

12+

АРХИТЕКТУРА БЕЗОПАСНОГО ИИ

Риски, управление и защита
искусственного интеллекта

ГАРИК ДАВТЯН

Гарик Давтян

**Архитектура безопасного ИИ.
Риски, управление и защита
искусственного интеллекта**

«Издательские решения»

Давтян Г.

Архитектура безопасного ИИ. Риски, управление и защита
искусственного интеллекта / Г. Давтян — «Издательские
решения»,

Русскоязычное практическое руководство по безопасности ИИ. Книга
охватывает угрозы LLM-приложениям и автономным агентам, архитектуру
защиты, управление рисками, безопасную разработку, тестирование,
реагирование на инциденты и программу внедрения за 365 дней. Опирается
на ISO/IEC 42001, NIST AI RMF, OWASP Top 10 for LLM, MITRE ATLAS
и EU AI Act. Для CISO, архитекторов безопасности, руководителей
и специалистов, которые внедряют ИИ и хотят управлять рисками.

© Давтян Г.

© Издательские решения

Содержание

Предисловие	7
Часть I. ИИ-система как объект защиты	9
Глава 1. Анатомия ИИ-системы	10
1.1. Компоненты ИИ-системы: данные, модели, приложения, инфраструктура	10
1.2. Агенты, инструменты и интерфейсы взаимодействия	12
1.3. Чем ИИ-система отличается от традиционного программного обеспечения	15
1.4. ИИ как социотехническая система	18
1.5. Активы и границы системы	20
Источники и дальнейшее чтение к главе 1	23
Глава 2. Типы ИИ-систем и их архитектурные особенности	24
2.1. Большие языковые модели и генерация текста	25
2.2. Компьютерное зрение и обработка изображений	27
2.3. Генерация и анализ аудио и речи	29
2.4. Генерация и анализ видео	31
2.5. Мультимодальные модели	32
2.6. Предиктивная аналитика, рекомендательные системы и классификация	34
2.7. Специфика безопасности каждого типа систем	36
Источники и дальнейшее чтение к главе 2	38
Глава 3. Жизненный цикл и границы ответственности	40
3.1. Стадии жизненного цикла ИИ-системы	40
3.2. Участники и их роли на каждой стадии	42
3.3. Распределение ответственности: поставщик, разработчик, оператор, пользователь	44
3.4. Границы ответственности организации и модель совместной ответственности	47
Источники и дальнейшее чтение к главе 3	49
Глава 4. Ландшафт рисков и свойства доверия	50
4.1. Понятия риска, угрозы, уязвимости и ущерба в контексте ИИ	50
4.2. Свойства доверия: безопасность, надёжность, устойчивость, справедливость	52
4.3. Прозрачность, объяснимость и подотчётность	55
4.4. Взаимозависимость свойств доверия и практические компромиссы	57
Источники и дальнейшее чтение к главе 4	60
Часть II. Управление и риски	62
Глава 5. Управление ИИ в организации	63
5.1. Зачем организации нужно управление ИИ	63
5.2. Политика ИИ и допустимые сценарии применения	65
5.3. Роли и ответственность: от руководства до владельца модели	67
5.4. Связь управления ИИ с действующими системами управления организации	70

5.5. Документирование решений и подотчётность	73
Источники и дальнейшее чтение к главе 5	75
Глава 6. Инвентаризация и классификация ИИ-систем	77
6.1. Реестр ИИ-систем: что учитывать и как вести	77
6.2. Теневой ИИ: как обнаружить и взять под контроль	79
6.3. Классификация по уровню риска: критерии и методы	82
Конец ознакомительного фрагмента.	83

Архитектура безопасного ИИ Риски, управление и защита искусственного интеллекта

Гарик Давтян

© Гарик Давтян, 2026

ISBN 978-5-0071-0007-6

Создано в интеллектуальной издательской системе Ridero

Предисловие

Я начал работать над этой книгой, когда понял, что безопасность искусственного интеллекта перестала быть темой для узкого круга исследователей. Она стала повседневной задачей для тех, кто отвечает за защиту организаций. За последние несколько лет искусственный интеллект прошёл путь от экспериментов в лабораториях до рабочего инструмента, встроенного в бизнес-процессы. Модели принимают решения по кредитным заявкам. Генеративные системы составляют документы и ведут переписку от имени сотрудников. Автономные агенты выполняют действия в корпоративных системах, вызывая API и обрабатывая чувствительные данные. Это происходит прямо сейчас, в организациях разного масштаба и разной зрелости.

Скорость, с которой организации внедряют ИИ, значительно опережает их готовность управлять рисками этих технологий. Я наблюдаю это в собственной практике и вижу подтверждение в отраслевых исследованиях. Большинство организаций признают необходимость защищать свои ИИ-системы, но не знают, с чего начать. У них нет реестра ИИ-систем. Нет ясности, кто отвечает за безопасность модели и кто принимает решение о допустимом уровне риска. Нет процессов тестирования, мониторинга и реагирования на инциденты, в которых участвует ИИ. Привычные инструменты информационной безопасности покрывают инфраструктуру, сети и приложения, но не учитывают специфику данных, моделей, промптов и автономных агентов.

Проблема ещё и в том, что безопасность ИИ оказалась на стыке нескольких дисциплин. Она требует понимания архитектуры систем машинного обучения и одновременно знания управления рисками, нормативных требований, защиты данных и организационного управления. Специалист по информационной безопасности, который всю карьеру защищал корпоративные сети, сталкивается с незнакомыми концепциями: отравление обучающих данных, уклоняющиеся воздействия (evasion attacks), инъекция промпта, дрейф поведения модели. Специалист по машинному обучению, в свою очередь, редко задумывается о моделировании угроз, управлении доступом для нечеловеческих субъектов или криминалистике инцидентов. Руководитель видит стратегическую возможность в ИИ, но ему не хватает инструментов для принятия обоснованных решений о допустимом уровне риска. Каждый из них владеет частью картины, но полную картину собрать не удаётся.

Эта книга написана для того, чтобы такую картину создать.

Я адресую её руководителям, CISO, архитекторам информационной безопасности, CIO и CTO, специалистам по рискам и соответствию требованиям, владельцам ИИ-продуктов, разработчикам и архитекторам ИИ-систем, специалистам по данным и MLOps, юристам и руководителям подразделений, которые внедряют ИИ в свои процессы. Мне было важно, чтобы текст оставался понятным руководителю и при этом сохранял практическую ценность для технического специалиста. Это непростой баланс, но именно он определяет полезность книги: если руководитель и архитектор говорят на разных языках, безопасность ИИ остаётся набором благих пожеланий.

Слово «архитектура» в названии книги выбрано намеренно. Безопасность ИИ нельзя свести к перечню технических контролей или к набору политик. Она требует целостной системы, которая охватывает управленческие решения, роли, процессы, данные, модели, приложения, инфраструктуру, механизмы контроля, аудита и подотчётности. Архитектура безопасного ИИ — это способ организовать все эти элементы так, чтобы они работали вместе на протяжении всего жизненного цикла системы.

Книга выстроена по логике, которая отражает путь организации. Первая часть формирует основу: состав ИИ-системы, типы технологий, жизненный цикл, свойства доверия. Вторая часть выстраивает управление: политики, инвентаризация, оценка рисков, норматив-

ные требования. Третья часть систематизирует угрозы: от отравления данных до компрометации автономных агентов. Четвёртая часть описывает архитектуру защиты: зоны доверия, защиту данных, управление доступом, безопасность моделей и конвейеров. Пятая часть охватывает безопасную разработку и эксплуатацию: моделирование угроз, тестирование, мониторинг. Шестая часть посвящена реагированию на инциденты и восстановлению. Седьмая часть переводит содержание книги в практическую программу действий с приоритетами на первый год.

Я опирался на международные стандарты и методологии: ISO/IEC 42001, ISO/IEC 23894, NIST AI RMF, NIST AI 600—1, OWASP Top 10 for LLM Applications, MITRE ATLAS, CSA AI Controls Matrix, рекомендации NCSC и CISA по безопасной разработке ИИ, EU AI Act и другие документы. Но эта книга — не пересказ стандартов. Стандарты дают структуру и требования. Книга показывает, как эти требования работают в организации, какие ограничения возникают на практике и какие решения приходится принимать руководителю и архитектору.

Технологии ИИ продолжают развиваться быстро. Модели становятся мощнее, агенты получают больше автономности, мультимодальные системы объединяют текст, изображение, аудио и видео в единый поток обработки. Часть конкретных инструментов и протоколов, упомянутых в книге, может измениться. Но принципы архитектуры безопасности, управления рисками, эшелонированной защиты и человеческого контроля останутся применимыми. Именно поэтому я сосредоточился на принципах и подходах, которые можно адаптировать к новым технологиям, а не на описании текущих версий конкретных продуктов.

У меня нет иллюзий, что одна книга решит все вопросы безопасности ИИ. Но я уверен, что она поможет вам выстроить систему координат: понять, какие активы нуждаются в защите, где возникают угрозы, как распределить ответственность, какие контроли применить и с чего начать. Если после прочтения вы сможете провести оценку рисков для ИИ-системы в своей организации, выстроить архитектуру защиты и сформировать программу первых практических шагов, книга выполнит свою задачу.



Часть I. ИИ-система как объект защиты

Прежде чем защищать ИИ-систему, нужно понять, что именно мы защищаем. Вопрос звучит просто, но на практике ответ на него вызывает затруднения. Руководитель видит ИИ как бизнес-инструмент, который ускоряет принятие решений. Разработчик воспринимает его как модель, обученную на определённых данных и встроенную в приложение. Специалист по информационной безопасности привычно смотрит на инфраструктуру, сети и контроль доступа. Каждый из них прав, но ни одна из этих точек зрения по отдельности не даёт полной картины.

ИИ-система — это не модель и не алгоритм. Это совокупность данных, моделей, приложений, интерфейсов, инструментов, агентов, инфраструктуры, людей и процессов, которые вместе создают ценность для организации. Каждый из этих компонентов несёт собственные риски, и связи между ними порождают риски, которых нет ни у одного компонента в отдельности. Модель, безопасная в лабораторной среде, становится уязвимой, когда получает доступ к корпоративным данным через систему дополненной генерации. Агент, корректно решающий тестовые задачи, может выйти за границы полномочий в продуктивной среде, где ему доступны внешние сервисы и инструменты.

При этом ИИ-системы принципиально отличаются от традиционного программного обеспечения. Их поведение определяется не только кодом, но и данными, на которых обучена модель, и контекстом, в котором она работает. Они могут вести себя по-разному при одинаковых входных данных. Их результаты не всегда воспроизводимы и не всегда объяснимы. Они способны меняться со временем без явного обновления, если изменяются данные или условия среды. Всё это создаёт новый ландшафт рисков, который требует собственного понятийного аппарата и собственных подходов к защите.

Эта часть книги формирует основу, на которой строится всё остальное. Первая глава разбирает анатомию ИИ-системы: из каких компонентов она состоит, какие активы в ней нужно учитывать и где проходят её границы. Вторая глава раскрывает типы ИИ-систем по модальности и технологии: текстовые, визуальные, аудио, видео, мультимодальные, предиктивные. Каждый тип несёт специфические риски, и понимание этих различий определяет выбор мер защиты. Третья глава описывает жизненный цикл ИИ-системы и распределение ответственности между участниками: поставщиками, разработчиками, операторами и пользователями. Четвёртая глава вводит систему понятий для управления рисками и раскрывает свойства, которые формируют доверие к ИИ-системе: безопасность, надёжность, устойчивость, справедливость, прозрачность и подотчётность.

После прочтения этой части вы сможете описать любую ИИ-систему в своей организации как целостный объект управления и защиты: определить её состав, тип, стадию жизненного цикла, участников, активы, границы и применимые свойства доверия. Это та отправная точка, без которой оценка рисков, построение архитектуры защиты и формирование программы безопасности остаются невозможными.

Глава 1. Анатомия ИИ-системы

1.1. Компоненты ИИ-системы: данные, модели, приложения, инфраструктура

Данные

Данные — это исходный материал, из которого ИИ-система извлекает закономерности и на основании которого принимает решения. В традиционном программном обеспечении данные являются объектом обработки. В ИИ-системе они формируют саму логику работы. Модель, обученная на искажённых данных, будет выдавать искажённые результаты, даже если её архитектура и код безупречны.

