

Разрешение противоречия «общая инфраструктура ↔ изоляция арендаторов» в мультитенантной SaaS- платформе

Демонстрационный отчёт для IT-архитекторов: ТРИЗ-анализ архитектурной задачи, расчёты стоимости изоляции и радиуса поражения, экономика решений и шаблоны для архитектурного ревью.

Тип задачи	Методология	Дата анализа	Формат
Инженерная / архитектура ПО	ТРИЗ: 9 фаз, 4 стратегии	28 сентября 2026	Демо-кейс для сайта

1. Резюме

B2B SaaS-платформа электронного документооборота обслуживает 320 арендаторов на одном общем кластере и одной общей базе данных. Пока клиентов было мало, это было дешево и просто. Теперь крупные арендаторы «шумят» и роняют скорость для остальных, а корпоративные клиенты при продлении и в тендерах требуют выделенную изоляцию данных, собственные ключи шифрования и гарантированный SLA. Дать каждому арендатору отдельную инфраструктуру невозможно: 320 отдельных контуров стоили бы 57,6 млн ₽ в месяц при выручке 31,1 млн ₽. Это классическое ТРИЗ-противоречие: инфраструктура должна быть общей и отдельной одновременно.

Агент Trizly прошёл полный девятифазный цикл анализа для инженерной задачи: сформулировал два технических противоречия (по обоим направлениям) и физическое противоречие, определил идеальный конечный результат (ИКР), провёл ресурсный аудит, сгенерировал решения по четырём стратегиям, объединил их, проверил «скептиком» (технический директор, руководитель SRE, руководитель информационной безопасности, финансовый директор) и ранжировал по составному критерию. Для архитекторов добавлены расчёты стоимости режимов изоляции, радиуса поражения и справедливых долей ресурсов, оценка экономики с честным диапазоном окупаемости и шаблоны для ревью.

Итог: из 5 проверенных решений приоритет — «гарантированные доли ресурсов» (квоты и справедливое планирование с отдельным пулом для тяжёлых задач) и «уровни изоляции как конфигурация» (общий пул, выделенная схема или база внутри общей ячейки, выделенная ячейка — при одной кодовой базе); далее — криптографическая изоляция со слоем защиты от ошибок доступа. Расчёт: изоляция уровня «bridge» для 30 крупных арендаторов обходится примерно в 0,7 млн ₽ в месяц против 5,4 млн ₽ за отдельные стеки, то есть в 7,8 раза дешевле; окупаемость пакета — от ≈5 до ≈21 месяца в зависимости от числа выигранных крупных сделок.

2. Описание задачи (кейс)

Контекст. Платформа электронного документооборота для компаний. Единая кодовая база, общий кластер приложений в Kubernetes, общая база PostgreSQL с колонкой `tenant_id` и фильтрацией на уровне кода, общая очередь фоновых задач. Команда: 9 SRE и 6 продуктовых команд. Кейс учебный: числа приближены к типичным и служат для иллюстрации методики.

Арендаторов: 320, из них 290 малых и средних (≈45 тыс. ₽/мес) и 30 крупных (≈600 тыс. ₽/мес; 58% выручки)

Выручка: ≈31,1 млн ₽/мес (≈373 млн ₽/год)

Инфраструктура и эксплуатация: 4,4 млн ₽/мес (≈14,2% выручки, ≈13,75 тыс. ₽ на арендатора)

Отдельные установки: 2 арендатора, вручную; ≈180 тыс. ₽/мес каждая; каждый релиз для них +2 дня работы SRE

Симптом	Данные
«Шумный сосед»	9 инцидентов в квартал, в среднем 55 минут; p95 API растёт с 250 мс до 2,1 с; радиус поражения — все 320 арендаторов; в ≈70% случаев источник — один из 5 крупнейших арендаторов (импорт или отчёты в последние 3 дня месяца, до 70% пула соединений БД)
Межарендаторный доступ	1 инцидент за год: ошибка в фильтре <code>tenant_id</code> в отчётном модуле, затронуты 2 арендатора, выявлено за 40 минут; компенсации ≈1,2 млн ₽
Требования крупных клиентов	11 из 30 крупных при продлении просят выделенную изоляцию данных, собственные ключи шифрования (BYOK) и локализацию; выручка под риском ≈79,2 млн ₽/год
Проигранные тендеры	3 сделки по ≈14 млн ₽ годовой выручки каждая (всего ≈42 млн ₽/год) из-за отсутствия выделенной изоляции
SLA-кредиты и разбор инцидентов	≈0,54 млн ₽/мес кредитов (в среднем 6 крупных арендаторов × 5% платежа за инцидент) + ≈0,45 млн ₽/мес работы инженеров

