ИНТЕЛЛЕКТ
ФИНАНСЫ
ДОВЕРИЕ
РЕАЛЬНЫЕ ДЕЙСТВИЯ

КОТОРЫМ
МОЖНО ДОВЕРЯТЬ

РЕГУЛИРУЕМЫЙ АГЕНТНЫЙ ФИНТЕХ

КАК СТРОИТЬ АВТОНОМНЫЕ AI-СИСТЕМЫ, КОТОРЫМ МОЖНО ДЕЛЕГИРОВАТЬ ДЕЙСТВИЯ, А НЕ ТОЛЬКО ОТВЕТЫ

АРХИТЕКТУРА
БЕЗОПАСНОСТЬ
УПРАВЛЕНИЕ
ДОКАЗУЕМОСТЬ
МАСШТАБИРОВАНИЕ

БАНКИ
ПЛАТЕЖИ
ИНВЕСТИЦИИ
СООТВЕТСТВИЕ
УСТОЙЧИВОСТЬ

Калачев Павел Александрович

2025

Павел Калачев

Регулируемый агентный финтех: как строить автономные AI-системы, которым можно делегировать действия, а не только ответы

https://litres.ru/74492109
SelfPub; 2026

Аннотация

В книге исследуется переход финансовых организаций от генеративных моделей к агентным системам, способным автономно изменять защищаемые состояния: денежные обязательства, права доступа, рыночные позиции и инфраструктуру. Автор предлагает концепцию регулируемой автономности, при которой полномочие агента задаётся как ограниченный транзакционный бюджет, а допустимость действия обеспечивается связью намерения, плана, политики, исполнения и наблюдаемого результата. Центральными категориями выступают доказательное равенство, побочный эффект, окно обратимости, радиус воздействия и долг доказуемости. На стыке инженерии, бухгалтерского контроля, права и риск-

менеджмента формируется архитектура допуска, аудита и остановки. В книге систематизируются нормативные требования, практические инциденты и эксплуатационные механизмы, предлагая финансовым организациям методологию безопасного масштабирования агентных систем в современной регулируемой цифровой среде.

Павел Калачев
Регулируемый агентный финтех: как строить автономные AI-системы, которым можно делегировать действия, а не только ответы

Предисловие. Право на действие

В настоящее время все больше задач делегируется программам. Человек получает должностные полномочия, проходит обучение и действует внутри процесса. Программа исполняет заранее определённые правила. Агентная система занимает промежуточное положение. Она понимает естественный запрос, строит план, выбирает инструмент и меняет план после наблюдения. Именно это сочетание делает её полезной и создаёт новую инженерную проблему.

Разговор об искусственном интеллекте долго строился вокруг качества ответа. Для банка этого уже недостаточно. Ответ можно проверить до использования. Действие способно изменить деньги, обязательство, рыночную позицию, право

доступа или состояние инфраструктуры до того, как человек увидит промежуточное решение. Вопрос доверия перемещается от текста к полномочию.

В этой книге предлагается контур регулируемой автономности, сокращённо КРА. Его основная идея проста: автономность следует задавать в форме конечного бюджета конкретной транзакции. Система получает право не «работать самостоятельно», а выполнить определённый план действий с заданными объектами при известных условиях и в течение ограниченного времени. Право уменьшается с каждым действием. Существенное изменение цели, плана или контекста требует нового допуска.

Второй элемент подхода называется доказательным равенством. Намерение, разрешённый план, фактическое действие и наблюдаемое хозяйственное состояние должны образовать воспроизводимую цепочку. Равенство здесь не означает совпадение текстов. Оно означает отсутствие необъяснённого разрыва между поручением, правом, исполнением и результатом.

Бухгалтерская перспектива проходит через всю книгу не как частный пример, а как дисциплина связи. Бухгалтер отличает документ от хозяйственного события, проект проводки от проведённой записи, платёжное поручение от расчёта, техническую отмену от сторно и последующей сверки. Эти различия помогают увидеть то, что чатовый интерфейс скрывает: агент действует не в мире фраз, а в мире состоя-

ний и доказательств.

Книга опирается на правовые акты, материалы регуляторов, стандарты, технические спецификации, исследования и опубликованные разборы инцидентов. Факт, расчёт и авторский вывод различаются. Реальные инциденты, которые не были вызваны языковыми агентами, используются только для анализа механизмов распространения ошибки. Такое ограничение прямо отмечается в соответствующих местах.

В первом разделе устанавливается новая единица риска: автономное действие с побочным эффектом. Во втором собирается референсная архитектура из намерения, плана, допуска, исполнения и доказательства. В третьем исследуются воспроизводимость, инженерные испытания, роль человека и переход от одного агента к сети специализированных участников. Четвёртый раздел связывает контроль с экономикой, масштабированием и дорожной картой банка.

Основополагающий вопрос книги звучит следующим образом: что должна уметь организация, прежде чем она передаст системе право действовать. Предлагаемый ответ не требует веры в безошибочность модели. Он требует ограниченного радиуса воздействия, исполняемой политики, наблюдаемого результата и доказательства, созданного вместе с действием.

Раздел I. От чат-бота к цифровому субъекту: новая инже-

1.1. Три перехода: генерация → инструменты → автономные цепочки действий

В конце месяца бухгалтер видит не разговор с машиной, а очередь незавершённых обязательств. У одного поставщика изменились банковские реквизиты. По другой операции накладная пришла после закрытия реестра. В третьем случае сумма в акте не совпала с суммой счёта. Обычный языковой помощник способен объяснить, как провести сверку. Система с инструментами может найти договор, открыть карточку контрагента и рассчитать расхождение. Агент способен пойти дальше: выбрать последовательность операций, запросить недостающие сведения, изменить статус документа, сформировать проводку, подготовить платёж и передать его на исполнение. В интерфейсе все три системы могут выглядеть как одно окно чата. В устройстве ответственности между ними лежат разные инженерные миры.

Различие начинается не с интеллекта модели. Оно начинается с границы, за которой текст превращается в действие. Пока система формирует ответ, её результат остаётся ин-

формационным объектом. Человек может прочитать его, отвергнуть, исправить или не заметить. Когда система вызывает инструмент, она получает возможность воздействовать на внешнее состояние. Когда она связывает несколько вызовов в автономную цепочку, она приобретает способность создавать последствия раньше, чем человек увидит промежуточные решения. Это и есть три перехода, вокруг которых строится первая часть книги.

Первый переход ведёт от генерации к структурированному ответу. Языковая модель получает контекст и формирует последовательность токенов. Даже если ответ содержит расчёт, проект платёжного поручения или фрагмент программного кода, сам по себе он ещё ничего не изменяет во внешней системе. Ошибка на этом уровне опасна, но обычно сохраняет окно для проверки. Бухгалтер может обнаружить неверный счёт до проведения документа. Инженер может прочитать команду до запуска. Специалист по комплаенсу может проверить ссылку на норму до включения вывода в заключение.

Эта стадия породила привычный язык оценки. Мы спрашиваем, верен ли ответ, соответствует ли он инструкции, содержит ли вымышленные сведения, не раскрывает ли конфиденциальные данные. Такие вопросы нужны и для агента, но их недостаточно. Точность ответа измеряет качество информационного продукта. Она не измеряет допустимость действия. Между фразой «поставщику следует перечислить

один миллион рублей» и фактическим списанием миллиона рублей существует не количественная, а категориальная граница.

Второй переход начинается, когда модель получает инструмент. Инструментом может быть поиск, калькулятор, система управления взаимоотношениями с клиентами, банковский интерфейс, хранилище документов, среда выполнения кода или корпоративный программный интерфейс. Исследование Toolformer описало модель, которая учится решать, какой внешний программный интерфейс вызвать, когда это сделать, какие аргументы передать и как использовать результат при последующем формировании текста. Работа ReAct объединила рассуждение и действие в чередующемся цикле, где наблюдение из среды меняет последующий ход решения. Эти публикации важны не как готовые производственные архитектуры, а как фиксация смены предмета: модель начинает не только вычислять продолжение текста, но также выбирать точку контакта с внешней средой. Toolformer и ReAct дают первичные описания этих подходов.

Инструмент устраняет часть слабостей языковой модели. Калькулятор выполняет арифметику по определённым правилам. Поиск возвращает актуальный документ. Учётная система предоставляет фактический остаток. Однако инструмент одновременно открывает новый класс ошибок. Модель может выбрать верную функцию с неверными аргументами.

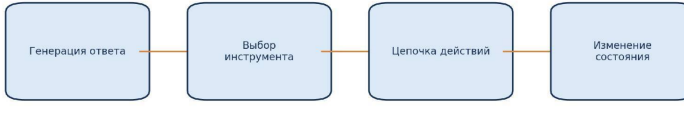
Она может вызвать функцию в неправильном контексте. Она может принять служебное описание инструмента за достаточное основание для действия. Она может корректно получить данные и неверно понять их хозяйственный смысл.

Для бухгалтера это различие знакомо. Доступ к форме проводки не означает права признать операцию. Техническая возможность изменить реквизит не доказывает полномочие контрагента. Наличие остатка на счёте не отвечает на вопрос, наступил ли срок платежа. Инструмент сообщает системе, что она способна сделать. Политика должна установить, что ей разрешено сделать. Экономический контекст должен объяснить, зачем действие совершается. Доказательный слой должен сохранить основание.

Третий переход возникает, когда отдельные вызовы образуют цепочку с обратной связью. Агент получает цель, формирует или уточняет план, вызывает инструмент, наблюдает результат, изменяет план и продолжает работу до условия завершения. В практическом руководстве OpenAI агент определяется как система, которая самостоятельно выполняет задачи от имени пользователя, а цикл запуска продолжается до результата, ошибки, вызова завершающего инструмента или иного условия остановки. Это определение подчёркивает операционную самостоятельность, но для регулируемого финтеха требуется дополнительное уточнение: автономным следует считать не тот цикл, который долго рассуждает, а тот, который может без нового решения человека изменить за-

щищаемое состояние. Практическое руководство по созданию агентов описывает цикл, инструменты и варианты оркестрации.

Переходы удобно рассмотреть как последовательное расширение объекта контроля. Сначала меняется форма результата, затем появляется вызов внешней функции, после чего отдельные вызовы соединяются в план и приводят к последствию. Эта логика представлена ниже через четыре состояния.



Добавление кнопок к языковой модели ещё не создаёт управляемого агента, и схема помогает увидеть причину. Кнопка обозначает инструмент, но ничего не говорит о происхождении цели, сроке полномочия, зависимостях меж-

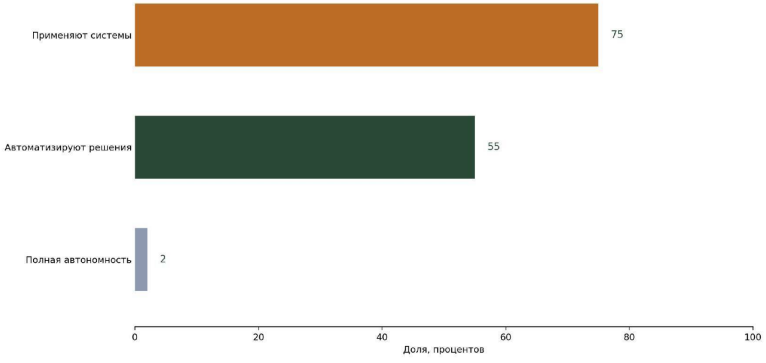
ду шагами и критериях завершения. Если модель получила функцию «создать платёж», инженер должен знать, кто определил сумму, откуда взяты реквизиты, можно ли менять получателя, требуется ли разделение ролей, какое событие прекращает полномочие и что должно попасть в журнал. Без этих ответов кнопка скрывает риск.

Развитие агентных систем часто описывают через увеличение способности модели рассуждать. Для финансовой организации полезнее другая ось: увеличение самостоятельности в управлении состоянием. На этой оси поисковый помощник и агент с доступом к платежам могут использовать одну модель, но относиться к разным классам риска. Первая система читает данные и предлагает человеку решение. Вторая выбирает и исполняет действие. Третья может наблюдать результат платежа, обнаружить отказ, исправить маршрут и повторить попытку. Каждая ступень сокращает число человеческих решений внутри процесса и одновременно увеличивает количество состояний, которые система способна создать.

Такой взгляд согласуется с наблюдаемой осторожностью финансового сектора. В совместном исследовании Банка Англии и Управления по финансовому регулированию участвовали 118 организаций. По данным отчёта, 75 процентов респондентов уже применяли искусственный интеллект. Некоторая степень автоматического принятия решений присутствовала в 55 процентах заявленных сценариев, тогда как

полностью автономные решения составляли 2 процента. Эти показатели имеют разные знаменатели и не должны складываться. Их совместное чтение показывает разрыв между распространением технологии и передачей полного операционного усмотрения. Исследование искусственного интеллекта в финансовых услугах Великобритании публикует методику и агрегированные результаты.

На следующем рисунке этот разрыв представлен без попытки превратить категории в части одной суммы. Столбцы отвечают на разные вопросы отчёта. Первый показывает долю организаций; второй отражает долю сценариев с некоторой автоматизацией решений; третий относится к доле сценариев с полной автономностью.



Цифры не доказывают безопасность конкретной архитектуры. Они задают масштаб инженерного перехода. Большинство организаций уже имеет дело с моделями, но лишь малая доля сценариев получила полную самостоятельность. Следовательно, рынок находится не в точке простого внедрения искусственного интеллекта, а в точке переговоров о границе делегирования.

Для таких переговоров полезно разделить три уровня по объекту контроля. Таблица не ранжирует их по качеству. Она показывает, какой новый вопрос возникает при каждом переходе.