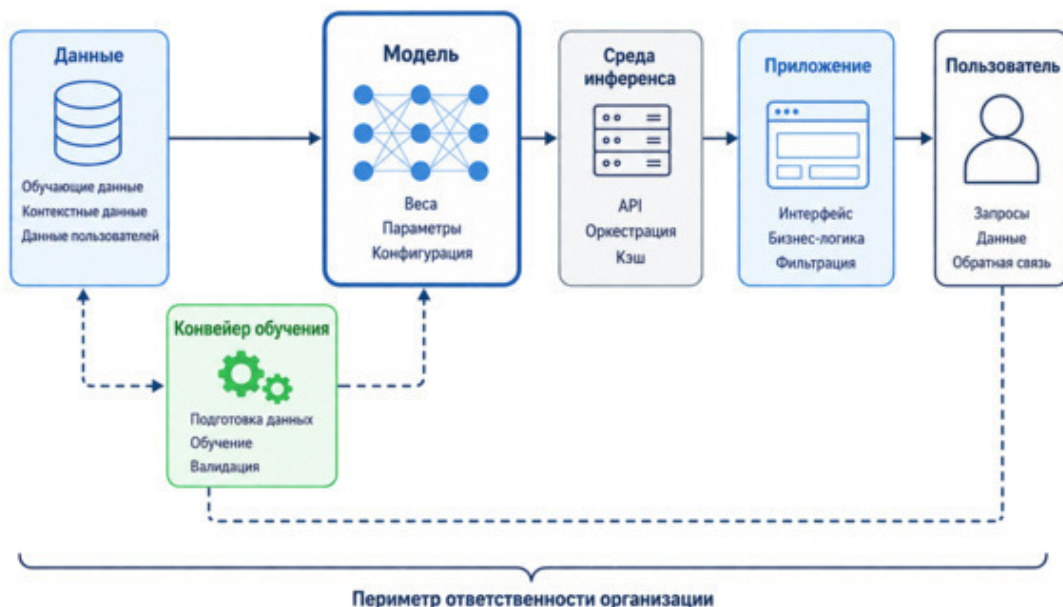


Рисунок 1. Архитектурная схема ИИ-системы: данные, модель, программные компоненты, инфраструктура, интерфейсы и контуры управления.

В жизненном цикле ИИ-системы данные выполняют разные роли. Обучающие данные используются для тренировки модели и определяют, какие закономерности она усвоит. Данные для тонкой настройки адаптируют предобученную модель к конкретной задаче или области. Оценочные данные применяются для проверки качества модели перед развёртыванием. Контекстные данные подаются модели в момент запроса, например через системы дополненной генерации (RAG), и формируют основу для ответа. Входные данные поступают от пользователей и других систем в ходе повседневной работы. Выходные данные — это результаты работы модели: тексты, классификации, прогнозы, изображения, решения.

Каждая из этих категорий несёт собственные риски. Обучающие данные могут быть отравлены злоумышленником или содержать скрытые смещения. Контекстные данные в RAG-системе могут включать чувствительную информацию, которую модель раскроет в ответе.

Входные данные могут содержать инъекции, направленные на изменение поведения модели. Выходные данные могут содержать конфиденциальные фрагменты, извлечённые из обучающего набора. Организация, которая не различает эти категории и не применяет к каждой из них соответствующие меры классификации, контроля доступа и мониторинга, оставляет открытыми сразу несколько направлений атаки.

Модели

Модель — это математическая структура, обученная на данных и способная выполнять задачу: генерировать текст, распознавать изображение, классифицировать транзакцию, прогнозировать спрос. С точки зрения безопасности модель является одновременно активом и потенциальным источником риска.

Как актив модель представляет значительную ценность. Она содержит знания, извлечённые из обучающих данных, и может воплощать месяцы работы и миллионы единиц вычислительных ресурсов. Извлечение модели злоумышленником означает потерю интеллектуальной собственности. Подмена модели приводит к тому, что все последующие решения принимаются на скомпрометированной основе.

Как источник риска модель способна вести себя непредсказуемо. Её поведение определяется обучающими данными и архитектурой, но конкретный результат для конкретного запроса не всегда воспроизводим и не всегда объясним. Модель может уверенно генерировать несуществующие факты. Она может раскрыть фрагменты обучающих данных в своих ответах. Она может изменить поведение после обновления без явного намерения разработчика.

На практике организация работает с моделями разного происхождения. Базовые модели (foundation models) обучаются крупными поставщиками на огромных объёмах данных и предоставляются через API или для локального развёртывания. Дообученные модели адаптируются к конкретной задаче на корпоративных данных. Собственные модели обучаются организацией с нуля. Каждый из этих вариантов влияет на профиль риска: при использовании внешней модели через API организация не контролирует её обучающие данные, архитектуру и поведение при обновлении. При локальном развёртывании сторонней модели она принимает на себя ответственность за среду выполнения и контроль доступа. При собственной разработке прибавляется ответственность за весь конвейер обучения и качество данных.

Приложения

Модель сама по себе не взаимодействует с пользователем и не выполняет бизнес-задачу. Это делает приложение — программный слой, который связывает модель с внешним миром. Приложение принимает запросы, формирует промпты, вызывает модель, обрабатывает результаты, передаёт их пользователю или другой системе. В генеративных ИИ-системах этот слой включает оркестрацию: управление контекстом, обращение к базам знаний, вызов внешних инструментов, цепочки рассуждений.

Приложение определяет, насколько опасным может быть результат работы модели. Одна и та же модель, отвечающая на вопросы в изолированном интерфейсе без доступа к корпоративным системам, представляет одну степень риска. Та же модель, интегрированная в рабочий процесс с правом записи в базу данных, отправки сообщений и вызова API, представляет принципиально иную. Именно на уровне приложения формируются полномочия модели: к каким данным она имеет доступ, какие действия способна выполнить, какие ограничения наложены на её поведение.

С точки зрения безопасности приложение — это пространство, где пересекаются традиционные уязвимости программного обеспечения и специфические риски ИИ. Инъекции в промпты проходят через приложение. Небезопасная обработка результатов модели происходит в приложении. Утечка системного промпта, раскрытие чувствительных данных, передача избыточных полномочий агенту — всё это вопросы архитектуры и логики приложения. Орга-

низация, которая сосредоточена исключительно на модели и не уделяет внимания приложению, упускает одну из самых широких поверхностей атаки.

Инфраструктура

Под инфраструктурой ИИ-системы понимается всё, что обеспечивает её работу на физическом и программном уровне: вычислительные ресурсы, хранилища данных, сети, среды выполнения, системы оркестрации контейнеров, средства мониторинга. Для обучения моделей требуются значительные вычислительные мощности: графические процессоры, тензорные процессоры, специализированные ускорители. Для продуктивной эксплуатации нужны среды инференса, балансировщики нагрузки, системы очередей. Для хранения данных и моделей используются объектные хранилища, векторные базы данных, реестры артефактов.

Инфраструктура ИИ может размещаться в публичном облаке, в частном облаке, на локальных серверах или в гибридной конфигурации. Каждый вариант определяет распределение ответственности между организацией и поставщиком. В публичном облаке поставщик отвечает за физическую безопасность, виртуализацию и часть программного стека, а организация — за конфигурацию, управление доступом, шифрование данных и безопасность приложений. В локальной инфраструктуре вся ответственность лежит на организации. В гибридных сценариях границы становятся сложнее, и без чёткого документирования распределения ответственности возникают зоны, которые не контролирует никто.

Существенная особенность инфраструктуры ИИ — разделение сред. Среда обучения, где модель тренируется на больших объёмах данных, обычно отличается от среды инференса, где модель обрабатывает запросы пользователей. Среда разработки, где исследователи экспериментируют с моделями и данными, часто защищена слабее продуктивных систем. Между тем именно в среде разработки сосредоточены ценные активы: обучающие данные, промежуточные версии моделей, ключи доступа к хранилищам и API. Компрометация этой среды может привести к отравлению данных, подмене модели или утечке интеллектуальной собственности задолго до того, как система попадёт в продуктивную эксплуатацию.

Ни один из четырёх компонентов не существует изолированно. Данные питают модель. Модель встраивается в приложение. Приложение работает на инфраструктуре. Изменение в одном компоненте отражается на остальных. Обновление обучающих данных меняет поведение модели. Новая версия модели требует пересмотра логики приложения. Миграция инфраструктуры из локальной среды в облако перестраивает модель доступа и распределение ответственности.

Для специалиста по безопасности это означает, что защита одного компонента без учёта остальных недостаточна. Шифрование данных в хранилище не спасёт от утечки через ответы модели. Контроль доступа к API приложения не предотвратит компрометацию обучающего конвейера. Защита инфраструктуры не снимет проблему избыточных полномочий агента. Безопасность ИИ-системы требует целостного взгляда, учитывающего все четыре компонента и взаимодействие между ними. Именно этому посвящена оставшаяся часть главы и книги в целом.

1.2. Агенты, инструменты и интерфейсы взаимодействия

В предыдущем разделе мы описали четыре базовых компонента ИИ-системы. Но современные системы всё чаще включают элементы, которые не укладываются в классическую схему «данные — модель — приложение — инфраструктура». Речь идёт об автономных агентах, внешних инструментах и протоколах взаимодействия, которые превращают модель из генератора ответов в участника рабочих процессов, способного выполнять действия в информационных системах организации.

Модель получает запрос и возвращает результат. Агент получает задачу и самостоятельно определяет последовательность шагов для её выполнения. Он может разбить задачу на подзадачи, обратиться к внешним источникам данных, вызвать инструменты, проанализировать промежуточные результаты и скорректировать свой план. В отличие от модели, которая обрабатывает один запрос за один проход, агент ведёт многоходовое взаимодействие с окружением и сохраняет состояние между шагами.

Архитектура типичного агента включает несколько взаимосвязанных компонентов. Восприятие отвечает за приём входных данных: текстового запроса, содержимого документа, сигнала от другой системы. Рассуждение опирается на языковую модель и определяет, какие действия нужно выполнить. Оркестрация управляет последовательностью шагов, координирует вызовы инструментов и при необходимости делегирует подзадачи другим агентам. Память позволяет агенту сохранять контекст разговора, промежуточные результаты и сведения, полученные на предыдущих шагах. Инструменты дают агенту возможность взаимодействовать с внешним миром: читать файлы, выполнять запросы к базам данных, отправлять сообщения, вызывать API.

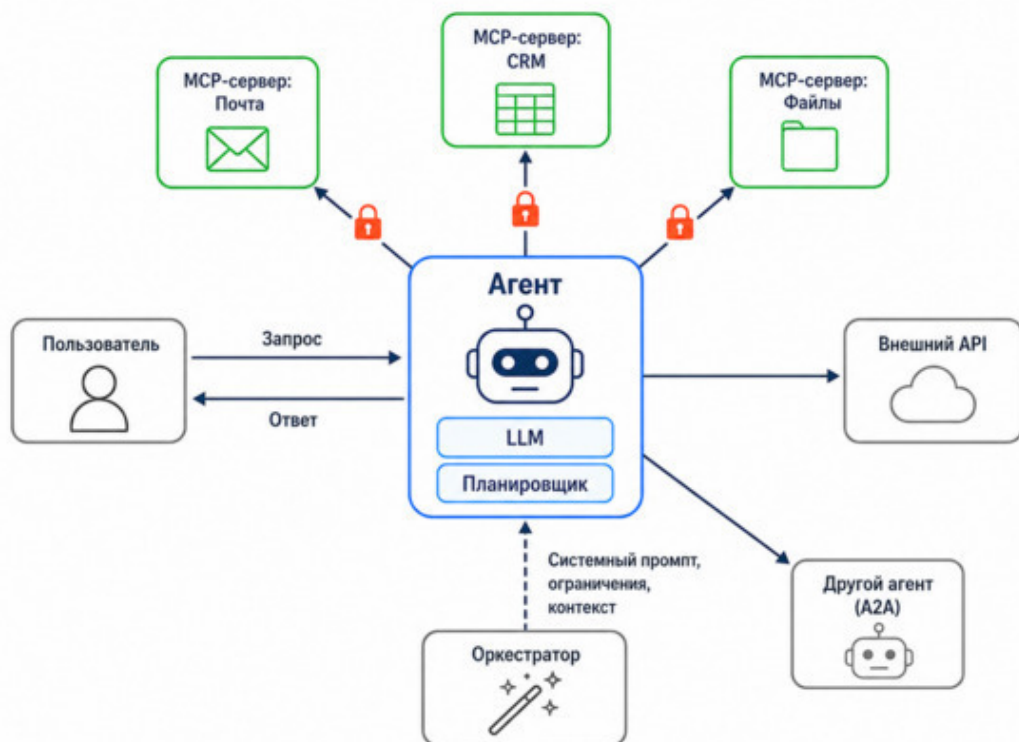


Рисунок 2. Схема агентной архитектуры: модель, память, планирование, инструменты, внешние системы и протоколы взаимодействия MCP и A2A.

Именно сочетание автономности и доступа к инструментам делает агентов принципиально новым объектом с точки зрения безопасности. Модель, которая генерирует текст в изолированном окне чата, ограничена в своём воздействии. Агент, который по результатам рассуждения обновляет запись в CRM, отправляет электронное письмо от имени сотрудника и создаёт заявку в системе управления задачами, воздействует на информационные системы организации с последствиями, сопоставимыми с действиями человека.

Инструменты — это внешние функции, к которым агент обращается для выполнения задач. Спектр инструментов практически не ограничен: поиск в интернете, чтение и запись файлов, выполнение программного кода, запросы к базам данных, вызовы корпоративных

API, отправка сообщений в мессенджерах, управление записями в учётных системах. Каждый инструмент расширяет возможности агента, но одновременно расширяет и поверхность атаки.

