



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Платёжная система с нуля: PCI DSS, токенизация, мульти-PSP

Как мы строим платёжный контур: score под SAQ A по PCI DSS v4.0.1, authorize/capture, идемпоте...

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Бизнес принимает оплату через одного PSP с жёсткой привязкой к его API: рост комиссии, даунтайм в пик или блокировка аккаунта останавливают продажи, а ошибка в платёжной логике стоит живых денег. Мы разворачиваем платёжный контур с минимальным PCI-score (SAQ A), абстрактным слоем над несколькими PSP, идемпотентностью и ежедневной сверкой — деньги не списываются дважды и не тер...

Почему это важно бизнесу

- Даунтайм единственного PSP = остановка выручки; fallback-провайдер возвращает продажи за минуты, а не за недели переинтеграции
- Двойные списания и потерянные вебхуки — прямые убытки и чарджбэки; идемпотентность и сверка закрывают это архитектурно
- Полный аудит PCI DSS Level 1 — миллионы рублей в год; грамотная токенизация сводит обязательства к самооценке SAQ A
- PAN на своих серверах втягивает весь контур в карточный score: инцидент — штрафы эквайера и потеря приёма карт
- Расхождение с PSP, найденное через месяц, почти невозможно; ежедневная reconciliation ловит его за сутки



Ключевые параметры реализации

4.0.1

актуальная версия стандарта: v4.0
отозвана, future-dated требования
обязательны с 31.03.2025

PCI SSC, PCI DSS v4.0.1

SAQ A

целевой уровень отчётности: PAN
вводится только в iframe/SDK PSP,
наши серверы карт не видят

PCI SSC, SAQ A v4.0.1 r1

60 с

TTL Redis-лока на Idempotency-Key
(SET NX EX) против параллельных
дублей; ответ кэшируем 24 ч

наш стандарт

7 дней

максимум окна authorize→capture:
дальше банк снимает hold, поэтому
храним capture_deadline

наш стандарт

TLS 1.2+

минимум для передачи карточных
данных и вебхуков; early TLS и
слабые шифры запрещены

PCI DSS v4.0.1, req 4.2.1

12 мес

глубина хранения журналов аудита
платёжных операций, последние 3
месяца — доступны сразу

PCI DSS v4.0.1, req 10.5.1



Минимизация scope PCI DSS: токенизация через iframe PSP

Что настраиваем

Checkout интернет-магазина/маркетплейса: фронт оплаты + backend API, карты не покидают домен провайдера

Как мы это делаем

- 1 Встраиваем платёжную форму iframe/JS SDK провайдера: карта вводится в контуре PSP, к нам на backend уходит только одноразовый payment_token
- 2 Для сохранённых карт храним recurring_token зашифрованным (ключ в отдельном KMS) — без merchant ID и API-ключа он бесполезен
- 3 Инвентаризуем скрипты платёжной страницы и включаем CSP — с 31.03.2025 это критерий применимости SAQ A по v4.0.1
- 4 Вычищаем PAN из логов и трасс: маскирование на уровне логгера, максимум BIN (6-8 цифр) + последние 4
- 5 Фиксируем соответствие: SAQ A v4.0.1 r1 (январь 2025) + ежеквартальный ASV-скан внешнего периметра

РЕЗУЛЬТАТ

Серверы клиента остаются вне карточного scope: вместо полного аудита Level 1 с ежегодным QSA-ассесментом — самооценка SAQ A. Утечка нашей БД не раскрывает ни одного номера карты: в ней только токены, бесполезные без ключей PSP.

КЛЮЧЕВОЙ НЮАНС

С января 2025 SAQ A переиздана (r1): требования 6.4.3/11.6.1 из анкеты убрали, но добавили критерий «сайт не подвержен скрипт-атакам» — без CSP и контроля скриптов право на SAQ A теряется, остаётся SAQ A-EP ил...



