

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: https://doi.org/10.60797/IRJ.2026.169.55 EDN: RTRGLR

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОБЕСПЕЧЕНИЯ ИДЕМПОТЕНТНОСТИ В КРЕДИТНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Научная статья

Мезенцева Е.М.[1, *], Масагутов К.А.[2], Тимофеев А.В.[3]
[1] ORCID : 0000-0001-8274-1600;
[2] ORCID : 0009-0008-7068-8820;
[3] ORCID : 0000-0002-5585-9774;
[1, 2, 3] Самарский национальный исследовательский университет имени академика С. П. Королёва, Самара, Российская Федерация

* Корреспондирующий автор (katya-mem[at]mail.ru)

Предложена: 03.04.2026; Принята: 11.06.2026; Опубликована: 17.07.2026

Аннотация

В статье рассматривается проблема обеспечения идемпотентности в кредитных информационных системах, функционирующих в условиях распределённой обработки данных, повторных сетевых вызовов и асинхронного обмена сообщениями. Актуальность исследования обусловлена тем, что повторная обработка кредитной операции может приводить к дублированию заявок, повторному резервированию лимитов, некорректной фиксации статусов и искажению финансово значимых данных. Объектом исследования являются процессы обработки кредитных операций в распределённых информационных системах, а предметом — методы обеспечения идемпотентности и их способность предотвращать повторную обработку на техническом, интеграционном и доменном уровнях.

Цель исследования состоит в теоретическом и эмпирическом сравнении трёх подходов: idempotency key, бизнес-ключа операции и контроля состояния бизнес-объекта. Методология включает формализацию уровней идемпотентности, многокритериальную экспертную оценку, анализ чувствительности весов и модельный эксперимент. По результатам оценки выяснилось, что метод контроля состояния и ожидаемого бизнес-эффекта показал наибольшую эффективность в рамках принятой модели оценки, так как он больше всего направлен на предотвращение повторных финансово важных результатов. В то же время показано, что для кредитных информационных систем наиболее уместно использовать несколько подходов вместе. Это помогает защититься от дублирования на уровне запроса, операции и итогового бизнес-результата. Практическая ценность работы состоит в том, что она обосновывает выбор методов обеспечения идемпотентности в кредитных информационных системах.

Ключевые слова: повторная обработка операций, бизнес-ключ операции, межканальные повторы, распределённые транзакции, финансово значимые данные, межсервисное взаимодействие, контроль состояния бизнес-объекта, повторный бизнес-эффект.

A COMPARATIVE ANALYSIS OF METHODS FOR ENSURING IDEMPOTENCY IN CREDIT INFORMATION SYSTEMS

Research article

Mezenceva E.M.[1, *], Masagutov K.A.[2], Timofeev A.V.[3]
[1] ORCID : 0000-0001-8274-1600;
[2] ORCID : 0009-0008-7068-8820;
[3] ORCID : 0000-0002-5585-9774;
[1, 2, 3] Samara National Research University, Samara, Russian Federation

* Corresponding author (katya-mem[at]mail.ru)

Suggested: 03.04.2026; Accepted: 11.06.2026; Published: 17.07.2026

Abstract

The article examines the problem of ensuring idempotency in credit information systems operating in environments characterised by distributed data processing, repeated network calls and asynchronous messaging. The relevance of the research is due to the fact that the repeated processing of a credit transaction can lead to the duplication of applications, the repeated reservation of limits, the incorrect recording of statuses and the distortion of financially significant data. The object of the study is the processing of credit transactions in distributed information systems and the subject is the methods for ensuring idempotence and their ability to prevent re-processing at the technical, integration and domain levels.

The aim of the research is to conduct a theoretical and empirical comparison of three approaches: the idempotency key, the business operation key and the business object state check. The methodology includes the formalisation of idempotency levels, multicriteria expert assessment, sensitivity analysis of weights, and a model experiment. The results of the evaluation showed that the method of monitoring the state and the expected business effect proved to be the most effective within the adopted evaluation model, as it is primarily aimed at preventing the duplication of financially significant results. At the same time, it has been demonstrated that for credit information systems, it is most appropriate to use several approaches in combination. This helps to protect against duplication at the level of the request, the operation and the final business outcome. The practical

value of this work lies in the fact that it substantiates the choice of methods for ensuring idempotence in credit information systems.

