Микросервис анализа транзакционной активности клиента (Anti-fraud / Scoring)

Обобщенная цель проекта: смоделировать антифрод/скоринговый сервис, который анализирует транзакции клиента, выявляя подозрительные операции и выставляя им “оценку риска” или “фрод-маркер”.

Оглавление

Кузнецов Арсений
email: ababaan365@gmail.com
Phone: +7 (901)701-80-33
Telegram: @cantsmoke

Описание бизнес-задачи (Problem Statement)

Контекст:

Банк ежедневно обрабатывает сотни тысяч транзакций. С ростом числа операций увеличилось кол-во мошеннических действий, которые включают новые/сложные схемы. Текущая система анализа транзакций клиентов недостаточно быстрая и не способна эффективно справляться с современными угрозами.

Проблема:

Банк несет убытки из-за:

1) Потери лояльных клиентов из-за невозможности защитить их от мошенников или из-за ложных блокировок.
2) Увеличения операционных издержек на ручные проверки подозрительных транзакций.
3) Штрафов за нарушение норм AML (Anti-Money Laundering) и KYC (Know Your Customer).

Цель проекта:

Разработать микросервис, который:

1) Снизит потери от мошенников за счет быстрого автоматизированного анализа транзакций в реальном времени.
2) Сократит количество ложных срабатываний.
3) Будет интегрирован с существующими системами.

Стейкхолдеры:

Заинтересованное лицо	Интерес
Бизнес	Удержание клиентов, улучшение сервиса
Блок антифрода	Минимизировать потери от мошенничества
IT-департамент	Простая интеграция сервиса
Блок комплаенс	Соответствовать требованиям регулятора
Клиенты банка	Получить безопасный сервис без ложных блокировок, не терять деньги из-за мошенников

Клиент банка является конечным потребителем проекта, так как от качества антифрод-сервиса зависит безопасность его средств и удобство использования финансовых сервисов. Хотя клиент не участвует в постановке задач, его интересы должны учитываться при проектировании решений (снижение ложных срабатываний, скорость восстановления доступа, защита от потери средств).

Ожидаемые эффекты (KPI/Метрики):

Показатель	Текущий	Цель
Убытки от фродов (за квартал)	10 млн ₽	< 1 млн ₽
Ложные срабатывания (%)	15%	< 3%
Среднее время анализа транзакции (сек)	5 сек	< 1 сек
Кол-во жалоб клиентов	Высокое	Сократить на 60%

Ограничения:

1) Интеграция только с текущей банковской IT-экосистемой.
2) Соблюдать регуляторные требования по AML, KYC и персональным данным.
3) Опираться только на данные, которые уже есть внутри банковской системы.

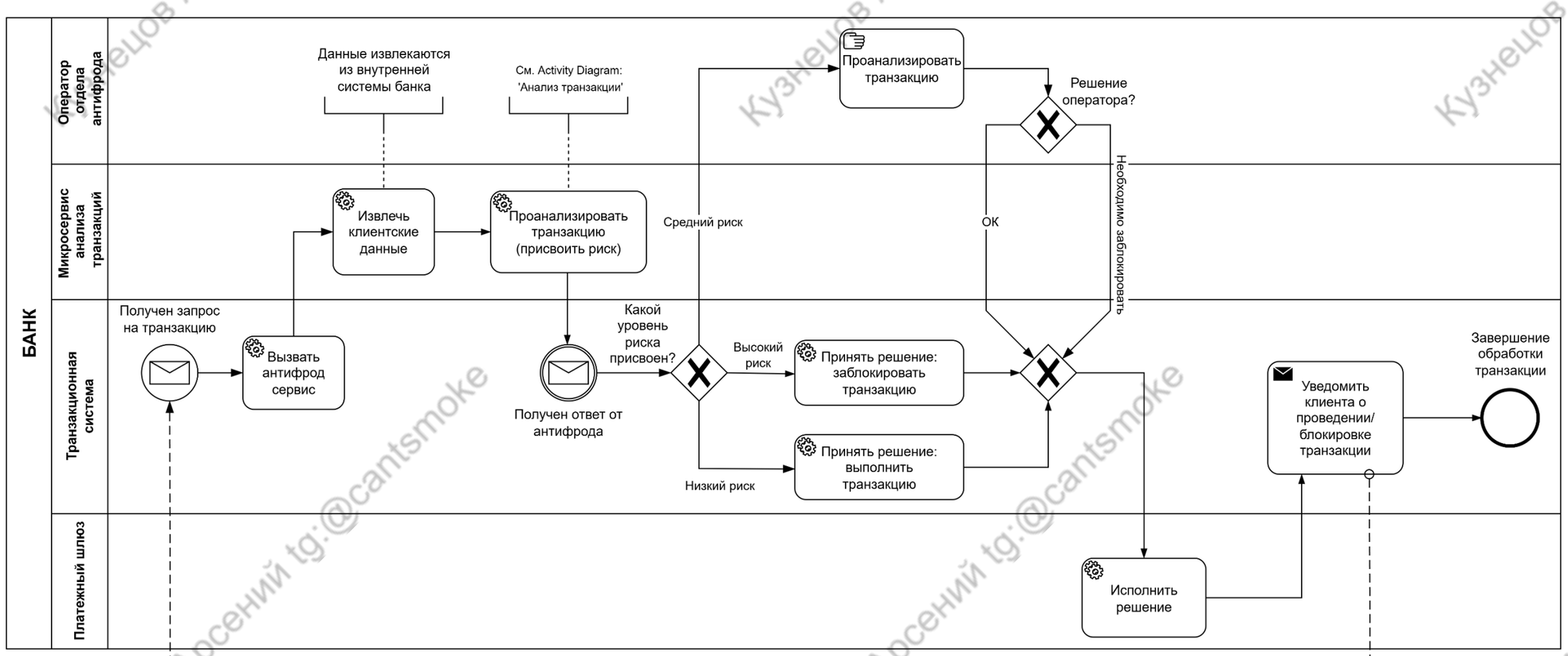

Рисунок 1. BPMN диаграмма бизнес-процесса

Участники процесса:

Пул	Роль	Описание
Банк	Транзакционная система	"Ядро", которое отслеживает все движения денег по счетам клиента, обрабатывает внутренние и внешние переводы, ведёт учёт остатка и т. д.
	Микросервис анализа транзакций	Микросервис, который анализирует транзакции на предмет мошенничества и проводит оценку.
	Платежный шлюз	Интерфейс между внешним миром (магазины, сайты, другие банки, платёжные системы) и внутренней системой банка.
	Оператор отдела антифрода	Сотрудник банка, который в случае среднего уровня риска имеет права на ручной анализ транзакции.
Клиент	Клиент	Клиент банка.