Уровень	Непосредственный результат	Основной объект проверки	Пример в работе бухгалтера	Новое условие допуска
Генерация	Текст, расчёт или проект документа	Корректность содержания	Объяснение расхождения по счёту	Проверка человеком до использования
Инструменты	Вызов внешней функции и полученный ответ	Корректность функции, аргументов и контекста	Получение оборотов по счёту из учётной системы	Минимальное право чтения и проверка назначения
Автономная цепочка	Последовательность решений с изменением состояния	Цель, план, полномочие, исполнение и доказательство	Формирование проводки, сверка, подготовка платежа и изменение статуса	Транзакционный допуск к конкретному плану

Различие между режимами работы сведено ниже. Автоматизация заранее известного маршрута и агентное выполнение неодинаковы. В обычном процессе последовательность шагов определена разработчиком. Условие ветвления фор-

мализовано. Допустимые параметры перечислены. Агент получает часть усмотрения: он выбирает следующий шаг на основании контекста, который не был полностью описан при разработке. Именно усмотрение делает систему полезной для неструктурированной работы. Оно же не позволяет считать прежний контроль процесса достаточным.

Исследовательские испытания подтверждают, что способность действовать нельзя выводить из качества ответов. AgentBench оценивает модели в восьми интерактивных средах и связывает типичные неудачи с долгосрочным рассуждением, принятием решений и следованием инструкциям. GAIA строит задачи, для которых нужны рассуждение, работа с несколькими форматами, поиск и инструменты. Первичная публикация GAIA показала заметный разрыв между результатами людей и исследованной конфигурацией модели с подключаемыми средствами. Конкретные показатели ранних моделей не следует переносить на системы 2026 года. Сохраняется методологический вывод: испытание ответа и испытание траектории являются разными задачами. AgentBench и GAIA дают описание сред и ограничений оценки.

Для регулируемого финтеха траектория важнее отдельной реплики. Правильное действие может быть получено недопустимым способом. Агент может найти верную сумму в документе, к которому не имел права обращаться. Он может сформировать корректный платёж после подмены назначе-

ния. Он может завершить сверку, раскрыв данные другому инструменту. Оценка только итогового результата пропускает нарушение, которое возникло по дороге.

Здесь я предлагаю сменить единицу регулирования. Ею должен стать не ответ модели, а переход защищаемого состояния. В такой переход входят исходное состояние, цель, план, полномочие, действие, новое состояние и доказательство связи между ними. Если хотя бы одного элемента нет, организация знает, что произошло, но уже не может объяснить допустимость результата.

Это положение меняет и роль бухгалтера. В традиционном представлении бухгалтер проверяет документ после хозяйственного события. В агентной системе профессиональное знание должно войти в условия действия до события. Признак закрытого периода, статус первичного документа, правило двойного подтверждения, допустимый счёт учёта, лимит суммы и связь с договором становятся не примечаниями к инструкции, а исполняемыми ограничениями. Бухгалтерский контроль перестаёт быть только последующей проверкой и становится частью архитектуры допуска.

Само слово «агент» ещё ничего не говорит о качестве управления. Решающее различие появляется там, где система получает право менять внешнее состояние. Я предлагаю считать этот переход сменой предмета ответственности. Сначала организация отвечает за использование ответа, затем за границу доступа, а после появления цепочки дей-

ствий уже за весь путь к новому состоянию. Чтобы продолжить рассуждение, нужно внимательнее посмотреть на то, что остаётся в системе после вызова инструмента.

1.2. Side effects как принципиальное отличие agentic AI

В программной инженерии побочным эффектом называют изменение, которое выходит за пределы вычисления возвращаемого значения. Функция может записать данные, отправить сообщение, изменить право доступа, списать деньги, разместить приказ или запустить другой процесс. Слово «побочный» иногда создаёт неверное впечатление, будто речь идёт о случайном или второстепенном результате. Для агента именно эффект часто является целью. Пользователь обращается к системе не ради текста «платёж подготовлен», а ради изменения статуса обязательства и, при наличии полномочия, ради движения денежных средств.

Эта особенность требует смены основной единицы безопасности. Для модели без инструментов естественно анализировать вход и выход. Для агента необходимо анализировать переход состояния. Он задаётся как минимум пятью элементами: состояние до действия, намерение, операция с параметрами, состояние после действия и доказательство

связи между ними. Возвращённое инструментом сообщение об успехе является лишь одним наблюдением. Оно не заменяет проверку результата в системе учёта, у платёжного провайдера или в журнале прав доступа.

Рассмотрим простую команду: «Оплати утверждённые счета поставщиков». Её словесная ясность обманчива. Агент должен определить, какие счета относятся к текущему юридическому лицу, что означает утверждение, не изменились ли реквизиты, наступил ли срок, не было ли частичной оплаты, открыт ли период, хватает ли ликвидности, нужен ли второй подписант и какая система является источником истины для статуса платежа. Каждый ответ может быть технически корректным и одновременно семантически непригодным. Например, счёт действительно утверждён менеджером закупок, но ещё не прошёл налоговую проверку. Реквизит существует в карточке поставщика, но был изменён утром после письма с подменённого адреса.

Побочный эффект превращает такую неоднозначность в убыток. Пока агент составляет список, ошибку можно исправить без движения денег. После передачи платёжного поручения появляется внешнее обязательство. После расчёта через платёжную систему возможность отмены зависит уже не только от организации. Она зависит от правил системы, времени, юрисдикции, действий банка получателя и правовой квалификации перевода. Поэтому уверенность модели нельзя использовать как прямую меру разрешения. Даже высо-

кая внутренняя оценка вероятности не создаёт юридического основания.

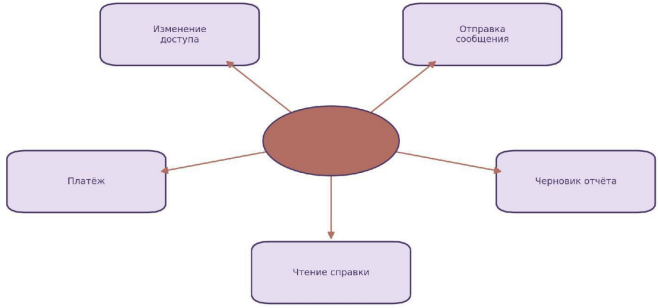
Протоколы распределённых систем давно различают безопасные и идемпотентные операции. Стандарт семантики протокола передачи гипертекста определяет безопасный метод как метод, запрошенная семантика которого по существу связана с чтением, а идемпотентный метод как операцию, для которой несколько одинаковых запросов имеют тот же предполагаемый эффект, что и один запрос. Это различие полезно для агентной архитектуры, но его нельзя переносить механически. Повторный запрос на создание платежа не становится безопасным только потому, что программный интерфейс принял ключ идемпотентности. Ключ защищает от части технических повторов. Он не проверяет назначение, сумму и полномочие. Стандарт RFC 9110 даёт нормативные определения безопасных и идемпотентных методов.

Побочный эффект имеет несколько измерений. Первое измерение связано с объектом изменения. Агент может менять деньги, позицию, договорное обязательство, персональные данные, право доступа, конфигурацию или публичное сообщение. Второе измерение связано с направлением: создание, изменение, удаление, передача, блокировка. Третье описывает обратимость. Четвёртое фиксирует масштаб. Пятое отражает связанность, то есть число последующих процессов, которые примут новое состояние как достоверное.

Последнее измерение часто недооценивают. Ошибочная

бухгалтерская проводка может автоматически попасть в управленческий отчёт, расчёт ковенанта, налоговый регистр и модель ликвидности. Каждая следующая система действует рационально относительно неверного входа. Первичная ошибка становится источником вторичных решений. Если агент видит только успешный ответ функции, он не замечает распространение эффекта по зависимостям.

Для наглядности расположим типовые действия по двум осям. Положение точек отражает качественную модель автора, а не статистическую оценку. Оно показывает, почему чтение справки и проведение платежа требуют разных режимов допуска даже при одинаковой вероятности ошибки модели.



Матрица не должна превращаться в универсальный ката-

лог. Обратимость зависит от момента и контекста. Черновик отчёта легко удалить до отправки, но после публикации он может повлиять на рынок. Право доступа можно отозвать, но нельзя гарантированно вернуть конфиденциальность данных, которые уже были прочитаны. Платёж иногда удаётся возвратить, но само обращение за возвратом создаёт операционные расходы и правовой риск. По этой причине контролировать следует не название функции, а конкретный переход с учётом времени.

Введём понятие окна обратимости. Это период, в течение которого результат можно отменить без несоразмерного ущерба, нарушения обязательства или потери юридической позиции. Для черновика окно может существовать до публикации. Для задания на изменение инфраструктуры оно может закрыться после распространения конфигурации. Для платёжного поручения граница зависит от статуса обработки. Окно является переменной, которую агент должен учитывать до действия, а доказательный слой должен сохранять после него.

Из окна обратимости следует практическое правило. Чем быстрее оно закрывается, тем раньше должен срабатывать контроль. Проверка после исполнения полезна для расследования, но не заменяет допуск до исполнения. Этот вывод связывает агентную инженерию с бухгалтерской логикой предварительного и последующего контроля. Последующий контроль устанавливает правильность отражения события.

Предварительный контроль предотвращает событие, для которого нет основания. Автономное действие требует обоих контуров.

Регулирование операционной устойчивости поддерживает такой взгляд, хотя не использует терминологию этой книги. Регламент Европейского союза о цифровой операционной устойчивости финансового сектора требует управления риском информационных технологий, сообщения об инцидентах, испытаний устойчивости и контроля риска внешних поставщиков. Базельский комитет определяет операционную устойчивость через способность банка продолжать критические операции при нарушении и предлагает заранее устанавливать допустимый уровень нарушения. Агент должен распознать критическую операцию до вызова инструмента. Иначе система определит её значимость только после сбоя. Регламент 2022/2554 и принципы Базельского комитета задают официальный нормативный и надзорный контекст.

Побочный эффект создаёт также проблему неполного наблюдения. Программный интерфейс может ответить, что запрос принят, но это ещё не означает завершения операции. Ответ может обозначать постановку в очередь. Платёж может перейти в промежуточный статус. Изменение доступа может распространиться с задержкой. Агент, который интерпретирует подтверждение приёма как подтверждение результата, способен выполнить следующий шаг на ложном основании.

В бухгалтерском процессе эта ошибка проявляется как разрыв между событием и признанием. Подготовка проводки, её проведение и включение в закрытый период являются разными состояниями. Создание файла для банка, отправка файла, приём банком и расчёт также различаются. Агент должен наблюдать не произвольный ответ инструмента, а доменный признак завершения. Для платежа таким признаком может быть определённый статус системы расчётов. Для доступа это проверка фактического разрешения. Для проводки это идентификатор проведённого документа и контрольные суммы затронутых регистров.

Проектные последствия побочного эффекта раскрывает следующее сопоставление. Оно показывает, почему общего разрешения на запись недостаточно.

Параметр	Проверочный вопрос	Пример риска	Инженерный контроль	Доказательство
Объект	Какое защищаемое состояние изменяется	Платёж ошибочному получателю	Белый список типов операций и получателей	Идентификатор объекта до и после действия
Масштаб	Какова верхняя граница последствия	Серия платежей по ошибочному правилу	Лимит суммы, числа операций и периода	Использованный лимит и остаток бюджета
Обратимость	До какого момента результат можно отменить	Закрытие окна отзыва	Отложенное исполнение и контроль статуса	Время создания, подтверждения и расчёта
Связанность	Какие процессы примут результат	Ошибка попадает в отчётность и прогноз	Изоляция, этап черновика, сверка зависимостей	Перечень затронутых систем и версий данных
Повторяемость	Что произойдёт при повторном вызове	Двойное списание после прерывания ожидания ответа	Ключ идемпотентности и проверка результата	Ключ запроса, ответ и итоговый статус
Полномочие	Кто разрешил именно этот переход	Использование общего технического аккаунта	Краткоживущий допуск к плану и параметрам	Субъект, политика, срок и область допуска

Из таблицы видно, что ограничение агента нельзя свести к запрету отдельных инструментов. Одна функция может быть безопасной и опасной в разных режимах. Чтение остатка обычно имеет низкий прямой эффект, но массовое чтение счетов создаёт риск конфиденциальности. Создание платежа на один рубль и создание платежа на сто миллионов используют одинаковый интерфейс, но имеют разный масштаб. Изменение роли для тестовой записи и изменение роли администратора отличаются объектом и связанностью.

Следовательно, полномочие должно прикрепляться не к имени инструмента, а к предикату действия. Такой предикат может включать тип операции, сумму, валюту, получателя,

юридическое лицо, назначение, источник реквизитов, время, состояние согласования и допустимую последовательность. Если параметр меняется, меняется и смысл разрешённого перехода.

Здесь появляется первое основание для транзакционного допуска. Обычный токен отвечает на вопрос, может ли субъект обратиться к ресурсу. Для регулируемого агента нужен ответ на более узкий вопрос: может ли конкретный исполнитель выполнить этот план действий с указанными объектами при установленных условиях и в заданный срок. Спецификация авторизации протокола контекста модели требует привязки токена к целевому ресурсу, проверки аудитории и минимизации областей доступа. Эти меры снижают риск передачи токена не тому серверу и чрезмерного полномочия. Однако допустимость суммы или получателя с точки зрения хозяйственного процесса должна быть выражена дополнительной политикой. Спецификация авторизации от 28 июля 2026 года фиксирует требования транспортного уровня.

Наличие допуска ещё не решает проблему наблюдения. Агент может действовать в разрешённой области и получить неожиданный результат. Поэтому после каждого значимого эффекта система должна сопоставить ожидаемое и фактическое состояние. Если различие выходит за пределы допуска, цепочка останавливается. Она не должна импровизировать с повышением полномочий, новым получателем или обходным инструментом только ради выполнения цели.

Это правило противостоит распространённой логике успешности задачи. В обычной оценке агент получает положительный балл, если достиг цели. В регулируемой системе достижение цели вне разрешённой траектории является отказом. Платёж правильному поставщику с использованием реквизитов из непроверенного письма нельзя считать успехом. Исправление остатка через прямую запись в базу вместо утверждённого документа нельзя считать успехом. Завершение операции не отменяет нарушения процесса.

Из этого различия рождается второй тезис. Побочный эффект не добавляет ещё один риск к обычной работе модели. Он меняет форму ответственности за систему. Разрешение приходится связывать с переходом состояния, а результат наблюдать на языке конкретного процесса. Без такой связи агент остаётся моделью с технической возможностью действия, но не становится регулируемым исполнителем.