Keywords: reprocessing of transactions, transaction business key, cross-channel duplicates, distributed transactions, financially significant data, cross-service interaction, business object state monitoring, repeated business effect.

Введение

Кредитные информационные системы работают с распределённой обработкой данных и внешними интеграциями. В такой архитектуре повторное выполнение одной и той же операции является типовой проблемой: запрос может быть отправлен повторно в случае потери ответа, хотя операция уже обработана.

В обычной информационной системе дубликат часто рассматривается как техническая ошибка обработки. В кредитном домене его значение шире: повторное создание заявки, повторное резервирование лимита или повторная фиксация договора способны изменить финансовое состояние клиента и кредитной организации. Поэтому задача сводится не только к обнаружению одинакового входного сообщения, но и к защите результата, который должен возникнуть ровно один раз.

Практика использования idempotency key показывает, что безопасный повтор запроса после сбоя связи является обязательным элементом современных API [1], [2]. Однако этот механизм решает только часть проблемы, поскольку распределённая система содержит не один, а несколько уровней повторов: сетевой, интеграционный, прикладной и бизнес-уровень. Для асинхронного обмена характерна семантика at-least-once delivery, при которой обработчик обязан корректно воспринимать повторную доставку сообщения [3], [4].

Объектом исследования являются процессы обработки кредитных операций в распределённых информационных системах. Предметом исследования выступают методы обеспечения идемпотентности и их способность предотвращать повторную обработку запросов, сообщений и доменных переходов состояния. Цель статьи — сравнить методы обеспечения идемпотентности в кредитных информационных системах и определить условия их рационального сочетания. Для достижения цели решаются следующие задачи: раскрыть специфику повторной обработки операций в кредитном домене; выделить уровни идемпотентности в распределённой архитектуре; сравнить подходы на основе idempotency key, бизнес-ключа операции и контроля состояния бизнес-объекта; выполнить многокритериальную оценку методов; проверить устойчивость выводов с помощью анализа чувствительности и модельного эксперимента. Научная новизна работы состоит в разделении уровней идемпотентности, введении критерия повторного бизнес-эффекта и дополнении экспертной модели результатами имитационной проверки. Практическая значимость заключается в формировании рекомендаций для проектирования кредитных сервисов, в которых недопустимо повторное выполнение финансово значимой операции.

Методы и принципы исследования

2.1. Теоретические основания идемпотентности в распределённых кредитных системах

В классическом понимании операция является идемпотентной, если её повторное выполнение не изменяет результат сверх первого применения. Для HTTP это означает, что несколько одинаковых запросов должны приводить к тому же наблюдаемому состоянию ресурса, что и один запрос [5]. В прикладных финансовых системах такое определение требует расширения, поскольку одинаковость запроса и одинаковость бизнес-операции не совпадают.

Теоретически команду можно представить как пару $c = (a, p)$, где a — тип действия, а p — набор параметров. Состояние доменного объекта обозначим через s , а переход состояния — через функцию $\delta(\mathrm{s}, \mathrm{c})$. . На уровне бизнес-эффекта требуется, чтобы для повторной или эквивалентной команды c' выполнялось условие $\delta\left(\delta(s,c), c'\right) = \delta(s,c)$. Иными словами, второй вызов не должен создавать новый финансовый результат, если первый уже сформировал требуемое состояние.

Для кредитного домена важно различать три вида совпадения. Техническое совпадение означает, что повторился тот же запрос с тем же идентификатором. Операционное совпадение фиксирует, что разные сообщения описывают одну и ту же смысловую операцию: например, одну заявку клиента на один кредитный продукт. Результативное совпадение связано с состоянием объекта: если лимит уже зарезервирован или договор активирован, повторная команда не должна порождать дополнительное изменение.