Когда агент вызывает инструмент, он фактически выполняет действие от имени пользователя или от собственного имени в среде, которой доверяет организация. Возникает вопрос: какими полномочиями обладает агент при вызове каждого инструмента? Если агент использует тот же API-ключ, что и администратор системы, он может выполнить любую операцию, включая те, которые для его задачи не нужны. Если инструмент возвращает агенту данные, которые содержат встроенные инструкции (например, текст документа с инъекцией), агент может интерпретировать их как команды и изменить своё поведение. Если среда выполнения кода, вызываемого агентом, не изолирована, компрометация одного инструмента может привести к доступу ко всей инфраструктуре.

Вопрос изоляции среды выполнения заслуживает пристального внимания. Организации, которые позволяют агенту выполнять произвольный код в общей среде, создают условия, при которых злоумышленник может через манипуляцию входными данными заставить агента запустить вредоносный скрипт, получить доступ к файловой системе или сети. Песочницы, контейнеры с ограниченными правами, детерминированные вызовы заранее определённых функций вместо произвольного выполнения кода — всё это архитектурные решения, которые существенно снижают риск.

По мере распространения агентов возникла потребность в стандартизации того, как они подключаются к внешним системам и взаимодействуют друг с другом. Два протокола, получивших широкое распространение, заслуживают упоминания.

Model Context Protocol (MCP) стандартизирует подключение агентов к внешним источникам данных, инструментам и корпоративным системам. Вместо создания уникальной интеграции для каждой пары «агент — сервис» MCP позволяет агенту выступать клиентом, который обращается к серверам, предоставляющим доступ к базам данных, календарям, почте, CRM и другим ресурсам через единый интерфейс. Это снижает трудозатраты на интеграцию и ускоряет развёртывание. Но одновременно MCP вводит в периметр организации внешние зависимости, каждая из которых может стать каналом атаки: скомпрометированный MCP-сервер способен передать агенту искажённые данные или вредоносные инструкции.

Agent-to-Agent protocol (A2A) решает другую задачу: он позволяет агентам обнаруживать друг друга, обмениваться описаниями своих возможностей, делегировать задачи и координировать действия. A2A вводит понятие карточки агента (agent card) — машиночитаемого описания идентичности, навыков, конечных точек и требований аутентификации агента. Это делает возможным построение многоагентных систем, в которых несколько агентов от разных поставщиков совместно решают сложную задачу.

С точки зрения безопасности оба протокола создают новый класс проблем. MCP расширяет поверхность атаки за счёт множества внешних подключений, каждое из которых требует аутентификации, авторизации и валидации данных. A2A порождает вопросы доверия между агентами: как убедиться, что агент, заявляющий определённые возможности, действительно является тем, за кого себя выдаёт? Как предотвратить ситуацию, в которой один агент передаёт другому задачу с полномочиями, превышающими необходимые? Как обеспечить аудит действий в цепочке из нескольких агентов, если каждый из них принимает решения автономно?

Стоит отметить, что эти протоколы пока находятся на ранней стадии зрелости. В отличие от устоявшихся сетевых стандартов, выросших из десятилетий открытого сотрудничества и стандартизации, MCP и A2A больше напоминают прикладные фреймворки, чем полноценные протоколы с проработанной моделью безопасности. Организации, внедряющие агентные системы, должны учитывать эту незрелость и выстраивать дополнительные контроли поверх протоколов, а не полагаться на их встроенные механизмы защиты.

ИИ-система взаимодействует с внешним миром через интерфейсы, и их разнообразие продолжает расти. Пользовательские интерфейсы включают чат-окна, голосовых ассистентов, формы ввода, панели управления. Программные интерфейсы (API) обеспечивают интеграцию с другими приложениями и сервисами. Межагентные интерфейсы, описанные выше, связывают агентов между собой и с инструментами.

Каждый интерфейс — это точка входа в систему и одновременно точка потенциальной атаки. Через пользовательский интерфейс поступают запросы, которые могут содержать прямые инъекции. Через API приходят данные от других систем, которые могут быть скомпрометированы. Через межагентные интерфейсы поступают инструкции от агентов, чьё поведение организация не контролирует полностью.

Для специалиста по безопасности критически важно знать, какие интерфейсы есть у конкретной ИИ-системы и кто через них взаимодействует. Модель, доступная только через внутренний API для одного приложения, имеет ограниченную поверхность атаки. Та же модель, доступная через публичный чат, интегрированная с десятком корпоративных систем через MCP и взаимодействующая с агентами партнёров через A2A, представляет собой систему с множеством точек входа, каждая из которых требует собственных контролей. Инвентаризация интерфейсов — такая же обязательная часть оценки рисков, как инвентаризация данных и моделей.

Появление агентов, инструментов и протоколов взаимодействия означает, что ИИ-система перестаёт быть замкнутым контуром, в котором данные поступают на вход, а результат появляется на выходе. Она становится активным участником информационной среды организации, способным читать, писать, отправлять, вызывать, делегировать и координировать. Привычная модель безопасности, построенная на контроле периметра и статических правилах доступа, оказывается недостаточной. Организации необходимо переходить к модели, в которой каждое взаимодействие агента проверяется, каждый вызов инструмента авторизуется, каждое действие фиксируется и каждая цепочка решений может быть восстановлена при расследовании инцидента. Это требует архитектурных решений, которые мы подробно разберём в четвертой части книги.

1.3. Чем ИИ-система отличается от традиционного программного обеспечения

Организации, которые впервые сталкиваются с задачей защиты ИИ, часто начинают с привычного подхода: берут существующие контроли информационной безопасности и распространяют их на новую систему. Контроль доступа, шифрование, сегментация сети, управление уязвимостями, журналирование — всё это применимо и необходимо. Проблема в том, что этих мер недостаточно. ИИ-системы обладают рядом свойств, которых нет у традиционного программного обеспечения, и эти свойства порождают риски, для которых привычные инструменты не предназначены.

Первое и, пожалуй, наиболее значимое отличие состоит в том, что поведение ИИ определяется данными, а не кодом. В традиционном ПО логика работы зафиксирована в исходном коде. Если разработчик написал правило «отклонить заявку при сумме свыше миллиона», система будет следовать этому правилу. Можно провести аудит кода, убедиться, что правило реализовано корректно, и точно предсказать поведение системы для любого входного значения.

В ИИ-системе поведение формируется на основе данных, использованных при обучении. Модель самостоятельно выявляет закономерности, и разработчик не всегда может точно объяснить, какие именно признаки определяют конкретное решение. Если обучающие данные содержат скрытые смещения, модель усвоит их и воспроизведёт в своих результатах. Если дан-

ные неполны или не соответствуют условиям эксплуатации, модель будет ошибаться в ситуациях, которых не видела при обучении. Аудит исходного кода нейронной сети не раскроет логику принятия решений, потому что логика хранится не в коде, а в миллиардах числовых параметров модели.

Для безопасности это означает фундаментальный сдвиг. Защита данных из задачи конфиденциальности превращается в задачу целостности самой системы. Компрометация обучающих данных — это компрометация логики принятия решений.

Из зависимости от данных вытекает второе отличие: недетерминированное поведение. Классическое ПО при одном и том же входе даёт один и тот же выход. Это свойство делает возможным предсказуемое тестирование и верификацию. ИИ-системы, особенно построенные на больших языковых моделях, ведут себя иначе. При одном и том же запросе модель может сгенерировать разные ответы. Это следствие архитектуры: на этапе генерации текста модель выбирает следующий токен из вероятностного распределения, и параметры выборки вносят элемент случайности.

Недетерминированность усложняет тестирование. Проверка, которая прошла успешно сегодня, может дать иной результат завтра без каких-либо изменений в системе. Это означает, что традиционные методы верификации — контрольные примеры с фиксированным ожидаемым результатом — работают ограниченно. Организации вынуждены переходить к статистическим методам оценки, измеряя не точный результат, а вероятность того, что система ведёт себя в рамках допустимых границ. Для специалиста по безопасности это создаёт конкретную проблему: как определить, что поведение системы изменилось из-за атаки, а не из-за естественной вариативности?

К недетерминированности добавляется ограниченная объяснимость. Обычное приложение можно проследить шаг за шагом: от входных данных через последовательность операций к выходному результату. В сложных нейронных сетях этот путь непрозрачен. Модель с сотнями миллиардов параметров принимает решение через множество слоёв преобразований, и извлечь из этого процесса понятное человеку объяснение удаётся далеко не всегда.

Ограниченная объяснимость создаёт трудности на нескольких уровнях. Для регулятора, который требует обоснования автоматизированных решений, затрагивающих права граждан. Для аудитора, который должен оценить корректность системы. Для команды реагирования на инциденты, которая пытается понять, почему модель выдала ошибочный или опасный результат. Для руководителя, который принимает решение о допустимости использования модели в конкретном бизнес-процессе. Методы повышения объяснимости существуют, но они дают приближённые интерпретации, а не точное воспроизведение логики решения.



Рисунок 3. Традиционное программное обеспечение и ИИ-система: различия в логике работы, поведении, зависимости от данных и характере ошибок.

Есть ещё одно свойство, которое отличает ИИ от классических систем и часто застаёт организации врасплох. Обычное приложение ведёт себя одинаково до тех пор, пока кто-то не обновит код. ИИ-система может деградировать без каких-либо изменений в ней самой, если меняется среда, в которой она работает. Это явление называют дрейфом.

Дрейф данных возникает, когда статистические характеристики входных данных в продуктивной среде начинают отличаться от тех, на которых модель обучалась. Модель кредитного скоринга, обученная в период стабильной экономики, начинает ошибаться при резком изменении макроэкономических условий, хотя ни одна строка кода и ни один параметр модели не изменились. Дрейф поведения проявляется в том, что результаты модели постепенно смещаются за пределы приемлемого диапазона. Концептуальный дрейф означает, что сама связь между входными данными и целевой переменной изменилась: признаки, которые раньше предсказывали результат, перестали работать.

Для организации дрейф означает, что модель, прошедшую все проверки перед развёртыванием, нельзя считать безопасной навсегда. Необходим непрерывный мониторинг поведения и регулярная переоценка. Модель, которая вчера работала корректно, сегодня может создавать риски, которых при запуске не было.

Наконец, сложные ИИ-системы способны демонстрировать поведение, которое не было явно заложено разработчиком. Большие языковые модели проявляют способности, которые не наблюдались у моделей меньшего масштаба и не были целью обучения. Агентные системы, объединяющие несколько моделей и инструментов, могут порождать результаты, которые невозможно предсказать, анализируя каждый компонент в отдельности. Такое эмерджентное поведение не обязательно опасно, но оно принципиально непредсказуемо. Организация не может гарантировать, что модель не продемонстрирует неожиданную способность, которая выйдет за рамки её назначения. Это создаёт ситуацию, в которой перечень угроз невозможно состав-

вить исчерпывающе, и защита должна опираться на архитектурные ограничения, а не только на знание конкретных атак.

Все перечисленные свойства в совокупности формируют поверхности атаки, которых нет у традиционного ПО. Отравление обучающих данных влияет на логику системы через данные, а не через код. Уклоняющиеся воздействия заставляют модель принимать ошибочные решения при минимальных, незаметных для человека изменениях входных данных. Инъекция в промпт манипулирует поведением модели через текст запроса. Извлечение модели позволяет атакующему восстановить интеллектуальную собственность, отправляя множество запросов и анализируя ответы. Обратное восстановление данных даёт возможность извлечь фрагменты обучающих данных из ответов модели.

Ни одна из этих атак не использует уязвимость в коде в традиционном смысле. Они эксплуатируют свойства самой модели: её зависимость от данных, её вероятностную природу, её неспособность отличить легитимную инструкцию от вредоносной. Традиционные сканеры уязвимостей, межсетевые экраны и системы обнаружения вторжений не предназначены для выявления таких атак. Организации необходимы специализированные инструменты и процессы: тестирование устойчивости к враждебным воздействиям, мониторинг поведения модели, фильтрация входов и выходов, ограничение полномочий.

Перечисленные отличия не отменяют традиционные практики безопасности. Инфраструктура ИИ-системы по-прежнему нуждается в защите сети, управлении доступом, обновлениях и мониторинге. Но поверх этого фундамента организация должна выстроить дополнительный слой контролей, учитывающих вероятностную природу модели, зависимость от данных, ограниченную объяснимость, дрейф и возможность эмерджентного поведения. Именно этот дополнительный слой и составляет предмет книги.

1.4. ИИ как социотехническая система

До сих пор мы рассматривали компоненты ИИ-системы преимущественно с технической стороны: данные, модели, приложения, инфраструктура, агенты, инструменты. Но ИИ-система не работает в вакууме. Она существует внутри организации, обслуживает людей, создаётся людьми и управляется людьми. Решения о том, какие данные использовать для обучения, какой уровень автономии предоставить модели, кому доверить контроль результатов, принимают конкретные сотрудники в конкретном организационном контексте. Именно этот контекст определяет, как система будет работать на практике и какие риски возникнут при её эксплуатации.

Понятие социотехнической системы пришло из инженерной и организационной науки. Оно описывает системы, в которых технические и социальные элементы неразрывно связаны и совместно определяют результат. Автомобиль сам по себе — техническая система. Автомобиль, водитель, правила дорожного движения, дорожная инфраструктура и культура вождения — социотехническая система. Авария может произойти из-за технической неисправности, но гораздо чаще она происходит из-за поведения водителя, плохой видимости или неудачного проектирования перекрёстка.