Пока результат существует только в тексте, его можно спокойно перечитать и отвергнуть. После записи, отправки или списания такой роскоши уже нет. Значение имеет не обещание инструмента об успехе, а подтверждённое изменение в учёте, расчётах или правах доступа. Отсюда естественно возникает вопрос о цене: одинаковая ошибка модели способна оставить совершенно разные последствия.

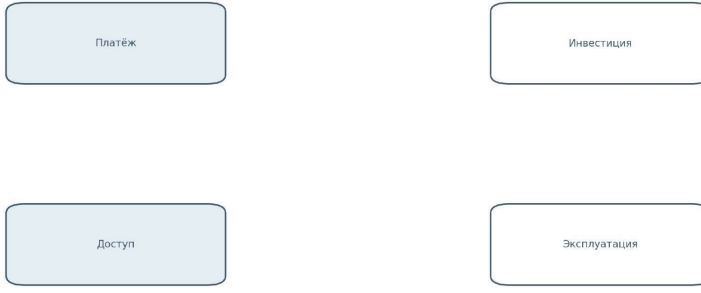
1.3. Цена ошибки в платеже, инвестиции, доступе и эксплуатации

Цена ошибки редко совпадает с суммой неправильной операции. Платёж создаёт прямое движение денег, но за ним следуют возврат, расследование, спор с получателем, корректировка учёта и проверка контроля. Ошибка инвестиционного действия изменяет рыночную позицию и может сама повлиять на цену. Ошибка доступа раскрывает возможность, а не только данные. Ошибка эксплуатации нарушает работу зависимых систем. Во всех четырёх случаях первоначальное действие образует цепочку последствий, часть которых проявляется позже.

Поэтому оценивать агента только по вероятности правильного ответа экономически недостаточно. Пусть вероятность ошибочного действия равна одному проценту. Для чтения внутренней справки это может означать лишнюю проверку. Для перевода средств тот же процент может быть неприемлем. Риск определяется произведением вероятности и тяжести, но агентная система требует ещё двух поправок: скорость накопления действий и связанность среды. Ошибка, повторяемая тысячу раз в минуту, создаёт иной профиль, чем единичная ошибка с тем же ожидаемым ущербом.

Четыре канала цены удобно рассмотреть по отдельности,

не разрывая их связь. Ниже собраны объекты, через которые ошибка становится наблюдаемой для финансовой организации. Количественной шкалы нет, поскольку суммы из разных случаев нельзя складывать без общей методики.



Платёжный канал кажется понятным, поскольку имеет денежную сумму. Реальный случай Citibank показывает, почему одной суммы недостаточно. В августе 2020 года банк должен был выплатить держателям долга Revlon промежуточные проценты и направить сумму основного долга около 894 миллионов долларов на внутренний технический счёт. Для подавления внешнего перевода требовалось выбрать три поля в системе. Участники процесса выбрали только одно. Процедура предусматривала три роли проверки, но все три

человека одинаково поняли интерфейс. Предупреждение сообщило, что деньги будут отправлены из банка, однако не показало сумму и не отделило запланированные проценты от основного долга. В результате банк перечислил 894 миллиона долларов основного долга и 7,8 миллиона долларов процентов. Около 385 миллионов долларов получатели вернули, а получатели суммы примерно в 500 миллионов долларов первоначально отказались возвращать средства. Апелляционный суд впоследствии отменил решение первой инстанции и направил дело на новое рассмотрение. Решение Апелляционного суда второго округа содержит описание процесса, суммы, трёх ролей и последующей сверки.

Этот случай не связан с языковой моделью. Именно поэтому он полезен. Он показывает, что участие нескольких людей не гарантирует смысловой независимости контроля. Создатель операции, проверяющий и утверждающий разделяли одну неверную модель поведения интерфейса. Формально процедура существовала. Семантически все участники проверили одну и ту же гипотезу. Агент, встроенный в такой процесс без независимого расчёта ожидаемого движения денег, способен сделать процедуру быстрее, не сделав её надёжнее.

С позиции бухгалтера решающей оказалась утренняя сверка. Несоответствие между ожидаемыми и фактическими движениями денежных средств выявило ошибку после исполнения. Для агентной архитектуры из этого следует

двойное требование. До платежа нужно вычислить ожидаемую сумму внешнего списания и сопоставить её с деловой целью. После платежа нужно получить сведения о фактическом движении средств и сопоставить их с ожидаемым результатом. Если система видит только заполненные поля формы, она контролирует интерфейс. Если она видит экономическое движение, она контролирует операцию.

Инвестиционный канал добавляет скорость и влияние на рынок. В августе 2012 года дефектное программное обеспечение Knight Capital обработало 212 входящих родительских заявок и отправило миллионы дочерних заявок. По данным Комиссии по ценным бумагам и биржам США, за период около 45 минут произошло 4 миллиона исполнений по 154 акциям общим объёмом более 397 миллионов акций. Компания получила чистую длинную позицию около 3,5 миллиарда долларов по акциям 80 компаний и короткую позицию около 3,15 миллиарда долларов по акциям 74 компаний. Реализованный убыток составил 460 миллионов долларов. Постановление Комиссии по делу Knight Capital описывает факты и недостатки контроля.

Выполним несколько расчётов только на основании опубликованных показателей. Четыре миллиона исполнений за 45 минут дают примерно 88 889 исполнений в минуту. Деление ещё на 60 даёт около 1 481 исполнения в секунду. Средний объём одного исполнения, рассчитанный как результат деления 397 миллионов акций на 4 миллиона испол-

нений, составляет около 99 акций. Если распределить реализованный убыток по времени инцидента, получится около 10,22 миллиона долларов за минуту. Последний показатель не описывает фактическую траекторию убытка и используется только как нормированная оценка скорости накопления последствия.

Формулы расчёта имеют следующий вид:

v_мин = 4 000 000 / 45 ≈ 88 889

v_сек = 4 000 000 / (45 × 60) ≈ 1 481

q_исп = 397 000 000 / 4 000 000 ≈ 99,25

l_мин = 460 000 000 / 45 ≈ 10 222 222 долл./мин

Здесь (v_{мин}) обозначает число исполнений в минуту; (v_{сек}) соответствует числу исполнений в секунду; (q_{исп}) показывает среднее число акций на одно исполнение; (l_{мин}) выражает условный убыток за минуту. Исходные данные взяты из постановления Комиссии. Все полученные значения являются авторскими расчётами.

Случай Knight Capital также показывает значение сигнала, который не был включён в операционный контроль. До открытия рынка система сформировала 97 автоматических сообщений, связанных с ошибкой. По материалам Комиссии, эти сообщения не были спроектированы как системные предупреждения, а сотрудники обычно не просматривали их при получении. Наличие записи не равно наличию контроля. Чтобы журнал снижал риск, событие должно иметь владельца, уровень существенности, маршрут реакции и усло-

вие остановки.

Для агента этот урок имеет прямое значение. Генеративная система способна производить много объяснений и предупреждений. Их количество может создать впечатление наблюдаемости. Однако сообщение, которое не влияет на допуск или остановку, остаётся повествованием вокруг ошибки. Регулируемая система должна уметь переводить сигнал в изменение права на последующее действие.

Канал доступа отличается тем, что ущерб может возникнуть без немедленного движения денег. Ошибка выдаёт субъекту возможность читать, изменять или передавать защищаемые объекты. После чтения конфиденциальных данных отзыв права не возвращает исходное состояние. После получения учётных данных злоумышленник может использовать их в другом контуре. Поэтому обратимость доступа асимметрична: разрешение можно удалить, но уже раскрытое знание удалить нельзя.

Правоприменительный материал по Capital One не относится к агенту и не доказывает причинную связь между искусственным интеллектом и инцидентом. Он показывает цену слабой оценки риска доступа при переходе к новой технологической среде. В августе 2020 года Управление контролёра денежного обращения США назначило штраф 80 миллионов долларов. Орган указал на неспособность банка создать эффективные процедуры оценки риска до переноса существенных операций в публичное облако и своевременно

исправить недостатки. Сообщение Управления контролёра денежного обращения содержит сумму и основание меры.

В агентной системе технологический переход имеет сходную форму. Организация может перенести уже существующий токен доступа в новый оркестратор и считать, что риск не изменился. Но меняется способ использования полномочия. Человек обращается к системе с локальной целью. Агент может связать несколько источников, повторить запрос и передать результат следующему инструменту. Поэтому прежняя роль пользователя не всегда является допустимой ролью агента. Необходим отдельный субъект, отдельный срок полномочия и отдельная область действия.

Эксплуатационный канал показывает, как ошибка распространяется через однородную инфраструктуру. 19 июля 2024 года CrowdStrike выпустила обновление содержимого для датчика на системах Windows. Компания сообщила, что проблемное содержимое привело к аварийному завершению работы системы. В область воздействия входили узлы с определёнными версиями датчика, которые были в сети между 04:09 и 05:27 по всемирному координированному времени и получили обновление. Дефект был отозван в 05:27. В предварительном разборе компания указала, что одна конфигурация прошла проверку вследствие ошибки валидатора, а затем предложила поэтапное развёртывание, начальное развёртывание на ограниченной группе, дополнительные проверки и расширенные средства контроля для клиен-

тов. Предварительный разбор инцидента является источником этих фактов.

Этот инцидент также не был связан с действием языкового агента. Он демонстрирует свойство, которое агент наследует от любой автоматизированной системы доставки: единичное решение получает большой радиус через скорость и однородность. Если агент способен менять конфигурацию многих узлов, его логическая ошибка соединяется с механизмом распространения. Ограничение числа узлов, поэтапный выпуск и автоматическая остановка по показателям работоспособности снижают радиус независимо от качества модели.

Сопоставим четыре случая без смешения их правовой природы. Таблица использует официальные материалы для фактов и авторскую классификацию для инженерных выводов.

Канал	Наблюдаемый случай	Подтверждённый масштаб	Что не сработало	Ограничение для агента
Платёж	Ошибочный перевод Citibank по долгу Revlon	894 миллиона долларов основного долга и 7,8 миллиона долларов процентов	Три участника одинаково поняли интерфейс, предупреждение не показало экономическую сумму	Независимый расчёт внешнего списания, сверка цели и суммы, проверка после исполнения
Инвестиция	Инцидент Knight Capital	4 миллиона исполнений, более 397 миллионов акций, убыток 460 миллионов долларов за период около 45 минут	Дефектный код, недостаточный предторговый контроль, сигналы без реакции	Лимит позиции, числа действий и скорости, автоматическая остановка по отклонению
Доступ	Надзорная мера по Capital One	Штраф 80 миллионов долларов	Недостаточная оценка риска до переноса операций и несвоевременное исправление недостатков	Отдельное полномочие агента, минимальная область, ограниченный срок и учёт распространения данных
Эксплуатация	Ошибочное обновление конфигурации	Затронуты совместимые узлы, получившие	Ошибка валидатора и широкое распространение	Ограниченная начальная группа, поэтапное

| Канал | Наблюдаемый случай | Подтверждённый масштаб | Что не сработало | Ограничение для агента |

Из сопоставления возникает важное наблюдение. В каждом случае в организации действовали определённые механизмы контроля. Citibank применял разделение на три роли. В Knight Capital формировались автоматические сообщения. Capital One действовала в регулируемой банковской среде. CrowdStrike использовала валидатор содержимого и

ранее проверенный тип шаблона. Проблема заключалась не в полном отсутствии правил. Контроль не совпал с реальным механизмом распространения ошибки.

Для платежа механизмом была семантика полей и внешнее движение средств. Для торговли механизмом стала скорость автоматического исполнения. Для доступа им было изменение технологической среды и сохранение раскрытого знания. Для эксплуатации им стала массовая доставка однородной конфигурации. Следовательно, контроль агента должен строиться вокруг механизма воздействия, а не вокруг организационного названия процесса.

Введём простую расчётную модель для предварительного сравнения сценариев. Она не заменяет капитал под операционный риск и не является надзорной формулой. Её задача состоит в инженерной сортировке действий перед проектированием контроля.

$$P = B \times T \times C \times H$$

Здесь (P) обозначает относительный риск действия, (B) вероятность семантической ошибки, (T) тяжесть единичного последствия, (C) коэффициент скорости накопления, (H) коэффициент необратимости. Каждый множитель оценивается по утверждённой шкале внутри одной организации. Значения разных организаций нельзя сравнивать без общей калибровки.

Модель намеренно выделяет скорость и необратимость. Обычное произведение вероятности и ущерба может дать

одинаковую оценку единичному крупному платежу и серии малых операций. Но способы контроля будут разными. Для единичного платежа важны подтверждение получателя и суммы. Для серии важны лимит частоты, накопительный предел и остановка. Необратимость также меняет время контроля. Если результат легко отменить, часть проверки можно перенести после действия. Если окно закрывается быстро, проверка должна предшествовать исполнению.

Для бухгалтера к этой модели следует добавить стоимость доказательного восстановления. После инцидента организация должна восстановить цепочку документов, проводок, решений и полномочий. Если журнал неполон, трудозатраты растут, а правовая позиция слабеет. Эта стоимость не всегда видна в момент проектирования. Она появляется при сверке, аудите, споре с клиентом или запросе регулятора.

Назовём разницу между созданным последствием и собранным доказательством долгом доказуемости. Система создаёт такой долг, если выполняет действие сейчас, а основание, версии политик или подтверждение результата обещает записать позже. Для чтения малочувствительной справки небольшой временной разрыв может быть допустим. Для необратимого платежа долг должен быть близок к нулю. Доказательство формируется в той же транзакционной границе, что и допуск к действию.

Цена ошибки имеет ещё один слой: утрата доверия внутри организации. Если сотрудники не понимают, почему агент

выполнил действие, они начинают перепроверять все результаты. Экономия времени исчезает. Если проверка превращается в формальное нажатие кнопки, риск возвращается. Следовательно, интерес совета директоров не должен ограничиваться долей автоматизированных задач. Нужны показатели разрешённых действий, предотвращённых нарушений, величины потенциального радиуса, времени восстановления и полноты доказательств. Эти показатели будут разработаны в главах 8 и 11.