Описание процесса:

1. Клиент инициирует транзакцию.
2. Транзакционная система направляет её в микросервис антифрода для скоринга.
3. Микросервис извлекает клиентские данные (необходимы для анализа).
4. Микросервис оценивает риск: низкий, средний или высокий.
5. При низком или высоком риске транзакционная система сразу принимает решение: проводить транзакцию или нет.
6. При среднем риске операция передается оператору на ручную проверку.
7. Оператор выносит решение о допуске или отклонении транзакции.
8. После решения транзакционной системы или оператора транзакция исполняется или отклоняется, клиент получает уведомление.

Описание задачи “Проанализировать транзакцию”:

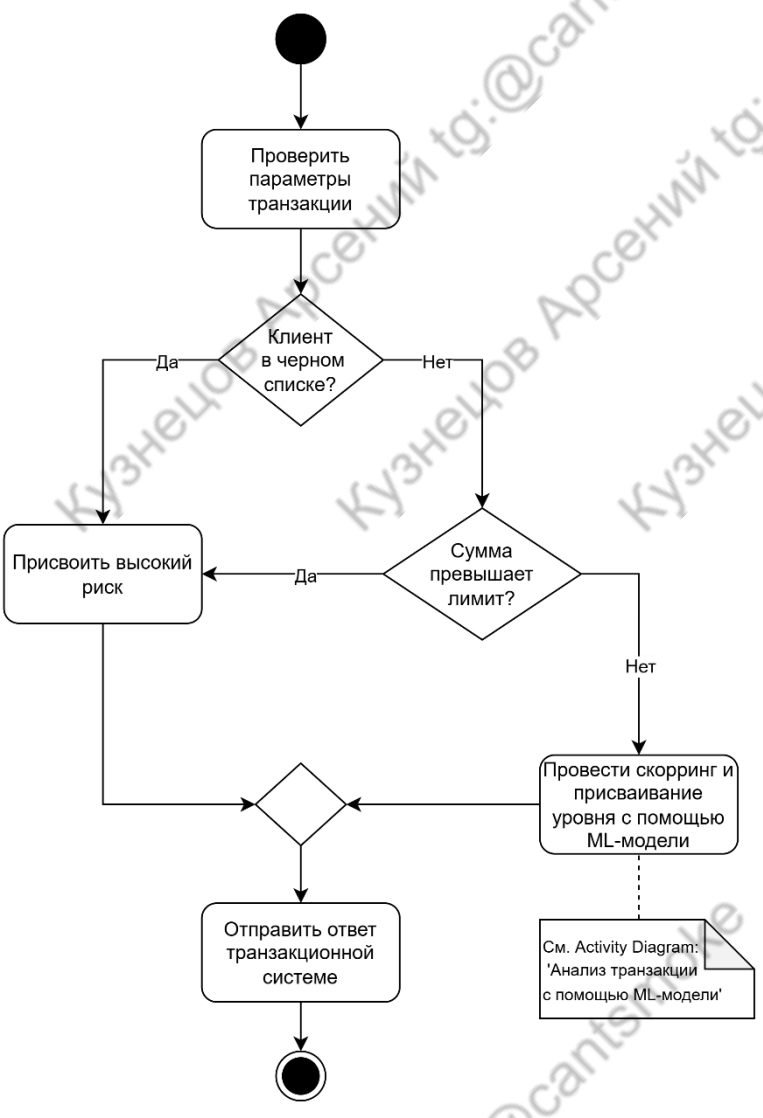

Рисунок 2. UML Activity Diagram "Анализ транзакции"

Описание действия “Провести скорринг и присваивание уровня с помощью ML-модели”:

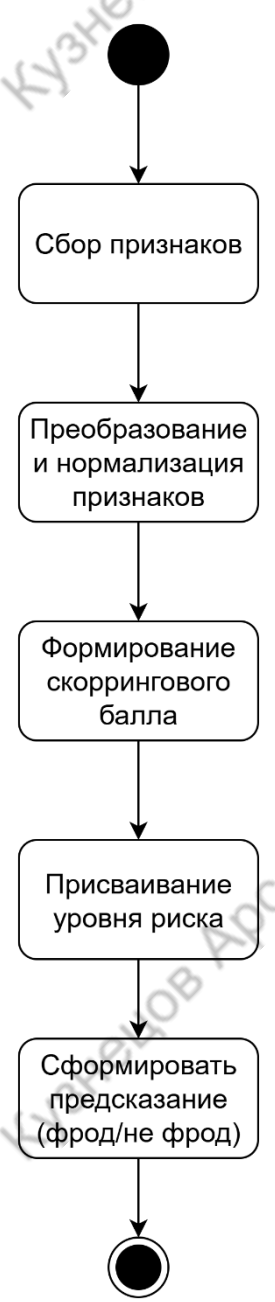

Рисунок 3. UML Activity Diagram "Анализ транзакции с помощью ML-модели"

Сбор признаков: “сырые данные” на входе, такие как отправитель, получатель, сумма, информация об устройстве и т. д., преобразуются в категориальные и числовые признаки.

Преобразование и нормализация признаков: признаки масштабируются, кодируются, обрабатываются пропуски, выбросы.

Формирование скорингового балла: с помощью обученной модели (формула/алгоритм/ML-модель) вычисляется итоговое значение - скор, вероятность или класс.