С ИИ-системами происходит то же самое. Модель может работать корректно в лабораторных условиях и давать опасные результаты в продуктивной среде, потому что пользователи применяют её способом, который разработчик не предусмотрел. Система может пройти все технические проверки и при этом причинить ущерб, потому что организация не определила, в каких ситуациях решение модели должно проверяться человеком. Инцидент может остаться незамеченным, потому что сотрудники не обучены распознавать признаки некорректного поведения ИИ.

NIST AI Risk Management Framework прямо указывает на социотехническую природу ИИ-систем и подчёркивает, что управление рисками требует учёта не только технических, но и человеческих, организационных и социальных факторов. ISO/IEC 42001:2023 включает в систему управления ИИ требования к компетенциям персонала, осведомлённости и коммуникации — признавая, что технические контроли работают только в организации, которая понимает их назначение и способна их поддерживать.

В жизненном цикле ИИ-системы участвует множество ролей: исследователи данных, разработчики моделей, инженеры по данным, архитекторы, специалисты по развёртыванию, операторы, тестировщики, аудиторы, владельцы продукта, конечные пользователи, руководители, юристы, специалисты по этике и защите данных. Каждый из них принимает решения, которые влияют на безопасность системы.

Исследователь данных выбирает обучающую выборку и определяет, какие признаки включить в модель. Его выбор может закрепить в модели скрытые смещения, если выборка не представляет все группы пользователей. Разработчик приложения определяет, какие данные передавать модели в контексте запроса и как обрабатывать её ответы. Его решение определяет, сможет ли модель раскрыть чувствительную информацию или выполнить нежелательное действие. Оператор настраивает среду выполнения и права доступа. Конечный пользователь формулирует запросы и интерпретирует результаты, иногда принимая на их основе решения с серьёзными последствиями.

Каждое из этих решений несёт собственные когнитивные искажения. Люди склонны доверять результатам системы, которая обычно работает правильно, и постепенно ослаблять собственный контроль. Это явление называют автоматизационным смещением. Оно приводит к тому, что ошибку модели замечают позже, чем ошибку человека, потому что ожидание корректного ответа притупляет критическое восприятие. Организация, которая ввела человеческий контроль над решениями модели, но не обеспечила условия для его реальной работы — обучение, время на проверку, право отклонить решение системы без последствий — получает формальный контроль, который не работает на практике.

Организационная среда определяет, как ИИ-система применяется и какие риски при этом возникают. Культура организации влияет на готовность сотрудников сообщать о проблемах с ИИ. В организации, где скорость вывода продукта на рынок ценится выше безопасности, оценка рисков превращается в формальность. В организации, где нет чёткого распределения ответственности за ИИ-систему, инциденты остаются без владельца.

Структура стимулов играет не менее важную роль. Если команда разработки вознаграждается за точность модели, но не за её устойчивость к враждебным воздействиям, устойчивость останется за пределами приоритетов. Если владелец бизнес-процесса отвечает за эффективность, но не за последствия ошибочных решений ИИ, он будет расширять автономию модели без оценки рисков. Если юридический отдел не вовлечён в проектирование ИИ-системы, вопросы защиты данных и соответствия требованиям всплывут после инцидента, а не до него.

Процессы организации тоже формируют профиль риска. Есть ли процедура утверждения новых сценариев применения ИИ? Проводится ли оценка воздействия перед запуском системы, которая затрагивает клиентов? Существует ли механизм обратной связи, позволяющий пользователям сообщить о подозрительном поведении модели? Включены ли ИИ-инциденты в общий план реагирования организации? Если ответ на эти вопросы отрицательный, технические контроли остаются изолированными мерами, не встроенными в управленческую ткань организации.

Взаимодействие человека и ИИ

Особого внимания требует точка взаимодействия человека с ИИ-системой. Здесь пересекаются свойства модели и свойства человека, и результат этого пересечения не всегда предсказуем. Исследования показывают, что при совместной работе человека и ИИ в задачах, тре-

бующих суждения, система может усиливать когнитивные искажения человека. Если модель подтверждает предположение оператора, он принимает решение быстрее и с большей уверенностью, даже если предположение ошибочно. Если модель даёт результат, противоречащий интуиции оператора, он может отвергнуть правильный ответ, опираясь на собственный опыт.

Конфигурации взаимодействия варьируются от полностью автономных до полностью ручных. Между ними лежит широкий спектр вариантов: модель предлагает решение, человек утверждает; модель действует автономно, человек контролирует по запросу; модель выполняет рутинные операции, человек включается при отклонениях. Выбор конфигурации — это управленческое решение, которое должно учитывать уровень риска, скорость принятия решений, квалификацию оператора и последствия ошибки. Архитектура системы должна поддерживать выбранную конфигурацию: если предусмотрено утверждение человеком, система должна предоставлять ему достаточно информации для принятия осмысленного решения, а не просто кнопку подтверждения.

Что это означает для безопасности

Социотехнический взгляд на ИИ-систему меняет подход к безопасности. Угрозы возникают не только со стороны внешнего злоумышленника и не только в техническом слое. Сотрудник, который обходит процедуру утверждения и подключает внешнюю модель к корпоративным данным, создаёт риск, сопоставимый с внешней атакой. Руководитель, который принимает решение о запуске ИИ-системы без оценки воздействия, несёт ответственность за последствия, которые мог бы предотвратить. Организация, которая не инвестирует в обучение персонала, получает систему, в которой дорогостоящие технические контроли обесцениваются человеческим фактором.

Защита ИИ-системы требует работы на трёх уровнях одновременно: технология, люди и процессы. Технические контроли ограничивают возможности системы. Обучение и осведомлённость формируют способность людей правильно использовать систему и распознавать проблемы. Процессы и управление создают структуру, в которой решения принимаются обоснованно, ответственность распределена, а обратная связь работает. Ни один из этих уровней не заменяет другой. Организация, которая инвестирует только в технологию, защищает систему, но не людей и процессы вокруг неё. Организация, которая выстраивает только процессы, получает документы без реальных контролей.

Эта книга последовательно рассматривает все три уровня. Технические меры защиты описаны в четвёртой части. Процессы управления — во второй и пятой. Роли, ответственность и человеческий контроль проходят сквозной темой через все части, потому что ни одна архитектурная схема не работает без людей, которые её поддерживают.

1.5. Активы и границы системы

Любая программа безопасности начинается с двух вопросов: что мы защищаем и где проходят границы того, что мы контролируем. В традиционной информационной безопасности ответы на эти вопросы давно формализованы. Организация ведёт реестр активов, определяет периметр сети, классифицирует данные, описывает зоны ответственности. Для ИИ-систем те же вопросы требуют пересмотра, потому что состав активов шире привычного, а границы системы сложнее и подвижнее.

Активы ИИ-системы

Активом является всё, что имеет ценность для организации и требует защиты. В ИИ-системе перечень активов выходит за рамки серверов, баз данных и исходного кода. Рассмотрим основные категории.



Рисунок 4. Карта активов ИИ-системы и расширенный периметр защиты.

Данные в ИИ-системе представляют собой актив особого рода. Обучающие данные воплощают месяцы работы по сбору, очистке, разметке и подготовке. Их утрата или компрометация означает, что модель придётся переобучать. Данные для тонкой настройки содержат специфические знания организации и часто включают конфиденциальную информацию. Контекстные данные в RAG-системах — базы знаний, документы, векторные хранилища — определяют качество и точность ответов. Входные и выходные данные, проходящие через систему при эксплуатации, могут содержать персональные данные, коммерческую тайну и другую чувствительную информацию. Журналы запросов и ответов фиксируют взаимодействие пользователей с моделью и сами по себе могут быть ценным и чувствительным ресурсом.

Модель — один из наиболее ценных активов. Веса обученной модели представляют интеллектуальную собственность организации. Их кража позволяет воспроизвести возможности системы без затрат на обучение. Для организаций, создающих собственные модели, это прямая потеря конкурентного преимущества. Но и при использовании сторонних моделей дообученная версия, адаптированная к задачам организации, становится самостоятельным активом. Кроме весов, к активам модели относятся её архитектура, гиперпараметры, конфигурация инференса и метаданные обучения.

Промпты и системные инструкции заслуживают отдельного внимания. В LLM-приложениях системный промпт определяет поведение модели: ограничения, стиль ответов, доступные функции, правила обработки запросов. Утечка системного промпта раскрывает логику работы приложения и может облегчить злоумышленнику поиск способов обхода ограничений. Библиотеки промптов, шаблоны для типовых задач и цепочки рассуждений, настроенные под бизнес-процессы организации, тоже представляют ценность.

Конвейеры обработки и оркестрации — это код и конфигурации, которые управляют потоком данных от входа до выхода: препроцессинг запросов, выбор модели, обращение к базе знаний, вызов инструментов, постобработка результатов. Компрометация конвейера позволяет атакующему повлиять на поведение системы, не затрагивая саму модель.

Инфраструктурные активы включают вычислительные ресурсы (GPU-кластеры, серверы инференса), хранилища данных и моделей, реестры артефактов, среды разработки и тестирования, ключи и токены доступа к API. Среда разработки, где исследователи экспериментируют с моделями, часто содержит наиболее ценные активы в наименее защищённом окружении.

Отдельный класс активов — это доверие пользователей и репутация организации. ИИ-система, которая принимает ошибочное решение, генерирует оскорбительный контент или раскрывает персональные данные, причиняет репутационный ущерб, который сложно измерить, но легко ощутить.

Где проходят границы системы

Определение границ ИИ-системы — задача более сложная, чем для традиционного приложения. В классической архитектуре граница проходит по периметру сети, по контуру приложения или по зоне ответственности команды. В ИИ-системе границы размываются по нескольким причинам.

Первая причина — зависимость от внешних моделей и сервисов. Когда организация использует модель через API облачного поставщика, часть системы работает за пределами её инфраструктуры. Организация контролирует запросы и ответы, но не контролирует саму модель, данные, на которых она обучена, и момент её обновления. Граница системы проходит по интерфейсу API, но риски возникают по обе стороны этой границы.

Вторая причина — интеграция с корпоративными системами. ИИ-ассистент, подключённый через MCP к почте, календарю, CRM и файловому хранилищу, фактически получает доступ к значительной части информационной среды организации. Формально он остаётся одним приложением. На практике его периметр воздействия охватывает все системы, к которым он подключён. Определение границы должно учитывать не только саму ИИ-систему, но и все ресурсы, к которым она имеет доступ.

Третья причина — участие внешних данных. RAG-система, обращающаяся к внешним источникам для формирования контекста, вводит в периметр данные, которые организация не производила и не контролирует. Документ, загруженный из интернета и добавленный в базу знаний, может содержать вредоносные инструкции, которые модель интерпретирует как команды. Граница доверия к данным перестаёт совпадать с границей инфраструктуры.

Четвёртая причина — многоагентное взаимодействие. Когда агент организации делегирует подзадачу агенту партнёра через протокол A2A, цепочка действий выходит за пределы контролируемой среды. Результат, возвращённый внешним агентом, может содержать искажённые данные или побочные инструкции. Определение границ в таких сценариях требует явного описания того, каким внешним компонентам организация доверяет, в какой мере и при каких условиях.

Нечёткое определение активов приводит к тому, что ценные ресурсы остаются без защиты. Организация защищает серверы и базы данных, но не учитывает системные промпты, библиотеки шаблонов и конфигурации конвейеров. Она классифицирует корпоративные данные, но не распространяет классификацию на обучающие выборки и векторные хранилища. Она ведёт реестр приложений, но не включает в него модели и агентов.

Нечёткое определение границ приводит к появлению зон, за которые никто не отвечает. Облачный поставщик считает, что безопасность данных — ответственность организации. Организация считает, что безопасность модели — ответственность поставщика. Между ними остаётся пространство, которое не покрыто контролями ни одной из сторон. В третьей главе, посвящённой распределению ответственности, мы подробно разберём модели совместной ответственности для различных сценариев развёртывания.

Практический вывод из этого раздела прост. Прежде чем строить архитектуру защиты, организация должна провести инвентаризацию активов ИИ-системы, выходящую за рамки привычного реестра ИТ-активов. Она должна включить в него данные всех категорий, модели,

промтты, конвейеры, интерфейсы и ключи доступа. И она должна определить границы каждой ИИ-системы с учётом внешних зависимостей, интеграций и агентного взаимодействия. Без этой работы оценка рисков, моделирование угроз и выбор мер защиты строятся на неполной картине, а неполная картина неизбежно оставляет пробелы.

Источники и дальнейшее чтение к главе 1

- Model Context Protocol — официальная спецификация, Anthropic / открытый стандарт, modelcontextprotocol.io.
- Agent2Agent Protocol (A2A) — официальная спецификация, Linux Foundation (изначально Google).
- NIST AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology, январь 2023.
- ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system, раздел 7.3 (компетентность и осведомлённость).
- NIST AI 100—2. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. National Institute of Standards and Technology.
- OWASP Top 10 for LLM Applications. OWASP Foundation.
- OWASP Agentic AI — Threats and Mitigations. OWASP Gen AI Security Project.
- MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). MITRE Corporation.