Четыре канала сходятся в одном наблюдении. Цена ошибки возникает не внутри модели, а в инфраструктуре полномочий. Модель предлагает или выбирает действие, тогда как масштаб определяют лимиты, скорость, связанность, режим исполнения и окно обратимости. Поэтому безопасность не может оставаться делом одной команды разработчиков. За неё вместе отвечают владелец процесса, бухгалтер, специалист по риску, архитектор, служба безопасности и внутренний аудит.

Теперь можно ответить на вопрос заглавия. Агент не сводится к языковой модели с кнопками, потому что кнопка описывает доступную функцию, а агент создаёт траекторию изменений. Его предметом становится не текст, а состояние обязательств, денег, прав и систем. Управлять такой системой означает управлять намерением, планом, контекстом, полномочием и доказательством каждого значимого перехода.

В этот момент становится ясно, почему одной точности модели недостаточно. Организация имеет дело сразу с деньгами, рыночной позицией, полномочиями и устойчивостью процесса. Для каждого объекта действует своя логика ущерба и восстановления. Значит, в следующей главе риск придётся разложить по источнику ошибки, ширине воздействия и возможности вернуть систему в допустимое состояние.

Глава 2. Модель риска автономного действия

2.1. Риск намерения, риск плана, риск инструмента и риск контекста

В начале рабочего дня бухгалтер получает просьбу оплатить просроченный счёт поставщика. Фраза выглядит определённой лишь до тех пор, пока её не требуется превратить в проводку и движение денег. Какое юридическое лицо должно платить? Какой договор породил обязательство? Подтверждена ли поставка? Относится ли просрочка к этому счёту или к другому документу? Не изменились ли реквизиты? Доступен ли период для отражения операции? Каждая проверка уточняет значение первоначальной просьбы. Для языкового помощника неоднозначность служит поводом задать вопрос. Для автономного агента она становится источником риска действия.

В первой главе риск рассматривался через последствия. Теперь нужно обратиться к происхождению ошибки. Фраза «модель ошиблась» слишком широка для инженерного решения. Она не показывает, где должен находиться контроль. Если система неверно поняла хозяйственную цель, провер-

ка программного интерфейса не поможет. Если цель понята верно, но план нарушает последовательность согласований, повторная генерация ответа не исправит архитектуру. Если план допустим, но инструмент изменил иной объект, требуется сверка результата. Если системе передали документ другого клиента, формально верное рассуждение остаётся недопустимым.

Предлагаемая модель разделяет четыре источника: риск намерения, риск плана, риск инструмента и риск контекста. Это авторская декомпозиция для систем, которые способны создавать последствия. Она не заменяет классификации NIST, банковские категории операционного риска или юридическую квалификацию. Её практическая задача состоит в том, чтобы связать каждый отказ с точкой наблюдения, владельцем контроля и доказательством.

Риск намерения возникает, когда системе поставлена неверная, неполная или недопустимая хозяйственная цель. Источник ошибки может находиться до модели. Пользователь просит «закрыть все долги», хотя часть обязательств оспаривается. Входящее письмо требует срочно заменить реквизиты, но не прошло независимую проверку. Показатель эффективности требует сократить очередь заявок, однако не определяет, какие проверки нельзя пропускать. Агент способен точно реализовать такую цель и усилить исходную ошибку.

Намерение нельзя отождествлять с текстом запроса. В ре-

гулируемом процессе оно включает субъект, объект, деловую цель, правовое основание, ограничения и критерий завершения. Для платежа недостаточно фразы об оплате. Нужна связь с обязательством, получателем, суммой, сроком и источником реквизитов. Для изменения доступа требуется связь с ролью сотрудника, задачей, сроком и владельцем данных. Для инвестиционного действия требуется портфельный мандат, инструмент, лимит, рынок и допустимый риск. Нормализация этих элементов не делает решение правильным, но превращает скрытую предпосылку в проверяемый объект.

Риск плана появляется после того, как цель уже определена. План может содержать неверную последовательность, пропущенную зависимость, недопустимое объединение ролей или ошибочное условие завершения. Бухгалтерская операция хорошо показывает эту разницу. Цель отразить подтверждённую поставку допустима. План, который сначала закрывает период, а затем пытается провести документ, нарушает порядок. Цель оплатить обязательство допустима. План, который меняет реквизиты и сразу отправляет платёж без независимого подтверждения, соединяет два действия, каждое из которых по отдельности может быть разрешено.

В агентной цепочке риск плана имеет накопительный характер. Ошибка раннего шага меняет наблюдение для позднего. Если система неверно сопоставила договор, последующий расчёт суммы может быть арифметически точным. Если

она получила неполный перечень документов, сверка может завершиться без расхождений внутри неполного набора. Если агент принял создание платёжного поручения за исполнение платежа, он способен закрыть задачу до расчёта. Поэтому план следует проверять не только до первого вызова. Его допущения нужно пересматривать после значимых наблюдений.

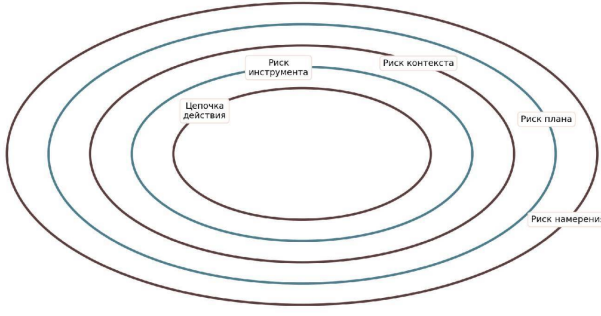
Риск инструмента относится к границе между решением и внешней системой. Он включает неверный выбор функции, ошибочные аргументы, несоответствие версии, неоднозначную семантику ответа, сбой идемпотентности и чрезмерное полномочие. Инструмент может вернуть технический успех, хотя хозяйственное действие ещё не завершилось. Он может принять дату в другом формате. Он может обработать повторный запрос как новую операцию. Он может позволить изменить больше полей, чем требовалось цели.

Часть риска инструмента детерминирована и потому должна контролироваться вне модели. Типы данных, диапазоны, допустимые значения, ключи идемпотентности, схемы ответа и ограничения частоты поддаются программной проверке. Нет основания просить языковую модель каждый раз решать, допустима ли отрицательная сумма платежа или существует ли получатель в утверждённом справочнике. Генеративная система полезна там, где нужно интерпретировать неоднозначный текст. Там, где правило можно выразить точно, его следует исполнять точно.

Риск контекста возникает, когда решение строится на неполных, устаревших, избыточных, чужих или неподтверждённых данных. Этот риск шире качества обучающей выборки. Агент работает с динамическим контекстом: письмом, договором, результатом поиска, историей чата, описанием инструмента, профилем пользователя и состоянием процесса. Каждый элемент имеет происхождение, срок актуальности и область допустимого использования.

Контекст является частью доказательства. Сумма из письма не равна сумме из подписанного акта. Реквизиты из карточки контрагента не равны реквизитам из неподтверждённого вложения. Остаток на начало дня не равен остатку в момент отправки платежа. Документ другого юридического лица может иметь сходное название, но не относится к операции. Если агент получает все эти фрагменты как равноправный текст, он теряет различие между сведением и основанием.

Связь четырёх рисков показана на рисунке. Стрелки обозначают не хронологию, а взаимное усиление. Неверный контекст может изменить намерение. Ошибка инструмента создаёт новое наблюдение и заставляет перестроить план. План выбирает, какой контекст будет запрошен. Намерение определяет, какое действие считается завершением.



Такое разделение согласуется с логикой Рамочной системы управления рисками искусственного интеллекта NIST, хотя вводит иную единицу анализа. NIST предлагает функции управления, картирования, измерения и обработки риска. При картировании организация должна документировать деловой контекст, границы знаний, область применения, издержки ошибок, человеческий надзор и риски компонентов, включая внешние данные и программное обеспечение. В части измерения требуется оценивать систему в условиях, сходных со средой применения, документировать метрики и отслеживать возникающие риски. Рамочная система NIST является добровольной и не создаёт отраслевого разрешения на финансовое действие.

Регламент Европейского союза 2024/1689 также показы-

вает, что риск нельзя считать свойством только модели. Статья 9 требует для систем высокого риска непрерывного итеративного процесса управления риском в течение жизненного цикла. Статья 12 требует технической возможности автоматической регистрации событий. Статья 14 требует проектировать эффективный человеческий надзор. Применимость конкретных статей зависит от роли организации, назначения и классификации системы. Поэтому эти положения нельзя автоматически распространить на каждый финансовый агент. Они важны как нормативное подтверждение жизненного цикла, журналирования и наблюдаемого надзора. Текст Регламента 2024/1689 позволяет проверить формулировки и область действия.

Четыре риска требуют разных проверочных вопросов. Таблица связывает источник ошибки с моментом контроля. Она составлена как инженерная схема и не является регуляторной классификацией.

Источник	Проверочный вопрос	Наблюдаемый признак	Момент контроля	Основное доказательство
Намерение	Какое хозяйственное состояние должно измениться и на каком основании	Цель не связана с обязательством, мандатом или ролью	До планирования и при изменении цели	Нормализованная цель, субъект, объект, основание и ограничения
План	Ведёт ли последовательность к цели допустимым путём	Пропуск согласования, неверный порядок, конфликт ролей	До исполнения и после значимого наблюдения	Версия плана, зависимости, условия остановки и результаты проверок
Инструмент	Соответствуют ли функция, параметры и ответ ожидаемой семантике	Неверный объект, повтор операции, ложное подтверждение завершения	Перед вызовом и после ответа	Схема вызова, параметры, ключ повторяемости и фактический результат
Контекст	Можно ли использовать эти данные для данной цели сейчас	Устаревший документ, смешение клиентов, неподтверждённый источник	При получении каждого фрагмента и перед действием	Происхождение, время, владелец, классификация и назначение данных

Практическая ценность декомпозиции проявляется при расследовании. Если ошибочный платёж отнести к общему «риску модели», команда может заменить модель или изменить инструкции модели. Однако корень может находиться в контексте реквизитов, в политике допуска, в семантике статуса банка или в плане, который объединил изменение справочника и перечисление денег. Исправление не того слоя создаёт видимость реакции и сохраняет механизм отказа.

Для каждой операции полезно проводить разбор в обратном порядке. Сначала устанавливается фактическое изменение состояния. Затем проверяется ответ инструмента и переданные параметры. После этого восстанавливается план с

версиями и развилками. Далее определяется контекст, использованный на каждом шаге. В конце сопоставляется нормализованное намерение с первоначальным запросом и деловым основанием. Такая последовательность не предполагает вины модели. Она ищет место, где система потеряла соответствие между смыслом и состоянием.

Из этого возникает критерий полноты контроля. Риск считается локализованным только тогда, когда для него известны объект наблюдения, условие отказа, точка остановки, владелец решения и сохраняемое доказательство. Простого списка опасностей недостаточно. Реестр, который нельзя связать с исполняемой цепочкой, описывает проблему после события.

Риск автономного действия удобнее читать как историю сбоя, а не как одно число. Ошибка может родиться в намерении, плане, инструменте или контексте, после чего несколько причин усиливают друг друга. Такая картина сразу меняет вопрос инженера. Нужно выяснить не только вероятность промаха, но и то, насколько далеко он успеет распространиться.

2.2. Blast radius и необратимость как базовые переменные

Когда бухгалтер исправляет черновик проводки, ошибка обычно остаётся внутри одного документа. Когда то же пра-

вило применяется к пакету закрытия периода, она затрагивает множество счетов и отчётов. Когда результат уходит в регуляторную отчётность или расчёт с контрагентом, исправление требует нового события, а не простого удаления. Разница между этими ситуациями состоит не в красноречии модели. Её создают радиус воздействия и необратимость.

Термин «blast radius» пришёл из инженерного языка аварий и безопасности. В этой книге он обозначает множество объектов, процессов и сторон, на которые способно повлиять одно ошибочное решение до остановки. Русское выражение «радиус воздействия» передаёт смысл точнее, когда речь идёт не только о техническом сбое, но также о деньгах, правах, отчётности и клиентах. В названии подраздела использовано русское выражение «радиус воздействия».

Радиус имеет несколько измерений. Количественное измерение отвечает на вопрос о числе затронутых объектов. Денежное показывает предел суммы или позиции. Организационное охватывает юридические лица, подразделения и роли. Временное показывает, сколько периодов или будущих действий зависят от результата. Сетевое отражает распространение через поставщиков, рынки и общие сервисы. Правовое измерение учитывает число обязательств и субъектов, чьи права могут быть затронуты.

Одно число редко описывает все измерения честно. Сто платежей по тысяче рублей имеют иной профиль, чем один платёж на сто тысяч. Сумма совпадает, но различаются чис-

ло получателей, возможность возврата, стоимость расследования и вероятность повторяющейся причины. Один неверный реквизит в справочнике способен влиять на будущие платежи до обнаружения. Поэтому радиус нужно хранить как вектор лимитов, а не как общий балл.

Для агента важен потенциальный радиус, доступный до обнаружения. Фактически затронутое множество становится известно после события. Проектировщик должен оценить верхнюю границу заранее: сколько вызовов агент способен сделать, какую сумму накопить, к скольким клиентам обратиться, какой объём данных прочитать и сколько систем изменить до принудительной остановки. Если эта граница определяется только терпением модели или сроком действия общего токена, система не имеет управляемого радиуса.

Необратимость описывает стоимость и полноту возвращения к допустимому состоянию. Её нельзя свести к ответу «да» или «нет». Платёж можно попытаться вернуть, но время, комиссии, спор и риск неполного возврата сохраняются. Право доступа можно отозвать, но прочитанные сведения остаются раскрытыми. Проводку можно сторнировать, однако закрытый период и выпущенная отчётность создают новые обязательства. Сообщение можно удалить из внутренней очереди, но нельзя гарантировать удаление после отправки внешнему адресату.