Две попытки «выбрать сторону». Отдельный контур для каждого: стоимость ≈180 тыс. ₽ в месяц на арендатора, то есть в 4 раза больше платежа малого арендатора; 320 контуров = 57,6 млн ₽ в месяц при выручке 31,1 млн ₽, а команда из 9 SRE не сможет обновлять 320 окружений. **Общая платформа с грубым лимитом запросов:** глобальный лимит на API ограничил и крупных клиентов с законными пиками, и не защитил от ошибок доступа к данным.

Цель. Сократить инциденты «шумного соседа» с 9 до ≤2 в квартал, исключить межарендаторные утечки, дать

крупным арендаторам выделенную изоляцию и собственные ключи при единой кодовой базе и росте затрат на инфраструктуру не более чем до ≈17% выручки.

Кейс сконструирован как репрезентативный пример класса задач «мультиарендантность: изоляция против экономики общей инфраструктуры», типичного для B2B SaaS, платформ данных, платёжных и маркетинговых систем — данные приближены к типичным порядкам величин и приведены для демонстрации работы агента.

3. Классификация типа задачи ФАЗА 0 — INTAKE+

ИНЖЕНЕРНАЯ ЗАДАЧА — архитектура ПО, мультиарендантность, безопасность данных, управление ресурсами.

Границы системы: шлюз и маршрутизация, приложения, база данных, очередь фоновых задач, ключи шифрования, процесс релизов

Главная полезная функция: обслуживать всех арендаторов надёжно и безопасно при экономике общей инфраструктуры

Условия эксплуатации: пики импорта и отчётов в конце месяца, арендаторы разных размеров и требований, 9 SRE на все окружения

Ограничения: без остановки сервиса; единая кодовая база (без форков); бюджет ≈36 человеко-месяцев за 6 месяцев; требования к локализации данных

4. Техническое противоречие ФАЗА 1

ТП-1: Улучшение изоляции арендатора и надёжности (аналог парам. 27 — надёжность) → рост сложности платформы и эксплуатации (аналог парам. 36 — сложность устройства).

По аналогии с матрицей (27 × 36) справочник агента рекомендует принципы:

№13 Наоборот

№28 Замена механической схемы

№35 Изменение параметров

№10 Предварительное действие

ТП-2 (обратное): Улучшение эффективности использования ресурсов и плотности арендаторов (аналог парам. 39 — производительность) → снижение изоляции и надёжности (парам. 27).

По аналогии с матрицей (39 × 27) справочник агента рекомендует принципы:

№1 Сегментация

№10 Предварительное действие

№35 Изменение параметров

№28 Замена механической схемы

Для программных систем соответствие параметрам матрицы — аналогия (парам. 27 — изоляция и устойчивость к отказам, парам. 36 — сложность платформы и эксплуатации, парам. 39 — плотность и экономичность); принципы фиксируются в Фазе 1 и применяются в Фазе 4.

Цепочка «почему» (углубление до корня)

- Почему общая инфраструктура вредит изоляции?** Все арендаторы делят пул соединений БД, очередь задач и процессор кластера, а изоляция данных держится на одном логическом условии в коде (tenant_id). Обе изоляции — по производительности и по доступу к данным — зависят от дисциплины кода и от честности соседа.
- Почему нельзя дать каждому отдельную инфраструктуру?** Минимальный отдельный контур стоит ≈180 тыс. ₽ в месяц (вычисления, база, эксплуатация) — в 4 раза больше платежа малого арендатора; 320 контуров обходятся в 57,6 млн ₽ при выручке 31,1 млн ₽, а обновлять 320 окружений команда из 9 SRE не сможет.
- Почему изоляция «всё или ничего»?** Платформа умеет только два режима: общий кластер или отдельная установка «вручную». Архитектура не выделяет изоляцию как набор регулируемых параметров (данные, вычисления, ключи, сеть, квоты), которые можно менять по арендатору без нового релизного процесса.