Распределённые транзакции усложняют применение этого правила. В микросервисной архитектуре данные кредитной заявки, расчёта лимита, договора и уведомлений могут находиться в разных сервисах. Глобальная атомарность часто заменяется локальными транзакциями, обменом событиями и компенсационными действиями [6]. Поэтому в практическом проектировании более корректно говорить не об exactly-once delivery, а об exactly-once effect: сообщение может быть получено несколько раз, но бизнес-эффект должен быть зафиксирован однократно.

С теоретической точки зрения идемпотентность опирается на инварианты предметной области. Инвариант кредитной заявки может запрещать повторную регистрацию активной заявки с тем же внешним номером; инвариант договора — повторную активацию уже активированного договора; инвариант лимита — повторное резервирование одной и той же суммы по одной операции. Такие правила ближе к доменной модели, чем к транспортному протоколу, и требуют проектирования конечного автомата состояний [7], [8].

2.2. Источники повторов и модель риска

Повторы в кредитных информационных системах возникают не только из-за ошибок пользователя. Наиболее типичны четыре группы причин: потеря ответа после успешной обработки, повторная доставка сообщения брокером, параллельное поступление заявки из разных каналов и повторная команда после частичного завершения процесса. Во всех случаях входные данные могут отличаться, хотя экономический смысл операции остаётся тем же.

В данной работе повторный бизнес-эффект понимается как нежелательное повторное изменение финансово значимого состояния: создание второго договора вместо одного, двойное резервирование лимита, повторная отправка

команды на выдачу кредита или повторная фиксация статуса, запускающего дальнейший процесс. Такой эффект отличается от технического дубля: технический дубль может быть безопасным, если система возвращает прежний результат и не меняет доменное состояние.

Для анализа принята многоуровневая модель риска. На первом уровне оценивается способность метода распознать повторный входной запрос. На втором — способность связать разные технические сообщения с одной бизнес-операцией. На третьем — способность проверить, что требуемое состояние уже достигнуто. Чем выше уровень, тем больше метод зависит от предметной логики, но тем точнее он защищает финансово значимый результат.

2.3. Методы обеспечения идемпотентности

Для сравнительного анализа выделены три базовых метода: idempotency key, бизнес-ключ операции и контроль состояния с учётом ожидаемого бизнес-эффекта.

Метод idempotency key основан на сохранении уникального идентификатора запроса вместе с результатом его обработки. Если запрос с таким ключом уже обрабатывался, система не выполняет операцию повторно, а возвращает ранее зафиксированный результат [1]. Этот подход удобен для синхронных API-вызовов и позволяет выявлять дубликат до выполнения основной бизнес-логики [2]. Недостаток метода состоит в том, что защита зависит от генерации, передачи и хранения ключа [9]. Ограничение проявляется в межканальных сценариях: новая техническая команда может иметь другой ключ, но соответствовать уже обработанной кредитной операции.

Бизнес-ключ операции строится из доменных атрибутов, позволяющих идентифицировать смысл операции. Для кредитной заявки таким ключом могут быть идентификатор клиента, продукт, внешний номер заявки, канал-партнёр, сумма, валюта и дата операционного окна. В отличие от технического ключа, бизнес-ключ устойчив к изменению транспорта и помогает распознавать дубликаты, пришедшие из мобильного приложения, веб-интерфейса, операторского контура или партнёрской системы [10], [11]. Основной риск бизнес-ключа связан с выбором состава атрибутов. Слишком узкий ключ пропускает реальные дубли, поскольку разные сообщения не будут сопоставлены. Слишком широкий ключ приводит к ложной блокировке допустимых операций: например, клиент действительно подал две похожие заявки в разные моменты времени. Поэтому бизнес-ключ должен проектироваться как часть предметной модели, а не как произвольный технический индекс.