Глава 2. Типы ИИ-систем и их архитектурные особенности

В первой главе мы разобрали общую анатомию ИИ-системы: компоненты, агенты, отличия от традиционного ПО, социотехническую природу и активы. Но за словами «ИИ-система» скрывается широкий спектр технологий, и каждая из них работает с информацией по-своему. Языковая модель, генерирующая текст, устроена иначе, чем система компьютерного зрения, распознающая дефекты на конвейере. Модель, синтезирующая речь, создаёт риски, которых нет у модели кредитного скоринга. Мультимодальная система, принимающая на вход текст, изображение и аудио одновременно, расширяет поверхность атаки за счёт каждого канала.

Для специалиста по безопасности различия между типами ИИ-систем имеют практическое значение. Уклоняющиеся воздействия на модели компьютерного зрения строятся на пиксельных возмущениях, незаметных человеческому глазу. Инъекции в промпт эксплуатируют способность языковой модели интерпретировать текст как инструкцию. Дипфейки используют генеративные модели изображений и видео для создания убедительных подделок. Атака, эффективная против одного типа системы, может быть бессмысленна для другого. Моделирование угроз, выбор контролей и тестирование безопасности должны учитывать конкретную технологию, а не абстрактный «ИИ».

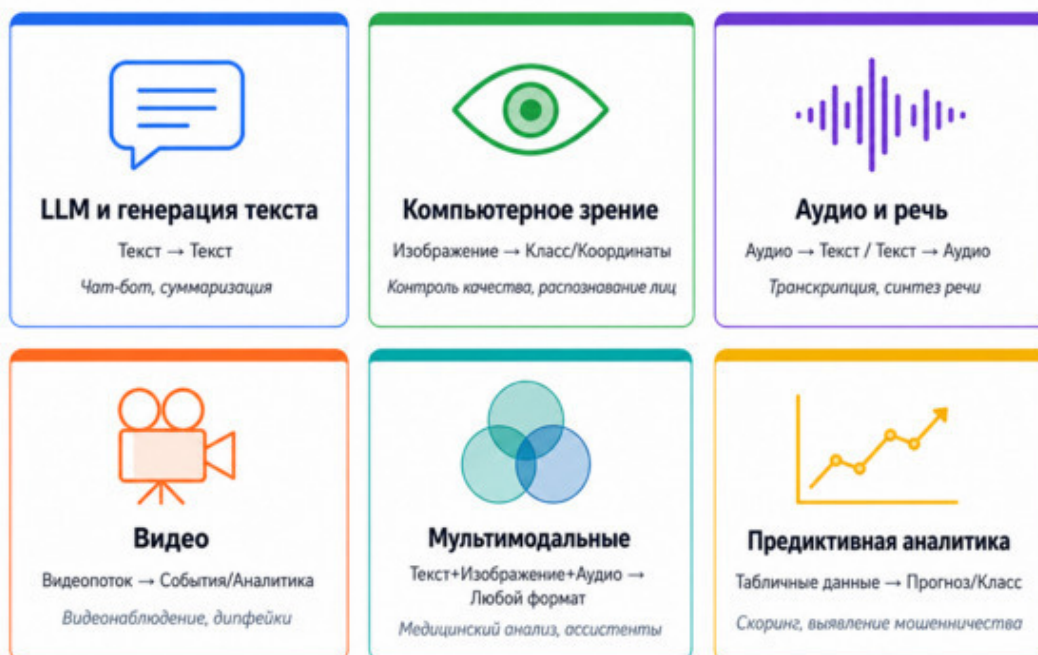


Рисунок 5. Таксономия типов ИИ-систем по назначению, архитектуре, степени автономности и характеру взаимодействия с пользователем.

Эта глава последовательно рассматривает основные типы ИИ-систем, с которыми сталкиваются организации: большие языковые модели, системы компьютерного зрения, модели генерации и анализа аудио, видеосистемы, мультимодальные модели и системы предиктивной аналитики. Для каждого типа мы разбираем архитектурные особенности, характерные сценарии применения и специфические риски безопасности. Завершающий раздел сводит эти раз-

личия в единую картину, которая помогает определить профиль угроз для конкретной системы в конкретной организации.

2.1. Большие языковые модели и генерация текста

Большие языковые модели (Large Language Models, LLM) стали наиболее обсуждаемой и наиболее быстро внедряемой категорией ИИ-систем. Когда руководители говорят о внедрении ИИ, в большинстве случаев они имеют в виду именно LLM: чат-ассистенты для сотрудников и клиентов, генерация документов, анализ текстов, автоматизация переписки, суммаризация отчётов, помощь в написании кода. Масштаб распространения этих систем определяет и масштаб рисков, которые они несут. Поэтому мы начинаем обзор типов ИИ-систем именно с них.

Архитектурной основой современных LLM является трансформер — тип нейронной сети, предложенный в 2017 году. Трансформер обрабатывает текст не последовательно, слово за словом, а параллельно, используя механизм внимания (attention), который позволяет модели учитывать связи между любыми элементами текста независимо от их расположения. Входной текст разбивается на токены — минимальные единицы обработки, которые могут соответствовать словам, частям слов или отдельным символам. Каждый токен преобразуется в числовой вектор, проходит через десятки или сотни слоёв трансформера и на выходе формирует вероятностное распределение следующего токена. Модель генерирует текст последовательно, токен за токеном, каждый раз выбирая продолжение на основе всего предшествующего контекста.

Размеры современных LLM измеряются миллиардами и сотнями миллиардов параметров. Контекстное окно — объём текста, который модель может учитывать при генерации ответа — варьируется от сотен тысяч до нескольких миллионов токенов. Эти масштабы определяют как возможности моделей, так и требования к инфраструктуре: обучение требует кластеров из тысяч специализированных процессоров, а продуктивный инференс — значительных вычислительных ресурсов для обработки каждого запроса.

Организации используют LLM в нескольких конфигурациях, и каждая из них формирует собственный профиль риска. Вызов внешней модели через API поставщика — наиболее распространённый вариант. Организация отправляет запрос и получает ответ, не контролируя внутреннее устройство модели, данные, на которых она обучена, и момент обновления. Запросы и ответы проходят через инфраструктуру поставщика, и вопрос о том, сохраняются ли они, используются ли для дообучения и кто имеет к ним доступ, регулируется условиями сервиса, которые организация не всегда изучает до начала использования.

Локальное развёртывание модели с открытыми весами даёт организации контроль над средой выполнения, но возлагает на неё ответственность за безопасность инфраструктуры, управление доступом и проверку целостности модели. Модель, загруженная из открытого репозитория, может содержать вредоносный код в сериализованных весах или скрытые закладки, внедрённые при обучении. Организации, выбирающие этот путь, должны проверять происхождение модели, валидировать контрольные суммы и ограничивать доступ к среде, в которой модель работает.

Дообучение (fine-tuning) предобученной модели на корпоративных данных создаёт гибрид: модель наследует общие способности базовой модели и приобретает специфические знания из данных организации. С точки зрения безопасности дообучение вводит дополнительный риск. Корпоративные данные, использованные при дообучении, могут быть извлечены из ответов модели. Процесс дообучения сам по себе может стать точкой атаки, если злоумышленник получает возможность повлиять на обучающую выборку.

Отдельная архитектурная конфигурация — LLM-приложение с дополненной генерацией (RAG). В этом случае модель не хранит все нужные знания в своих параметрах, а получает контекст из внешней базы знаний в момент запроса. Текст запроса дополняется релевантными

фрагментами документов, извлечёнными из векторного хранилища, и модель формирует ответ с учётом этого контекста. RAG расширяет возможности модели, но одновременно расширяет поверхность атаки: через внешние документы в контекст модели может попасть вредоносная инструкция, а ответ может содержать фрагменты конфиденциальных документов, к которым пользователь не должен иметь доступа.

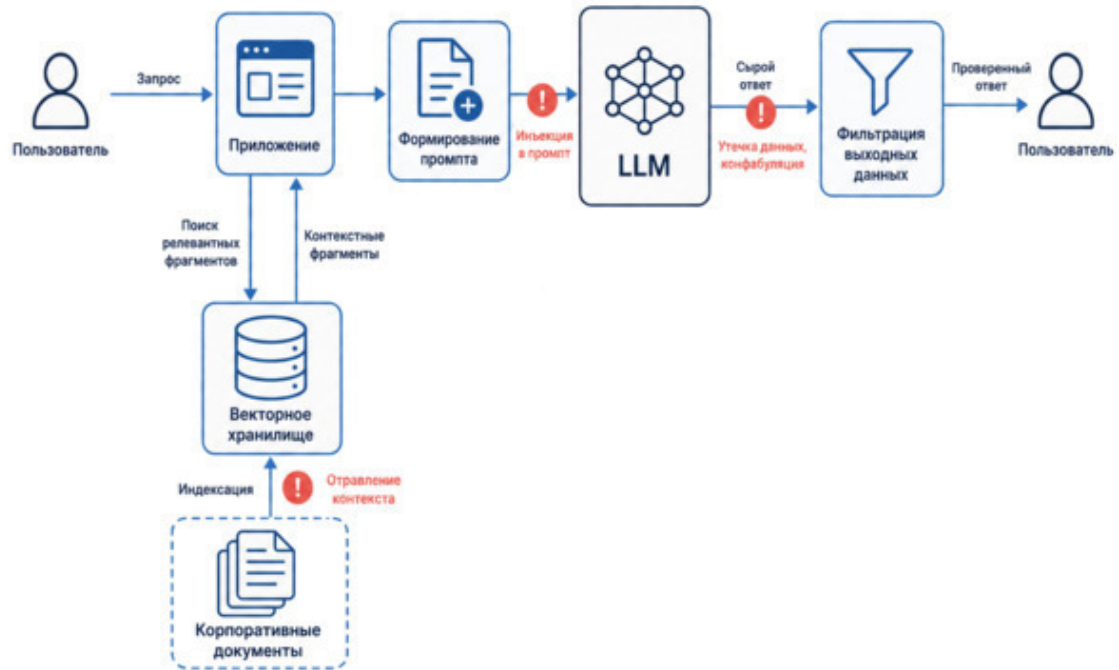


Рисунок 6. Архитектура LLM-приложения с дополненной генерацией на основе поиска: запрос, оркестрация, поиск, векторное хранилище, контекст и языковая модель.

С точки зрения безопасности LLM обладают набором специфических уязвимостей, которые не встречаются в других типах ИИ-систем. Инъекция в промпт (prompt injection) позволяет злоумышленнику изменить поведение модели через специально сформированный текст. Прямая инъекция вводится в запросе пользователя. Косвенная — встраивается в документ, письмо или веб-страницу, которые модель обрабатывает как часть контекста. Модель не различает инструкцию разработчика и текст, поступивший извне, потому что и то, и другое для неё — последовательность токенов.

Утечка системного промпта раскрывает внутренние инструкции приложения: правила поведения, ограничения, описание доступных инструментов, иногда даже ключи и идентификаторы. Конфабуляция — способность модели уверенно генерировать несуществующие факты, ссылки и данные — создаёт риски при использовании LLM в юридических, медицинских и финансовых контекстах, где ошибка может привести к конкретному ущербу. Неограниченное потребление ресурсов при обработке сложных или злонамеренно сформированных запросов может привести к отказу в обслуживании или значительным финансовым затратам.

Каждая из этих уязвимостей подробно рассматривается в десятой главе. Здесь важно зафиксировать принципиальный момент: LLM — это системы, чьё поведение управляется текстом, и именно текст становится основным вектором атаки. Защита LLM-приложения требует контролей на уровне входных и выходных данных, ограничения полномочий модели, изоляции

контекста и непрерывного мониторинга того, что модель генерирует и какие действия выполняет от имени пользователя.

2.2. Компьютерное зрение и обработка изображений

Если большие языковые модели работают с текстом, то системы компьютерного зрения работают с визуальной информацией: фотографиями, видеопотоками, медицинскими снимками, спутниковыми изображениями, сканами документов. Организации используют их для распознавания лиц при аутентификации, контроля качества продукции на конвейере, анализа медицинских изображений, мониторинга территорий через камеры наблюдения, автоматической обработки документов и классификации визуального контента. Автономные транспортные средства полагаются на компьютерное зрение для распознавания пешеходов, знаков и других объектов в режиме реального времени. Масштаб применения широк, и во многих из этих сценариев ошибка модели может привести к последствиям для здоровья, безопасности или прав человека.

Архитектурную основу большинства задач компьютерного зрения составляют свёрточные нейронные сети (Convolutional Neural Networks, CNN). Свёрточная сеть обрабатывает изображение, пропуская его через серию фильтров, каждый из которых выделяет определённые визуальные признаки: границы, текстуры, формы, сочетания элементов. Ранние слои сети распознают простые паттерны, такие как горизонтальные и вертикальные линии. Более глубокие слои комбинируют эти паттерны в сложные структуры: контуры объектов, лица, символы. На выходе сеть формирует классификацию изображения, определяет координаты объектов или создаёт попиксельную разметку сцены.