Физическое противоречие (Уровень 3)

Инфраструктура арендатора должна быть **общей** (экономика, простота эксплуатации, один код) — функция 1, и одновременно должна быть **отдельной** (изоляция данных, производительности и ключей; требования безопасности и SLA) — функция 2.

Зона конфликта: по месту — слой данных и вычислений (база, очередь фоновых задач, пул соединений, кэш); по времени — пики импорта и отчётов в последние 3 дня месяца; по условию — крупные арендаторы с требованиями BYOK, локализации и повышенного SLA.

5. Идеальный конечный результат (ИКР) ФАЗА 2

ИКР-1 (функциональный): Каждый арендатор получает ровно тот уровень изоляции, который ему нужен и за который он платит, при общей инфраструктуре и одном коде, без роста затрат на эксплуатацию.

ИКР-2 (ресурсный): Используя только существующие ресурсы (единую кодовую базу, оркестратор, шифрование, метрики по арендаторам, простаивающие мощности вне пиков), изоляция создаётся программно — квотами, ключами, маршрутизацией и ячейками, а не отдельным железом.

ИКР-3 (предельный — устранение элемента): Развилка «общий или отдельный» исчезает: арендатор получает «виртуально отдельную» среду (собственные ключи, квоты, ячейку), а физически делит с другими только то, что делить безопасно.

6. Ресурсный анализ ФАЗА 3

Тип ресурса	Что доступно	Статус
Вещество (компоненты, данные)	Единая кодовая база; кластер Kubernetes; PostgreSQL; очередь задач; 9 SRE; опыт двух ручных отдельных установок	Отдельные установки не автоматизированы
Поле (нагрузки, ограничения, ключи)	Квоты процессора и ввода-вывода; политики доступа; ключи шифрования; нагрузка по арендаторам	Квоты и ключи на арендатора не используются
Пространство	Возможность размещать арендаторов по ячейкам, схемам базы и регионам; пул воркеров	Не используется
Время	Пики импорта и отчётов в последние 3 дня месяца; ночные окна; циклы продления контрактов	Тяжёлые задачи не разнесены по времени
Информация	Метрики и трассировки с tenant_id, профили нагрузки, тарифы, требования из тендеров, журнал инцидентов	Метрики есть, но не управляют распределением ресурсов
Функция/свойство	Шлюз уже знает арендатора; оркестратор умеет создавать изолированные пространства; крупные клиенты готовы платить за изоляцию	Не используется

7. Генерация решений — 4 параллельные стратегии ФАЗА 4

Ниже — необработанные решения по каждой стратегии до объединения (Фаза 5) и проверки (Фаза 6).

А. Принципы + матрица противоречий

- **№13 Наоборот** (ТП-1, по аналогии с матрицей) — вместо «изолировать физически ценой сложности» изолировать логически и криптографически, а физически разделять только критичное; вместо «дать арендатору свою среду» вынуть тяжёлую нагрузку арендатора из общего пути.
- **№28 Замена механической схемы** (ТП-1 и ТП-2) — заменить физическое разделение оборудования программной изоляцией: квоты, ключи шифрования, сетевые политики, отдельные пулы.
- **№35 Изменение параметров** (ТП-1 и ТП-2) — изоляция как параметр арендатора: режим изоляции, квоты, ключи и регион задаются конфигурацией.
- **№10 Предварительное действие** (ТП-1 и ТП-2) — заранее назначить арендаторам ячейки и квоты, подготовить «прогретые» шаблоны окружений.
- **№1 Сегментация** (ТП-2) — сегментировать арендаторов по уровням изоляции (общий пул, выделенная схема или база, выделенная ячейка) и по ячейкам.
- **№3 Местное качество** (дополнительно) — разные режимы для разных арендаторов и компонентов.
- **№15 Динамичность** (дополнительно) — перенос арендатора между уровнями и ячейками без простоя; динамические квоты.
- **№24 Посредник** (дополнительно) — маршрутизатор арендаторов и обязательный слой доступа к данным.
- **№7 Вложение** (дополнительно) — арендаторы внутри ячейки, ячейки внутри региона.
- **№2 Извлечение** (дополнительно) — вынести тяжёлые задачи в отдельные пулы воркеров.
- **№19 Периодическое действие** (дополнительно) — пакетные задачи в окна низкой нагрузки.