Контроль состояния и ожидаемого бизнес-эффекта использует конечный автомат объекта. Он основан на контроле состояния бизнес-объекта и ожидаемого бизнес-эффекта. Повторная обработка не допускается, если система уже достигла нужного результата. Такой метод ближе всего к доменным инвариантам: он проверяет не совпадение сообщений, а допустимость перехода состояния. Преимущество этого метода в том, что он тесно связан с бизнес-логикой, но для него нужно иметь чёткую модель состояний и правил перехода между ними [7]. Недостаток контроля состояния состоит в большей сложности внедрения. Требуется формальная модель состояний, перечень допустимых переходов, правила обработки конкурентных команд и защита от гонок на уровне транзакций или версий объекта.

2.4. Методика сравнительной оценки

Сравнение методов выполнено по шести критериям: С1 — точность выявления дублей; С2 — устойчивость к межканальным повторам; С3 — сложность внедрения; С4 — независимость от предметной логики; С5 — надёжность предотвращения повторного бизнес-эффекта; С6 — удобство сопровождения. Оценка каждого метода задаётся по шкале от 1 до 5, где 1 означает низкое соответствие критерию, а 5 — высокое.

Интегральная оценка метода определяется как взвешенная сумма частных оценок:

$$I_m = \sum_{i=1}^{6} w_i \cdot x_{mi}, \quad \sum_{i=1}^{6} w_i = 1, \quad x_{mi} \in \{1,2,3,4,5\}.$$

Здесь $I(m)$ — итоговая оценка метода m, w_i — вес критерия i, x_{mi} — экспертная оценка метода по критерию. Веса заданы с учётом риска кредитного процесса: наибольший вес получает С5, поскольку именно повторный бизнес-эффект напрямую влияет на финансовую корректность операции. Критерии С3, С4 и С6 отражают эксплуатационные свойства и имеют меньшую значимость, чем предотвращение ошибочного результата. Веса и обоснования представлены в таблице 1.

Таблица 1 - Весовые коэффициенты критериев оценки

DOI: https://doi.org/10.60797/IRJ.2026.169.55.1

Критерий	Содержание критерия	Вес	Обоснование
С1	Точность выявления дублей	0,20	Влияет на пропуск дубликатов и на ошибочную блокировку новых операций
С2	Устойчивость к межканальным повторам	0,15	Одна кредитная операция может прийти через разные каналы обслуживания
С3	Сложность внедрения	0,10	Определяет стоимость разработки, тестирования и сопровождения
С4	Независимость от	0,10	Чем выше

Критерий	Содержание критерия	Вес	Обоснование
	предметной логики		независимость, тем проще повторное использование метода
C5	Предотвращение повторного бизнес-эффекта	0,30	Главный риск кредитного домена связан с повторным финансовым результатом
C6	Удобство сопровождения	0,15	Важны прозрачность правил, аудит и возможность изменения бизнес-процесса

Представленные в таблице 1 критерии и весовые коэффициенты используются далее для расчёта интегральной оценки каждого метода обеспечения идемпотентности.

Для метода idempotency key интегральная оценка рассчитывается следующим образом:

$$Q_1 = 0,20 \cdot 4 + 0,15 \cdot 2 + 0,10 \cdot 5 + 0,10 \cdot 5 + 0,30 \cdot 3 + 0,15 \cdot 4 = 3,60 .$$

Для метода бизнес-ключа операции:

$$Q_2 = 0,20 \cdot 4 + 0,15 \cdot 5 + 0,10 \cdot 3 + 0,10 \cdot 3 + 0,30 \cdot 4 + 0,15 \cdot 3 = 3,80$$

Для метода контроля состояния и ожидаемого бизнес-эффекта:

$$Q_3 = 0,20 \cdot 5 + 0,15 \cdot 4 + 0,10 \cdot 2 + 0,10 \cdot 2 + 0,30 \cdot 5 + 0,15 \cdot 3 = 3,95$$

Результаты сравнительной оценки представлены в таблице 2.