Задачи компьютерного зрения различаются по сложности. Классификация определяет, что изображено на фотографии: кошка, автомобиль, дефект на детали. Детекция находит объекты на изображении и определяет их координаты. Сегментация разделяет изображение на области, относящиеся к разным объектам или категориям. Распознавание лиц сопоставляет изображение лица с хранящимся образцом. Генерация изображений создаёт новые визуальные данные на основе текстового описания, шаблона или случайного начального состояния. Каждая из этих задач создаёт собственный профиль рисков, и защитные меры, эффективные для одной, могут оказаться бесполезными для другой.

Генерация изображений заслуживает отдельного внимания, потому что за последние годы она перешла из области исследований в массовое применение. Модели на основе диффузии (diffusion models) способны создавать фотореалистичные изображения по текстовому описанию. Процесс начинается с случайного шума, который модель постепенно преобразует в изображение, соответствующее запросу. Генеративно-сопоставительные сети (GAN) используют другой подход: две нейронные сети конкурируют друг с другом, одна генерирует изображение, другая оценивает его правдоподобность, и в результате качество генерации улучшается с каждой итерацией.

С точки зрения безопасности системы компьютерного зрения подвержены набору угроз, которые существенно отличаются от угроз текстовым моделям. Наиболее изученная категория — уклоняющиеся воздействия, или адверсариальные атаки. Суть их в том, что минимальное, незаметное человеческому глазу изменение изображения заставляет модель принять ошибочное решение. Исследователи продемонстрировали, что добавление специально рассчитанного шума к фотографии панды заставляет классификатор с высокой уверенностью определить её как гиббона. Изменения настолько малы, что два изображения визуально неотличимы для человека, но модель видит их совершенно по-разному.

На практике эти атаки принимают разные формы. Адверсариальные патчи — небольшие наклейки или изображения, размещённые в поле зрения камеры, которые заставляют

модель игнорировать объект или классифицировать его неправильно. В экспериментах удавалось добиться того, что система автопилота переставала распознавать дорожный знак при наклеивании на него специально подобранного стикера. Модификация физических объектов — нанесение определённого узора на одежду или аксессуары, которое помогает человеку уклониться от распознавания системой видеонаблюдения. Эти сценарии выходят далеко за пределы лабораторных экспериментов и имеют прямое значение для организаций, использующих компьютерное зрение в системах физической безопасности, контроля доступа и критической инфраструктуры.

Системы распознавания лиц несут дополнительные риски, связанные с приватностью и правами человека. Биометрические данные — одна из наиболее чувствительных категорий персональных данных, и их обработка регулируется строгими нормами в большинстве юрисдикций. EU AI Act относит системы биометрической идентификации в реальном времени в общественных пространствах к категории запрещённых или высокорисковых сценариев. Помимо регуляторных требований, системы распознавания лиц демонстрируют различную точность для разных демографических групп: исследования показывали, что модели распознавания лиц ошибаются значительно чаще при работе с определёнными этническими группами и возрастными категориями. Если организация использует такую систему для принятия решений, затрагивающих людей, неравномерная точность превращается в дискриминационный риск.

Обход биометрической аутентификации — ещё один вектор атаки. Злоумышленники используют фотографии, загруженные из открытых источников, и превращают их в поддельные видеозаписи, демонстрирующие движения лица: моргание, повороты головы, мимику. Виртуальные камеры подменяют видеопоток, и система принимает поддельное видео за живого человека. Этот метод был успешно применён для обхода систем аутентификации в финансовых и государственных сервисах. Защита от него требует продвинутых методов определения подлинности: анализ текстуры кожи, микродвижений, глубины изображения, аппаратная привязка к доверенной камере.

Генеративные модели изображений создают отдельный класс рисков, связанный с дипфейками и синтетическим контентом. Организация может столкнуться с дипфейком руководителя, используемым для социальной инженерии. Сгенерированные изображения могут применяться для создания поддельных документов, удостоверений, отчётов. Синтетический контент может нарушать авторские права, воспроизводя стиль или элементы защищённых произведений. Маркировка ИИ-сгенерированного контента, водяные знаки и средства проверки происхождения (provenance) становятся важными элементами защиты, но ни один из них пока не даёт абсолютных гарантий.

Есть и ещё одно свойство, отличающее системы компьютерного зрения от текстовых моделей: их часто развёртывают на периферийных устройствах с ограниченными ресурсами. Камера видеонаблюдения, дрон, промышленный контроллер, мобильный терминал — всё это среды, где модель работает локально, вне защищённого периметра организации. Компрометация такого устройства может дать атакующему прямой доступ к модели, её весам и обрабатываемым данным. Защита периферийных развёртываний требует шифрования модели на устройстве, аппаратной аттестации, контроля целостности и мониторинга поведения модели на удалённых точках.

Для организации, внедряющей системы компьютерного зрения, из этого следует практический вывод. Оценка рисков должна учитывать не только точность модели, но и её устойчивость к враждебным воздействиям, справедливость для разных групп пользователей, условия развёртывания и потенциальное применение генерируемых данных. Модель, прошедшая проверку точности на тестовом наборе, может оказаться уязвимой к адверсариальному патчу, предвзятой к определённой демографической группе или небезопасной при развёртывании на

незащищённом устройстве. Каждый из этих аспектов требует отдельной проверки и отдельного набора контролей.

2.3. Генерация и анализ аудио и речи

Звук — один из наиболее естественных каналов взаимодействия человека с технологией. Голосовые ассистенты принимают команды и отвечают на вопросы. Системы распознавания речи транскрибируют телефонные разговоры, записи совещаний и обращения клиентов. Модели синтеза речи озвучивают навигаторы, справочные системы и автоматические уведомления. Системы голосовой биометрии идентифицируют звонящего по характеристикам голоса. За каждой из этих функций стоят ИИ-модели, и каждая из них создаёт риски, которые организации привыкли недооценивать.

Технологически работа с аудио и речью делится на два направления. Распознавание речи (Automatic Speech Recognition, ASR) преобразует звуковой сигнал в текст. Модель принимает на вход аудиозапись, разбивает её на фрагменты, извлекает акустические признаки и сопоставляет их с языковой моделью, чтобы сформировать наиболее вероятную текстовую расшифровку. Современные ASR-системы построены на архитектуре трансформера и способны работать с десятками языков, распознавать речь в условиях шума, различать нескольких говорящих и адаптироваться к акценту. Организации используют ASR для автоматической транскрипции звонков в контакт-центрах, протоколирования совещаний, голосового управления интерфейсами и анализа переговоров в целях контроля соответствия требованиям.

Синтез речи (Text-to-Speech, TTS) работает в обратном направлении: преобразует текст в звуковой сигнал, имитирующий человеческую речь. Ранние системы звучали механически и узнаваемо. Современные модели генерируют речь, которую сложно отличить от записи живого человека. Они воспроизводят интонации, паузы, эмоциональную окраску и индивидуальные особенности произношения. Организации используют TTS для автоматизации клиентских коммуникаций, озвучивания контента, создания голосовых интерфейсов и локализации продуктов на разных языках.

Между этими двумя направлениями лежит область, которая быстро развивается и порождает наиболее острые вопросы безопасности: клонирование голоса. Современные модели способны воспроизвести голос конкретного человека по нескольким секундам записи. Они анализируют тембр, ритм, манеру говорить и создают синтетическую копию, которая звучит убедительно даже для знакомых людей. Технология находит легитимное применение: дубляж фильмов, восстановление голоса для людей, утративших способность говорить, персонализация голосовых интерфейсов. Но она же создаёт инструмент для мошенничества и социальной инженерии, масштаб которого трудно переоценить.

Инциденты, связанные с голосовыми дипфейками, перестали быть теоретической угрозой. В одном из широко описанных случаев злоумышленники воспроизвели голос генерального директора европейской энергетической организации и по телефону убедили подчинённого перевести около 220 тысяч евро на подконтрольный счёт. Сотрудник был уверен, что разговаривает с руководителем, поскольку голос, интонации и манера речи совпадали. В другом случае, в Гонконге, участники видеоконференции оказались сгенерированными дипфейками: изображение и голос нескольких руководителей были синтезированы в реальном времени, что привело к переводу значительных средств на мошеннический счёт. Эти случаи показывают, что голосовой дипфейк способен преодолеть один из наиболее устойчивых механизмов доверия в организации: узнавание голоса знакомого человека.

На тёмных рынках появились сервисы «дипфейк как услуга» (deepfake-as-a-service), которые значительно снизили порог входа для злоумышленников. Заказчик предоставляет образец голоса, сценарий и получает готовую аудиозапись в течение нескольких часов. Сто-

имость таких услуг варьируется, но остаётся вполне доступной. Некоторые платформы предлагают готовые голосовые клоны публичных персон, которые можно использовать сразу, без индивидуальной настройки. Это означает, что атаки с использованием голосовых дипфейков перестали требовать глубокой технической экспертизы и доступны широкому кругу злоумышленников.

Системы голосовой биометрии, используемые для аутентификации в банковских и государственных сервисах, оказываются уязвимы к тем же технологиям. Если система сравнивает голос звонящего с хранящимся образцом, синтетический голос, созданный по записи из открытых источников, может пройти проверку. Исследователи демонстрировали успешный обход голосовой аутентификации с помощью клонированного голоса в лабораторных условиях, и есть основания полагать, что подобные атаки уже применяются на практике. Защита требует дополнительных уровней верификации: анализ спектральных характеристик, выявление артефактов синтеза, проверка канала связи, комбинирование голосовой биометрии с другими факторами аутентификации.

Помимо мошенничества, аудиосистемы подвержены адверсариальным атакам, аналогичным тем, которые применяются к компьютерному зрению. Исследователи показали, что в аудиозапись можно встроить возмущения, незаметные человеческому уху, но заставляющие модель распознавания речи выдать произвольный текст. Звуковой файл, который человек воспринимает как обычную музыку, может содержать скрытую команду, которую голосовой ассистент интерпретирует как инструкцию: открыть веб-страницу, отправить сообщение, выполнить действие. Эта категория атак особенно опасна в сценариях, где голосовой интерфейс используется для управления устройствами, транзакциями или доступом к системам.

Отдельного внимания заслуживают вопросы приватности. Аудиоданные содержат не только содержание речи, но и биометрические характеристики говорящего, эмоциональное состояние, фоновые звуки, позволяющие определить местоположение. Организация, которая транскрибирует звонки клиентов с помощью ASR-модели, обрабатывает чувствительные данные. Если эта обработка происходит через внешний API, аудиозаписи передаются поставщику, и условия их хранения, использования и удаления определяются договором, а не техническими мерами организации. Если записи используются для дообучения модели, голоса конкретных людей могут оказаться встроенными в параметры модели и теоретически могут быть извлечены.

Генерация музыки и звуковых эффектов с помощью ИИ добавляет ещё одно измерение рисков. Модели, обученные на корпусах музыкальных произведений, способны генерировать композиции, стилистически близкие к работам конкретных исполнителей. Вопросы авторских прав, возникающие при этом, пока не получили однозначного правового разрешения в большинстве юрисдикций. Организация, использующая генеративную аудиомодель для создания контента, должна учитывать юридические риски, связанные с потенциальным сходством сгенерированных произведений с защищёнными оригиналами.

Для организации практический вывод состоит в следующем. Аудио и речевые системы работают с одним из самых чувствительных типов данных и одновременно подвержены одним из наиболее убедительных форм мошенничества. Оценка рисков должна учитывать три направления: защиту от голосовых дипфейков в процессах, где решения принимаются на основании телефонного звонка или голосовой команды; защиту голосовой биометрии от обхода с помощью синтетического голоса; и защиту приватности аудиоданных, обрабатываемых ИИ-моделями. Организации, в которых голос руководителя является достаточным основанием для санкционирования финансовой операции, должны пересмотреть этот процесс с учётом того, что голос больше не является надёжным идентификатором.

2.4. Генерация и анализ видео

Видео объединяет визуальный ряд, звук и временную последовательность, и именно эта комбинация делает его наиболее убедительным медиаформатом. Человек доверяет видеозаписи больше, чем тексту или отдельному изображению, потому что подделка видео долгое время требовала профессиональных навыков и значительных ресурсов. ИИ изменил это соотношение. Современные модели генерируют видео по текстовому описанию, превращают статичные фотографии в движущееся изображение, подменяют лица и голоса в записях и создают полностью синтетические сцены, которые сложно отличить от реальных. Для организаций это означает одновременно новые возможности и новый уровень угроз.

Анализ видео с помощью ИИ — более зрелая область, чем генерация. Организации используют видеоаналитику для наблюдения за территорией, подсчёта посетителей, распознавания номерных знаков, контроля соблюдения техники безопасности на производстве, анализа поведения покупателей в торговых залах. Архитектурно видеоаналитика строится на тех же свёрточных нейронных сетях, что и обработка изображений, но добавляет временное измерение: модель должна отслеживать объекты между кадрами, распознавать действия и последовательности событий, а не только статичные сцены. Рекуррентные нейронные сети и трансформеры для видео позволяют учитывать контекст движения и изменения на протяжении длительных фрагментов записи.