В. Су-поле + 76 стандартов

Модель: S1 (общий ресурс платформы: база, воркеры) — F (нагрузка и доступ) → S2 (арендатор). Поле не разделено: действие одного арендатора (нагрузка или ошибка доступа) через общий S1 передаётся другим.

- **Стандарт 2.1.1** — ввести S3, поглощающий вредное воздействие между арендаторами: лимитеры, взвешенные очереди, «переборки» (отдельные пулы и ячейки).
- **Стандарт 1.2.7** — разделить S2 на части, взаимодействующие независимо: ячейки арендаторов.
- **Стандарт 1.2.5** — заменить плохо управляемое поле («кто успел») управляемым: планировщик с приоритетами и справедливыми долями.
- **Стандарт 3.1.1** — перейти к двойной системе: общая плоскость управления и изолированные плоскости данных.
- **Стандарт 3.1.6** — сделать систему динамичной и адаптивной: перенос арендатора между уровнями и ячейками по нагрузке и тарифу.
- **Стандарт 5.1.2** — ввести «вещество» временно: временные выделенные воркеры под тяжёлый импорт, исчезающие после задачи.
- **Стандарты 4.1.1 и 5.2.1** — обнаруживать нарушение изоляции по уже имеющимся полям: метки tenant_id в трассировках и метриках; синтетический арендатор-«канарейка».

С. Аналогии физических эффектов

- **Экранирование и гальваническая развязка** — изоляция без физического разделения проводов; аналог — криптографическая изоляция: ключ на арендатора.
- **Водонепроницаемые переборки судна** — пробоина заливает один отсек, а не весь корабль; аналог — ячейки арендаторов.
- **Демпфер и ограничитель тока (предохранитель)** — аналог квот и лимитеров на арендатора.

D. ARIZ-85V

Не потребовалось — стратегии А-С дали достаточное покрытие решения; переход к Фазе 5 без полного цикла ARIZ.

Принципы разделения противоречивых свойств

- **По времени:** тяжёлые операции — вне пиков или во временных выделенных воркерах; ограничения ресурсов включаются на время пика.
- **По месту:** плоскость управления общая, плоскость данных — по арендаторам и ячейкам, ключи — на арендатора.
- **По условию:** уровень изоляции зависит от тарифа и профиля риска: малым и средним — общий пул с квотами, крупным — выделенная схема или база и свои ключи, регулируемым — выделенная ячейка.
- **Целое/часть:** платформа — целое (единый код и конвейер релизов), арендаторы и ячейки — части со своими границами.

8. Синтез решений ФАЗА 5

1. Ячейки арендаторов

Платформа делится на независимые ячейки (каждая со своим стеком приложений и базой), в каждой не более 40 арендаторов; маршрутизатор направляет арендатора в его ячейку; релизы выкатываются поэтапно по ячейкам, рост — добавлением ячеек. Блок отказа — одна ячейка: при 8 ячейках затрагивается ≤12,5% арендаторов вместо 100%. Принципы №1, №7; стандарты 1.2.7, 2.1.1.

2. Гарантированные доли ресурсов

Квоты и справедливое планирование на арендатора: лимит соединений с базой (не более 8% пула), взвешенные очереди задач по тарифам, отдельный пул воркеров для тяжёлых задач (импорт, отчёты) с ограничением и приоритетами, окна для пакетных операций. Принципы №13, №2, №19, №35; стандарты 1.2.5, 2.1.1.

3. Уровни изоляции как конфигурация

Изоляция — параметр арендатора при единой кодовой базе и одном конвейере релизов: малые и средние — общий пул; крупные — отдельная схема или база внутри общей ячейки со своим пулом соединений и ключами (уровень «bridge»); регулируемые — выделенная ячейка или регион (уровень «silo»). Арендатора можно перенести между уровнями без простоя. Принципы №35, №1, №15; стандарты 3.1.1, 3.1.6.