Поэтому следует различать техническую отмену и хозяйственное восстановление. Техническая отмена возвращает

состояние конкретной системы. Хозяйственное восстановление возвращает согласованность денег, прав, документов и отчётности. Отмена записи в базе не возвращает деньги. Сторно не делает исходную проводку несуществующей. Отзыв роли не отменяет факт доступа. Для регулируемого агента критерием служит не наличие функции отмены, а подтверждённое восстановление доменного состояния.

Радиус и необратимость образуют пространство решений, представленное ниже. Положение точек отражает качественную классификацию автора, а не статистику. Действие не закреплено навсегда в одной координате. Контекст способен переместить его: черновик становится внешним обязательством после публикации, а локальное изменение доступа приобретает сетевой масштаб при использовании общего профиля.

Инцидент Knight Capital даёт проверяемый пример быстрого расширения радиуса. Комиссия по ценным бумагам и биржам США установила, что за период около 45 минут система создала около четырёх миллионов исполнений по акциям 154 компаний общим объёмом более 397 миллионов акций. Реализованный убыток составил 460 миллионов долларов. Эти показатели описывают фактическое последствие конкретного программного отказа, а не языкового агента. Для нашей модели важен механизм: автоматическая скорость превратила ошибку одной конфигурации в портфель множества рыночных позиций. Постановление Комиссии служит первичным источником чисел.

Из опубликованных данных можно получить показатель плотности воздействия. Разделим число исполнений на число затронутых акций:

$d = 4\,000\,000 / 154 \approx 25\,974$

В среднем на одну позицию в перечне из 154 акций пришлось около 25 974 исполнений. Эта величина не раскрывает распределение и не описывает отдельную ценную бумагу. Она показывает, что радиус имел одновременно ширину по числу инструментов и плотность внутри каждого инструмента. Для агента аналогом будет не только число клиентов, но и число действий на клиента. Ограничение одного измерения не заменяет ограничение другого.

Случай Citibank показывает необратимость другого рода. Судебное решение фиксирует перевод 894 миллионов долларов основного долга и 7,8 миллиона долларов процентов. Около 385 миллионов долларов было возвращено, а получатели суммы примерно в 500 миллионов долларов первоначально отказались вернуть средства. Если сопоставить возвращённую сумму только с основной суммой, получим приблизительную долю:

$$k = 385 / 894 \times 100\ \% \approx 43{,}1\ \%$$

Расчёт не описывает окончательный финансовый исход дела и не включает проценты. Он показывает состояние возврата, изложенное в судебном решении на соответствующем этапе. Более половины основной суммы не было добровольно возвращено сразу. Следовательно, функция перевода не имела симметричной функции отмены. Решение Апелляционного суда второго округа содержит суммы и процессуальную историю.

Для бухгалтерского контроля оба случая дают общий урок. Радиус измеряется до исполнения, а восстановление подтверждается после него. До исполнения нужны накопительные лимиты, область объектов и время действия допуска. После исполнения нужны сверка, подтверждение внешней стороны и оценка остаточного последствия. Если система умеет только остановить будущие вызовы, она ограничивает рост, но не восстанавливает уже изменённое состояние.

В нормативном языке сходную логику можно увидеть в

принципах операционной устойчивости Базельского комитета. Банк должен определить критические операции, установить допустимый уровень нарушения и картировать зависимости, необходимые для выполнения этих операций. Документ не посвящён агентам, однако его единица анализа полезна: организация защищает непрерывность критической операции, а не отдельный компонент. Агент может работать исправно как программный компонент и одновременно нарушать критическую операцию неверной последовательностью допустимых вызовов. Принципы операционной устойчивости задают основу такого взгляда.

Совет по финансовой стабильности в 2024 году выделил зависимости от внешних поставщиков, рыночные корреляции, киберриск, риск моделей, качество данных и управление как уязвимости, через которые искусственный интеллект способен усиливать системный риск. Этот вывод переводит радиус с уровня одной организации на уровень финансовой сети. Если многие участники используют общую модель, общий источник данных или сходную стратегию, отдельные действия могут стать коррелированными. Материал Совета по финансовой стабильности прямо перечисляет эти каналы.

Для проектирования полезно разложить радиус и необратимость на наблюдаемые параметры. Таблица показывает минимальный набор. Значения должны извлекаться из реального процесса, договоров, архитектуры и статистики операций. Их нельзя назначать по впечатлению разработчика.

Измерение	Единица	Верхняя граница до действия	Проверка после действия	Возможный ограничитель
Денежное	Валюта расчёта	Сумма одного действия и накопительная сумма	Фактическое списание, позиция или обязательство	Лимит суммы, дневной предел, резерв
Количественное	Число объектов и вызовов	Число клиентов, счетов, узлов или документов	Реальный перечень затронутых объектов	Пакет, квота, ограничение частоты
Временное	Минуты, часы, отчётные периоды	Срок допуска и продолжительность цепочки	Время обнаружения и восстановления	Истечение допуска, контрольная точка
Организационное	Юридические лица, роли, подразделения	Разрешённая область субъекта	Владельцы затронутых процессов	Изоляция полномочий и данных
Сетевое	Поставщики, рынки, внешние стороны	Число зависимостей и каналов распространения	Подтверждения внешних систем	Начальная группа, этапность, разрыв контура
Необратимость	Стоимость, время и полнота восстановления	Окно отмены и доступный способ компенсации	Остаточное последствие после исправления	Задержка исполнения, двойное подтверждение, запрет

Таблица позволяет сформулировать правило размещения контроля. При малом радиусе и сохранённом окне восстановления допустима проверка после действия. По мере роста одного из измерений контроль смещается к моменту допуска. При сочетании широкого радиуса и высокой необратимости одного согласования недостаточно. Требуются технический предел, независимая проверка и автоматическая

остановка по фактическому состоянию.

Из правила не следует запрет автономности. Напротив, управляемый радиус позволяет передавать агенту больше повторяющейся работы, поскольку организация заранее ограничивает последствия единичной ошибки. Вместо общего вопроса «можно ли доверять модели» появляется проверяемый вопрос «какое состояние она способна изменить до остановки». Ответ на него можно выразить в политике и измерить в эксплуатации.

У безопасности здесь есть две практические координаты. Первая отвечает за ширину затронутой области. Вторая показывает, сколько усилий потребуется для возвращения к допустимому состоянию. Вместе они превращают неопределённую тревогу перед автономностью в ограничиваемый ресурс. Теперь можно спросить, кому, на какой срок и в каком объёме этот ресурс допустимо выдавать.

2.3. Автономность как бюджет, а не как фича

В интерфейсе программного продукта автономность часто выглядит как переключатель. Пользователь разрешает системе действовать самостоятельно или оставляет подтверждение каждого шага. Для регулируемого финтеха такая двоичная модель слишком груба. Она смешивает разные полномочия: прочитать данные, сформировать проект, из-

менить запись, отправить сообщение, провести платёж и перестроить план после ошибки. Разрешив «автономный режим», организация не понимает, какой риск она выдала системе и когда это право закончится.

Предлагаемый подход рассматривает автономность как бюджет. Бюджет здесь означает заранее установленный набор пределов, внутри которых агент может действовать без нового решения человека. Он расходуется при выполнении цепочки, уменьшается после каждого значимого действия и прекращается при достижении любого ограничения. Бюджет относится не к модели вообще, а к конкретному субъекту, цели, плану, набору объектов и промежутку времени.

Слово «бюджет» важно по трём причинам. Первая состоит в конечности ресурса. Полномочие не должно существовать бессрочно только потому, что агент однажды прошёл проверку. Вторая связана с распределением между видами ресурса. Денежный предел не заменяет ограничение числа действий, а срок не заменяет ограничение данных. Третья причина состоит в наблюдаемости расхода и возможности объяснить его владельцу процесса на языке операции.

Бюджет автономности включает по крайней мере пять компонентов. Денежный компонент ограничивает единичную и накопительную сумму. Количественный задаёт число вызовов, документов, клиентов или объектов. Временной определяет срок полномочия и продолжительность цепочки. Контекстный ограничивает объём и классы доступных дан-

ных. Плановый устанавливает, насколько агент вправе изменять утверждённую последовательность без нового допуска.

Эти компоненты нельзя свободно обменивать. Остаток денежного лимита не даёт права прочитать больше персональных данных. Неиспользованное время не разрешает добавить нового получателя. Малое число действий не делает допустимым изменение роли администратора. Поэтому бюджет представляет собой вектор, а остановка происходит, когда исчерпан любой критический компонент.

Обозначим бюджет задачи как множество:

Б = {Д, К, В, О, П}

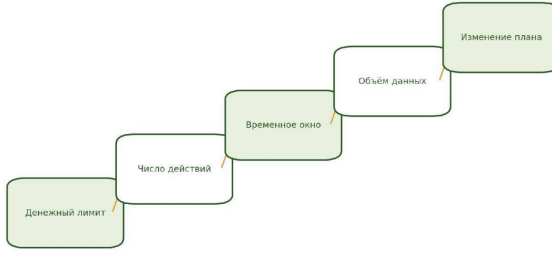
Здесь (Д) обозначает денежные пределы; (К) соответствует количественным пределам; (В) обозначает временные пределы; (О) задаёт область данных и объектов; (П) определяет свободу изменения плана. После каждого действия система вычисляет остаток по каждому компоненту. Для количественных величин можно использовать простое выражение:

Б_i^ост = Б_i^нач − Σ(j=1…n) p_ij

В этой формуле (Б_{i}^{нач}) является начальным пределом компонента (i), а (p_{ij}) обозначает расход этого компонента действием (j). Для областей данных и прав вместо вычитания применяется проверка принадлежности разрешённому множеству. Эту формулу я использую как конструкцию управления допуском. Она не предназначена для расчёта регуляторного капитала или бухгалтерского бюдже-

та.

Состав бюджета ниже дан без условных чисел. Все компоненты сходятся в одном решении: хватает ли остатка для действия и сохранилась ли исходная цель. Числовой шкалы здесь нет сознательно. Реальные пределы должны опираться на политику организации и статистику конкретного процесса.



Для бухгалтерского процесса бюджет можно связать с объектом учёта. Агент получает право обработать определённый реестр документов конкретного юридического лица до установленного времени. Он может сопоставлять договоры и акты, создавать проекты проводок и формировать перечень исключений. Право провести документ выдаётся от-

дельно. Право сформировать платёж не включает право изменить карточку получателя. Право отправить платёж ограничивается утверждённой суммой и идентификатором обязательства.

Такое разделение не означает постоянного ручного согласования. Напротив, человек утверждает границы пакета, а не каждую повторяющуюся операцию. Если реестр однороден, источники подтверждены, суммы укладываются в лимиты, а фактические результаты сверяются, агент способен обработать весь пакет. При появлении нового получателя, расхождения документа, закрытого периода или изменения плана бюджет соответствующего компонента обнуляется. Задача переходит в режим уточнения.

Ключевое различие проходит между назначением бюджета и его расходованием. Назначение является решением о риске. Его принимает владелец полномочия или исполняемая политика на основании утверждённых правил. Расходование является техническим фактом. Его фиксирует система при каждом действии. Модель может предложить размер бюджета, но не должна сама увеличивать собственный предел в ходе той же цепочки.

Из этого следует требование разделения ролей. Компонент, который планирует действие, не должен единолично выдавать себе дополнительную сумму, новые данные или продление времени. Если возникла необходимость расширения, формируется новый запрос допуска с объяснением из-

менения. Старый бюджет сохраняется в журнале и не переписывается. Такая конструкция препятствует постепенному расширению полномочия через серию малых решений.

Подход, основанный на оценке риска, поддерживается действующими руководствами, хотя понятие бюджета автономности является авторским. В апреле 2026 года Федеральная резервная система США опубликовала пересмотренное руководство по управлению модельным риском. Оно заменило письма 2011 и 2021 годов и подчёркивает настройку управления с учётом профиля моделей, размера и сложности операций банка. Положения письма особенно значимы для регулируемых Федеральной резервной системой банковских организаций с активами свыше 30 миллиардов долларов. Письмо SR 26 2 содержит дату, область релевантности и указание на адаптацию контроля к уровню риска.

Эту логику нельзя переносить механически на каждую организацию и каждое действие. Она подтверждает принцип соразмерности. Интенсивность контроля зависит от использования, профиля риска и сложности. Бюджет автономности делает соразмерность исполняемой. Вместо общего класса «высокий риск» система получает конкретные пределы, которые проверяются перед вызовом и уменьшаются после него.

Рамочная система NIST также связывает объём управления с допустимостью риска организации. Функция управления требует определить необходимый уровень деятельно-

сти по управлению риском с учётом допустимого для организации риска. Функция обработки предлагает приоритизировать ответы по воздействию, вероятности и доступным ресурсам, а также предусматривать отключение систем, результаты которых не соответствуют назначению. Рамочная система NIST поддерживает идею конечного и наблюдаемого допуска, хотя не задаёт предложенный здесь вектор.

Компоненты бюджета переведены в наблюдаемые условия в следующей таблице. Она не содержит универсальных чисел. Каждый предел должен иметь источник: положение о полномочиях, платёжный регламент, матрицу доступа, договор, календарь закрытия, статистику инцидентов или решение органа управления.

Компонент	Что ограничивается	Источник реального предела	Событие расхода	Условие прекращения
Денежный	Сумма одного действия и накопительная сумма	Платёжный регламент, мандат, лимит позиции	Создание обязательства или подтверждённое исполнение	Превышение суммы, валюты, получателя или накопительного предела
Количественный	Число документов, клиентов, вызовов и изменений	Параметры процесса и допустимый радиус	Каждый вызов с побочным эффектом	Достижение квоты или неожиданный рост повторов
Временной	Срок полномочия и длительность цепочки	Рабочее окно, статус периода, срок согласования	Течение времени и смена процессного состояния	Истечение срока, закрытие периода, изменение статуса
Контекстный	Классы данных, юридические лица и цели обработки	Классификация данных и матрица полномочий	Получение или передача нового фрагмента	Выход за область, смена цели, потеря актуальности
Плановый	Допустимые развилки и число перестроений	Утверждённый шаблон процесса и политика риска	Изменение шага, порядка, инструмента или критерия завершения	Появление неутверждённой ветви или изменение экономического смысла

У бюджета должен быть владелец. Для платежа им может быть руководитель казначейства в пределах утверждённой матрицы. Для закрытия периода это владелец бухгалтерского процесса. Для доступа это владелец данных и служба безопасности. Для эксплуатации это владелец сервиса. Техническая платформа исполняет решение, но не подменяет хозяйственную ответственность.