Таблица 2 - Многокритериальная оценка методов обеспечения идемпотентности

DOI: https://doi.org/10.60797/IRJ.2026.169.55.2

Метод	Критерии						Оценка
	C1	C2	C3	C4	C5	C6	
Idempotency key	4	2	5	5	3	4	3,60
Бизнес-ключ операции	4	5	3	3	4	3	3,80
Контроль состояния и бизнес-эффекта	5	4	2	2	5	3	3,95

Разница не означает неэффективность технического ключа; она показывает, что один уровень защиты не закрывает все классы повторов. Полученные значения отражают результат при базовом распределении весов критериев. Однако итоговая оценка может зависеть от того, какие свойства метода считаются более значимыми: техническая простота, межканальное распознавание дублей или предотвращение повторного бизнес-эффекта. Поэтому далее проведён анализ чувствительности, позволяющий проверить, сохраняется ли вывод при изменении весовых коэффициентов. Результаты анализа представлены в таблице 3.

Таблица 3 - Анализ чувствительности результатов к изменению весов

DOI: https://doi.org/10.60797/IRJ.2026.169.55.3

Сценарий весов	Idempotency key	Бизнес-ключ	Контроль состояния	Интерпретация
Базовый	3,60	3,80	3,95	Контроль состояния имеет наибольшую оценку
Технический акцент	3,50	3,95	3,95	Бизнес-ключ и контроль состояния дают равный результат

Сценарий весов	Idempotency key	Бизнес-ключ	Контроль состояния	Интерпретация
Акцент на бизнес-эффекте	3,40	3,80	4,10	Контроль состояния усиливает преимущество

Анализ чувствительности показывает, что изменение весов влияет на соотношение методов, но не отменяет необходимости их комбинированного применения. При техническом акценте бизнес-ключ операции приближается к контролю состояния, поскольку лучше распознаёт межканальные повторы. При повышении значимости критерия, связанного с предотвращением повторного бизнес-эффекта, преимущество контроля состояния становится более выраженным. Следовательно, выбор метода должен определяться не только удобством реализации, но и уровнем риска, который возникает при повторной обработке кредитной операции.

Для наглядного сопоставления методов результаты многокритериальной оценки дополнительно представлены в графической форме. Лепестковая диаграмма позволяет сравнить сильные и слабые стороны каждого подхода по критериям и показать, по каким параметрам методы имеют наибольшие различия. Графическое представление результатов приведено на рисунке 1.

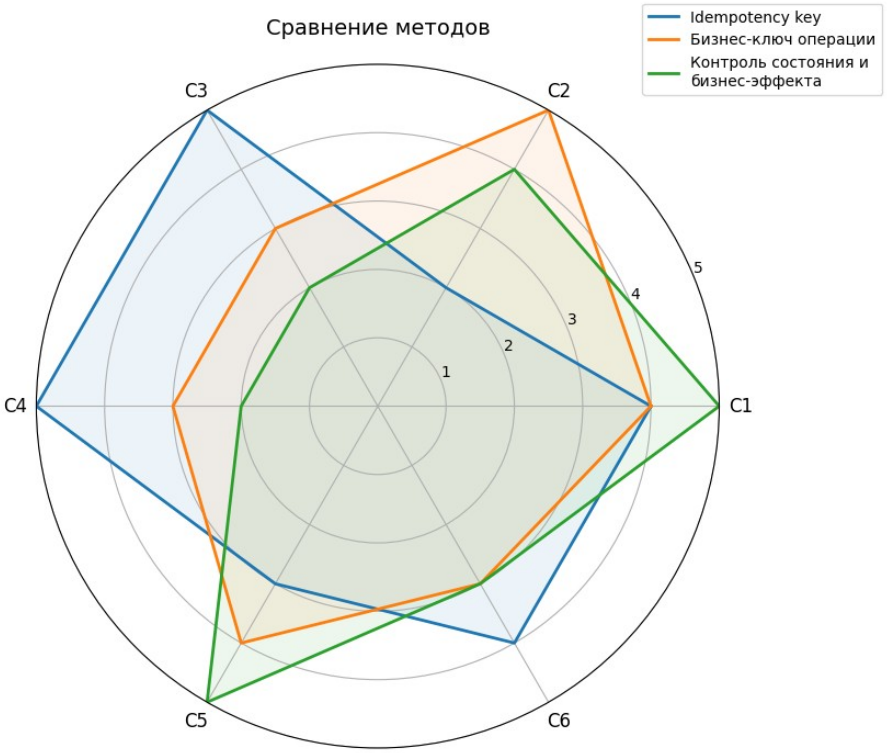

Рисунок 1 - Лепестковая диаграмма сравнительной оценки методов обеспечения идемпотентности
DOI: https://doi.org/10.60797/IRJ.2026.169.55.4