4. Криптографическая изоляция и защита от ошибок доступа

Данные шифруются ключом арендатора (envelope encryption; BYOK для крупных); слой доступа к данным сам подставляет tenant_id, так что запрос без него невозможен; политики строкового уровня в базе как вторая линия защиты; синтетический арендатор-«канарейка» непрерывно проверяет, что чужие данные недоступны. Принципы №13, №28, №24; стандарты 4.1.1, 5.2.1.

5. Шардирование воркеров с перемешиванием (shuffle sharding)

Каждому арендатору назначается уникальная комбинация из 3 воркеров из 12 в ячейке; «ядовитый» арендатор выводит из строя только совпадающие комбинации: полностью недоступным остаётся ≈0,45% арендаторов вместо всех. Принципы №1, №10; стандарт 2.1.1.

9. Адверсарная проверка ФАЗА 6

Каждое решение проверено «скептиком» (технический директор, руководитель SRE, руководитель информационной безопасности, финансовый директор) по 4 критериям: логика, экономика, реализуемость, побочные эффекты.

Решение	Статус	Реализуемость	Трудоёмкость	Ключевые риски
1. Ячейки арендаторов	С ДОРАБОТКОЙ	3,5/5	Высокая (≈14 чел.-мес.)	Рост фиксированных расходов (каждая ячейка — минимальный стек); сложность маршрутизации, биллинга и кросс-арендаторной аналитики; перенос арендаторов между ячейками; нужна автоматизация платформы
2. Гарантированные доли ресурсов	ПОДТВЕРЖДЕНО	4,5/5	Низкая (≈3 чел.-мес.)	Неверные квоты дают «искусственные» отказы законным пикам — нужны адаптивные лимиты и прозрачные метрики; не защищает от ошибок доступа
3. Уровни изоляции как конфигурация	С ДОРАБОТКОЙ	3,5/5	Средняя-высокая (≈8 чел.-мес.)	Рост тестовой матрицы (режимы x версии), риск «зоопарка» конфигураций; миграция без простоя; тарифы и SLA по уровням
4. Криптоизоляция и защита от ошибок доступа	С ДОРАБОТКОЙ	3,5/5	Средняя (≈6 чел.-мес.)	Управление ключами (ротация, отзыв), влияние на индексы и поиск при шифровании на уровне приложения, сложность BYOK; рефакторинг всех запросов под слой доступа
5. Shuffle sharding	С ДОРАБОТКОЙ	3/5	Средняя (≈3 чел.-мес.)	Сложнее отладка маршрутизации; работает, пока «ядовитый» запрос не проникает дальше воркеров (например, в общую базу); не решает проблему общей БД

10. Ранжирование по составному критерию ФАЗА 7

Composite = Инновационность×0,30 + Реализуемость×0,30 + Стоимость×0,20 + Универсальность×0,20

№1 · Гарантированные доли ресурсов — 0.74

Инновационность 0.40 · Реализуемость 0.90 · Стоимость 0.90 · Универсальность 0.85

№2 · Уровни изоляции как конфигурация — 0.71

Инновационность 0.70 · Реализуемость 0.70 · Стоимость 0.60 · Универсальность 0.85

№3 · Криптоизоляция и защита от ошибок доступа — 0.69

Инновационность 0.60 · Реализуемость 0.70 · Стоимость 0.65 · Универсальность 0.85

№4 · Ячейки арендаторов — 0.64

Инновационность 0.75 · Реализуемость 0.55 · Стоимость 0.35 · Универсальность 0.90

№5 · Shuffle sharding — 0.62

Инновационность 0.80 · Реализуемость 0.50 · Стоимость 0.60 · Универсальность 0.55

Рекомендуемая связка по горизонтам. Быстрый эффект (1–2 месяца): гарантированные доли ресурсов и защита от ошибок доступа. Средний срок (3–5 месяцев): уровни изоляции как конфигурация и криптоизоляция для крупных. Стратегически: ячейки и shuffle sharding — при росте свыше ≈600 арендаторов или требованиях SLA 99,95%, когда фиксированные расходы ячеек распределяются на большее число арендаторов.