Присваивание уровня риска: итоговая оценка сравнивается с пороговым значением, и на основе данного сравнения транзакции присваивается уровень риска.

На этом этапе ML-модель выдает итоговое решение - фрод или не фрод - на основе всех собранных и обработанных данных. Это решение она записывает в БД, но не делится им с транзакционной системой.

Описание требований

Функциональные требования к микросервису (FR):

ID	Требование	Критерий приемки / комментарий
FR-1	Микросервис должен принимать входящий HTTP POST-запрос с информацией о транзакции.	Структура запроса: формат JSON. Поля: сумма, валюта, отправитель, получатель, параметры устройства (+ IP-геолокация).
FR-2	Микросервис должен извлекать дополнительную информацию о клиенте из внутренней системы банка.	Данные профиля клиента, история транзакций и т. п.
FR-3	Микросервис должен обрабатывать входные данные, нормализовать их и преобразовывать в признаки для скоринга.	Категориальные, числовые признаки; обработка пропусков, выбросов.
FR-4	Микросервис должен определять уровень риска с использованием ML-модели и набора бизнес-правил.	На выходе категория: высокий, средний, низкий риск.
FR-5	Микросервис должен возвращать результат анализа в формате JSON.	Формат ответа: JSON. Поля: уровень риска, причина (Бизнес правила или ML-модель).
FR-6	Микросервис должен записывать все результаты анализа в хранилище для последующего аудита.	Хранение идентификатора транзакции, входных данных, результатов, времени анализа.
FR-7	Микросервис должен обеспечивать интеграцию с существующей ИТ-экосистемой банка по REST API.	Взаимодействие с транзакционной системой, системой ручной проверки, хранилищем данных.
FR-8	Микросервис должен иметь возможность подписываться на очередь с финальными решениями о проведении транзакций (Kafka).	Для использования сервисом фактических исходов для обучения модели.
FR-9	Микросервис должен сохранять поступающие данные для последующего переобучения модели.	Данные хранятся в базе / хранилище отдельно от транзакций.
FR-10	Микросервис должен предоставлять доступ к актуальным метрикам качества в Monitoring Service для их хранения и визуализации.	Метрики доступны в виде дашбордов в Prometheus, обновляются не реже 1 раза в 10 минут.
FR-11	Микросервис должен предоставлять endpoint, который инициирует загрузку предыдущей версии модели из Model Registry (например, MLflow) и замену активной модели Scoring Engine.	Данные о проведенных откатах хранятся в базе.
FR-12	Микросервис должен самостоятельно предсказывать является ли транзакции фродом/не фродом для последующего сравнения финального решения и предсказания.	Сформированное предсказание хранится в базе.

Нефункциональные требования (NFR):

Категория	Требование
Производительность	Время отклика не более 1 секунды для 95% запросов. Поддержка до 2000 RPS.
Масштабируемость	Микросервис должен поддерживать горизонтальное масштабирование.
Надежность	Доступность не менее 99,95% в месяц. Устойчивость к отказу одной или нескольких нод.
Надежность	В случае критического падения метрик (ROC-AUC < 0.85) микросервиса должна быть предусмотрена возможность отката на предыдущую версию модели по команде, полученной от внешнего клиента.
Безопасность	Обязательное использование HTTPS, авторизация JWT, ограничение по IP доступа.
Интеграция	Синхронная интеграция осуществляется с существующей системой банка через REST API для проверки транзакций в реальном времени.
Интеграция	Асинхронный Consumer очереди (Kafka) осуществляет сбор данных для переобучения и формирования метрик.
Соответствие нормам	Соблюдать требования AML, KYC, GDPR по хранению и защите персональных данных.
Стабильность	В случае сбоя должен поддерживаться механизм повторных попыток / компенсации операций.

Нефункциональные требование к ML-модели (NFR-ML):

ID	Требование
NFR-ML-1	Для обучения модели использовать исключительно внутренние данные банка: история транзакций, профили клиентов, данные о уже выявленных случаях мошенничества.
NFR-ML-2	Для обучения модели обязательно разделение данных на тренировочную, валидационную и тестовую выборки.
NFR-ML-3	Модель должна обеспечивать метрики качества не ниже заданных пороговых: • ROC - AUC > 0.90 • Precision > 0.75 • Recall > 0.80
NFR-ML-4	Модель должна переобучаться с использованием накопленных меток (раз в месяц/ по расписанию).

Функциональные требования к переобучению ML-модели (ML-ST-FR):

ID	Требование	Комментарий
ML-ST-FR-1	Микросервис должен иметь механизм периодического переобучения модели на основе накопленных данных о результатах транзакций.	Переобучение запускается по расписанию или при достижении заданного порога новых данных.
ML-ST-FR-2	Микросервис должен поддерживать конфигурируемый параметр расписания переобучения (например, раз в месяц/квартал/год).	Параметр настраивается через конфигурацию сервиса.
ML-ST-FR-3	Микросервис должен сохранять результаты переобучения, включая версию модели, метрики качества (ROC-AUC, Precision, Recall), дату обучения.	Информация сохраняется в базе / MLflow / аналогичном инструменте.
ML-ST-FR-4	Микросервис должен автоматически использовать новую переобученную модель для анализа новых транзакций.	Предыдущая версия модели должна оставаться доступной для отката.

Нефункциональные требования к переобучению ML-модели (ML-ST-NFR):

Категория	Требование
Обслуживаемость	Микросервис должен поддерживать автоматизированный контроль версий обученных моделей.
Надежность	В случае сбоя переобучения должна сохраняться текущая работоспособная версия модели.

User Story:

ID	User Story	Критерии приемки
US-01	Как транзакционная система банка, я хочу отправлять данные транзакции в антифрод-сервис, чтобы получать оценку риска по каждой операции.	1. API принимает корректный JSON-запрос. 2. В ответе приходит оценка риска (низкий/средний/высокий). 3. Ответ приходит в течение 1 секунды. 4. При среднем уровне риска создается задача на ручную проверку. 5. При высоком уровне риска транзакция немедленно блокируется.