С точки зрения безопасности видеоаналитика наследует все уязвимости компьютерного зрения, описанные в предыдущем разделе, и добавляет к ним специфические. Адверсариальные воздействия, эффективные для одного кадра, могут оказаться нестабильными при смене ракурса, освещения или позы объекта в следующем кадре. Это одновременно и преимущество, и проблема: модель сложнее обмануть устойчиво, но и защитные проверки, успешные для одного кадра, не гарантируют надёжности на всей последовательности. Атакующий, размещающий адверсариальный патч в поле зрения камеры видеонаблюдения, должен учитывать, что камера фиксирует объект под разными углами, при разном освещении и на разном расстоянии. Исследования показали, что такие устойчивые патчи возможны: специально подобранные узоры на одежде позволяли человеку оставаться незамеченным для систем детекции пешеходов на протяжении длительных фрагментов видеозаписи.

Генерация видео переживает стремительное развитие. Модели на основе диффузии, адаптированные для работы с временными последовательностями, способны создавать видеоролики продолжительностью в десятки секунд с реалистичной физикой движения, освещением и детализацией. Организации начинают использовать генерацию видео для создания маркетингового контента, обучающих материалов, прототипов рекламных роликов и презентаций. Стоимость производства видео с участием синтетических персонажей значительно ниже традиционных съёмок, и этот разрыв продолжает увеличиваться.

Но те же технологии создают инструментарий для злоумышленников. Видеодипфейки — синтетические видеозаписи, в которых лицо или тело одного человека заменено на другого — стали одной из наиболее обсуждаемых угроз. Инцидент в Гонконге, упомянутый в предыдущем разделе, показал, что дипфейк может быть достаточно убедительным для обмана участника видеоконференции. Злоумышленники синтезировали внешность и голос нескольких руководителей организации в реальном времени, и сотрудник, наблюдавший знакомые лица и слышавший знакомые голоса, выполнил финансовую операцию по их указанию. Этот случай продемонстрировал, что видеодипфейки способны преодолеть даже многоканальную верификацию: визуальное узнавание, голос и контекст рабочего совещания оказались недостаточными для выявления подделки.

Помимо мошенничества, видеодипфейки применяются для дискредитации, шантажа и манипулирования общественным мнением. Синтетические видеозаписи с участием руководителей организаций, политиков или публичных персон могут быть распространены в социальных сетях и нанести репутационный ущерб до того, как будет установлена их поддельность. Скорость распространения контента в интернете значительно превышает скорость его верификации, и организация может оказаться в ситуации, когда опровержение приходит слишком поздно.

Генерация видео поднимает и юридические вопросы, аналогичные тем, которые возникают при генерации изображений и аудио. Синтетическое видео может воспроизводить внешность конкретных людей без их согласия. Оно может содержать элементы, защищённые авторским правом: архитектуру зданий, дизайн продуктов, фрагменты фильмов или рекламных материалов. Организация, использующая генеративные видеомодели для производства контента, должна выстроить процесс проверки выходных материалов на предмет нарушения прав третьих лиц, а также обеспечить маркировку ИИ-сгенерированного контента в соответствии с требованиями, которые вводятся в ряде юрисдикций.

Средства обнаружения видеодипфейков развиваются параллельно с технологиями генерации, но пока отстают. Детекторы анализируют артефакты, характерные для синтетического видео: несогласованность освещения на лице и фоне, неестественные микродвижения губ, отсутствие характерных для живого человека пульсаций кровеносных сосудов под кожей, нарушения текстуры на границах подменённых областей. Проблема в том, что каждое новое поколение генеративных моделей устраняет артефакты, на которых были обучены детекторы предыдущего поколения. Возникает гонка между генерацией и обнаружением, в которой у генерации есть структурное преимущество: создать убедительный дипфейк проще, чем доказать, что видеозапись поддельна.

Организациям, которые полагаются на видеокоммуникации для принятия решений, необходимо пересмотреть свои процессы верификации. Видеоконференция больше не является надёжным подтверждением личности участника. Критические решения — финансовые операции, предоставление доступа, утверждение контрактов — требуют дополнительных каналов подтверждения, не зависящих от видео и голоса. Кодовые слова, обратные звонки по заранее согласованным номерам, двухфакторная аутентификация через независимый канал — всё это меры, которые добавляют трение в процесс, но снижают риск обмана через синтетические медиа.

Для организаций, работающих с видеоаналитикой на объектах критической инфраструктуры, практический вывод дополняется необходимостью тестирования устойчивости системы к адверсариальным воздействиям в физической среде. Камера на входе в защищённую зону, контролирующая доступ, должна быть проверена на устойчивость к патчам, подмене видеопотока и презентационным атакам. Эта проверка выходит за рамки стандартного тестирования точности модели и требует специализированных сценариев, приближённых к условиям реальной атаки.

2.5. Мультимодальные модели

В предыдущих разделах мы рассматривали системы, работающие с одной модальностью: текстом, изображением, аудио или видео. Но организации всё чаще сталкиваются с моделями, которые принимают на вход и генерируют данные нескольких типов одновременно. Пользователь загружает фотографию и задаёт вопрос текстом. Модель анализирует медицинский снимок и формирует текстовое заключение. Система принимает голосовую команду, интерпретирует изображение с камеры и отвечает синтезированной речью. Виртуальный ассистент читает документ, извлекает из него таблицу, строит график и объясняет результат в текстовом и голо-

совом формате. Такие модели называют мультимодальными, и их распространение меняет ландшафт рисков.

Архитектурно мультимодальные модели объединяют несколько кодировщиков, каждый из которых обрабатывает свой тип входных данных, и общее пространство представлений, в котором информация из разных модальностей приводится к единому формату. Текст, изображение и аудио преобразуются в числовые векторы, которые модель может сопоставлять, комбинировать и использовать для генерации результата в любой из поддерживаемых модальностей. Трансформерная архитектура, изначально разработанная для текста, оказалась достаточно гибкой, чтобы работать с токенами из разных источников: текстовыми, визуальными, аудио. Современные мультимодальные модели обрабатывают все модальности в едином потоке, и граница между «текстовой» и «визуальной» частью модели становится всё менее отчётливой.

Организации используют мультимодальные модели в сценариях, где одной модальности недостаточно. Медицинская система анализирует рентгеновский снимок совместно с историей болезни пациента и генерирует предварительное заключение. Система контроля качества на производстве совмещает изображение с камеры с показаниями датчиков и классифицирует дефект. Ассистент для работы с документами извлекает текст, таблицы и изображения из PDF-файла и отвечает на вопросы, учитывая все три типа содержимого. Чат-бот клиентской поддержки принимает фотографию повреждённого товара и текстовое описание проблемы, чтобы сформировать решение по обращению. Каждый из этих сценариев требует от модели способности устанавливать связи между данными разной природы и формировать ответ, основанный на их совокупности.

С точки зрения безопасности мультимодальные модели создают расширенную поверхность атаки, которая складывается из уязвимостей каждой отдельной модальности и новых уязвимостей, возникающих на их стыке. Инъекция в промпт, которая в текстовой модели передаётся через текст, в мультимодальной системе может быть встроена в изображение. Исследователи продемонстрировали, что текст, размещённый на изображении мелким шрифтом или скрытый за счёт цветовой маскировки, может быть прочитан моделью и интерпретирован как инструкция, хотя пользователь его не видит. Аналогичным образом вредоносные команды могут быть скрыты в аудиодорожке видеофайла: человек слышит обычную речь, а модель извлекает из звукового потока дополнительные инструкции.

Это означает, что для мультимодальной системы каждый канал входных данных становится потенциальным вектором атаки. Текст, изображение, аудио, видео, документы, таблицы — всё, что модель способна обработать, может содержать встроенные инструкции. При этом фильтрация входных данных усложняется: проверка текста на наличие инъекций — задача, для которой существуют отработанные подходы. Проверка изображения на наличие скрытого текста, который будет прочитан моделью, требует принципиально иных методов. Проверка аудиодорожки на наличие возмущений, незаметных для человека, но воспринимаемых моделью, — ещё более сложная задача. Организация, развёртывающая мультимодальную систему, должна выстраивать фильтрацию на каждом канале, а не только на текстовом входе.

Взаимодействие между модальностями порождает ещё один класс проблем: несогласованность. Модель может получить на вход изображение, содержащее одну информацию, и текстовый запрос, содержащий противоречащую информацию. Как она разрешит противоречие? Ответ зависит от архитектуры и обучения, и разработчик не всегда может предсказать результат. В контексте безопасности злоумышленник может использовать эту несогласованность: отправить изображение с одним содержанием и текст, описывающий другое, чтобы заставить модель сгенерировать результат, которого не дала бы ни одна модальность по отдельности. Это создаёт атаки, специфичные именно для мультимодальных систем и не имеющие аналогов в одномодальных.

Утечка информации через смену модальности — ещё одна угроза, характерная для мультимодальных систем. Модель, которая получает конфиденциальный документ в виде изображения и отвечает на вопросы о его содержании, может раскрыть чувствительные данные в текстовом ответе. Информация, защищённая в одной модальности (например, текст на изображении, которое не индексируется текстовыми поисковыми системами), становится доступной при переводе в другую модальность. Организация должна учитывать, что мультимодальная модель способна конвертировать данные между форматами, и классификация чувствительности должна распространяться на все модальности, через которые информация может быть извлечена.

Вопросы приватности приобретают дополнительную остроту. Мультимодальная модель, обрабатывающая видеозвонок, одновременно имеет доступ к изображению лица участника, его голосу, содержанию речи, фоновой обстановке и текстовым сообщениям в чате. Объём персональных данных, доступных модели в один момент времени, значительно превышает то, что получает одномодальная система. Если эти данные передаются внешнему поставщику для обработки, организация должна оценить риски с учётом всех модальностей одновременно, а не каждой по отдельности.

На практике мультимодальность добавляет сложности ко всем процессам безопасности, которые организация выстраивает для ИИ-систем. Моделирование угроз должно учитывать атаки через каждый канал ввода и их комбинации. Тестирование безопасности должно включать кросс-модальные сценарии: инъекции через изображение, скрытые команды в аудио, противоречия между модальностями. Мониторинг должен отслеживать поведение модели не только по текстовым входам и выходам, но и по визуальным, аудио и смешанным запросам. Классификация данных должна учитывать конвертацию между форматами. Всё это увеличивает стоимость и сложность программы безопасности, но игнорирование кросс-модальных рисков оставляет организацию с защитой, построенной для одномодального мира в мультимодальной реальности.

2.6. Предиктивная аналитика, рекомендательные системы и классификация

Генеративный ИИ привлекает основное внимание прессы и руководителей, но значительная часть ИИ-систем, работающих в организациях, решает более традиционные задачи. Модель предсказывает, какие клиенты уйдут в следующем квартале. Другая модель оценивает кредитоспособность заёмщика. Третья рекомендует товары на основе истории покупок. Четвёртая классифицирует транзакции как легитимные или мошеннические. Эти системы внедрены глубже, работают дольше и зачастую влияют на решения с более ощутимыми последствиями для конкретных людей, чем генеративные модели. При этом внимания к их безопасности уделяется меньше, потому что они воспринимаются как зрелые и понятные.

Предиктивная аналитика строится на моделях, обученных выявлять закономерности в исторических данных и переносить их на будущие события. Модель анализирует признаки — возраст клиента, частоту обращений, сумму покупок, историю платежей — и формирует прогноз: вероятность оттока, ожидаемый доход, риск дефолта. Архитектура таких моделей варьируется от классических алгоритмов машинного обучения — деревья решений, случайные леса, градиентный бустинг — до глубоких нейронных сетей. Многие из этих моделей работают в режиме реального времени, обрабатывая тысячи запросов в секунду и влияя на автоматические решения без участия человека.

Рекомендательные системы определяют, какой контент, товар или действие предложить пользователю. Они анализируют поведение конкретного пользователя (коллаборативная фильтрация), характеристики объектов (контентная фильтрация) или комбинируют оба подхода.

Эти системы формируют значительную часть пользовательского опыта в электронной коммерции, стриминговых сервисах, новостных платформах и социальных сетях. Их влияние на поведение людей трудно переоценить: рекомендация определяет, что человек увидит, прочитает, купит или посмотрит.

Классификация — одна из базовых задач машинного обучения. Модель относит входные данные к одной из заранее определённых категорий. Письмо классифицируется как легитимное или спам. Транзакция — как обычная или подозрительная. Медицинский снимок — как норма или патология. Резюме кандидата — как соответствующее требованиям или нет. Обращение клиента — как жалоба, запрос информации или заявка на обслуживание. Простота формулировки задачи маскирует сложность последствий: ошибочная классификация транзакции может заблокировать легитимную операцию или пропустить мошенническую, а ошибочная классификация медицинского снимка — задержать лечение или назначить ненужное вмешательство.

С точки зрения безопасности эти системы подвержены ряду угроз, часть которых пересекается с угрозами генеративным моделям, а часть — специфична. Отравление обучающих данных работает здесь даже эффективнее, чем в LLM, потому что модели предиктивной аналитики обычно обучаются на меньших и более контролируемых наборах данных. Внедрение относительно небольшого количества искажённых записей может сместить границу решения модели. Классическая иллюстрация: злоумышленник, знающий структуру данных системы выявления мошенничества, целенаправленно совершает серию транзакций с определёнными характеристиками, чтобы при следующем цикле обучения модель перестала считать такие транзакции подозрительными.