11. Анализ эволюции и системный оператор ФАЗА 8

	Прошлое	Настоящее	Будущее
Надсистема	Установка продукта у каждого клиента	Общая SaaS-платформа для арендаторов разных размеров и требований	Платформа с политиками изоляции как продуктом: клиент выбирает уровень изоляции и региона
Система	Отдельная установка на арендатора	Общий кластер и общая база → конфликт экономики и изоляции	Ячейки арендаторов и уровни изоляции при одной кодовой базе
Подсистема	Физическое разделение серверов	Логическое разделение условием tenant_id	Криптографическая изоляция, квоты и автоматические проверки изоляции

Закон эволюции: переход от физического разделения к программному и полевому (квоты, ключи, политики); переход на микроуровень (ячейки, отдельные схемы и ключи на арендатора); рост управляемости и динамичности (арендатор переносится между уровнями по тарифу и нагрузке).

Следующее вероятное новое противоречие: «гибкость изоляции по арендатору ↔ простота и предсказуемость платформы»: чем больше режимов и параметров изоляции, тем сложнее тестировать, поддерживать и объяснять поведение системы; выросла бы стоимость эксплуатации и риск ошибок конфигурации.

12. Образец расчётов для архитектурного ревью

Методический образец: значения стоимости и нагрузки приняты для иллюстрации. В реальном проекте их берут из биллинга облака, метрик и нагрузочных тестов и проверяют экспериментом.

12.1. Стоимость режимов изоляции

Режим	Что изолировано	Стоимость на арендатора, €/мес	Для 30 крупных, млн €/мес
Pool (общий пул, квоты, RLS)	Нагрузка (квоты), данные (логически)	≈13,75 тыс. (среднее по сети)	—
Bridge (схема или база в общей ячейке, свой пул соединений, ключи)	Данные, соединения, ключи	+≈23 тыс. к базовой стоимости (≈37 тыс. в сумме)	≈0,7 (30 × 23 тыс.)
Silo (отдельный стек)	Всё, включая вычисления и сеть	≈180 тыс.	≈5,4 (30 × 180 тыс.)

Silo для всех: 320 × 180 тыс. € = 57,6 млн €/мес при выручке 31,05 млн € (290 × 45 тыс. + 30 × 600 тыс.) — экономически невозможно

Bridge вместо Silo для 30 крупных: 5,4 / 0,69 ≈ 7,8 раза дешевле

Доля инфраструктуры сейчас: 4,4 / 31,05 ≈ 14,2% выручки

12.2. Радиус поражения

Ячейки: 320 арендаторов / 8 ячеек = 40 арендаторов в ячейке → затронуто 40 / 320 = 12,5% (вместо 100%)

Shuffle sharding: каждому арендатору 3 воркера из 12: $C(12, 3) = 220$ комбинаций

Полностью недоступен арендатор с той же комбинацией: $1 / 220 \approx 0,45\%$; в ячейке из 40 арендаторов ожидается 39 / 220 ≈ 0,18 такого арендатора (вместо 39)

Частично затронуты (делят хотя бы один воркер): $1 - C(9, 3) / C(12, 3) = 1 - 84 / 220 \approx 61,8\%$; они теряют 1–2 из 3 воркеров и остаются доступными

12.3. Квоты и справедливые доли

Пул соединений БД 240; лимит арендатора 8% = 19 соединений; «шумный сосед» сейчас занимает до 70% пула = 168 соединений

С лимитом остальным всегда доступно не менее 92% пула

Веса очередей: малые и средние — 1, крупные — 4; при конкуренции крупные в сумме гарантированно получают $30 \cdot 4 / (290 \cdot 1 + 30 \cdot 4) = 120 / 410 \approx 29,3\%$ мощности воркеров; на одного крупного — $4 / 410 \approx 0,98\%$, на одного малого — $1 / 410 \approx 0,24\%$

12.4. Ограничения модели

- Не учтены коррелированные отказы общих компонентов (общая база, брокер, сеть, ключевое хранилище), нагрузка на систему ключей при шифровании на уровне арендатора и стоимость кросс-арендаторной аналитики.
- Стоимость режимов изоляции, число воркеров и доли пула приняты для иллюстрации; результат чувствителен к ним и подтверждается нагрузочными и chaos-тестами.

