закрытие ссылки важнее быстрого перехода, при неизвестной свежести можно отказать. Это выбор команды, а не свойство кэша по умолчанию.

Нагрузка и статистика

Пусть за сутки 150 млн переходов. Средняя частота около 1736 в секунду. В первые секунды рассылок тестируем 12 000 в секунду. Если узлы выдачи закрывают 80% запросов, до приложения дойдут 2400 в секунду. При доле попаданий внутреннего кэша 90% это около 240 чтений базы в секунду. Обе доли измеряем в своих местах. При полном холодном старте без допуска до базы может попытаться дойти поток в десятки раз больше.

Создание ссылки — другой маршрут. Пусть на коротком участке до 40 созданий в секунду. Для 50 млн записей по 250 байт полезные поля займут примерно 12,5 ГБ. Отдельно посчитайте копии, журналы и индексы. Индекс — указатель для быстрого поиска: по коду ссылки он подскажет, где лежит её запись. Сам указатель тоже занимает место. Задайте срок хранения. Прежде чем разбивать базу, измерьте объём и скорость обычной схемы. Разделение по коду потребует другого индекса для кабинета владельца и плана переноса.

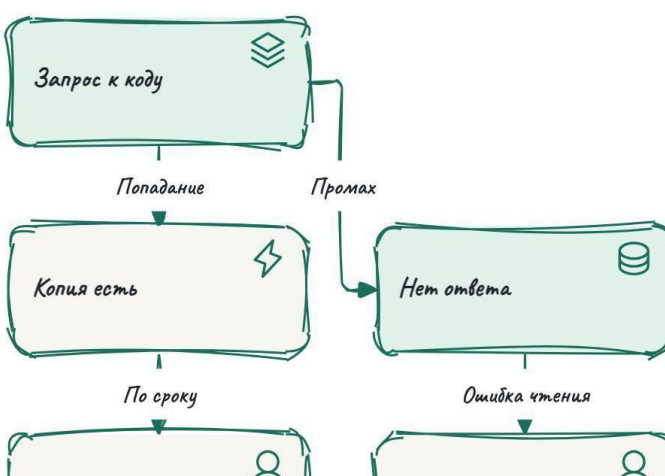


Счётчики могут запаздывать, а ссылка открываться.

Статистика считает события, которые сервис умеет увидеть. Если копия перенаправления выдана в узле, приложе-

ние не наблюдает этот запрос. Для подсчёта нужны журналы или события самих узлов. Дубли возможны при повторной публикации, поэтому для точной суммы внутри выбранного окна служит собственный идентификатор. Если считаем приблизительные переходы, можно искать потерю по доле событий и явно указывать задержку. Переход не ждёт записи статистики.

Сбой, который покажет границу



Временный отказ и отсутствующая ссылка дают разные ответы.

Отключите базу при пустой копии и при попадании. Во втором случае запишите, насколько свежа копия и относится ли она к активному коду. Для закрытых и истёкших ссылок полезен режим явного отказа. Для стабильных публичных ссылок с допустимым сроком можно обслуживать часть копий. В обоих случаях наблюдаем долю ответов, ошибки и

число чтений источника.

Сбой — Потерян ответ создания

Что сохраняется — Ссылка и ключ

Как проверить — Повтор не создаёт второй код

Сбой — Узел выдачи не успел очистить

Что сохраняется — Источник уже закрыт

Как проверить — Последний переход укладывается в договор

Сбой — Статистика отстаёт

Что сохраняется — Переадресация работает

Как проверить — Видны возраст событий и пропуски

Сбой — База недоступна

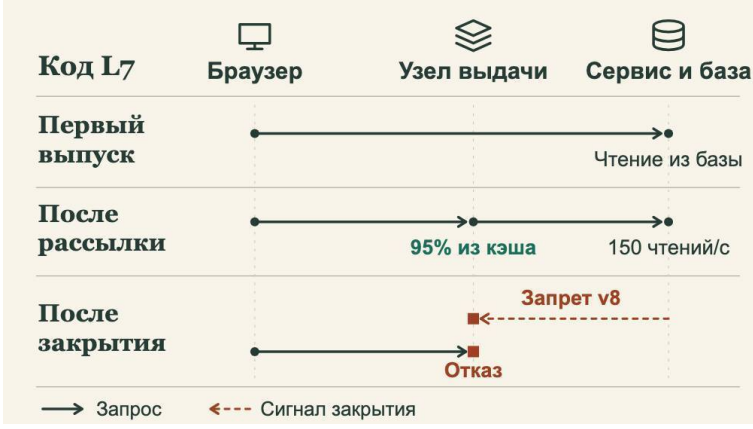
Что сохраняется — Допустимые копии

Как проверить — При промахе нет ложного 404

Одна ссылка получает 3 000 переходов в секунду

Проследите три версии сервиса на одном коротком коде. В час пик этот код получает 3 000 переходов в секунду. Узлы выдачи обслуживают 95%, кэш приложения пуст. Сколько чтений в секунду дойдёт до базы при таком распределении, если каждый промах вызывает отдельное чтение? Что изменится после блокировки?

На схеме короткая ссылка обозначена L7. Узел выдачи получил запрет версии 8. Следом вернулся ответ из базы с адресом версии 7. Куда теперь попадёт читатель, который откроет ссылку L7?



Как выросла одна короткая ссылка

В строке «После закрытия» узел выдачи помнит запрет версии 8, поэтому отбрасывает адрес версии 7. Браузер получает отказ.

Что произойдёт, если сигнал о закрытии вообще не дойдёт до узла?

До истечения разрешённого срока копии узел ещё может вернуть старый адрес. Затем ему понадобится свежее подтверждение от источника.

В приложение приходит $3000 \times 0{,}05 = 150$ запросов в секунду. Если каждый промах идёт к базе, столько же чтений она и получит. Объединение позволяет нескольким запросам использовать один результат, но не отменяет срок ожидания. После блокировки значение в источнике станет закрытым. Очистка, версия и срок должны перевести остальные копии в заданный режим. Отдельно испытайте всплеск обращений, пришедший в момент очистки.

Практика. Вернулась закрытая ссылка

Источник закрывает ссылку в момент 0. Узел выдачи должен прекратить перенаправления за 30 секунд. Пока каждое заполнение кэша даёт записи новые 30 секунд жизни.

Момент — −1 с

Что произошло — При промахе началось чтение, которое увидело версию 7

Момент — 0 с

Что произошло — Источник записал закрытие и версию 8

Момент — 3 с

Что произошло — Узел получил сигнал и удалил свою копию

Момент — 9 с

Что произошло — Результат старого чтения дошёл до узла

Когда перестанет действовать старый ответ, если узел снова положит его в кэш? Что оставить вместо пустого ключа? Как проверить узел, пропустивший сигнал?

Ход проверки. Версия 7 проживёт до 9 + 30 = 39 с. Это на 9 секунд дольше обещанного окна.

По сигналу оставьте отметку о закрытии с версией 8. При заполнении сравнивайте версии в той же операции, что записывает значение. Версия 7 проиграет. Отметку храните дольше максимального времени возврата старого чтения.

На случай потери сигнала источник задаёт абсолютный срок подтверждения активности. Узел использует его, даже когда копия попала в кэш с задержкой. По истечении сро-

ка нужна повторная проверка. При неизвестном результате узел отвечает без перенаправления.

В тесте задержите результат чтения версии 7 и отпустите его после закрытия. Повторите с потерей сигнала. На каждом узле измерьте время последнего ответа 302. К отметке 30 с перенаправления должны прекратиться.

Задание для оборота билета

Представьте, что короткую ссылку напечатали на бумаге, а через неделю вход перенесли. Длинный адрес с зашитыми координатами переживает исчезновение нашего сервера, но исправить уже напечатанные координаты нельзя. У короткой ссылки другая цена: владелец может заменить назначение, зато каждый переход зависит от нашей записи или её свежей копии. Для семейного чата можно просто написать новое сообщение. Для десяти тысяч бумажных приглашений экономия одного редактирования оправдывает целый маленький сервис.

Один ID для системы, другой для гостей

Код из семи символов удобно переслать. Внутри записи нужен стабильный идентификатор и после смены публичного кода. Последовательный номер удобно выдавать одной базой. При нескольких независимых создателях можно на ме-

сте генерировать UUID — длинный идентификатор, который разные устройства могут создавать независимо с очень малой вероятностью совпадения. У UUIDv7 начало содержит время с точностью до миллисекунды, что помогает группировать близкие вставки в упорядоченном индексе. В пределах одной миллисекунды разные узлы не получают из этого обещания общего порядка. Внешний код, который должен быть трудно угадать, выбирают отдельно. Ни один из этих идентификаторов не заменяет проверку права открыть закрытый документ.