2.5. Эмпирическая проверка на модельном наборе данных

Для дополнения экспертной оценки проведена имитационная проверка. Использован модельный журнал входящих событий, отражающий типовые сценарии кредитной системы: первичная подача заявки, повторный HTTP-запрос после потери ответа, межканальный дубль, повторная доставка интеграционного сообщения и команда, поступившая после достижения целевого состояния. Данные не являются промышленными логами и не содержат персональной информации; их назначение — воспроизводимо сравнить методы при контролируемых условиях.

Каждое событие содержало технический идентификатор запроса, канал поступления, бизнес-атрибуты операции, состояние объекта до обработки и признак допустимости новой операции. Для оценки качества выявления повторных операций использовались показатели точности и полноты. Точность отражает долю корректно выявленных дублей среди всех операций, которые метод классифицировал как дубли. Полнота показывает, какая часть всех фактических дублей была обнаружена методом.

$$\text{Precision} = \frac{TP}{TP+FP}, \ \text{Recall} = \frac{TP}{TP+FN},$$

где *TP* — количество корректно выявленных дублей; *FP* — количество операций, ошибочно отнесённых к дублям; *FN* — количество дублей, которые метод не выявил.

Для эмпирической проверки предложенных методов был сформирован модельный журнал событий, включающий как корректные операции, так и различные типы повторов. Состав журнала отражает основные сценарии, характерные для распределённой кредитной системы. Структура модельного набора данных представлена в таблице 4.

Таблица 4 - Состав модельного журнала событий

DOI: https://doi.org/10.60797/IRJ.2026.169.55.5

Тип события	Количество	Назначение в эксперименте
Исходные корректные операции	10 000	Базовый поток новых заявок и действий по кредитному процессу
Повторный HTTP-запрос с тем же idempotency key	600	Сценарий потери ответа после успешной обработки
Межканальный повтор с новым техническим ключом	350	Одна операция поступает из другого канала обслуживания
Повторная доставка интеграционного сообщения	250	Семантика at-least-once delivery в асинхронном обмене
Повтор после достижения целевого состояния	50	Команда приходит после регистрации заявки, активации договора или резервирования лимита
Допустимые похожие операции	200	Новые операции с близкими атрибутами, которые нельзя блокировать как дубли

На основе сформированного модельного журнала была выполнена сравнительная проверка методов. Для каждого подхода рассчитаны значения метрик. Дополнительно зафиксировано количество случаев, в которых повторная обработка приводила к повторному бизнес-эффекту. Результаты проверки представлены в таблице 5.

Таблица 5 - Результаты эмпирической проверки методов

DOI: https://doi.org/10.60797/IRJ.2026.169.55.6

Метод	TP	FN	FP	Precision, %	Recall, %	Повторный бизнес-эффект, случаев
Idempotency key	600	650	3	99,5	48,0	413
Бизнес-ключ операции	1 018	232	41	96,1	81,4	126
Контроль состояния и бизнес-эффекта	1 124	126	16	98,6	89,9	38
Комбинированная схема	1 239	11	9	99,3	99,1	4

Результаты показывают, что idempotency key почти не создаёт ложных блокировок, но обнаруживает только 48,0% дублей, так как межканальные и событийные повторы имеют другие технические параметры. Бизнес-ключ повышает полноту до 81,4%, однако увеличивает число FP: часть допустимых похожих операций ошибочно классифицируется как повтор. Контроль состояния достигает recall 89,9% и резко снижает число повторных бизнес-эффектов, потому что решение принимается по доменному состоянию, а не только по признакам входного сообщения.