Use Case:

Атрибут	Описание
Название Use Case	Проверка транзакции на факт мошенничества.
Цель	Оценить транзакцию на предмет фрод-риска и вернуть результат.
Актор	Транзакционная система банка.
Описание	Транзакционная система отправляет данные на проверку, получает уровень риска, принимает решение (проводить / блокировать / отправить на ручную проверку).
Триггер события	Поступает запрос на проведение новой транзакции от клиента банка.
Основной поток	1. Система отправляет данные транзакции в антифрод-сервис. 2. Сервис анализирует данные и присваивает уровень риска. 3. Сервис возвращает результат в формате JSON. 4. Транзакционная система принимает решение на основании уровня риска.
Альтернативные потоки	1. При низком уровне риска транзакция проводится. 2. При среднем уровне риска операция передается на ручную проверку оператору. 3. При высоком уровне риска транзакция немедленно отклоняется.
Результат	Транзакционная система знает уровень риска каждой транзакции и понимает, можно ли её проводить.

Decision Table (таблица бизнес-правил принятия решения по транзакции):

Условие	Клиент в черном списке?	Сумма превышает лимит?	ML-модель определяет высокий риск?	ML-модель определяет средний риск?	Решение
Простой фрод - клиент в черном списке	Да	-	-	-	Высокий риск
Простой фрод - сумма	Нет	Да	-	-	Высокий риск
Все простые (первичные) признаки в норме	Нет	Нет	Да	-	Высокий риск
	Нет	Нет	Нет	-	Низкий риск
	Нет	Нет	Нет	Да	Средний риск

Описание архитектуры системы

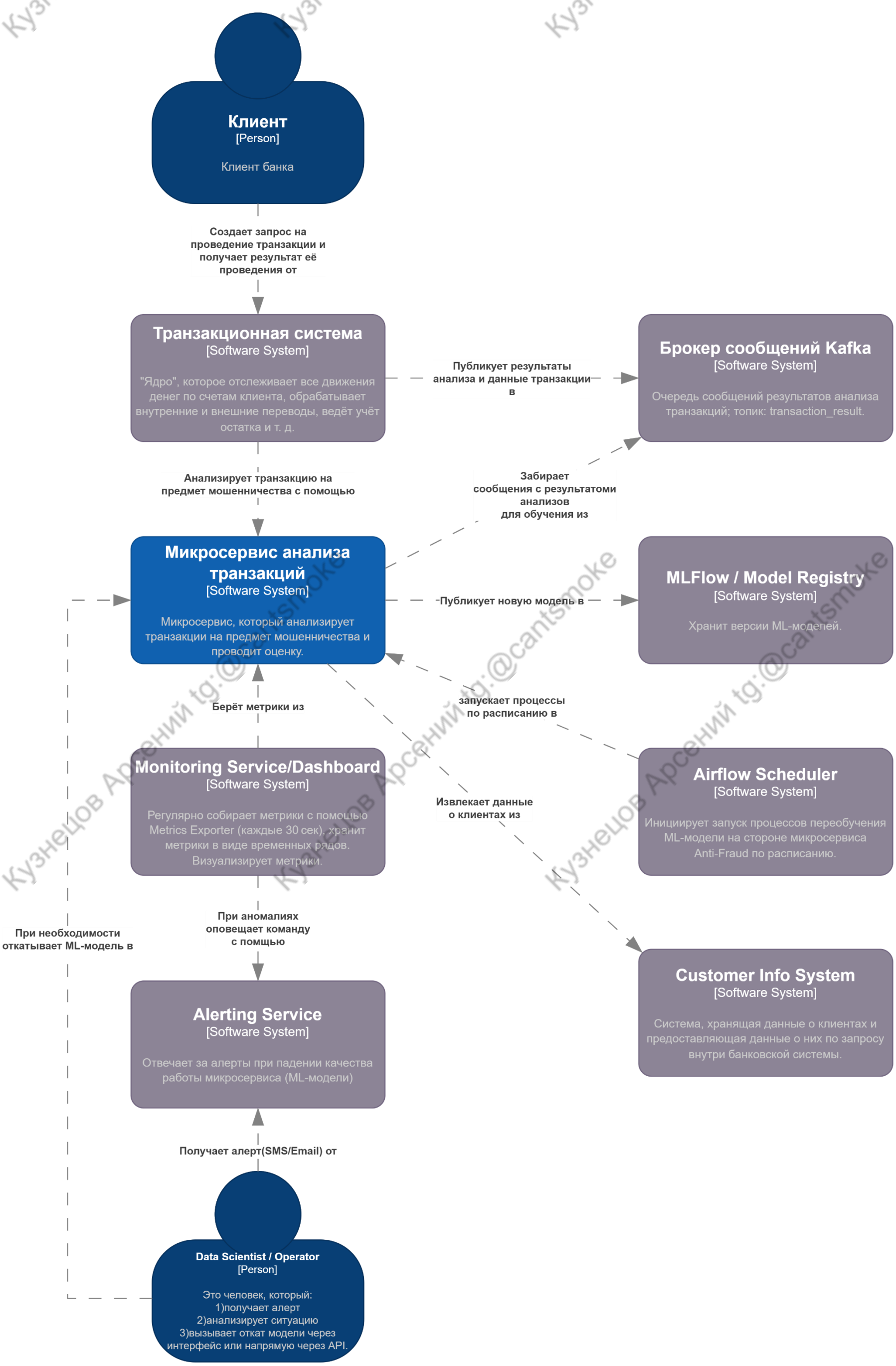

Рисунок 4. Контекстная диаграмма C4 на первом уровне (контекст)