Короткая ссылка всё ещё должна найти живой сервер, даже если он переехал. В главе 26 Оля откроет меню по коду на столике. Там и разберём, как телефон найдёт новый адрес.

Глава 3. Молоко уже взяли?

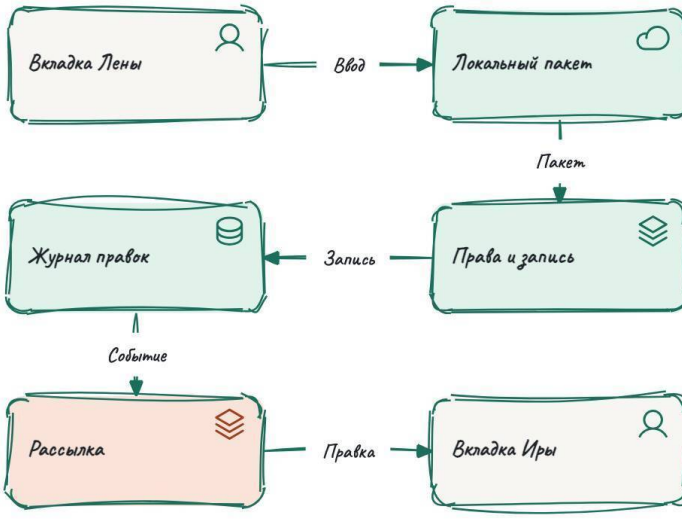
10:20 · Совместные списки и автономная запись

«Молоко уже взяли?» — спрашивает Глеб. Взяли. Остались лёд и лимоны. Глеб называет это «общей закупкой». Пока общими были только сообщения. Оля зачёркивает «лёд» на кассе, а Лиза в метро, ещё не видевшая изменения, дописывает «два пакета». Если приложение просто сохранит тот список, который пришёл последним, либо исчезнет уточнение, либо лёд снова окажется в покупках.

У пункта списка есть собственный номер. Тогда клиент может отправить «отметить купленным пункт 7», а не весь текст сразу. У количества и отметки о покупке появляются разные поля. Что делать, когда количество изменилось после покупки? Это решают люди. Наше приложение сохраняет обе правки и показывает «куплено 1 из 2». Чтобы так ответить, нужно хранить, сколько купили, а не одну зелёную галочку.



У списка есть и общая заметка. Чтобы увидеть, как редактор соединит две правки одного предложения, возьмём короткую запись Лены и Иры из соседнего чата. Обе открыли строку «Встреча в зале». Лена вставит « завтра» сразу после слова «Встреча», Ира добавит восклицательный знак после «зале». Потом вернёмся к лимонам.

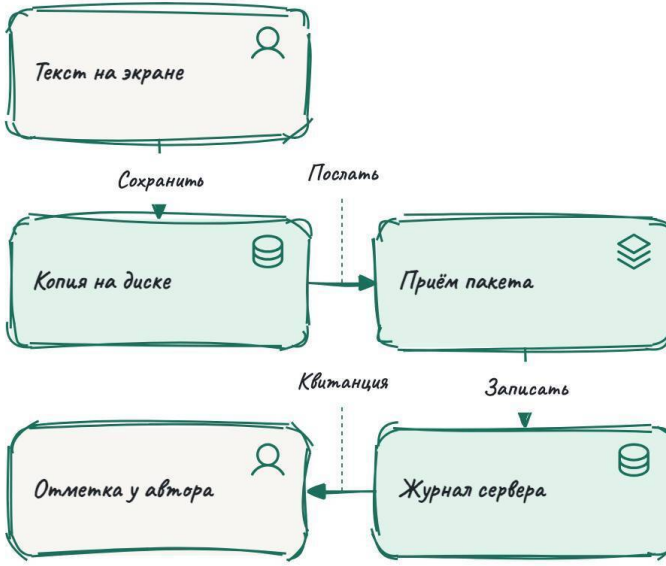


Лена сохраняет ввод на устройстве. После проверки доступа сервер записывает правку и рассылает её, даже если квитанция Лене потеряется.

Что переживёт закрытие вкладки

Лена видит слово «завтра» сразу после ввода. Пока оно только в памяти, закрытие вкладки может его уничтожить. Редактор записывает локальное изменение и пакет для отправки одной транзакцией в браузерное хранилище. Толь-

ко после успеха появляется «На устройстве». Если места не хватило, оставляем текст на экране, сообщаем об ошибке и даём скопировать черновик.



Вкладка меняет статус после подтверждения конкретного пакета.

У пакета есть идентификатор документа и собственный устойчивый номер, например Л7. В нашей модели сервер в

одной транзакции проверяет право автора на этот документ и фиксирует пакет с уникальным ключом. Изменение прав согласовано с этой же записью. Квитанция на Л7 уходит после фиксации по принятому в сервисе правилу долговечности. Когда соединение оборвалось на обратном пути, вкладка повторяет Л7. Сервер при действующем доступе находит прежнюю запись и возвращает её номер. Короткая квитанция подтверждает судьбу пакета независимо от того, прочитал ли правку кто-нибудь ещё.

В одной вкладке лежит неподтверждённый пакет с номером Л7. Связь оборвалась между серверной записью и ответом.



Квитанция потерялась

Автор видит «На устройстве». У коллеги уже может появиться новая строка. После повтора сервер находит запись Л7 и подтверждает её без повторной вставки.

Чью очередь можно теперь сократить?

Только очередь отправителя, получившего квитанцию на Л7. О доставке остальным читателям эта квитанция ничего не говорит.

Что хранится — Пакет Л7 для отправки

У кого — У Лены на устройстве

Когда можно освободить место — Когда получена квитанция именно на Л7

Что хранится — Принятая правка

У кого — В журнале документа

Когда можно освободить место — Когда проверенный снимок покрывает эту запись

Что хранится — Связь Л7 с результатом

У кого — В хранилище сервиса

Когда можно освободить место — По окончании оговорённого срока повтора

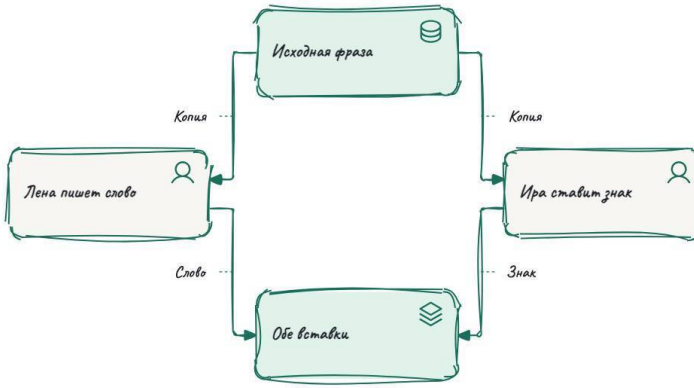
Что хранится — Текущая позиция курсора

У кого — В памяти комнаты

Когда можно освободить место — Когда пришла новая позиция или истёк срок присутствия

Как обе копии собирают одну фразу

В редакторе, который хранит только позиции символов от начала строки, инструкция «поставь знак по старому смещению» опасна. Лена вставила семь символов с пробелом, и конец строки сдвинулся. Для нашего редактора выберем CRDT — структуру данных, копии которой могут принимать изменения независимо, а после обмена сходятся к одному состоянию. В такой последовательности вставка может ссылаться на устойчивую позицию относительно знакомых элементов текста. Ирин знак встанет после последней буквы в слове «зале» даже в той копии, где появилось «завтра».



Каждый автор видит свой ввод сразу. После обмена обе копии содержат обе вставки.

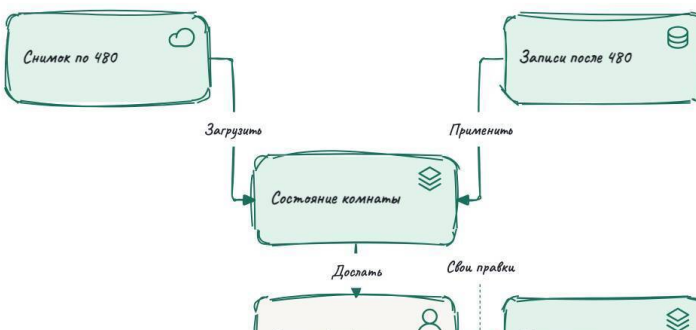
А если двое вставили в один промежуток по одному знаку? Правило структуры задаёт воспроизводимый порядок. Но программа не знает, какой смысл оба автора вкладывали в эти знаки. Вставив одновременно «?» и «!», люди могут получить «!?» или «?!» в зависимости от правила и идентификаторов. У всех, кто применил одни и те же изменения, порядок совпадёт. В Yjs, библиотеке для совместных редакторов, обновления можно получить в разном порядке и применить несколько раз. Это свойство пригодится при повторной доставке целого пакета.