Для бюджета должно действовать правило аннулирования остатка. Неиспользованное полномочие после завершения цели не является активом агента. Оно аннулируется. Повторная задача получает новый допуск, даже если параметры сов-

падают. Это правило мешает превращать разовый доступ в постоянную скрытую роль. Оно также облегчает аудит, поскольку каждый вызов связан с конкретной целью и сроком.

При таком устройстве автономность перестаёт быть маркетинговой характеристикой продукта. Она становится режимом конкретной транзакции. Один и тот же агент может автономно читать открытые справочники, готовить проекты проводок в заданном пакете и требовать подтверждения для внешнего платежа. Различается не интеллект, а выданный бюджет допустимого воздействия.

Бюджет автономности ограничивает право на действие, но не наделяет план хозяйственным смыслом. Система способна аккуратно уложиться во все лимиты и всё же оплатить не то обязательство. Поэтому после разговора о пределах приходится перейти к качеству выбранной траектории. Именно там обнаруживаются планы, которые безупречны по форме и ошибочны по существу.

2.4. Planning failures : когда план формально корректен, но семантически ошибочен

План может пройти все технические проверки и всё же привести организацию не туда. Типы параметров совпадут со схемой. Порядок вызовов будет допустим. Сумма останется в лимите. Токен окажется действительным. Каждый

инструмент вернёт успешный статус. Ошибка обнаружится позже, когда бухгалтер увидит, что операция относится не к тому обязательству, период закрыт на неверном основании или платёж отправлен по документу, который не подтверждал поставку.

Такой отказ будем называть семантической ошибкой плана. Формальная корректность отвечает на вопрос, можно ли исполнить последовательность. Семантическая корректность отвечает на другой вопрос: ведёт ли эта последовательность к хозяйственной цели на допустимом основании и сохраняет ли смысл цели после каждого наблюдения. Различие знакомо любой регулируемой организации. Правильно заполненная форма не доказывает законность операции. Наличие подписи не доказывает, что подписан нужный документ. Сходство суммы не доказывает связь с обязательством.

Семантическая ошибка часто возникает в промежутке между двумя системами описания. Пользователь говорит языком цели. Модель преобразует цель в шаги. Программный интерфейс принимает поля. Учётная система фиксирует документ. Банк исполняет перевод. На каждой границе часть смысла сворачивается в идентификатор, код или статус. Если связь не сохранена, поздний шаг остаётся формально валидным и теряет исходное основание.

Рассмотрим типовые формы такого отказа. Первая состоит в подмене объекта. Агент находит документ с совпадающими суммой и названием контрагента, но иной датой или

юридическим лицом. Вторая связана с подменой состояния. Система считает созданное поручение исполненным платежом. Третья возникает при подмене основания. Письмо используется как доказательство изменения реквизитов, хотя процесс требует независимого подтверждения. Четвёртая появляется при подмене критерия завершения. Агент закрывает задачу после успешного вызова, хотя хозяйственный результат ещё не сверён.

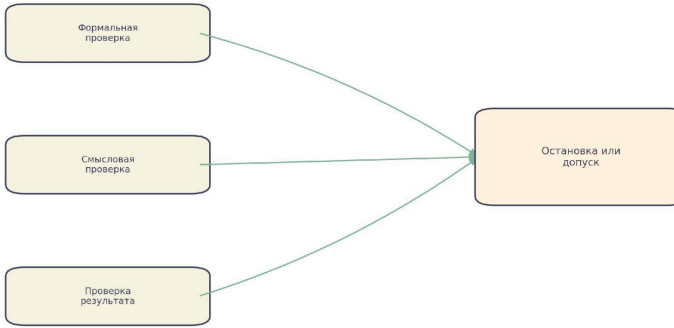
Пятая форма связана с локальной оптимизацией. Цель сократить просрочку превращается в выбор платежей по возрасту, но план не учитывает спорность обязательства, санкционные проверки, доступность ликвидности или приоритет критического поставщика. Все шаги могут быть разрешены, однако выбранная последовательность противоречит более широкой политике. Шестая форма возникает при соединении допустимых действий. Изменить справочник получателя и оплатить счёт можно при разных условиях. Выполненные одной цепочкой на основании одного источника, они устраняют независимость контроля.

Седьмая форма появляется после неожиданного наблюдения. Агент получает отказ инструмента и выбирает обходной путь, который формально доступен. Например, он меняет способ отражения операции, выбирает другой счёт или повторяет действие через иной канал. Локальная цель сохраняется, но причина отказа не исследована. Отказ, который должен был остановить цепочку, превращается в сигнал для

поиска обхода.

Проблема не решается требованием показать рассуждение модели. Текстовое объяснение может быть убедительным и не соответствовать фактической причинной цепочке. Оно также способно рационализировать уже выбранное действие. Для контроля нужен план как структурированный объект: шаги, зависимости, ожидаемые состояния, источники данных, проверяемые предпосылки, условия остановки и допустимые развилки.

Три независимые проверки образуют путь к решению. Первая устанавливает типы и порядок. Вторая связывает цель с основанием. Третья сопоставляет ожидаемое экономическое состояние с фактическим. Отказ на любом шаге ведёт к остановке или новому допуску. Общая стрелка не означает, что все проверки выполняет один компонент.



Формальная проверка должна быть детерминированной. Она проверяет существование шага, схему аргументов, типы, разрешённые зависимости, отсутствие циклов без предела, срок действия допуска и совместимость версий. Если операция требует сначала получить документ, план не может ссылаться на его идентификатор до соответствующего шага. Если политика запрещает одному субъекту менять получателя и отправлять платёж, граф должен содержать независимую границу.

Смысловая проверка требует доменной модели. Для бухгалтерской операции она сопоставляет документ, обязательство, период, юридическое лицо, валюту, налоговый режим и полномочие. Проверка не обязана быть полностью генеративной. Значительную часть можно выразить предикатами: идентификатор договора связан с контрагентом, сумма документа согласована, период открыт, источник реквизитов подтверждён, назначение соответствует типу обязательства. Модель нужна для извлечения и сопоставления неструктурированного смысла, но решение допуска принимает политика над проверенными атрибутами.

Проверка результата замыкает цепь. До действия план содержит ожидаемое состояние. После действия система получает независимое наблюдение и сравнивает его с ожиданием. Для проводки это затронутые счета и суммы. Для платежа

это фактический статус, получатель и списание. Для доступа это эффективное разрешение, а не только ответ на запрос изменения. Расхождение не следует автоматически исправлять новой импровизацией. Оно меняет состояние риска и требует остановки либо нового плана.

Профиль генеративного искусственного интеллекта NIST выделяет конфабуляцию как отдельный риск и рекомендует проверять заявления о возможностях эмпирическими методами, избегать переноса выводов из узких и несистематических оценок, а также проверять источники и ссылки в выходах системы во время испытаний и мониторинга. Эти рекомендации относятся к генеративным системам в широком смысле. Для агентного плана из них следует более узкое требование: утверждение модели не становится предпосылкой действия, пока не связано с проверяемым источником или доменным правилом. Профиль NIST содержит перечисленные действия.

Судебный материал по переводу Citibank показывает семантический отказ без генеративной модели. Три роли участвовали в процессе, но одинаково поняли назначение полей интерфейса. Предупреждение сообщало об уходе средств, однако не показывало сумму. Формальные элементы контроля существовали, а экономический смысл внешнего списания не был представлен проверяющим. Решение Апелляционного суда позволяет восстановить эти обстоятельства.

Из этого случая не следует, что автоматическая проверка всегда надёжнее человека. Следует другое: независимость роли не равна независимости гипотезы. Если создатель, проверяющий и агент видят одну форму и одну интерпретацию, три подтверждения могут воспроизвести одну ошибку. Смысловой контроль должен получать иной признак. В платеже таким признаком является ожидаемая сумма внешнего движения, рассчитанная независимо от выбранных полей.

Для проверки планов предлагается использовать набор инвариантов. Инвариант представляет условие, которое должно оставаться истинным на всём пути. Бухгалтерское равенство дебета и кредита является известным примером, но его недостаточно: ошибочная проводка тоже способна быть сбалансированной. Дополнительные инварианты связывают документ с юридическим лицом, обязательство с получателем, изменение справочника с независимым подтверждением, сумму пакета с лимитом, а закрытие задачи с фактической сверкой.

Типы смысловой ошибки и способы их обнаружения сопоставлены в следующей таблице. Она составлена автором на основе анализа процессов и указанных источников.

Тип ошибки	Формально допустимое действие	Потерянный смысл	Проверочный инвариант	Реакция
Подмена объекта	Обработка существующего документа	Документ относится к другому обязательству или лицу	Объект связан с утверждённым основанием и субъектом	Остановка и уточнение сопоставления
Подмена состояния	Успешный ответ программного интерфейса	Запрос принят, но хозяйственный результат не наступил	Завершение подтверждено доменным статусом и сверкой	Ожидание либо расследование
Подмена основания	Использование доступного источника	Источник не имеет требуемой доказательной силы	Каждый критический атрибут имеет допустимое происхождение	Запрос подтверждённого источника
Локальная оптимизация	Выбор действия в пределах лимита	Игнорируется ограничение процесса или портфеля	Локальная цель не нарушает общую политику	Перестроение с новым допуском
Соединение ролей	Последовательность разрешённых вызовов	Один субъект меняет условие и исполняет зависимое действие	Конфликтующие полномочия разделены независимой границей	Передача решения другой роли
Обход отказа	Повтор через доступный канал	Причина запрета не устранена	Отказ класса контроля запрещает альтернативный маршрут	Остановка и разбор причины
Ложное завершение	Закрытие после последнего вызова	Ожидаемое состояние не сопоставлено с фактическим	Задача завершается только после доказательного равенства	Сверка и фиксация расхождения

Последняя строка вводит понятие доказательного равенства. Оно достигается, когда намерение, разрешённый план, фактическое действие и наблюдаемое хозяйственное состояние связаны одной воспроизводимой цепочкой. Равенство не означает буквальное совпадение текстов. Оно означает отсутствие необъяснённого смыслового разрыва между тем, что требовалось, что разрешили, что исполнили и что полу-

чилось.

Доказательное равенство можно проверять на каждом значимом переходе. Перед действием ожидаемое состояние выводится из цели и плана. После действия фактическое состояние получается из независимого источника. Политика сравнивает существенные атрибуты. Если получатель, сумма, объект, статус или период отличаются, цепочка не продолжает работу. В журнал попадает не только ошибка, но и конкретное нарушенное равенство.

Этот подход меняет предмет тестирования. Обычный сценарий спрашивает, выполнил ли агент задачу. Сценарий смысловой проверки спрашивает, сохранил ли он основание задачи при изменении контекста и ответов инструментов. Можно намеренно подать два похожих документа разных юридических лиц, промежуточный статус вместо итогового, устаревшие реквизиты рядом с подтверждёнными или отказ, который легко обойти. Успехом считается не завершение любой ценой, а корректная остановка при невозможности доказать смысл.

На уровне всей главы происхождение ошибки соединяется с последствием. Риск намерения, плана, инструмента и контекста становится управляемым только при конечном радиусе, учёте необратимости и расходуемом бюджете автономности. Семантические инварианты удерживают цель внутри этих границ. Вместе элементы образуют модель автономного действия, которую можно перевести в архитектуру,

политики и испытания.

Так складывается полная модель автономного риска. Источник ошибки объясняет, где возникло отклонение. Радиус и необратимость описывают возможное последствие. Бюджет сдерживает распространение, а смысловые инварианты не позволяют технически допустимому плану подменить хозяйственную цель. Дальше эту модель нужно перевести из языка анализа в язык норм, политик и доказательств.

Глава 3. Регуляторная инженерия вместо «compliance after the fact»

3.1. AI governance, финансовые нормы и operational resilience

В финансовой организации новый агент редко встречает пустое нормативное поле. До его появления уже существуют правила платежей, управления доступом, аутсорсинга, защиты данных, модельного риска, информационной безопасности, внутреннего контроля и непрерывности деятельности. Команда проекта обычно добавляет к этому перечню рамочную систему управления искусственным интеллектом. Проблема возникает не из отсутствия документов, а из того, что каждый документ описывает иной объект и говорит с иной профессиональной ролью.

Специалист по управлению искусственным интеллектом спрашивает о назначении модели, данных, ограничениях, тестировании и человеческом надзоре. Финансовая функция спрашивает об обязательстве, полномочии, проводке, лимите и отчётности. Служба операционной устойчивости спрашивает о критической операции, зависимости, допустимом нарушении и восстановлении. Агент соединяет все три об-

ласти в одном действии. Он интерпретирует цель с помощью модели, изменяет финансовое состояние и зависит от цифровой цепочки исполнения.

Если области проверяются последовательно, между ними образуются разрывы. Модель может пройти оценку качества, а платёжный процесс получить разрешение на автоматизацию. Инфраструктура может пройти испытание восстановления. Но никто не проверит, разрешено ли этой версии модели использовать этот контекст для этого плана и этой суммы в данном состоянии процесса. Каждый контроль будет правильным внутри своей дисциплины, а совокупное действие останется без владельца.

В этой книге регуляторная инженерия означает перевод правовых, надзорных и внутренних требований в архитектурные ограничения, исполняемые политики, события контроля и доказательства. Термин не предполагает, что право можно полностью превратить в код. Часть норм требует толкования и человеческого решения. Инженерная задача состоит в другом: явно определить, какая часть требования должна влиять на систему до действия, какая проверяется после него и какие сведения позволяют доказать соблюдение.