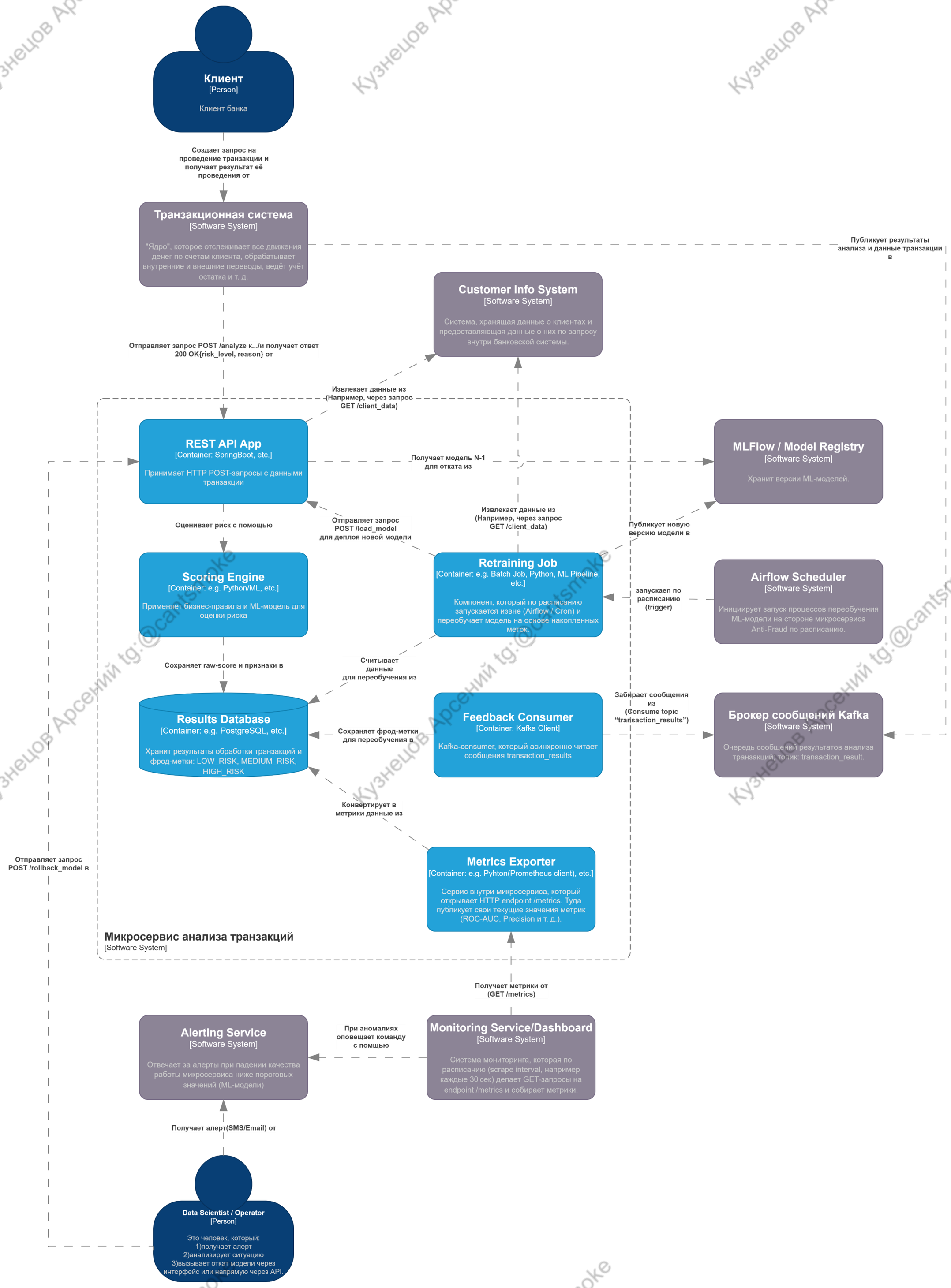

Рисунок 5. Контекстная диаграмма C4 на втором уровне (контейнеры)