Цена этого выбора видна в хранилище. Каждая реплика, то есть отдельная копия документа, хранит не только ви-

димые буквы. Ей нужна информация о структуре документа. Есть и более простой продуктовый договор: разрешить публикацию только из текущей версии и показать конфликт остальным авторам. Для короткой формы согласования это может быть выгоднее, чем поддерживать длительное автономное редактирование.

Вкладка вернулась через неделю

Пока Лена ехала, в журнале появились сотни записей. Чтобы заново открыть комнату, сервер берёт снимок состояния после записи 480 и применяет записи с 481 до текущей. Число 480 обозначает границу серверного журнала. Свою правку, которая ещё лежит в телефоне, Лена в эту нумерацию не включала.



Состояние для восстановления включает внутреннюю структуру совместного документа.

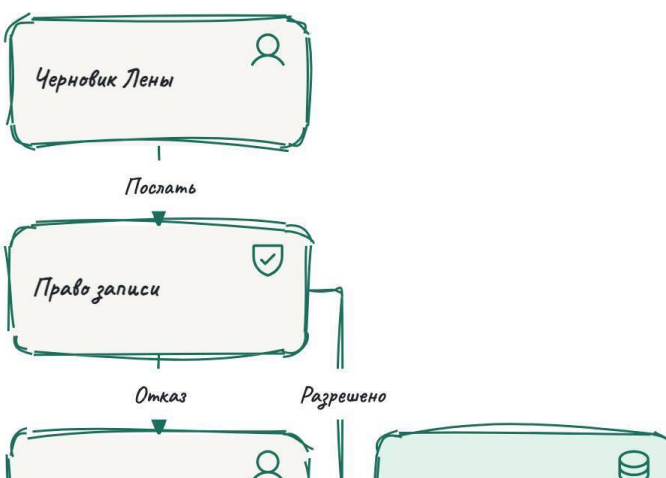
Сборщик читает согласованное состояние на выбранной границе, сохраняет снимок совместного документа вместе с его внутренними идентификаторами и проверяет, что снимок открывается. Затем одним неделимым действием публикует ссылку на файл вместе с границей 480. Такую смену называют атомарной: читатель увидит либо прежнюю пару значений, либо новую. Записи до этой границы можно убрать после проверки. Новые записи с 481 остаются в журнале. Если процесс упал до публикации указателя, восстановление пойдёт с прежнего снимка и его полного хвоста.

Экспорт в обычный текст покажет верные буквы, но у нового документа будут другие внутренние позиции. Поэтому

проверка снимка включает старую вкладку с неопубликованной вставкой. В Yjs стороны могут обменяться векторами состояния и вычислить недостающие обновления. Передаём изменения в обе стороны, включая удаления. Номер 480 сам по себе не описывает, что успел записать конкретный телефон.

Право редактирования истекло в дороге

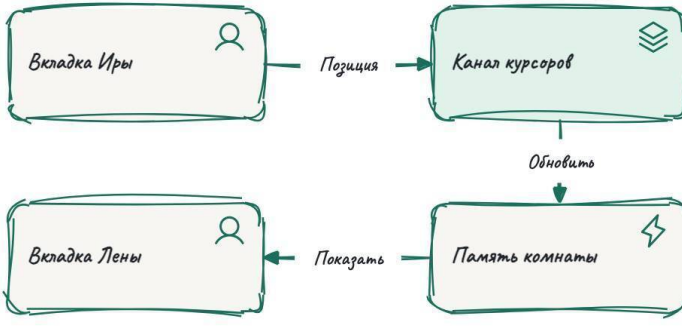
Лена открыла документ утром как редактор. Днём владелец оставил ей только чтение. При подключении Лена получает свежий текст, но её пакет сервер отклоняет. Вкладка сохраняет локальный вариант отдельно и показывает, какие изменения остались у автора. Автоматически посылать их снова каждую секунду бессмысленно. Если отозвали и чтение, сервер также прекращает выдавать новые данные. Копия, уже побывавшая на телефоне, по одному серверному сигналу с него не исчезнет.



Личная копия и опубликованный документ могут закончить день с разным текстом.

Курсорам подходит более короткая жизнь. При движении мыши Ира посылает актуальную позицию в комнату документа. Для медленного читателя достаточно последней известной позиции. После разрыва связи маркер гаснет по тайм-ауту: если от вкладки слишком долго ничего не слышно, её курсор убирают. Прокручивать вчерашнюю траекто-

рию мыши при открытии файла незачем.

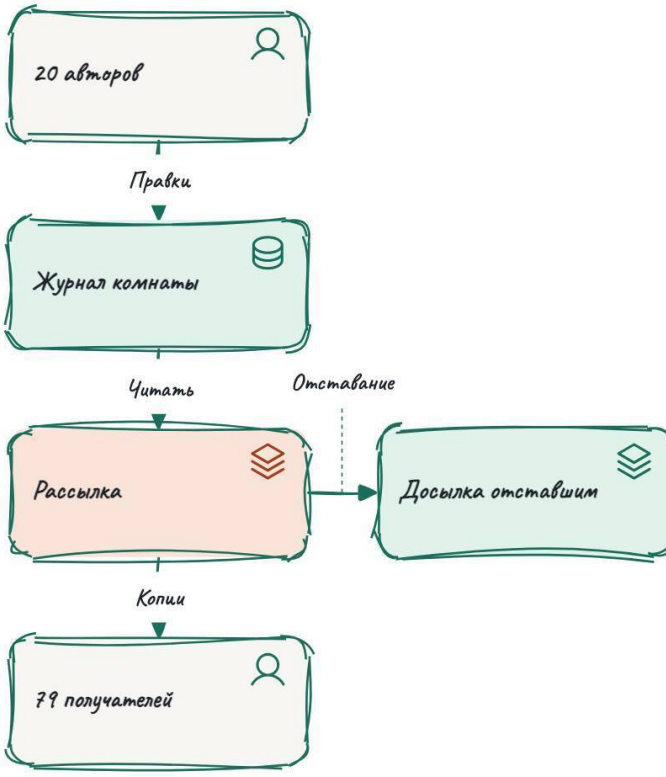


Ира сдвигает курсор. Лена видит его через канал комнаты. После обрыва позиция исчезает по тайм-ауту.

Когда тормозит один документ

Проверим тот же редактор на более шумной встрече. Объявление открыли 80 человек. Двадцать из них пишут и отправляют по три пакета в секунду. Журнал получает 60 пакетов в секунду. Если каждому пакету положены 79 других получателей, комнате нужно 4740 доставок в секунду. При полезном размере 250 байт выходит 1,185 МБ/с, ещё до транспорта, квитанций и повторов. Та же группа из восьмидесяти человек, разбитая на восемь отдельных документов по де-

сять участников, при том же суммарном вводе создаст лишь 540 доставок в секунду.



Расход на рассылку определяется числом получателей в комнате.

Комнаты можно распределить по серверам по идентификатору документа. Одна популярная комната от этого легче не станет. Следим за временем от ввода до квитанции и отдельно — за отставанием получателя от журнала. Если очередь одной вкладки переполнена, завершаем её поток и при подключении досылаем недостающее из устойчивого состояния. Остальные продолжают писать. Для присутствия достаточно выбросить устаревшие позиции. Проверьте это на двух медленных вкладках: в одной двигайте мышь, во второй печатайте. После восстановления в тексте нужны все буквы. Курсор можно нарисовать там, где вкладка покажет его сейчас.

Практика. Закрыли вкладку после обрыва

У Лены на устройстве сохранён пакет Л7 со словом «завтра». Сеть пропала до получения квитанции. Лена закрыла вкладку. Утром сервис восстановил комнату из снимка по запись 480 и хвоста с 481-й. Право записи всё ещё действует, срок хранения результатов повтора Л7 ещё не истёк. Ирин восклицательный знак уже входит в снимок.

Рассмотрите два исхода. В первом сервер успел записать Л7 под номером 479. Во втором он вообще не видел пакет.

Что покажет вкладка до подключения? Что сделает сервер с повтором? Какую строку в итоге должны увидеть оба автора?

Ход проверки. До подключения Лена видит свою локальную правку со статусом «На устройстве». Квитанции у неё нет в обоих случаях. Сначала восстановите её пакет с диска, затем откройте соединение и проверьте права.