Первый контур образует управление искусственным интеллектом. Рамочная система NIST организует работу через функции управления, картирования, измерения и обработки риска. Она требует понимать законодательные требования,

определять роли, документировать деловой контекст, оценивать систему в условиях применения, отслеживать новые риски и предусматривать отключение при несоответствии назначению. Этот подход полезен для агента, поскольку не ограничивается показателем точности модели. Рамочная система NIST остаётся добровольной и не подменяет обязательное финансовое регулирование.

Регламент Европейского союза 2024/1689 создаёт обязательства для определённых участников и категорий систем. Его применимость должна устанавливаться юридически для конкретного назначения. Для систем высокого риска статья 9 требует документированного и поддерживаемого процесса управления риском на протяжении жизненного цикла. Статья 12 требует технической возможности автоматической регистрации событий. Статья 14 закрепляет эффективный человеческий надзор. Эти требования не дают универсального разрешения агенту действовать. Они показывают, что риск, журнал и надзор должны быть свойствами системы, а не дополнением к отчёту о внедрении. Регламент 2024/1689 является первичным правовым источником.

Второй контур образуют финансовые нормы. Они зависят от юрисдикции, лицензии, продукта и роли организации. Здесь нельзя составить один мировой список и объявить его полным. Платёжная организация, банк, инвестиционная компания и поставщик технологии несут разные обязанности. Даже внутри банка требования меняются между плате-

жом, кредитованием, торговлей, поддержкой клиента и внутренней эксплуатацией. Поэтому архитектура должна принимать правовую применимость как входной атрибут, а не зашивать одну классификацию в модель.

Обновление американского руководства по модельному риску в апреле 2026 года показывает движение к соразмерности, основанной на оценке риска. Совместное руководство Федеральной резервной системы, OCC и FDIC заменило прежние письма и подчёркивает настройку управления с учётом профиля моделей, размера и сложности операций. Его нельзя считать единым правилом для всех стран или всех агентов. Для нашей архитектуры важен принцип: интенсивность независимой проверки и мониторинга связана с фактическим использованием, а не только с названием технологии. Письмо Федеральной резервной системы SR 26 2 фиксирует дату, область релевантности и замену прежнего руководства.

Третий контур образует операционная устойчивость. Регламент Европейского союза 2022/2554 устанавливает требования к управлению риском информационных технологий, сообщениям об инцидентах, испытаниям устойчивости и управлению риском внешних поставщиков для охватываемых финансовых организаций. Его вводная часть подчёркивает взаимосвязанность финансового сектора и возможность распространения локального цифрового инцидента через финансовые каналы. Регламент 2022/2554 содержит

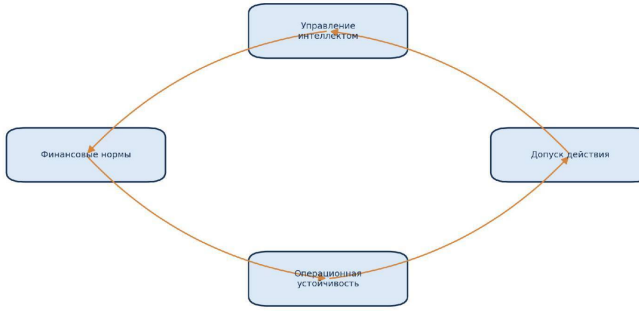
полный перечень субъектов и обязательств.

Базельский комитет предлагает смотреть на способность банка продолжать критические операции при нарушении. Организация определяет критические операции, допустимый уровень нарушения, зависимости и сценарии отказа. Агентное действие должно входить в эту карту как источник решения и как зависимость. Если агент участвует в обработке платежей, недостаточно восстановить модельный сервис. Нужно знать, может ли платежная операция продолжаться вручную или через иной маршрут, сохранены ли очереди, не повторятся ли уже отправленные поручения и можно ли воспроизвести состояние допуска. Принципы операционной устойчивости дают надзорную основу.

Совет по финансовой стабильности в июне 2026 года опубликовал консультационный доклад с двенадцатью практиками ответственного внедрения искусственного интеллекта в финансовых организациях. Документ прямо охватывает сложные формы, включая генеративные и агентные системы, предлагает применять практики соразмерно и адресует их советам директоров и старшему руководству. На дату этой книги это консультационный материал, а не окончательный обязательный стандарт. Страница консультационного доклада указывает статус, дату и срок консультации.

Пересечение трёх контуров показано на рисунке. Центр обозначает не отдельный комитет, а решение о допуске конкретного действия. Управление искусственным интеллек-

том отвечает за свойства системы. Финансовые нормы задают допустимость хозяйственного результата. Операционная устойчивость ограничивает зависимость и последствия нарушения.



Для бухгалтера пересечение проявляется в обычном документе. Система должна понять содержание акта. Эта задача относится к управлению моделью и контекстом. Затем система определяет возможность признания обязательства и проведения записи, опираясь на учётную политику и финансовый контроль. Наконец, она должна сохранить возможность продолжить закрытие периода при отказе модели или поставщика, что относится к операционной устойчивости. Если один из контуров отсутствует, документ либо нельзя

доверить системе, либо невозможно доказать корректность процесса.

Следующая таблица разделяет обязанности нормативных контуров и показывает точку их соединения. Она не заменяет юридическую карту требований конкретной организации.

Контур	Основной вопрос	Объект контроля	обязательного результата	Вклад в допуск
Управление искусственным интеллектом	Соответствует ли система назначению и пределам применения	Модель, данные, оценка, мониторинг, участие человека	Документированные ограничения, испытания и роли	Разрешённый способ интерпретации и планирования
Финансовые нормы	Допустимо ли хозяйственное действие для данного субъекта и продукта	Обязательство, клиент, платёж, позиция, отчётность, полномочие	Выполнение применимых правовых и внутренних условий	Разрешённый экономический переход состояния
Операционная устойчивость	Сохранится ли критическая операция при отказе и распространении ошибки	Сервис, зависимость, поставщик, восстановление, инцидент	Предел нарушения, сценарий остановки и восстановления	Разрешённый радиус и маршрут продолжения
Объединённое решение	Можно ли выполнить этот план сейчас	Намерение, план, бюджет, контекст и	Транзакционный допуск с версией	Исполняемое разрешение или

Контур	Основной вопрос	Объект контроля	обязательного результата	Вклад в допуск

Пересечение нельзя реализовать общим комитетом, который рассматривает документы раз в квартал. Агент способен изменить состояние за секунды. Значимая часть реше-

ния должна находиться на пути исполнения. Комитет устанавливает политику, ответственность и допустимость классов действий. Система применяет эти решения к конкретным атрибутам и сохраняет доказательство.

Отсюда возникает принцип двойного масштаба. Управление действует на уровне жизненного цикла системы и на уровне отдельной транзакции. На уровне жизненного цикла утверждаются назначение, модели, данные, испытания и общие пределы. На уровне транзакции проверяются цель, объект, сумма, состояние, контекст и доступный бюджет. Одно без другого неполно. Хорошо управляемая модель может получить недопустимую транзакцию. Допустимая транзакция может быть передана версии модели, которая не прошла утверждение.

Три нормативных контура встречаются не в отчёте и не в организационной схеме. Они встречаются в конкретном действии, которое оставляет наблюдаемое последствие. Именно действие позволяет сопоставить требования к искусственному интеллекту, финансовому процессу и операционной устойчивости. Следующая задача состоит в том, чтобы превратить такое сопоставление в правило, способное сработать до исполнения.

3.2. Перевод требований в исполняемые политики

Нормативный текст редко говорит системе, какой вызов разрешить. Он устанавливает обязанность, принцип, результат или ответственность. Организация должна управлять риском, обеспечить надзор, защитить данные, сохранить устойчивость и контролировать внешних поставщиков. Между таким требованием и решением о конкретном платеже лежит цепочка интерпретаций. Если её не оформить, правило живёт в политике на бумаге, а программный код действует по иной логике.

Исполняемая политика является формальным условием, которое система может проверить на основе известных атрибутов до действия или во время него. Она не равна нормативной норме. Норма имеет юридический контекст, цели и область применимости. Политика реализует выбранное контрольное утверждение внутри конкретного процесса. Поэтому у каждой политики должны сохраняться происхождение, владелец, толкование, версия, область действия и дата пересмотра.

Перевод начинается с обязанности. Юрист и владелец процесса определяют, что требуется от организации и к каким субъектам, продуктам и ситуациям это относится. Затем обязанность превращается в контрольное утверждение. Оно

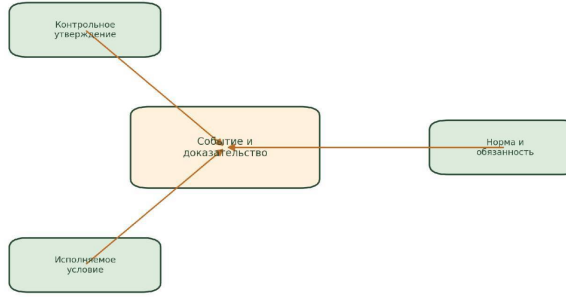
описывает наблюдаемый результат. Например, критическое изменение платёжных реквизитов должно быть подтверждено источником, независимым от канала запроса. Следующий шаг выражает утверждение через атрибуты и предикат. Система проверяет тип изменения, источник исходного запроса, источник подтверждения, субъект проверки и время.

После предиката определяется событие. Нужно решить, в какой точке процесса условие должно быть истинным. Проверка после отправки платежа не реализует предварительный запрет. Проверка при создании карточки может потерять актуальность к моменту исполнения. В агентной цепочке одно требование иногда превращается в несколько событий: при получении контекста, при изменении плана, перед вызовом и после результата.

Последний шаг определяет доказательство. Разрешение без сохранённого основания трудно отличить от обхода. Отказ без указания нарушенного условия трудно исправить. Политика должна выдавать не только логическое значение, но также идентификатор правила, его версию, проверенные атрибуты, время, решение и ссылку на допуск. Чувствительные значения могут храниться в защищённой форме, но их происхождение и контрольная сумма должны оставаться проверяемыми.

Цепочка перевода разворачивается слева направо, но допускает возврат к предыдущему шагу. Если исполняемое условие не отражает цель нормы или требует недоступ-

ных данных, контрольное утверждение приходится уточнять. Значение по умолчанию лишь скроет пробел.



Возьмём требование эффективного человеческого надзора. Если перевести его в правило «человек нажал кнопку», система получит формальное подтверждение без оценки качества надзора. Контрольное утверждение должно быть содержательнее: уполномоченный человек получил достаточную информацию, располагал временем, мог отклонить действие и не проверял собственное решение той же роли. Исполняемая часть может проверить полномочие, конфликт ролей, полноту пакета, срок и факт явного решения. Способность человека понять риск требует организационного обучения и оценки интерфейса, которые не сводятся к преди-

кату.

Другой пример даёт операционная устойчивость. Требование продолжать критическую операцию в пределах допустимого нарушения нельзя реализовать общим флагом доступности агента. Сначала определяется критическая операция и её предел. Затем выделяются зависимости: модель, оркестратор, хранилище контекста, сервис политик, инструмент, журнал и внешний поставщик. Исполняемая политика может запретить агенту начинать массовую цепочку, если недоступна сверка, журнал работает с потерями или отсутствует маршрут восстановления.

Для бухгалтера полезен принцип отсутствия тихого понижения контроля. Если сервис политик недоступен, система не должна считать это разрешением. Если система, содержащая договор, не отвечает, модель не должна заменять его похожим документом из истории. Если журнал доказательств не принимает событие, необратимое действие не должно продолжаться. Такое поведение называют закрытым отказом: неопределённость сохраняет запрет до восстановления требуемого контроля.

Однако закрытый отказ не должен остановить всю организацию без различия риска. Политика может разрешить низкорисковые действия в ограниченном режиме: чтение локального справочника, подготовку черновика, классификацию документов без изменения состояния. Для платежа, доступа и эксплуатации остаётся запрет. Это ещё одно при-

менение бюджета автономности: при деградации контроля уменьшается доступное полномочие.

Политика должна быть отделена от инструкций модели. Текстовая инструкция полезна для поведения, но не является надёжной границей полномочия. Она может быть вытеснена контекстом, неверно истолкована или изменена при обновлении. Критическое условие исполняет независимый компонент, который принимает структурированные атрибуты и возвращает решение. Модель не должна иметь возможность изменить этот компонент через обычный контекст задачи.

При этом политика не должна вручную дублировать данные хозяйственного процесса. Список допустимых получателей, статусы договоров, роли и лимиты существуют в системах учёта и управления. Сервис решения получает ссылки на авторитетные источники и их версии. Копия без синхронизации создаёт новый риск контекста. Архитектура должна различать правило и значение атрибута: правило меняется через управляемый жизненный цикл, значение поступает из доверенного источника в момент решения.

Четыре уровня перевода требования представлены на последовательных примерах. Формулировки являются авторской инженерной интерпретацией, а не цитатами нормативных актов.

Область	Исходная обязанность	Контрольное утверждение	Исполняемое условие	Доказательство
Управление риском	Риск управляется в течение жизненного цикла	Версия системы действует только в утверждённой области	Назначение, версия и класс действия присутствуют в реестре допуска	Идентификатор оценки, версия и срок решения
Человеческий надзор	Надзор должен быть эффективным	Уполномоченная независимая роль видит существенные параметры и может отказать	Роль допустима, конфликт отсутствует, пакет полон, решение явно	Кто, когда, что видел и какое решение принял
Устойчивость	Критическая операция сохраняется в пределах нарушения	Необратимое действие не начинается без журнала, сверки и	Сервисы контроля доступны, состояние не деградировало,	Состояние зависимостей и применённый режим

Область	Исходная обязанность	Контрольное утверждение	Исполняемое условие	Доказательство
		восстановления	проверен	
Финансовый контроль	Действие имеет хозяйственное основание и полномочие	Платёж связан с подтверждённым обязательством и получателем	Договор, документ, сумма, валюта, получатель и роли согласованы	Ссылки на объекты, контрольные суммы и решение политики

Качество перевода можно проверять четырьмя вопросами. Первый спрашивает о полноте: каждое ли существенное требование имеет владельца и точку реализации. Второй относится к точности: не стало ли правило шире или уже исходной обязанности. Третий проверяет исполнимость: доступны ли атрибуты в момент решения. Четвёртый относится к доказуемости: можно ли позже восстановить, какая версия правила сработала и на каких данных.