Платёжное ядро: authorize/capture, идемпотентность, мульти-PSP

Что настраиваем

Backend платёжного сервиса: конечный автомат платежа, Redis, абстрактный gateway на двух и более провайдеров

Как мы это делаем

- 1 Строим state machine статусов: pending → authorized → captured / voided / refunded / failed; переход валидируется до вызова PSP
- 2 Idempotency-Key обязателен на всех мутирующих эндпоинтах: Redis-лок SET NX EX 60, готовый ответ кэшируется на 86400 с
- 3 Абстрактный интерфейс PaymentGateway (authorize/capture/void/refund) — каждый PSP отдельным адаптером; суммы только Decimal в minor units
- 4 Роутинг: основной провайдер + fallback при недоступности, правила по сумме заказа и комиссии
- 5 Retry только идемпотентных вызовов: экспоненциальный backoff base 1 с, max 30 с, 3 попытки, jitter 10%

РЕЗУЛЬТАТ

Двойное списание исключено архитектурно: повтор запроса после сетевого таймаута возвращает сохранённый ответ, а не создаёт вторую транзакцию. Переключение на резервный PSP не требует правок бизнес-кода — меняется только конфиг роутинга.

КЛЮЧЕВОЙ НЮАНС

Capture контролируем по дедлайну: авторизация живёт около 7 дней, дальше банк снимает hold и capture падает. Поэтому capture_deadline лежит в записи платежа, за сутки до истечения — алерт.



Вебхуки, ежедневная сверка и аудит-контур

Что настраиваем

Интеграционный слой PSP → backend: webhook endpoint, reconciliation-джоба, append-only аудит-лог

Как мы это делаем

- 1 Webhook: проверяем HMAC-подпись, дедуплицируем по `webhook_id`, отвечаем 200 менее чем за 200 мс, обработку уносим в транзакцию асинхронно
- 2 Ежедневная reconciliation: сравниваем свои платежи с settlement-отчётом PSP по `transaction_id` и суммам, расхождение больше 1 ₽ — алерт финотделу
- 3 Аудит-лог append-only: DELETE/UPDATE запрещены на уровне БД, копия уходит в S3-архив на 5 лет
- 4 Чарджбэк из вебхука: автоматическая блокировка суммы на балансе продавца + тикет в поддержку
- 5 Курс валют фиксируем в момент авторизации (кэш котировок 15 мин) — capture идёт строго по зафиксированному курсу

РЕЗУЛЬТАТ

Потерянный или дублированный вебхук больше не искажает финансовую картину: статус доедет через сверку максимум за сутки. Финотдел видит расхождения с провайдером на следующее утро, а не при закрытии месяца.

КЛЮЧЕВОЙ НЮАНС

Вебхук — не источник истины: он теряется, дублируется и опаздывает. Истина — ежедневная сверка с отчётом PSP; вебхук лишь ускоряет обновление статуса платежа.

Подводные камни

✗ **Retry без Idempotency-Key**

Повтор после таймаута списывает деньги дважды. Ретраим только вызовы с идемпотентным ключом; создание платежа без ключа не ретраится никогда

✗ **PAN в логах и трассах**

Один access-лог с номером карты втягивает сервер в полный PCI-score. Маскирование на уровне логгера: максимум BIN(6-8) + последние 4 цифры

✗ **Вебхук как источник истины**

Вебхуки теряются и дублируются — статусы расходятся с деньгами. Обязательна ежедневная reconciliation по settlement-отчёту PSP

✗ **float для денежных сумм**

Двоичная арифметика теряет копейки на конвертации и комиссиях. Только Decimal и целые minor units, режим округления фиксирован (ROUND_HALF_UP)

✗ **SAQ A «по привычке»**

С 31.03.2025 у SAQ A новый критерий применимости: сайт не подвержен скрипт-атакам. Без CSP и контроля скриптов — уже SAQ A-EP или SAQ D

✗ **Жёсткая привязка к одному PSP**

Рост комиссии или даунтайм провайдера останавливает продажи. Абстрактный gateway + протестированный fallback переключают за часы, не недели