Что стало с Л7 — Запись 479 вошла в снимок

Действие сервера — Найти результат Л7 и вернуть прежнюю квитанцию

Что проверяет тест — В журнале нет второй записи для Л7

Что стало с Л7 — Сервер не видел Л7

Действие сервера — Зафиксировать пакет и выдать новую квитанцию

Что проверяет тест — Слово дошло и до Иры

После обмена прочтите документ на обоих устройствах. В каждой копии должна быть строка «Встреча завтра в зале!» с одним словом «завтра» и одним знаком. Перезапустите сервер ещё раз и откройте документ в пустой вкладке. Эта последняя проверка поймает правку, которую все видели, но которая так и осталась в памяти.

Две корзины

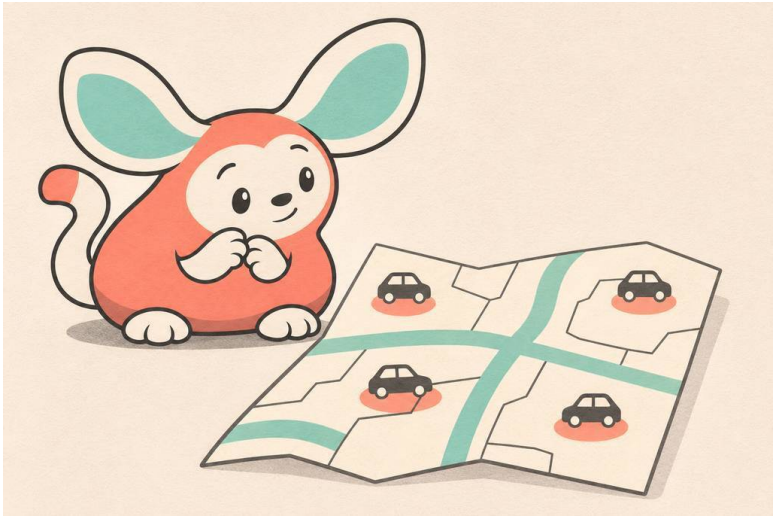
Вернёмся к покупкам Оли и Лизы. Для проверки возьмите с соседом по листку. На ваших листках одинаково написано «лимоны, нужно 3, куплено 0». Вы покупаете два, сосед один. Обмен сообщениями «теперь куплено 2» и «теперь куплено 1» теряет чью-то покупку при любом выборе победителя. Два разных события «добавить покупку 2» и «добавить покупку 1» дают три. Чтобы повторная доставка сообщения не превратила три лимона в пять, у каждого события есть свой устойчивый номер. По этой же причине у перевода за обед позже появится номер поручения.

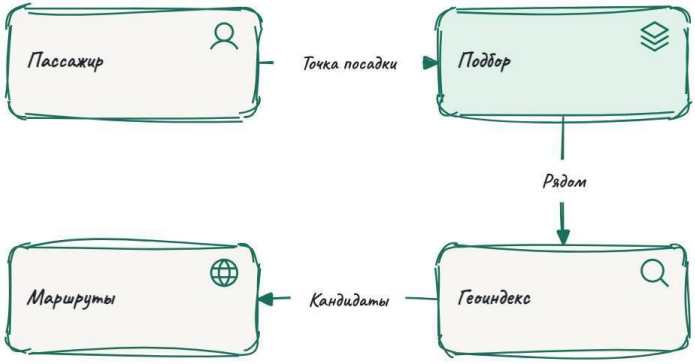
Глава 4. Машина в двух минутах. На другом берегу

10:50 · Такси, геопоиск и назначение

Оля проходит пешком. Глеб, который «вышел» семьдесят минут назад, вызывает такси. На карте свободная машина стоит почти рядом. Глеб уже видит машину через реку. Водитель тоже его видит. Они машут друг другу. До подачи восемь минут.

Начнём с двух решений. Для сотни машин диспетчерская может проверить все свежие позиции. Для большого города стоит сначала отбросить заведомо далёкие, а маршрутизатор спросить только про оставшихся. Сэкономленные вычисления имеют цену: если взять слишком узкую область, свободная машина на соседней улице выпадет ещё до сравнения. Даже отличная оценка пути пока никому ничего не обещает. Одна и та же машина может одновременно оказаться первой у двух пассажиров.



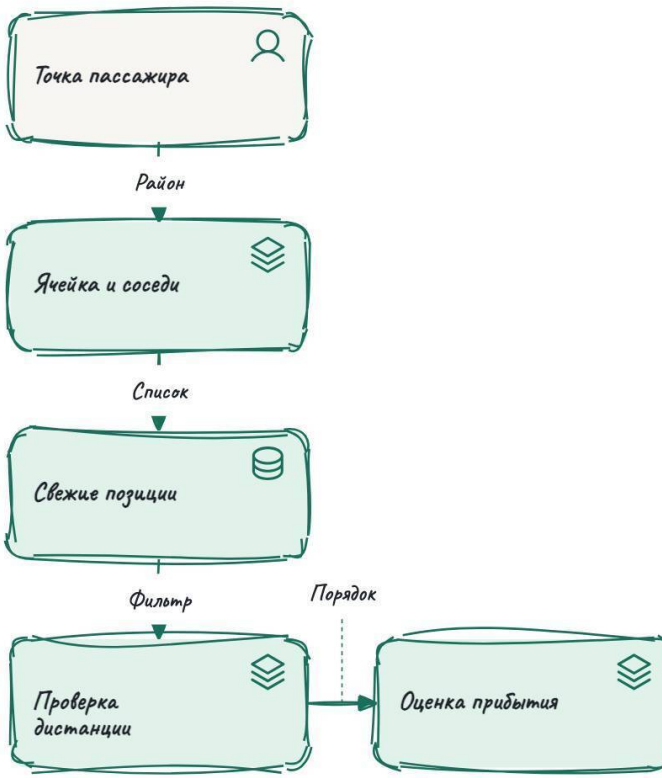

Пассажир
Точка посадки
Подбор
Рядом
Маршруты
Кандидаты
Геоиндекс

Поиск берёт свежие позиции из соседних ячеек. Дорожное время помогает выбрать кандидата.

Быстрый поиск и точный маршрут

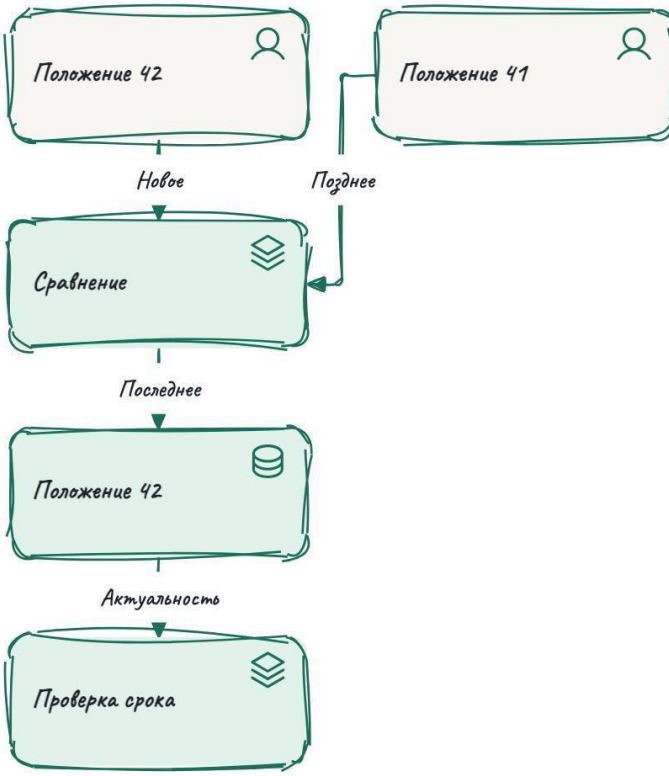
Сервис приёма геопозиций знает водителя из авторизованного сеанса. Каждое измерение имеет номер в этом сеансе, координаты, время устройства и точность. Используем серверное время приёма для ограничения срока жизни и отдельно отбрасываем явно старые измерения. Замена последней точки сравнивает номер и идентификатор сеанса. Сообщение № 34 не должно отменить № 36. Новый сеанс не принимает запоздалое обновление старого.