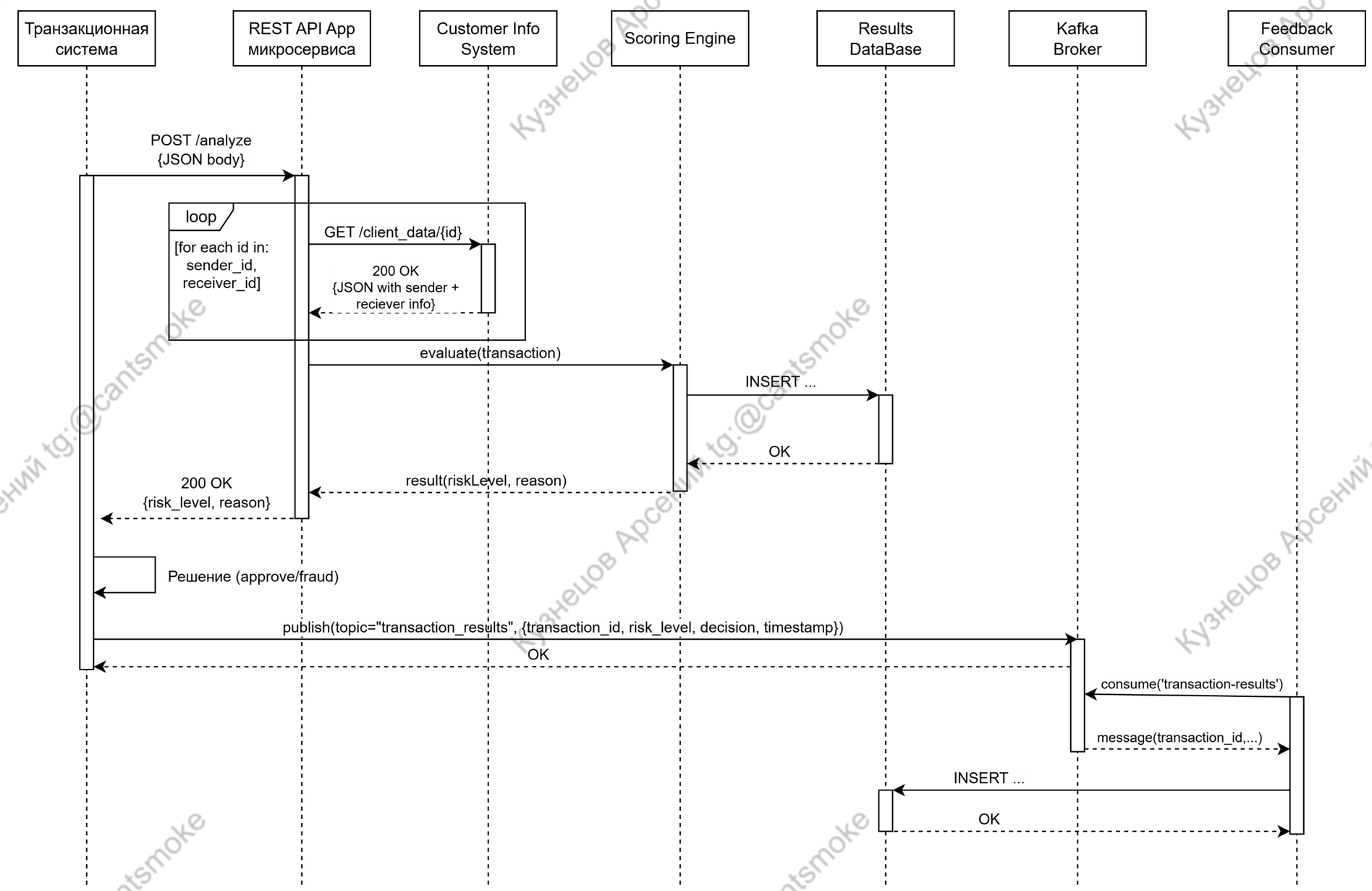

Рисунок 6. UML Sequence Diagram для анализа транзакции (антифрод)