Наилучший результат показала комбинированная схема. Она не заменяет три базовых метода, а последовательно применяет их: сначала проверяется idempotency key, затем бизнес-ключ, после чего выполняется контроль допустимости перехода состояния. В модельном журнале такая схема оставила 4 случая повторного бизнес-эффекта против 413 у одного idempotency key. Это подтверждает тезис о необходимости многоуровневой защиты в кредитных информационных системах.

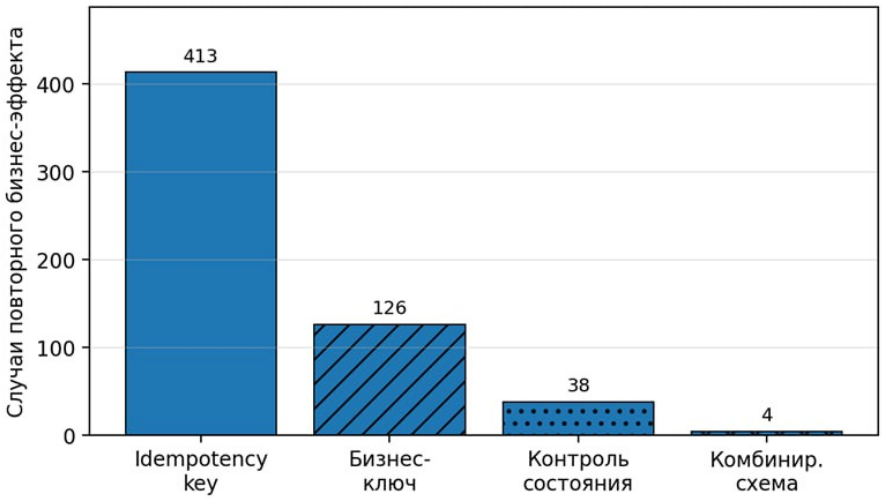

Рисунок 2 - Остаточные повторные бизнес-эффекты по результатам модельного эксперимента
DOI: https://doi.org/10.60797/IRJ.2026.169.55.7

Основные результаты

Полученные результаты позволяют уточнить архитектурные рекомендации. Idempotency key следует применять на границе внешнего API, где он устраняет повторные запросы клиента и снижает нагрузку на логику. Запись о ключе должна сохраняться атомарно с результатом обработки либо в рамках транзакции, либо через согласованный журнал состояния запроса.

Бизнес-ключ целесообразен на уровне прикладного сервиса и интеграционных сценариев. Его задача — связать технически разные сообщения, относящиеся к одной смысловой операции. В кредитной системе такой ключ должен быть согласован с правилами продукта: например, для заявки наличного кредита, кредитной карты и реструктуризации состав атрибутов может различаться. Поддержка бизнес-ключа требует версионирования правил, так как изменения кредитного процесса меняют критерии эквивалентности операций.

Контроль состояния должен располагаться в доменной модели и фиксировать допустимые переходы объекта. Для заявки это могут быть состояния «создана», «на проверке», «одобрена», «отклонена», «договор оформлен»; для лимита — «свободен», «зарезервирован», «использован», «освобождён». Команда, не меняющая состояние или повторяющая уже достигнутый результат, должна возвращать безопасный ответ без повторного финансового действия.

Практически наиболее устойчивой является схема «технический ключ + бизнес-ключ + состояние». Технический ключ защищает от потери ответа и повторного нажатия кнопки. Бизнес-ключ выявляет дубли, пришедшие по разным каналам. Контроль состояния предотвращает ошибочный результат в ситуациях, когда два первых уровня не дали полной уверенности. Такая комбинация снижает зависимость от одного механизма и повышает объяснимость решения при аудите.