На записи определяем ячейку геосетки. Для пассажира собираем ограниченный набор ячеек вокруг места посадки. Берём соседей, иначе автомобиль за линией границы выпадет. Из списка удаляем дубли, отбрасываем устаревшие координаты, неверный класс и недоступных водителей. При малом количестве расширяем область, пока остаётся бюджет времени. Зависимость от маршрутизации используем только для верхнего списка: например, из 120 точек выбираем до 20 по грубой оценке и рассчитываем ETA, предполагаемое время до места посадки, максимум для них. Набор ближайших по прямой здесь приближение. Пример с мостом нужен в контрольной выборке.



Ближняя машина может находиться в соседней ячейке. Дорожные данные могут устареть, а расчёт маршрута —

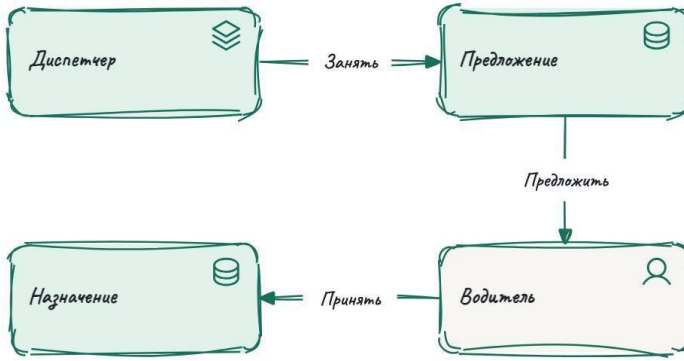
опоздать. Дедлайн здесь означает последний момент, когда ответ ещё пригодится для выбора машины. Допустимый резервный ответ — показывать приблизительное время, если это не нарушает принятый договор, и дать диспетчеру ограниченный радиус поиска. Когда уверенности не хватает, пользователь видит, что приложение продолжает поиск. Точные «4 минуты» без расчёта выглядят как обещание. У геоиндекса и сервиса маршрута разные метрики. Первый измеряет полноту списка кандидатов, второй — ошибку прогноза времени до посадки.



Старая позиция не заменяет более свежую.

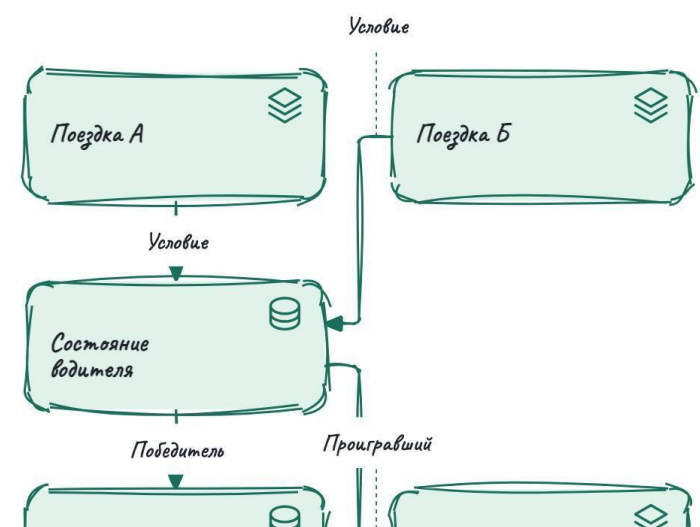
Как закрепить одного водителя за поездкой

Создание поездки использует устойчивый ключ. При перезапуске запроса смартфон получает прежний идентификатор. Назначение проходит три состояния: поиск, временное предложение, подтверждённая поездка. Хранилище владеет переходом и ограничивает активное назначение для каждого водителя. Индекс может опаздывать, серверная проверка перед предложением перечитывает запись назначения.



Найденную машину ещё надо закрепить. Окончательный статус появляется после проверки подтверждения водителя.

Пусть два пассажира одновременно получили одного кандидата. Каждый диспетчер пытается захватить предложение. В одной транзакции проверяем водителя и поездку, записываем ID предложения, владельца и срок действия. Один диспетчер выигрывает, другой пробует следующего водителя. Для подтверждения водителем проверяем тот же идентификатор, статус и серверный срок. Если старое предложение уже истекло и водитель получил новое, запоздалое принятие не должно изменить чужую запись.



Окончательное назначение защищает единая запись.

Если пригласить нескольких водителей сразу, двое могут согласиться почти одновременно. В нашей модели действует одно предложение на поездку. Замерим, сколько времени пассажир теряет в ожидании. Для параллельных приглашений потребуется выбрать одного водителя атомарно и отозвать остальные приглашения.

Водитель увидел два последовательных предложения на одном устройстве.



Водитель принял просроченное предложение

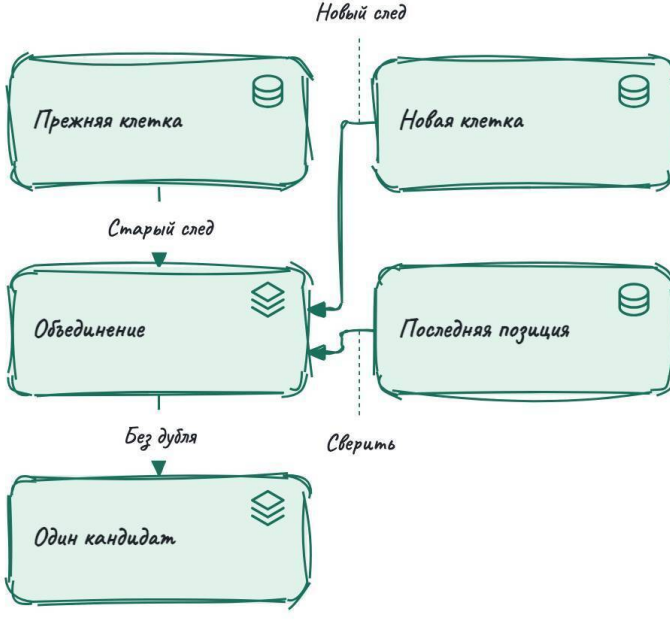
Назначение проверяет запись хранилища, ID и срок предложения. Координаты и близость не дают право занять поездку.

Кто получит поездку, если водитель нажмёт «Принять» в тот момент, когда пассажир нажмёт «Отменить»?

Если первой записали отмену, водитель увидит «Заказ отменён». Если первым пришло и зафиксировалось принятие, у Глеба уже есть машина. На команду «отменить поиск» приложение покажет её номер. От уже назначенной машины Глеб может отказаться отдельно.

Горячие районы и приватность

После концерта сотни людей нажимают кнопку на небольшой территории. Наиболее загруженной оказывается ячейка у выхода. Геосетка даёт представление пространства, её границы не служат гарантией равномерной нагрузки. В популярной ячейке можно делить водителей по ID, ограничивать частоту запросов к расчёту маршрутов и перепроверять смежные районы. Запрос в поиске не должен следовать за одинаковой машиной на каждом повторе. Для равных кандидатов применяем устойчивый порядок: например, время освобождения, затем ID.



Смена клетки не должна создавать двух водителей.

Координаты и история маршрута чувствительны. Пользователь видит только сведения о назначенном водителе в рамках текущей поездки. Трасса — запись того, как один запрос прошёл через систему. У диспетчера она хранит число кандидатов, причины отказа и округлённую область, когда подробные позиции не нужны. Поток геопозиций имеет от-

дельный срок жизни, доступ и аудит. Аналитический отчёт строится на агрегатах с учётом того, по каким малым выборкам можно установить личность.

Сколько работы в секунду

8 000 водителей с интервалом 4 секунды дают 2 000 обновлений/с. На входе возможны всплески после восстановления мобильной сети. Если в секунду поступает 100 поисков и для каждого проверяют не более двадцати машин, маршрутизатор получит до 2 000 пар «машина — пассажир» в секунду. Если отправлять пары одним пакетом, будет меньше RPC — обращений к программе на другом сервере. Дорогу для каждой пары всё равно придётся рассчитать. Чтобы понять, как долго ждут расчёта маршрута, замерим p95 его длительности: упорядочим 100 замеров от быстрого к медленному и возьмём 95-й. Пять самых медленных в это время могут не уложиться. Отдельно проверим возраст точки, долю отказов захвата и время до фактической поездки.


Ключ t7
Повтор
Проверка поездки
Читает
Водитель назначен

Проверка поездки восстанавливает видимое состояние.

Случай — Один водитель доступен для двух заказов

Что измерить и проверить — Выдано не больше одного активного назначения

Случай — Водитель принимает предложение после истечения срока

Что измерить и проверить — Проверка ID и срока на сервере

Случай — Ближайший по прямой за рекой

Что измерить и проверить — Порядок кандидатов после дорожной оценки

Случай — GPS отстал на сорок секунд

Что измерить и проверить — Водитель не проходит фильтр свежести

Случай — Маршрутный API деградирует

Что измерить и проверить — Верхняя граница параллельных расчётов соблюдена

Пассажир отменил поездку, пока машина подтверждала запрос. Повторите разбор в двух порядках. Если первой прошла отмена, предложение уже недействительно. Если води-

тель успел подтвердить поездку, действуют правила отмены подтверждённой поездки. Оба результата должны воспроизводиться из журнала. Время на телефоне для разрешения гонки мало помогает.