13. Экономика и целевые KPI (оценочная модель)

Показатель	Сейчас	Цель
Инциденты «шумного соседа»	9 в квартал (≈55 мин)	≤2 в квартал
p95 API при пиковых нагрузках	2,1 с (в норме 250 мс)	≤400 мс
Радиус поражения инцидента (доля арендаторов)	100%	≤25% после квот и пулов; ≤12,5% после ячеек
Межарендаторные утечки	1 инцидент за год	0; автоматическая проверка «канарейкой» для 100% API
Выделение изолированного контура крупному арендатору	≈6 недель, вручную	≤3 дней
Тендеры с требованием изоляции	0 из 3 выиграно	≥2 из 3 сопоставимых
Инфраструктура и эксплуатация	4,4 млн ₽/мес (14,2% выручки)	≈5,7 млн ₽/мес (≈17% выручки)

Сценарий	Результат, млн ₽/мес	Комментарий
Статус-кво	потери ≈4,6	Инциденты ≈1,0 (кредиты ≈0,54 + разбор ≈0,45) + упущенная маржа по проигранным сделкам ≈2,45 (3 × 14 млн ₽ в год × 70%) + ожидаемая потеря при продлении ≈1,15 (риск 25% выручки под риском 6,6 млн ₽/мес × 70%)
А. Отдельные контуры для всех	неосуществимо	Инфраструктура 57,6 при выручке 31,1
Б. Отдельные стеки только для 30 крупных	≈−0,8 относительно статус-кво	Инфраструктура ≈8,4 (+4,0); выигрыш: 2 сделки ≈1,63 + защита продлений ≈1,15 + меньше инцидентов ≈0,4; шум для малых арендаторов не решён; нагрузка на SRE от 30 ручных окружений не оценена
В. Гибрид: квоты + уровни изоляции + криптоизоляция	≈+0,6 ... +2,3 относительно статус-кво	Дополнительные расходы на инфраструктуру ≈1,3 (bridge ≈0,7 + пул для тяжёлых задач ≈0,35 + ключи ≈0,15 + квоты и наблюдаемость ≈0,1); выигрыш в зависимости от числа выигранных сделок и продлений

Статья	Оценка
Осторожный случай: инциденты –60% (≈0,6), 1 сделка (≈0,82), защита 10% продлений (≈0,46), минус расходы ≈1,3	≈+0,58 млн ₽/мес
Оптимистичный случай: инциденты –80% (≈0,8), 2 сделки (≈1,63), защита 25% продлений (≈1,15), минус расходы ≈1,3	≈+2,3 млн ₽/мес
Разовые затраты: 34 чел.-мес. × ≈0,3 млн ₽ (≈10,2) + инструменты (≈2,0)	≈12,2 млн ₽
Срок окупаемости	≈5 месяцев (оптимистичный случай) ... ≈21 месяц (осторожный)
Не включено в расчёт	Защита остальных 19 из 30 крупных клиентов, репутационный эффект, ускорение продаж корпоративным клиентам

Допущения модели (для иллюстрации): маржа с дополнительной выручки 70%; стоимость инженера ≈2 500 ₽/час, 60 часов на инцидент; стоимость работы команды ≈0,3 млн ₽ в человеко-месяц; выручка под риском при продлении — 11 крупных арендаторов по 600 тыс. ₽ в месяц. Широкий диапазон окупаемости показывает реальную неопределённость: она зависит не от техники, а от числа выигранных и удержанных крупных клиентов. Для реальной платформы модель нужно пересчитать по собственным данным.

14. Дорожная карта внедрения

Этап	Действие	Срок	Эффект
1	Измерения: метрики и трассировки по tenant_id, профили нагрузки, реестр инцидентов, карта требований тендеров, каталог общих ресурсов	2 недели	Базовая линия и целевые метрики
2	Гарантированные доли ресурсов: лимиты соединений, взвешенные очереди, отдельный пул для тяжёлых задач	4–6 недель	Снижение инцидентов «шумного соседа»
3	Защита от ошибок доступа: слой доступа с обязательным tenant_id, политики строкового уровня, «канарейка»	6–8 недель, параллельно	Исключение межарендаторных утечек
4	Уровни изоляции как конфигурация: схема или база на арендатора, ключи по арендатору (KMS), перенос без простоя	10–14 недель	Изоляция для крупных без отдельных стеков
5	Пилот BYOK и выделенной изоляции для 2–3 крупных клиентов	4 недели после этапа 4	Подтверждение спроса и цены
6	Ячейки и shuffle sharding — оценка при росте свыше ≈600 арендаторов или SLA 99,95%	по готовности	Ограничение радиуса поражения