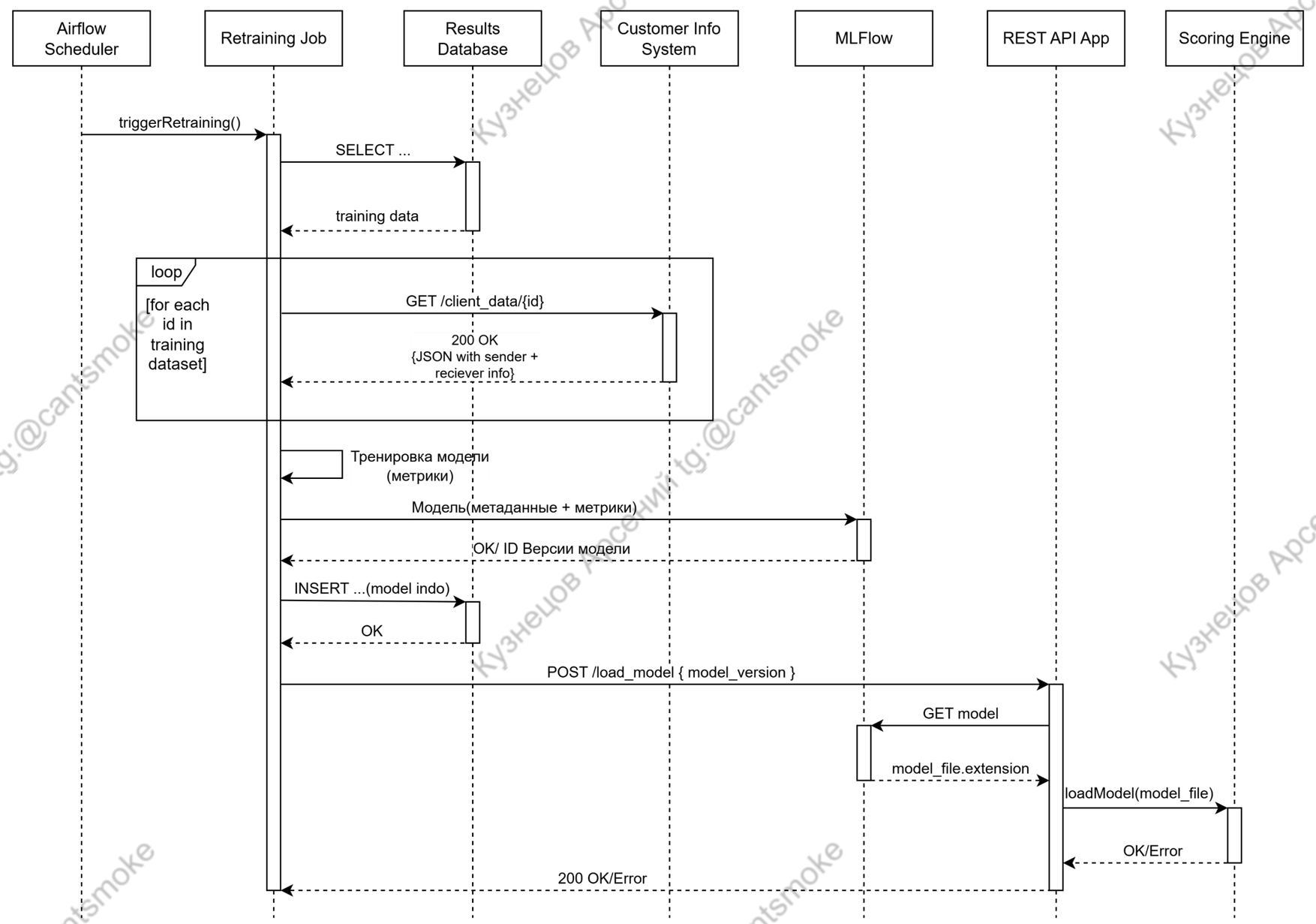

Рисунок 7. UML Sequence Diagram для переобучения модели

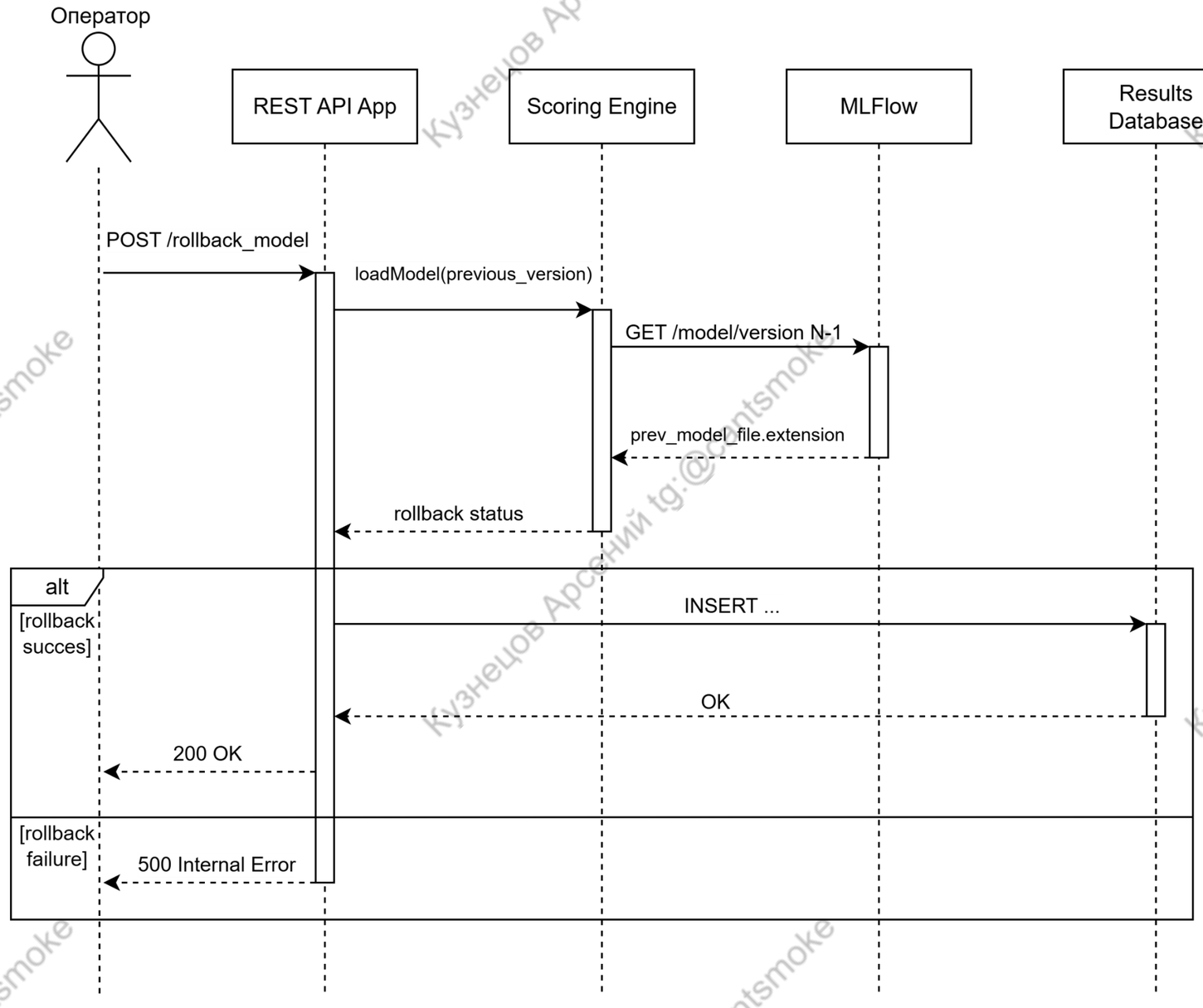

Рисунок 8. UML Sequence Diagram для отката ML-модели

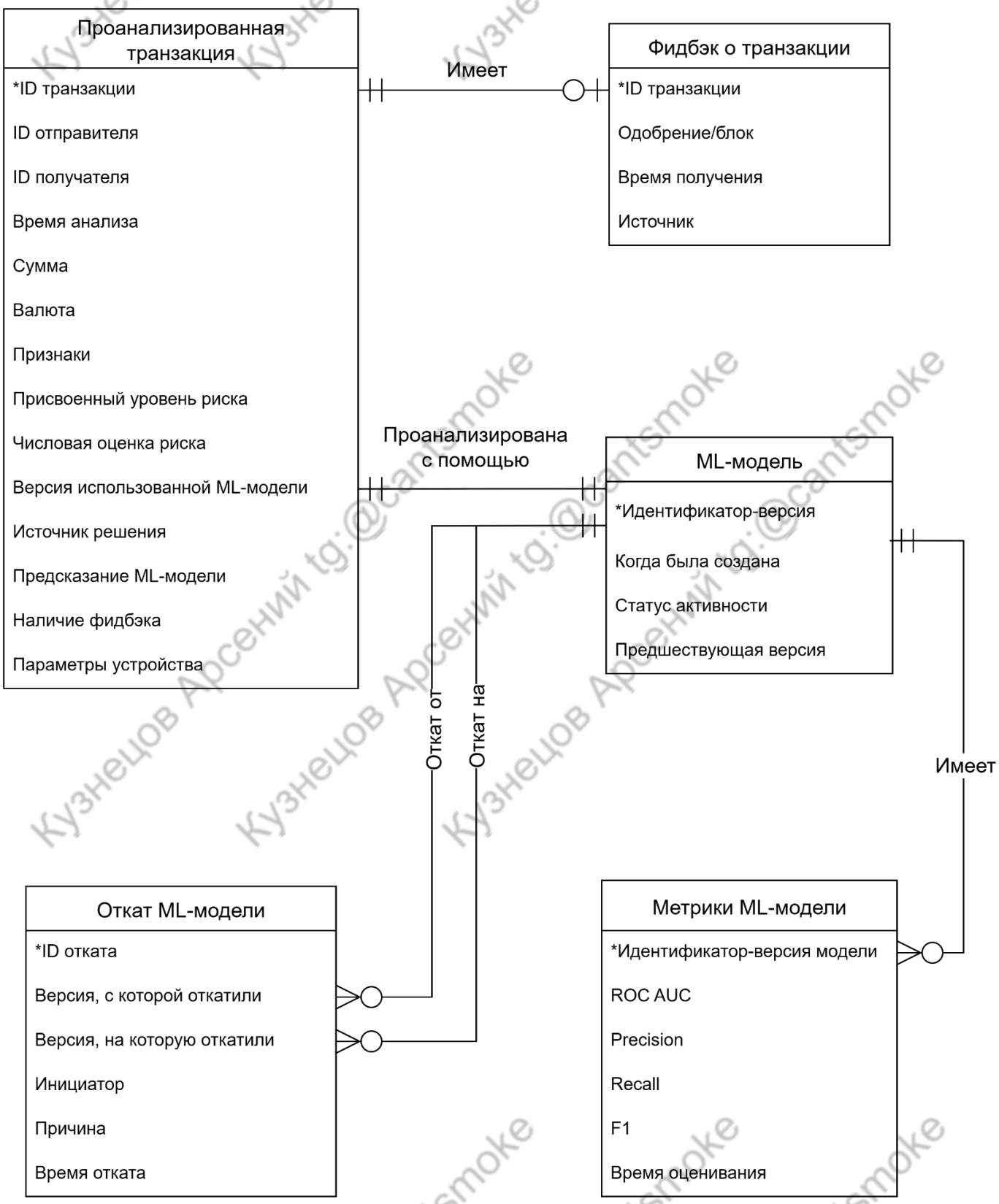

Рисунок 9. ERD на логическом уровне для Results Database

Пояснения к ER-модели:

1) Сущность “Проанализированная транзакции”

Атрибут	Комментарий
Время анализа	Время, когда была проанализирована транзакция
Сумма	Денежная сумма операции
Признаки	Сырые/подготовленные признаки
Присвоенный уровень риска	Низкий/Средний/Высокий
Числовая оценка риск	Результат скоринга
Источник решения	Уровень риска присвоен на основе бизнес-правил или по результатам скоринга ML-модели
Предсказание ML-модели	Фрод/Не фрод
Параметры устройства	Параметры устройства клиента, среди которых также содержатся тип устройства, ОС, IP-геолокация и т. п.

2) Сущность “Фидбэк о транзакции”

Атрибут	Комментарий
Фрод-статус	Финальное решение: фрод/не фрод
Одобрение/блок	Одобрена или заблокирована транзакция
Время получения	Время получения фидбэка
Источник	Транзакционная система/оператор/отдельный запрос клиента

3) Сущность “Метрики ML-модели”

Атрибут	Комментарий
ROC AUC	Метрика, измеряющая способность ML-модели различать классы
Precision	Точность ML-модели
Recall	Полнота ML-модели
F1	Гармоническое среднее между precision и recall
Время оценивание	Когда была проведена оценка
Источник данных	Тестовые данные/реальные данные и т. п.

4) Сущность “Откат ML-модели”

Атрибут	Комментарий
Инициатор	Оператор (+ какой-либо идентификатор оператора)
Причина	Причина отката, указанная оператором
Время отката	Когда был совершен откат

API Спецификация

Описание

API для анализа транзакций и управления ML-моделью в antifraud микросервисе. Все запросы требуют JWT-аутентификации. Версия: 1.1.0. Для подробного ознакомления со спецификацией рекомендуется использовать Swagger Editor (см. примечание №1)

Серверы

- https://api.bank.com/fraud - Основной REST API
- https://monitoring.bank.com/fraud_metrics - Сервис метрик

Аутентификация

Все методы защищены с помощью Bearer JWT токена.