Практика. Кто заберёт пассажира у моста

Пассажир ждёт у реки. Диспетчер принимает точки возрастом до 15 секунд включительно. Время до посадки рассчитано по дороге до пассажира. Все три машины подходят по классу.

Машина — А, за рекой

По прямой — 400 м

Возраст точки — 3 с

До посадки — 9 мин

Машина — Б, у выезда на мост

По прямой — 1 500 м

Возраст точки — 6 с

До посадки — 4 мин

Машина — В, у остановки

По прямой — 250 м

Возраст точки — 22 с

До посадки — 2 мин

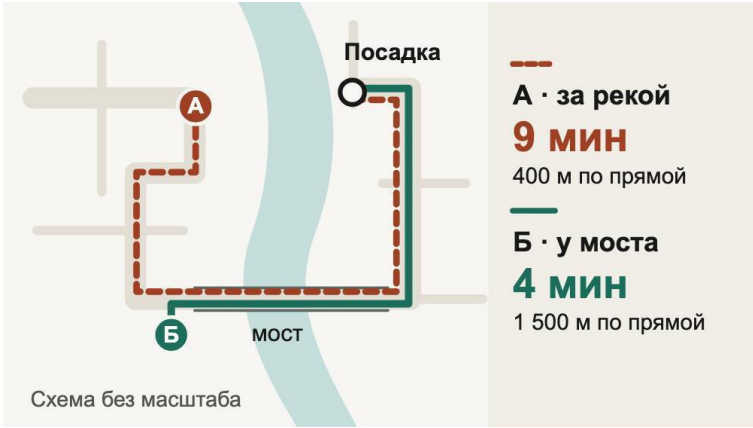
Кому первым предложить поездку? Пока шёл расчёт, водитель Б принял другой заказ. Что должен сделать сервис, если в поисковом индексе Б всё ещё свободен?

Ход проверки. Точку В исключаем по возрасту. Из оставшихся Б приедет на пять минут раньше А. Короткий список должен дойти до проверки занятости в порядке Б, А.

Попытка закрепить Б за этой поездкой обнаружит действующее назначение. Сервис перейдёт к А, заново проверит свежесть, срок поиска и занятость. Предложение можно послать после успешной фиксации. Пассажир увидит назначенного водителя после принятия предложения по правилам поездки.

Теперь ограничьте выборку одним ближайшим по прямой после фильтра свежести. В неё попадёт А. Система покажет девять минут даже тогда, когда Б свободен. Повторите опыт со свободным и занятым Б. Так можно заметить и неудачную выборку кандидатов, и ошибку двойного назначения.

Водитель А ближе к метке пассажира. До неё нужно пересечь реку.



По прямой и по дороге

Обе машины едут через один мост. Пунктир ведёт от А, сплошная линия от Б. Отбор только по прямой исключил бы Б до расчёта маршрута.

Что подскажут соседние ячейки?

Машина за условной границей может оказаться лучшим кандидатом.

Куда смотреть пассажиру

Точка на карте может отставать на две секунды. Глеб всё равно узнает машину у моста. А вот номер машины Глебу показывают только после подтверждения. Обведите на схе-

ме запись, где совпали поездка, водитель и принятое предложение.

Часть II. До обеда



Читаем городскую ленту, находим кофе с опечаткой в запросе и выбираем базу для кассы.

Глава 5. Кофе с сабой

11:15 · Поиск, фильтры и индексы

У Глеба три процента заряда и одна опечатка. Поиск всё равно предлагает ближайшее кафе. Оля уточняет: «Там есть столик на улице?» Географически подходящее место, релевантный текст и свежая карточка заведения — три разных причины показать результат. Начнём с трёх карточек, чтобы выбор можно было проверить без всякой машины.

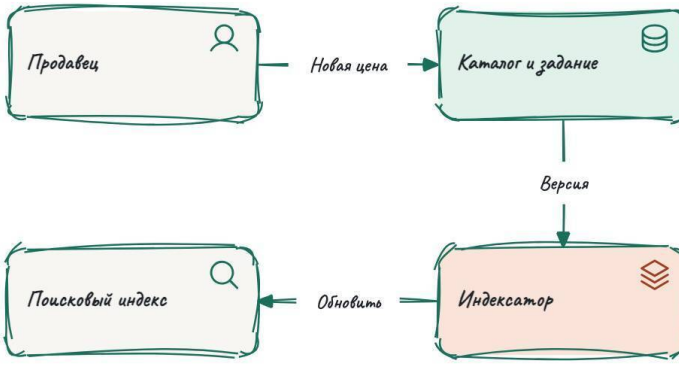
Первая кофейня открыта, до неё 400 метров, капучино стоит 220 рублей. Вторая за углом, но открывается через час. Третья работает и находится ближе первой, только в меню у неё кофе в зёрнах. Для запроса «выпить кофе сейчас» вторая проигрывает по расписанию, третья — по смыслу. Если пользователь вместо этого ищет «зёрна домой», правильный порядок станет другим. Одной сортировкой по расстоянию эти два обещания не выполнить.

От меню к индексу

Запишем для каждого нормализованного слова список карточек, в которых оно встретилось. Пропустим через ту же обработку запрос. Подходящие по тексту карточки проверим по фильтрам и отсортируем. Слово «сабой» в меню не встретилось. До знакомого «собой» одна замена буквы. По-

иск пробует и это написание, находит «кофе с собой» в первой кофейне и по-прежнему проверяет часы работы. Глебу хватит заряда дойти до открытой двери. В каталоге магазина у того же механизма есть проверка построже: все выбранные признаки должны относиться к одному продаваемому варианту товара. Разберём такую витрину, пока делают кофе.





Продавец меняет карточку. Вместе с правкой каталог сохраняет задание обновить поисковый индекс.

Что индексировать

Каталог хранит карточку товара и предложения продавцов. Если в одном документе перепутать атрибуты разных предложений, фильтр «42-й размер до 7 000 #» вернёт товар, у которого размер 42 есть за 9 000 #, а 6 000 # стоит размер 38. Для начала индексируем одно предложение на документ. Из похожих предложений собираем карточки на уровне выдачи с явно выбранным правилом группировки. Число карточек после группировки может оказаться меньше количества совпавших документов.

Поле — Название и описание

Представление — Анализируемый текст

Зачем — Поиск по словам, нормализация
Поле — Продавец, бренд, размер

Представление — Точные значения

Зачем — Фильтры и группировка
Поле — Цена в копейках

Представление — Целое число

Зачем — Диапазон и сортировка
Поле — Видимость и версия

Представление — Точное значение, число

Зачем — Право показа и защита от старых обновлений

Для примера «беговые кроссовки» сначала учитываем совпадение слов в заголовке. Фильтры применяем к тем же документам. Опечатки и синонимы добавляем после измерения качества на реальных запросах, иначе поправка «лук» повлияет и на продукты, и на одежду. Отдельная панель разбора показывает совпавшие слова, отфильтрованные пред-

ложения и вклад сигналов в порядок. Так можно объяснить, почему нужный результат оказался на двадцатой позиции.

На бирках двух предложений можно проверить этот запрос вручную: закройте одну бирку и посмотрите, подходят ли размер и цена на оставшейся.

В магазине одна модель кроссовок представлена двумя предложениями. Покупатель ищет 42-й размер по цене не выше 7 000 #.



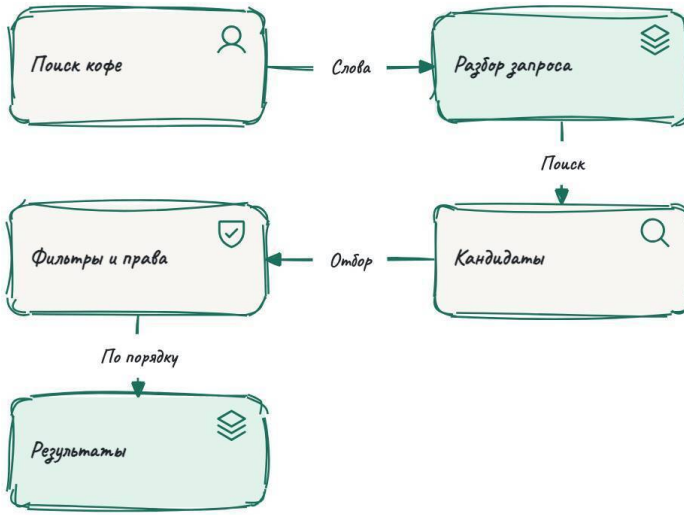
Размер и цена на одной бирке

У предложения А подходит размер, у Б — цена. Ни одна бирка не прошла обе проверки, поэтому карточка модели в выдачу не попадает. Если сложить все размеры в один список, а цены в другой, у товара появится сочетание, которого

магазин не продаёт.