✗ **Медленный webhook endpoint**

PSP считает endpoint мёртвым и прекращает доставку. Отвечаем 200 сразу (< 200 мс), тяжёлую обработку уносим в очередь

✗ **Capture после истечения hold**

Просроченный capture падает — товар отгружен, деньги не списаны. Храним capture_deadline в платеже и алертом за сутки до истечения

Как правильно

МИНИМУМ

- Карта только в iframe/SDK PSP — серверы вне карточного scope, цель SAQ A
- Idempotency-Key на всех мутирующих платёжных эндпоинтах
- TLS 1.2+, маскирование PAN в логах, ключи PSP prod/staging отдельно
- Ежедневная сверка с settlement-отчётом PSP, алерт при расхождении

НОРМАЛЬНО

- Второй PSP за абстрактным gateway, fallback протестирован на staging
- Вебхуки: HMAC-подпись, дедуп по webhook_id, асинхронная обработка
- Фрод-контур: velocity-лимиты, гео-проверки, 3-D Secure по скорингу
- Append-only аудит-лог, журналы 12 мес (3 мес — оперативный доступ)

ХОРОШО

- CSP и инвентаризация скриптов платёжной страницы (6.4.3/11.6.1)
- ASV-сканы ежеквартально, пентест ежегодно, крипто-инвентарь по 12.3.3
- ML-скоринг фрода поверх правил: пороги allow / 3-DS / block
- Роутинг PSP по сумме и комиссии + нагрузочный тест 1000 tx/мин



Чек-лист самопроверки

- | | |
|--|--|
| ☐ Карточные данные вводятся только в iframe/SDK PSP и не касаются наших серверов и логов? | ☐ Capture контролируется по сроку жизни авторизации (capture_deadline с алертом)? |
| ☐ Все мутирующие платёжные эндпоинты требуют Idempotency-Key и переживают повтор запроса без двойного списания? | ☐ SAQ актуализирована под критерии 2025 года: сайт защищён от скрипт-атак (CSP, контроль скриптов)? |
| ☐ Вебхуки проверяются по HMAC-подписи и дедуплицируются по webhook_id? | ☐ Курс валюты фиксируется при авторизации и не пересчитывается на capture? |
| ☐ Ежедневная сверка с settlement-отчётом PSP настроена, алерт при расхождении больше 1 ₽? | ☐ Аудит-лог append-only, production-ключи PSP отделены от staging? |
| ☐ Fallback-PSP подключён и реально протестирован на staging? | ☐ Фрод-лимиты включены со старта (velocity, гео, порог 3-DS) и мониторится refund rate? |

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Проектируем и внедряем платёжный контур под ключ: токенизация, authorize/capture, мульти-PSP gateway
- Сводим score к SAQ A: iframe-интеграция, CSP, маскирование PAN, пакет соответствия PCI DSS v4.0.1
- Строим вебхуки, reconciliation и append-only аудит под требования регулятора и финотдела
- Настраиваем фрод-правила, мониторинг refund rate и аномалий сумм, алерты в Telegram
- Проводим нагрузочное тестирование и go-live чеклист из 30 пунктов перед боевым запуском

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** PCI DSS v4.0.1 — Requirements and Testing Procedures (pcisecuritystandards.org — v4.0.1)
- 02** SAQ A v4.0.1 r1 — критерии применимости для e-commerce (pcisecuritystandards.org — 2025)
- 03** PCI DSS req 6.4.3 / 11.6.1 — контроль скриптов платёжной страницы (pcisecuritystandards.org — v4.0.1)
- 04** Redis SET (NX/EX) — атомарные блокировки идемпотентности (redis.io — 7.x)
- 05** Наш шаблон платёжного ядра: PaymentGateway ABC + idempotency middleware (itfresh.ru — 2026)
- 06** Наш go-live чеклист платёжного контура (30 пунктов) (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.