Особое внимание требуется исключениям. Ручное разрешение обхода политики иногда необходимо для инцидента или правовой обязанности. Однако исключение является новым решением о риске. Оно должно иметь узкую область, срок, уполномоченного владельца и последующую проверку. Общий режим аварийного доступа без связи с задачей превращает исключение в постоянный канал.

Изменение политики также является регулируемым действием. Если команда разработки может ослабить лимит и сразу использовать новое правило, разделение между исполнением и разрешением исчезает. Нужны версия, проверка, испытание на исторических и враждебных сценариях, утверждение и управляемое введение. Старые решения должны оставаться воспроизводимыми по старой версии.

Политика получает практический смысл только в момент решения. Она не заменяет правовую норму и не изображает автоматического юриста. Её роль конкретна: проверить условия, зафиксировать основание и не допустить действие, если обязательного свидетельства нет. После этого остаётся принципиальный для аудита вопрос: сможет ли система спустя время восстановить всю цепочку допуска.

3.3.
Evidence
-
by
-

design

: что система должна уметь доказать

После закрытия периода бухгалтер может показать, откуда появилась проводка. Существуют первичный документ, карточка контрагента, запись согласования, номер операции и след в регистре. Если сумма не сходится, цепочку восстанавливают от отчёта к записи и далее к основанию. Агент добавляет к этой цепочке новые объекты: версию модели, набор инструкций, найденные фрагменты, план, решения политик, вызовы инструментов и автоматические наблюдения. Если они не сохранены в момент работы, позднее объяснение модели не заменит доказательство.

Подход «доказательство, встроенное в архитектуру» означает, что возможность обосновать допустимость действия проектируется вместе с действием. В названии подраздела английская форма сохранена дословно. На русском языке речь идёт о доказательстве, встроенном в устройство системы. Журнал не является выгрузкой для аудитора после запуска. Он участвует в допуске: если обязательное доказательство нельзя сформировать, действие соответствующего класса не исполняется.

Система должна отвечать по крайней мере на семь вопросов: кто поставил цель; как цель была нормализована; какие данные использовались и откуда они получены; какой план был допущен; какие политики и версии обеспечили решение; что именно вызвал агент; какое фактическое состояние

возникло и кто его подтвердил. Ответы образуют цепочку, в которой каждый элемент связан с предыдущим идентификатором и временем.

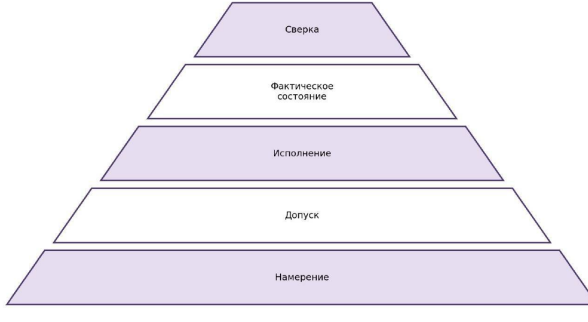
Этого перечня недостаточно без целостности. Запись, которую можно незаметно изменить, не подтверждает прошлое решение. Версия модели без версии конфигурации не воспроизводит поведение. Ссылка на страницу сайта без снимка существенного фрагмента не показывает, что видел агент. Текст инструкций модели без перечня доступных инструментов не определяет пространство действий. Поэтому доказательство состоит не только из содержимого, но и из идентификаторов, версий, контрольных сумм и связей происхождения.

Онтология происхождения W3C PROV различает сущности, виды деятельности и агентов, а также отношения порождения, использования и ответственности. Она предназначена для обмена сведениями о происхождении в разных системах. Для агентного финтеха эта модель полезна как нейтральный язык связи: документ является сущностью, вызов представляет вид деятельности, а пользователь, сервис или агент выступает участником. Рекомендация W3C PROV O даёт формальное описание классов и отношений.

Однако происхождение само по себе не доказывает допустимость. Оно отвечает, откуда появился объект и какие действия с ним связаны. Регулируемому агенту нужно дополнительно сохранить решение политики, область полномочия,

бюджет и сравнение ожидаемого состояния с фактическим. Эти элементы составляют доказательный пакет транзакции.

Замкнутый контур связывает намерение, допуск, исполнение и фактическое состояние. Сверка возвращает результат к исходному поручению. Ниже видно, при каком условии цепочка становится полной: система объясняет каждый переход и подтверждает его сохранёнными событиями.



Первое доказательство относится к намерению. Необходимо сохранить исходный запрос, идентичность субъекта и нормализованную цель. Если исходный текст содержит чувствительные данные, политика хранения может требовать защищённую форму или выделение только существенных атрибутов. Нельзя, однако, оставить только итоговую цель

без связи с запросом. Иначе невозможно установить, добавила ли система ограничение или потеряла часть поручения.

Второе доказательство относится к контексту. Для каждого значимого фрагмента нужны источник, время получения, владелец, классификация, область допустимого использования и контрольная сумма содержимого. Простая запись поискового запроса не подтверждает, какой результат был показан модели. Сохранение полного документа также не всегда допустимо по соображениям конфиденциальности и срока хранения. Архитектура должна уметь фиксировать ссылку, версию, существенный фрагмент и доказательство целостности с учётом политики данных.

Третье доказательство описывает план. Следует хранить не только итоговый список шагов, но и утверждённую версию, зависимости, ожидаемые состояния, условия остановки и разрешённые развилки. Если план изменился после ответа инструмента, новая версия связывается с наблюдением, которое вызвало изменение. Это позволяет отличить допустимую адаптацию от самовольного расширения цели.

Четвёртое доказательство относится к политике. Решение содержит идентификатор правила, версию, входные атрибуты, результат, причину и срок действия. Для разрешения фиксируется выданный бюджет. Для отказа сохраняется нарушенное условие. Не все входные значения следует помещать в открытый журнал. Часть может быть представлена ссылками и защищёнными контрольными суммами. Важна

возможность уполномоченной проверки.

Пятое доказательство описывает исполнение. Нужны точное имя и версия инструмента, целевой ресурс, параметры, ключ повторяемости, время вызова, ответ и связь с шагом плана. Секреты и токены не записываются в открытом виде. Вместо них сохраняются идентификатор полномочия, область, получатель и срок. Такой журнал позволяет проверить, что техническое право соответствовало решению политики.

Шестое доказательство относится к результату. Ответ «успешно» недостаточен. Система должна получить доменное подтверждение: идентификатор проведённого документа, фактическое списание, эффективную роль, состояние узла или иной признак. После этого сравниваются ожидаемые и фактические атрибуты. Результат сравнения становится отдельным событием.

Седьмое доказательство описывает человеческое участие. Если действие подтверждал человек, нужно знать его роль, доступный ему пакет, время, решение и конфликт интересов. Запись нажатия без содержимого экрана не показывает, что было предметом одобрения. При этом сбор данных о человеке должен быть соразмерным и соответствовать применимым нормам.

Регламент 2024/1689 для систем высокого риска требует автоматического журналирования событий в течение жизненного цикла и устанавливает связанные обязанности хра-

нения для охватываемых участников. Конкретные сроки и обязанности зависят от роли и применимости. Этот факт поддерживает принцип автоматического происхождения записей. Он не определяет полный состав доказательного пакета финансового агента. Статья 12 и связанные положения Регламента являются первичным источником.

Регламент 2022/2554 связывает управление цифровым риском с идентификацией, защитой, обнаружением, реагированием, восстановлением, сообщением об инцидентах, испытаниями и внешними поставщиками. Для доказательного слоя это означает, что журнал агента должен соединяться с процессом инцидента и устойчивости. Изолированный след модели, который нельзя сопоставить с сервисом, критической операцией и внешним провайдером, не помогает расследованию. Регламент 2022/2554 задаёт применимый контекст для охватываемых финансовых организаций.

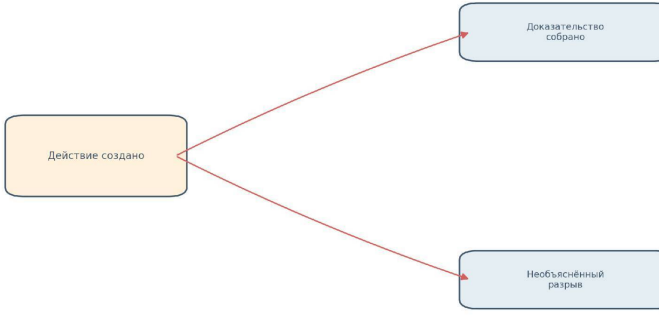
Минимальный состав доказательного пакета раскрывается через его функции. Слово «минимальный» означает необходимое ядро авторской архитектуры, а не установленный законом универсальный перечень.

Элемент	Что фиксируется	Зачем нужно	Контроль целостности	Ограничение данных
Намерение	Исходный запрос, субъект, нормализованная цель и основание	Проверить соответствие поручению	Идентификатор и контрольная сумма	Скрытие лишних персональных и секретных данных
Контекст	Источник, версия, время, фрагмент и назначение	Восстановить фактические сведения, доступные системе	Контрольная сумма и связь с источником	Срок хранения и цель обработки
План	Шаги, зависимости, ожидания, остановки и развилки	Проверить допустимость последовательности	Неизменяемая версия и связь с наблюдениями	Исключение секретов из описаний
Политика	Правило, версия, атрибуты, решение, причина и бюджет	Доказать основание допуска или отказа	Подпись решения и время	Защита значений атрибутов
Исполнение	Инструмент, ресурс, параметры, ключ и ответ	Воспроизвести фактический вызов	Идентификатор операции и контрольная сумма	Токены и секреты не сохраняются открыто
Результат	Доменное состояние и сравнение с ожиданием	Подтвердить хозяйственный исход	Независимый источник и время	Сохранение только требуемого объёма
Участие человека	Роль, пакет рассмотрения, решение и время	Оценить реальность надзора и разделение ролей	Связь с идентичностью и транзакцией	Соразмерность данных о сотруднике

Из состава пакета возникает принцип атомарности доказательства. Для необратимого действия критические события допуска и исполнения должны фиксироваться так, чтобы успешное изменение состояния не могло существовать без связанного свидетельства. Реализация зависит от систем: транзакционный журнал, очередь событий, подписанная квитанция или согласованная запись. Цель состоит не в выборе одной технологии, а в устранении временного разрыва.

Такой разрыв создаёт долг доказуемости. Агент совершает действие сейчас, а система обещает позже собрать версию политики, контекст или подтверждение. Пока долг не погашен, организация не может полностью объяснить собственное действие. Для обратимого черновика небольшой разрыв может быть принят. Для внешнего платежа, раскрытия данных или изменения продуктивной инфраструктуры допустимый долг близок к нулю.

Происхождение долга можно проследить по моменту появления двух объектов. Одновременное создание действия и доказательства не оставляет разрыва, если их связывает общий идентификатор. При задержке одной стороны появляется период, когда последствие уже существует, а основание ещё не подтверждено.



Долг можно измерять. Для каждой транзакции определяется перечень обязательных элементов и время их появления. Полнота равна доле сформированных элементов. Возраст долга равен времени от последствия до появления последнего обязательного доказательства:

$$П = (m / M) \times 100\ \%$$

$$A = t_{полн} - t_{действ}$$

Здесь (П) обозначает полноту, (m) число доступных обязательных элементов, (M) их установленное число, (A) возраст долга, ($t_{полн}$) момент достижения полноты, а ($t_{действ}$) момент последствия. Перечень (M) зависит от класса действия. Формулы являются авторскими операционными метриками и не имеют статуса нормативного расчёта.

Для бухгалтера эти метрики позволяют связать агентный журнал с привычной сверкой. Если проводка существует, но нет ссылки на первичный документ, полнота ниже установленного уровня. Если платёж исполнен, а подтверждение результата появляется через несколько часов, возраст долга показывает уязвимое окно. Если политика изменилась, но старое решение нельзя воспроизвести, долг становится постоянным.

Измерение не должно побуждать собирать все данные без ограничения. Избыточный журнал увеличивает риск конфиденциальности, стоимость хранения и сложность расследования. Принцип встроенной доказательности требует доста-

точности, а не тотальности. Для каждого элемента определяются назначение, доступ, срок и способ удаления. Контрольная сумма может подтвердить неизменность объекта, не раскрывая его всем потребителям.

Доказательная перспектива меняет понимание соответствия требованиям. Его нельзя считать состоянием системы, которое подтверждается сертификатом раз в год. Для автономного действия оно становится непрерывной способностью доказать связь между поручением, допустимостью, исполнением и результатом. Регуляторная инженерия превращает эту способность в часть транзакции.

Первый раздел приводит нас к простой, но требовательной мысли. Соответствие требованиям нельзя оставлять на время после запуска. Оно должно сопровождать действие от исходного поручения до сверенного результата. Во втором разделе эта мысль получит инженерную форму: намерение, план, допуск, исполнение и доказательство станут самостоятельными границами контроля.

Раздел II. Контур регулируемой автономности: референсная архитектура

Глава 4. Пять слоёв КРА

4.1. Intent layer : нормализация запроса и цели

Контур регулируемой автономности, далее КРА, начинается не с модели и не с инструмента. Он начинается с различия между просьбой и полномочием. Пользователь может написать «оплатите счёт сегодня», но эта фраза ещё не определяет юридическое лицо, обязательство, источник реквизитов, предельную сумму, срок действия поручения и допустимый способ исполнения. Пока элементы не установлены, система имеет текст, но не имеет цели, пригодной для делегирования.

КРА состоит из пяти слоёв: намерение, планирование, допуск, исполнение и доказательство. Слои задают логические границы, а не обязательное число сервисов. Их можно реализовать разными технологиями, но нельзя без потери контроля слить в одно неразличимое решение модели. Каждый слой получает определённый объект, выполняет проверку и создаёт результат, который может быть проверен следующим слоем.

Конец ознакомительного фрагмента.