Что изменится, если А подешевеет ровно до 7 000 #?

Предложение А пройдёт оба условия. В карточке для этого запроса можно показать его цену. Более дешёвое предложение Б по-прежнему относится к другому размеру.

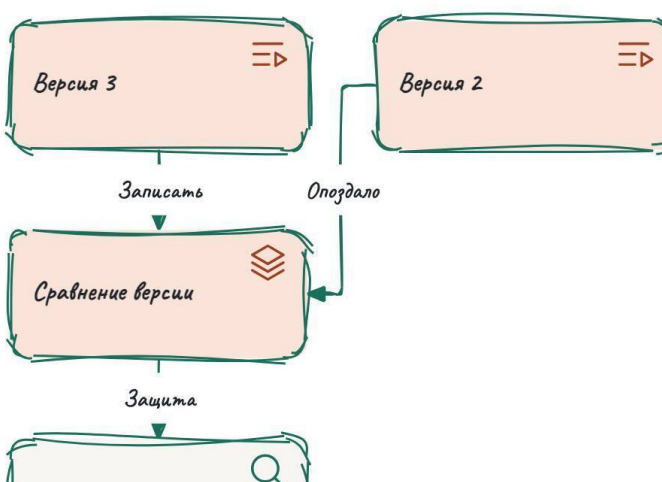


По словам из запроса поиск находит кандидатов. Фильтры исключают закрытые заведения и скрытые карточки до выдачи.

Запись и догоняющий индекс

В одной транзакции каталог сохраняет предложение и кладёт записку «передать версию в поиск» в специальную таблицу исходящих событий, outbox. После сбоя записка напомнит, что осталось отправить. Индексатор загружает снимок нужного предложения, по возможности группирует запись пачками и отмечает обработанную позицию после подтверждения. При обновлении одного предложения событие может нести полные индексируемые поля, но в этом случае потребуется поддерживать версию схемы события. Для обоих подходов индексатор сравнивает версию предложения атомарно с обновлением.

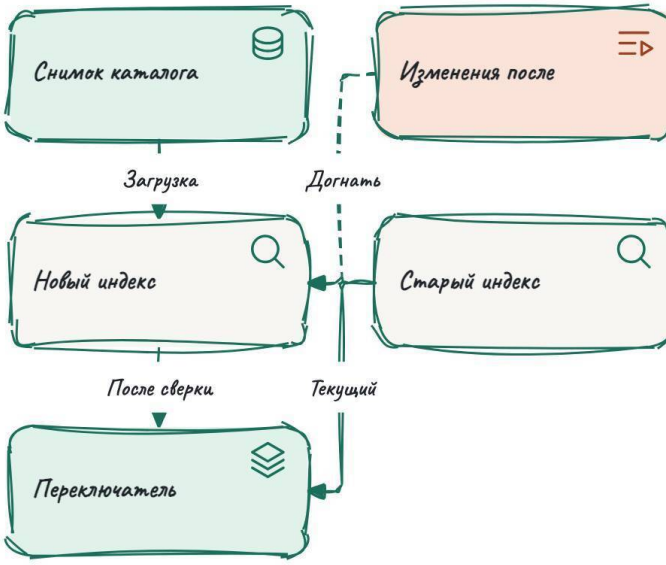
Пример. Событие № 105 подняло цену, № 106 сняло товар с продажи. После повтора обрабатывается 105. Запись со старой версией пропускается. Пока возможны старые повторы, индексатор или журнал хранит отметку об удалении с номером версии. Если стереть и эту отметку, запоздавшее событие № 105 сможет создать документ заново.



Индекс обновляет документ по версии.

Снятие видимости по жалобе проходит через небольшой актуальный реестр запретов. При выдаче сервис исключает запрещённые идентификаторы, при открытии карточки каталог применяет собственную проверку. Кэш ответа привязывается к версиям видимости либо отключается для маршрута, если недоступность реестра закрывает выдачу. Решение принимают с учётом цены ошибки. Например, магазин

может показывать устаревшую общую выдачу без персональных данных, но скрытое предложение показывать нельзя.

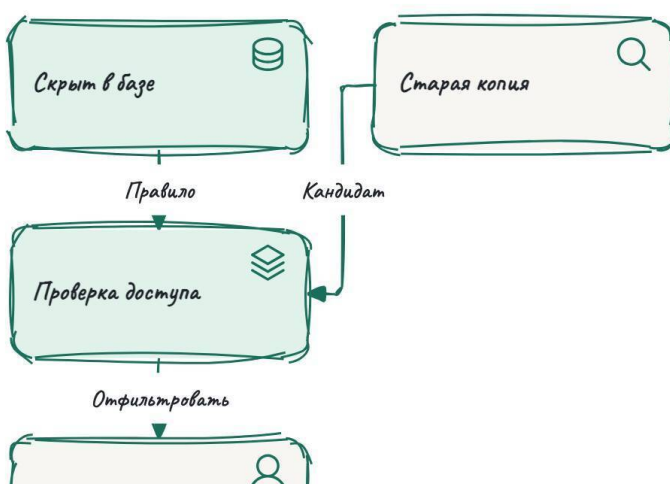


Старый индекс работает, пока новый догоняет изменения.

Как перейти на новый индекс между страницами

Результаты сортируются по значению релевантности и устойчивому идентификатору. Курсор несёт оба значения вместе с ID поискового снимка и отпечатком фильтров. Короткоживущий снимок защищает листание от переезда результата между страницами при обновлении индекса. Снимки имеют цену: держат старые сегменты, поэтому время жизни, размер страницы и глубина ограничены. После срока действия клиент начинает поиск заново. Очередь старых курсоров не должна мешать смене версии индекса.

Переиндексация стартует с согласованного снимка каталога и позиции журнала. Новый индекс получает полный снимок и затем догоняет журнал. Некоторое время обе версии обслуживают обновления. Перед переключением псевдонима сравниваем количество активных документов, контрольные версии и контрольные выдачи. Запрос с прежним снимком может завершиться в пределах его срока. После перехода сохраняем параллельное обновление старого индекса, пока нужен быстрый откат.

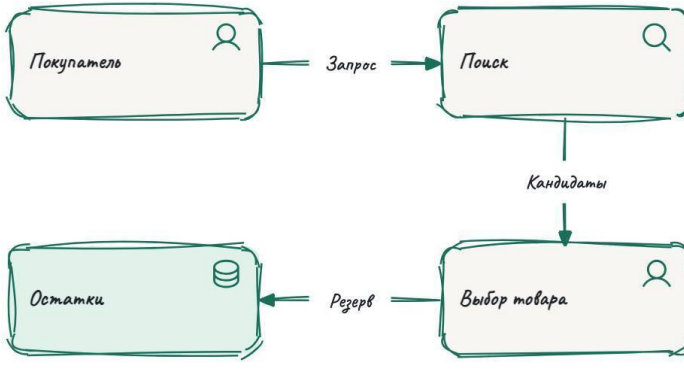


Проверка статуса закрывает окно отставания.