Заключение

В работе проведён сравнительный анализ методов обеспечения идемпотентности в кредитных информационных системах. Теоретическая часть показала, что идемпотентность следует рассматривать не только как свойство HTTP-запроса, но и как свойство бизнес-перехода, сохраняющего инварианты кредитного процесса. Поэтому защита должна охватывать технический уровень, уровень смысловой операции и уровень состояния доменного объекта.

Многокритериальная оценка показала наибольшую интегральную эффективность контроля состояния и ожидаемого бизнес-эффекта. Бизнес-ключ операции оказался наиболее полезен для межканальных повторов, а idempotency key — для ранней обработки повторных API-вызовов. Анализ чувствительности подтвердил, что при изменении весов меняется относительный отрыв методов, но сохраняется вывод о необходимости их сочетания.

Эмпирическая проверка на модельном журнале из 11 450 событий дополнила экспертные выводы количественными данными. Одиночный idempotency key выявил 48,0% дублей, бизнес-ключ — 81,4%, контроль состояния — 89,9%, а комбинированная схема — 99,1%. Число остаточных повторных бизнес-эффектов снизилось с 413 при использовании только idempotency key до 4 при многоуровневой защите.

Для кредитных информационных систем рекомендуется комбинированный подход: на входе API использовать idempotency key, в прикладной логике применять бизнес-ключ операции, а в доменной модели контролировать допустимость перехода состояния. Дальнейшее развитие исследования может быть связано с проверкой модели на промышленных данных, оценкой стоимости ложных блокировок и разработкой адаптивных правил бизнес-ключа для разных кредитных продуктов.

Конфликт интересов

Не указан.

Рецензия

Юнусова В.С., Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань Российская Федерация
DOI: https://doi.org/10.60797/IRJ.2026.169.55.8

Conflict of Interest

None declared.

Review

Yunusova V.S., Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan Russian Federation
DOI: https://doi.org/10.60797/IRJ.2026.169.55.8

Список литературы на английском языке / References in English

1. Idempotent requests // Stripe. — 2025. — URL: https://docs.stripe.com/api/idempotent_requests. (accessed: 29.03.26)
2. Lowery B. Designing robust and predictable APIs with idempotency / B. Lowery, D. Ponnapalli // Designing robust and predictable APIs with idempotency. — 2017. — URL: https://stripe.com/blog/idempotency. (accessed: 29.03.26)
3. Richardson C. Pattern: Idempotent Consumer / C. Richardson // microservices. — 2026. — URL: https://microservices.io/patterns/communication-style/idempotent-consumer.html. (accessed: 29.03.26)
4. Richardson C. Handling duplicate messages using the Idempotent Consumer pattern / C. Richardson // microservices. — 2020. — URL: https://microservices.io/post/microservices/patterns/2020/10/16/idempotent-consumer.html. (accessed: 29.03.26)
5. Fielding R. RFC 9110: HTTP Semantics / R. Fielding, M. Nottingham, J. Reschke // RFC 9110: HTTP Semantics. — 2022. — URL: https://www.rfc-editor.org/rfc/rfc9110.html . (accessed: 30.03.26)
6. Richardson C. Transactional Outbox / C. Richardson // microservices. — 2026. — URL: https://microservices.io/patterns/data/transactional-outbox.html. (accessed: 30.03.26)
7. Hohpe G. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions / G. Hohpe, B. Woolf. — Boston: Addison-Wesley, 2003. — 736 p. (accessed: 30.03.26).
8. Fowler M. Patterns of Enterprise Application Architecture / M. Fowler. — Boston: Addison-Wesley, 2002. — 560 p.
9. Jena J. The Idempotency-Key HTTP Header Field / J. Jena, S. Dalal // The Idempotency-Key HTTP Header Field. — 2025. — URL: https://datatracker.ietf.org/doc/draft-ietf-httpapi-idempotency-key-header. (accessed: 01.04.26)
10. Richardson C. Microservices Patterns: With examples in Java / C. Richardson. — Shelter Island: Manning, 2018. — 520 p.
11. Kleppmann M. Designing Data-Intensive Applications / M. Kleppmann. — O'Reilly Media: O'Reilly Media, 2017. — 616 p.