15. Как использовать подход в архитектурной работе

Образец ADR (запись архитектурного решения)

Раздел	Содержание
Название	ADR-021: изоляция арендатора как конфигурация (pool, bridge, silo) при единой кодовой базе
Контекст	320 арендаторов на общей базе; 9 инцидентов «шумного соседа» в квартал; 11 из 30 крупных клиентов требуют выделенную изоляцию; отдельные стеки для всех стоят 57,6 млн ₽/мес
Решение	Изоляция — параметр арендатора: pool для малых, bridge для крупных, silo для регулируемых; один конвейер релизов; перенос между режимами без простоя
Альтернативы	Отдельные стеки для всех (отвергнуто: экономика); только квоты (частично: не закрывает требования безопасности); форк кода для крупных (отвергнуто: стоимость поддержки)
Последствия	Плюс: изоляция по цене, гибкость тарифов. Минус: рост тестовой матрицы, необходимость автоматизации переносов и управления ключами
Как проверим	«Канарейка» и тесты изоляции в CI; нагрузочный тест шумного арендатора: p95 остальных ≤400 мс; выделение bridge-контура ≤3 дней

Чек-лист изоляции арендатора

Слой	Как изолируем	Как проверяем
Данные	Обязательный слой доступа с tenant_id, политики строкового уровня; для крупных — отдельная схема или база	Автотесты на утечки, «канарейка»
Вычисления	Квоты, взвешенные очереди, отдельные пулы для тяжёлых задач	Нагрузочный тест «шумного соседа»
Ключи	Шифрование ключом арендатора, ротация, отзыв; BYOK для крупных	Учения по отзыву ключа
Сеть	Сетевые политики между ячейками и уровнями, отдельные точки входа для silo	Тесты сегментации
Наблюдаемость	Метрики, логи и трассировки с tenant_id, отдельные панели и оповещения по арендатору	Проверка, что данные одного арендатора не видны другому
Релизы и миграции	Поэтапная выкатка по ячейкам, перенос арендатора без простоя	Канареечные релизы, откат
Локализация и аудит	Размещение данных по регионам, журнал доступа	Периодический аудит

Вопросы для архитектурного воркшопа

- Какие ресурсы делят арендаторы (соединения, очереди, кэш, диск, сеть) и какой арендатор может занять каждый из них?
- Что произойдёт при ошибке в фильтре tenant_id? Сколько независимых линий защиты?
- Какие требования к изоляции реально диктуют клиенты и регуляторы, а какие сформулированы «на всякий случай»?
- Какая часть арендаторов платит за изоляцию и сколько она готова платить?
- Можно ли перенести арендатора между режимами без простоя и сколько это стоит по времени SRE?
- Какова тестовая матрица при трёх режимах изоляции и как её автоматизировать?
- Как измерять «шум» и когда система должна автоматически ограничить арендатора?

Типичные ошибки: считать изоляцию по tenant_id достаточной; давать «выделенные» стеки вручную и без автоматизации; форкать код под крупного клиента; вводить квоты без наблюдаемости и прозрачных метрик; выбирать режим изоляции по громкости клиента, а не по требованиям и экономике. **Ограничения метода:** ТРИЗ-анализ структурирует поиск решений и даёт направления; выбор нужно подтверждать измерениями, нагрузочными и chaos-тестами и оценкой безопасности.

16. Приложение: инструменты ТРИЗ, использованные в анализе

Матрица противоречий (по аналогии для ПО)

40 принципов изобретательства

Су-польный анализ

76 стандартов

Принципы разделения (время/место/условие/целое-часть)

Аналогии физических эффектов

ИКР (3 уровня)

Ресурсный чек-лист (6 типов)

Системный оператор

Законы эволюции систем

Адверсарная верификация

Многокритериальное ранжирование