Нагрузка и проверка

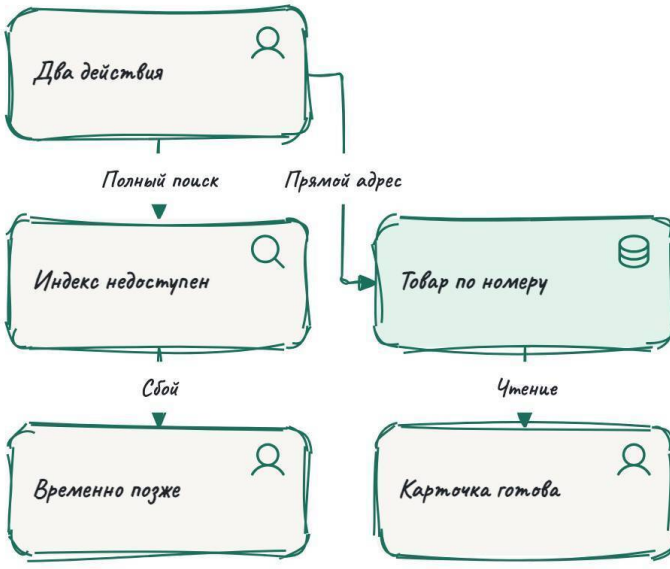
Если сериализованное предложение занимает около 1,2 КБ, десять миллионов записей дают 12 ГБ только этих данных. У каждого слова есть список карточек, в которых оно встретилось. Такие записи называют постингами. Для оценки места на диске придётся учесть эти списки, служебные

поля и сжатие. Объём индекса и пик при слиянии сегментов измеряем на пробной выгрузке. При миграции держим два набора шардов вместе с репликами. Шард хранит свою часть индекса, реплика хранит копию этой части. Если разрезать меню на «еду» и «напитки», получатся две части. Если снять копию страницы с напитками, второй обед от этого не появится. При 2 000 запросов/с и 6 затронутых шардах на запрос верхняя оценка — 12 000 обращений к шардам/с до повторов. Нужна проверка с реальными фильтрами и распределением ключей. Общее p95 не покажет, что один шард с популярной категорией занят 95% времени.



Индекс помогает найти подходящие предложения. Перед покупкой сервис проверяет конкретное предложение у вла-

дельца остатка.



Резервный путь покрывает только простые запросы.

Для оценки качества соберите контрольные запросы. Например, «кроссовки 42 до 7000» проверяет связь атрибутов, модель с известным артикулом — точное совпадение, запрос с опечаткой — устойчивость анализатора. Отдельно считайте долю нулевых выдач и актуальность цены в карточках.

CTR показывает, какая доля показов закончилась нажатием. Если по карточке нажали 12 раз за 100 показов, CTR равен 12%. Высокий CTR может поощрять яркое описание при малом числе релевантных предложений, поэтому часть запросов лучше размечать вручную.

Проверка — Два объявления с разными размерами и ценами

Ожидаемое поведение — Выдача не подменяет значения между предложениями

Проверка — Удаление приходит раньше обновления

Ожидаемое поведение — Обновление не воскрешает документ

Проверка — Курсор после переключения псевдонима

Ожидаемое поведение — Действующий снимок продолжает выдачу

Проверка — Журнал открыт после паузы в 25 минут

Ожидаемое поведение — Отставание наблюдаемо, запреты продолжают применяться

Принесите на обсуждение решение о цене на экране. Для поиска цена может отставать до согласованного предела, а перед покупкой нужен ответ из актуального каталога. Изменившуюся цену показываем пользователю до подтвержде-

ния заказа. Метрика расхождений подскажет, хватает ли 30 секунд или компромисс стоит пересмотреть.

Практика. Какую цену поставить в выдаче

Покупателю нужны кроссовки 42-го размера не дороже 7 000 #, которые есть в наличии. Таблица показывает один снимок индекса. В результатах нужна одна карточка на модель с минимальной ценой среди подходящих предложений этой модели.

Модель и предложение — Ветер · А

Размер — 42

Цена — 7 100 #

Остаток — 1
Модель и предложение — Ветер · Б

Размер — 41

Цена — 6 800 #

Остаток — 3
Модель и предложение — Шаг · В

Размер — 42

Цена — 7 000 #

Остаток — 1
Модель и предложение — Шаг · Г

Размер — 42

Цена — 6 500 #

Остаток — 0

Какая карточка попадёт в выдачу и какую цену увидит покупатель? Пока он открывает карточку, последнюю пару по предложению В покупает другой человек. Что должен проверить магазин при оформлении?

Ход проверки. У модели «Ветер» предложение А дороже потолка, у Б другой размер. Для «Шага» подходит только В: цена ровно на границе допустима. В выдаче одна карточка, «Шаг» за 7 000 #. Цена 6 500 # взята с бирки отсутствующей пары.

При оформлении магазин проверяет выбранное предложение В и атомарно пытается зарезервировать одну пару у владельца остатка. Если остаток уже нулевой, резерв не создаётся, а покупатель видит, что пару успели забрать. Успеш-

ное чтение из поискового индекса само по себе никому не обещало товар.

Событие версии 105 повторилось вслед за отменой видимости версии 106.



Старая запись после удаления

Индексатор сохраняет сведения о последней версии удалённых документов, пока возможны поздние события. Проверка выполняется при записи.

Зачем хранить отметку «удалено» (tombstone), если товара больше нет?

В отметке об удалении остаётся номер версии. Когда вечером придёт вчерашняя цена, индексатор увидит, что товар успели убрать, и карточка снова не появится в поиске.

У первой кофейни в карточке есть пометка «терраса». Оля нажимает «на улице» и видит её снова. Значит ли это, что четверым найдётся место прямо сейчас? Кафе сообщило, что у него есть столики во дворе. Свободны ли они, знает тот, кто рассаживает гостей. Для обещания «свободен стол на четверых» понадобятся данные зала, время последней проверки, а для гарантии места — ещё и бронь. Пока приложение показывает «есть терраса». Оля звонит в кафе.

Глава 6. Почему вчерашний кофе стоит дороже

11:40 · Как выбрать базу данных

Оля берёт на кассе кофе в зёрнах. Пачка стоит 459 рублей. Вчера она купила точно такую же за 499. Пока кассир ищет сдачу, Оля открывает старый чек. Хорошая новость для архитектора: оба числа правильные. Плохая для самой простой схемы: одной колонки «цена» на весь магазин теперь мало.

В карточке товара есть номер и название. В предложении магазина — этот номер, магазин, цена и начало действия цены. В строке продажи — номер чека, товар, количество и сумма, которую покупатель заплатил. Если завтра карточка поменяет название, бухгалтерии всё ещё понадобится прочесть вчерашний чек так, как его выдали на кассе. Хранить некоторые сведения второй раз здесь разумно: строки описывают разные события во времени.

Вопрос — Что в пачке со штрихкодом 705?

Запись, которая на него отвечает — Карточка товара и индекс по коду

Вопрос — Сколько пачка стоит здесь сегодня?

Запись, которая на него отвечает — Предложение магазина с периодом действия

Вопрос — За сколько вчера продали две пачки?

Запись, которая на него отвечает — Строки подтверждённого чека с фактической ценой

Вопрос — Сколько кофе продали за месяц?

Запись, которая на него отвечает — Сумма количеств из проведённых продаж с учётом возвратов

Как хранить чек, ценник и месячную выручку

У реляционной базы есть таблицы и связи. Можно связать строку с существующим товаром, задать уникальную пару «касса, номер чека» и фиксировать чек вместе со всеми строками одной транзакцией. Для десяти магазинов с обычным поиском это хороший исходный проект. Фотографию пачки положим отдельно, в базе оставим её адрес. Имя бренда не годится на роль ключа: упаковщик может переименовать кофе и по-прежнему продавать тот же товар.

Документной модели подходит законченный чек, который обычно читают целиком. Сохраняем документ с массивом строк и суммами. Чтение удобно; новый отчёт «продажи одного товара по городу за год» заставит искать внутри множества документов или строить дополнительное представление. Документ также может лежать в реляционной базе. Фор-

мат записи и свойства конкретного движка стоит оценивать отдельно.

Хранилище «ключ — значение» удобно для локального ответа кассы по штрихкоду. Если искать «все товары без сахара дешевле 200 рублей», известных ключей уже нет. Понадобится индекс или просмотр набора. Ставить такую задачу перед каждым терминалом накладно. Отдадим сложную выдачу каталогу. Касса пусть умеет быстро узнавать пакет молока.

Колоночное хранение интересно там, где отчёт читает несколько полей из миллионов продаж. Сканируем время, магазин и сумму, оставив адрес доставки и печатный текст в покое. Сначала измерим реальные отчёты. Если они занимают секунду ночью, отдельное хранилище придётся кормить и обслуживать совершенно зря. Графовая модель полезна для обходов связей: «чем заменить этот товар, если нужной марки нет у всех доступных поставщиков?» Кассиру с пачкой кофе хватает одного поиска по штрихкоду.

Один эксперимент до покупки сервера

Пусть рабочий каталог занимает 60 МБ, кафе обновляет за день 300 цен, одна запись обновления вместе с кодом занимает 100 байт. Полная копия для одной кассы стоит 60 МБ, дневные изменения — около 30 КБ без служебных расходов. Передача изменений экономит байты, но заставляет

помнить номер версии и уметь догонять пропуски. На первом пилоте с одной кассой полная копия раз в ночь может оказаться лучшей покупкой. Если требуется менять цену одновременно во всех магазинах в полдень, ночной копии уже недостаточно при любой её дешевизне.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.