Эндпоинты

• POST /analyze

Анализирует транзакцию в реальном времени

Пример тела запроса (application/json):

```
{
  "transaction_id": "abc123",
  "sender_id": "user_001",
  "receiver_id": "user_002",
  "amount": 1500.75,
  "currency": "RUB",
  "device_info": {
    "device_type": "mobile",
    "os": "iOS",
    "app_version": "1.3.7",
    "ip_address": "192.168.0.1",
    "geo_ip": {
      "country": "Russia",
      "region": "Moscow",
      "city": "Moscow"
    },
    "user_agent": "Mozilla/5.0..."
  }
}
```

Пример ответа (200 OK):

```
{
  "transaction_id": "abc123",
  "risk_level": "high",
```

```
  "reason": "ml_model_v4.2"
}
```

Возможные ответы:

- 200: Результат анализа транзакции
- 400: Неверный формат запроса или отсутствуют поля
- 401: Неавторизованный доступ
- 500: Внутренняя ошибка сервера

• POST /load_model

Загружает и активирует новую ML-модель

Пример тела запроса:

```
{
  "model_version": "v5.0.0"
}
```

Возможные ответы:

- 200: Новая модель успешно загружена и активирована
- 400: Неверный формат запроса
- 401: Неавторизованный доступ
- 500: Ошибка загрузки модели

• POST /rollback_model

Откатывает модель к предыдущей версии

Пример тела запроса:

```
{
  "reason": "Метрики значительно ухудшились"
}
```

Возможные ответы:

- 200: Модель успешно откатана
- 400: Откат невозможен (нет предыдущей версии)
- 401: Неавторизованный доступ
- 500: Внутренняя ошибка при откате

- GET /metrics

Возвращает метрики текущей модели. Этот эндпоинт предоставляется отдельным компонентом мониторинга (Metrics Exporter), не входит в REST API App. Используется для сбора метрик системой мониторинга (например, Prometheus).

Пример ответа:

```
{
  "model_version": "v5.0.0",
  "roc_auc": 0.92,
  "precision": 0.88,
  "recall": 0.85,
  "f1_score": 0.86,
  "timestamp": "2025-07-29T10:15:30Z"
}
```

Возможные ответы:

- 200: Метрики модели
- 401: Неавторизованный доступ
- 500: Ошибка получения метрик

Примечания

1) API-спецификация в формате Swagger/OpenApi (JSON-файл) https://github.com/cantsmoke/Anti-Fraud-Microservice/blob/main/AntiFraud_MicroserviceAPI_Specs.json