Уклоняющиеся воздействия применимы к любой модели классификации. Если злоумышленник понимает, какие признаки влияют на решение модели, он может модифицировать свои действия так, чтобы оставаться ниже порога обнаружения. Мошенник, знающий, что система блокирует транзакции свыше определённой суммы из определённого региона, разобьёт операцию на несколько меньших переводов из других географических точек. Отправитель спама, знающий, какие слова повышают оценку нежелательности, перефразирует сообщение, сохраняя смысл, но обходя фильтр. Это не гипотетические сценарии, а повседневная практика: модели выявления мошенничества и спама находятся в постоянной гонке с теми, кто пытается их обойти.

Извлечение модели представляет серьёзную угрозу для организаций, чья конкурентная позиция зависит от качества предиктивных моделей. Атакующий отправляет множество запросов к модели, доступной через API, и по совокупности ответов восстанавливает её поведение с достаточной точностью, чтобы построить собственную копию. Для систем кредитного скоринга, ценообразования или оценки рисков потеря модели означает потерю коммерческого преимущества. Защита от извлечения включает ограничение числа запросов, огрубление ответов (возвращение категории вместо точной вероятности), мониторинг аномальных паттернов обращений и ограничение объёма информации в ответе.

Вопросы справедливости и дискриминации стоят для этих систем острее, чем для генеративных моделей, потому что предиктивные модели часто принимают решения, непосредственно затрагивающие права и возможности людей. Модель кредитного скоринга, обученная на исторических данных, в которых определённые демографические группы получали одобрение реже, воспроизведёт этот паттерн как закономерность. Модель отбора резюме, обученная на данных о прошлых наймах, унаследует предпочтения, которые организация, возможно, хотела бы преодолеть. EU AI Act относит ИИ-системы, используемые для оценки кредитоспособности, подбора персонала, назначения социальных выплат и правоприменения, к категории высокого риска и предъявляет к ним повышенные требования к прозрачности, объяснимости и отсутствию дискриминации.

Дрейф данных, описанный в разделе 1.3, особенно опасен для предиктивных моделей, потому что они чувствительны к изменениям в распределении входных данных. Модель прогнозирования спроса, обученная до пандемии, перестала давать адекватные прогнозы при резком изменении потребительского поведения. Модель оценки кредитного риска, откалиброванная в условиях низкой инфляции, может систематически занижать риск при изменении макроэкономической обстановки. В отличие от LLM, которые вызываются через внешний API и обновляются поставщиком, предиктивные модели часто развёрнуты локально, и ответственность за обнаружение дрейфа и переобучение лежит целиком на организации.

Объяснимость в предиктивных моделях имеет прямое практическое и регуляторное значение. Когда модель отклоняет кредитную заявку, заёмщик имеет право знать причину. Когда система выявления мошенничества блокирует транзакцию, клиент вправе оспорить решение. Регуляторы требуют, чтобы организация могла объяснить, какие факторы повлияли на автоматизированное решение. Для линейных моделей и деревьев решений объяснимость достижима. Для сложных ансамблей и глубоких нейронных сетей она требует специальных методов интерпретации, которые дают приближённое, а не точное объяснение. Организация, выбирающая архитектуру модели, должна учитывать требования к объяснимости как архитектурное ограничение, а не как задачу, которую можно решить после развёртывания.

Для организации практический вывод из этого раздела состоит в том, что зрелость технологии не означает зрелость защиты. Предиктивные модели и классификаторы работают в организациях годами, но многие из них не проходили оценку устойчивости к враждебным воздействиям, не проверялись на справедливость по отношению к разным группам пользователей и не оснащены мониторингом дрейфа. Именно потому, что эти модели принимают решения с конкретными последствиями для людей и бизнеса, они требуют такого же внимания к безопасности, как и генеративные системы, привлекающие основное внимание.

2.7. Специфика безопасности каждого типа систем

На протяжении этой главы мы рассмотрели шесть типов ИИ-систем, и у каждого обнаружился собственный набор уязвимостей. Языковые модели атакуют через текст. Модели компьютерного зрения обманывают через пиксельные возмущения. Голосовые системы поддельвают через синтетическую речь. Видеомодели эксплуатируют через дипфейки. Мультимедальные системы открывают сразу несколько каналов атаки. Предиктивные модели уязвимы к отравлению данных и уклоняющимся воздействиям. Организация, которая подходит к защите всех этих систем одинаково, неизбежно оставляет пробелы.

Но из этого не следует, что для каждого типа нужно строить защиту с нуля. Угрозы различаются по механизму, но вопросы, которые организация должна задать перед развёртыванием любой ИИ-системы, остаются общими. Какие данные обрабатывает система и насколько они чувствительны? Какие решения принимает модель и каковы последствия ошибки? Кто может подать данные на вход и через какие каналы? Какие действия способна выполнить система в информационной среде организации? Как обнаружить, что модель ведёт себя аномально? Как откатить решение, если оно оказалось ошибочным?

Различия начинаются на уровне конкретных ответов. Для LLM-приложения основной канал атаки — текстовый ввод, и фильтрация входных промптов является первой линией защиты. Для системы компьютерного зрения текстовая фильтрация бесполезна: угроза приходит через модифицированные пиксели, адверсариальные патчи или подмену видеопотока. Для голосовой системы ключевой риск — клонирование голоса и обход биометрии, и защита строится на спектральном анализе и многофакторной верификации. Для предиктивной модели кредитного скоринга критичен дрейф данных и справедливость по отношению к демографи-

ческим группам, а не инъекции в промпт. Универсальный набор контролей, применённый без учёта типа системы, создаёт иллюзию безопасности.

	LLM	Компьютерное зрение	Аудио/речь	Видео	Мультимодальные	Предиктивная аналитика
Инъекция в промпт	Высокая	Низкая / неприменимо	Низкая / неприменимо	Низкая / неприменимо	Высокая	Низкая / неприменимо
Адверсарialные воздействия	Низкая / неприменимо	Высокая	Средняя	Средняя	Средняя	Средняя
Отравление данных	Средняя	Высокая	Средняя	Средняя	Высокая	Высокая
Дипфейки / синтетический контент	Низкая / неприменимо	Низкая / неприменимо	Высокая	Высокая	Низкая / неприменимо	Низкая / неприменимо
Извлечение модели	Высокая	Средняя	Средняя	Средняя	Высокая	Средняя
Утечка данных через выходы	Высокая	Низкая / неприменимо	Средняя	Средняя	Высокая	Средняя
Дискриминация / смещение	Средняя	Средняя	Низкая / неприменимо	Низкая / неприменимо	Средняя	Высокая
Дрейф модели	Средняя	Средняя	Средняя	Средняя	Средняя	Высокая

■ Высокая
 ■ Средняя
 ■ Низкая / неприменимо

Рисунок 7. Сравнение профилей угроз различных типов ИИ-систем.

Различается и характер ущерба. Ошибка генеративной модели может привести к репутационному инциденту: оскорбительный ответ чат-бота, выдуманная юридическая ссылка, раскрытие конфиденциального фрагмента. Ошибка модели классификации мошенничества влечёт прямые финансовые потери. Ошибка системы распознавания лиц на пункте пропуска создаёт физическую угрозу. Ошибка системы компьютерного зрения в автономном транспорте может стоить человеческой жизни. Уровень допустимого риска и строгость контролей должны соответствовать характеру ущерба, а он напрямую зависит от типа системы и её роли в бизнес-процессе.

Регуляторные требования тоже различаются по типам систем. EU AI Act относит биометрическую идентификацию в общественных пространствах и системы социального скоринга к запрещённым или высокорисковым категориям. Системы кредитного скоринга и отбора персонала требуют объяснимости и отсутствия дискриминации. Генеративные модели общего назначения подпадают под требования прозрачности и маркировки синтетического контента. Организация, применяющая ИИ в нескольких областях, может обнаружить, что одна из её систем регулируется строго, другая — мягко, а третья пока вообще не попадает в поле зрения регулятора. Карта применимости регуляторных требований, которую мы рассмотрим в восьмой главе, должна строиться с учётом типа каждой системы.

Тестирование безопасности для разных типов ИИ-систем требует разных компетенций и инструментов. AI красная команда для LLM-приложения предполагает навыки работы с промптами, понимание механизмов инъекций и умение провоцировать нежелательное поведение модели через текстовые манипуляции. Тестирование устойчивости модели компьютерного зрения требует генерации адверсарialных изображений, проверки реакции на патчи и оценки поведения при разных условиях освещения и ракурса. Тестирование голосовой биометрии требует образцов синтетической речи и инструментов для оценки устойчивости к клонированным голосам. Тестирование предиктивных моделей включает проверку справедливости,

анализ устойчивости к отравлению данных и оценку поведения при дрейфе. Команда, которая умеет тестировать только один тип систем, не сможет покрыть весь портфель ИИ организации.

Мониторинг в продуктивной среде тоже зависит от типа системы. Для LLM мониторинг включает анализ входных промптов на наличие инъекций, проверку выходных ответов на утечки данных и конфабуляции, отслеживание вызовов инструментов и действий агентов. Для системы компьютерного зрения мониторинг фокусируется на стабильности классификации, выявлении аномальных входных данных и проверке целостности видеопотока. Для предиктивных моделей мониторинг отслеживает дрейф распределения входных данных и смещение метрик качества. Единая система мониторинга, не учитывающая специфику модальности, пропустит сигналы, характерные для конкретного типа системы.

При всём разнообразии угроз существуют принципы, которые остаются применимыми независимо от типа ИИ-системы. Минимальные полномочия: модель, агент или приложение получают только тот доступ, который необходим для выполнения задачи. Эшелонированная защита: контроли на входе, внутри конвейера обработки и на выходе. Изоляция сред: разделение обучения, тестирования и продуктивной эксплуатации. Контроль происхождения: проверка источника модели, данных и зависимостей. Журналирование и наблюдаемость: фиксация входных данных, решений модели и выходных результатов в объёме, достаточном для расследования инцидента. Человеческий контроль: определение условий, при которых решение модели требует подтверждения человеком. Эти принципы составляют каркас, внутри которого организация адаптирует конкретные контроли к типу системы.

Практический вывод для организации состоит в том, что оценка рисков и выбор мер защиты начинаются с вопроса: какой тип ИИ-системы мы защищаем? Ответ на этот вопрос определяет, какие угрозы приоритетны, какие контроли применимы, какие компетенции нужны команде тестирования и какие метрики использовать для мониторинга. Реестр ИИ-систем, который мы будем строить в шестой главе, должен фиксировать тип каждой системы как одну из первых характеристик. Без этой информации оценка рисков останется абстрактной, а меры защиты — шаблонными.

Источники и дальнейшее чтение к главе 2

- Vaswani A. и др. Attention Is All You Need. NeurIPS, 2017 — основополагающая статья по архитектуре трансформера.
- Goodfellow I., Shlens J., Szegedy C. Explaining and Harnessing Adversarial Examples, 2014 — классический пример с пандой и гиббоном.
- Brown T. и др. Adversarial Patch, 2017 — адверсариальные патчи и наклейки.
- Eykholt K. и др. Robust Physical-World Attacks on Deep Learning Visual Classification, 2018 — атака на распознавание дорожных знаков.
- NIST IR 8280. Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects. National Institute of Standards and Technology.
- Buolamwini J., Gebru T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification, 2018.
- Coalition for Content Provenance and Authenticity (C2PA) — стандарт маркировки происхождения контента, c2pa.org.
- Regulation (EU) 2024/1689 (EU AI Act) — категории запрещённых и высокорисковых систем (биометрическая идентификация, кредитный скоринг, отбор персонала).
- Stupp C. Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case. The Wall Street Journal, 30 августа 2019 — голосовой дипфейк, €220 000.
- Magramo K. Finance worker pays out \$25 million after video call with deepfake «chief financial officer». CNN Business, 4 февраля 2024 — кейс Arup, Гонконг.

- Carlini N., Wagner D. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, 2018.
- Zhang G. и др. DolphinAttack: Inaudible Voice Commands, 2017 — скрытые голосовые команды.
- Thys S., Van Ranst W., Goedemé T. Fooling Automated Surveillance Cameras: Adversarial Patches to Attack Person Detection, 2019.
- Tramèr F. и др. Stealing Machine Learning Models via Prediction APIs. USENIX Security, 2016 — извлечение модели через API

Глава 3. Жизненный цикл и границы ответственности

В первых двух главах мы определили, из чего состоит ИИ-система и какие типы систем встречаются в организациях. Но ИИ-система — это не статичная конструкция. Она проходит через последовательность стадий: от первоначального замысла через сбор данных, обучение, тестирование и развёртывание до повседневной эксплуатации, обновления и, в конечном счёте, вывода из эксплуатации. На каждой стадии возникают свои риски, участвуют свои люди и применяются свои контроли. Организация, которая выстраивает безопасность только на этапе развёртывания, упускает угрозы, возникающие задолго до него, и проблемы, которые проявляются значительно позже.

Вопрос ответственности усложняет картину. Кто отвечает за безопасность модели, которую обучил один поставщик, дообучил второй, развернул в инфраструктуре третьего и предоставил пользователям четвёртый? Когда модель выдаёт ошибочный результат, причинивший ущерб, к кому обращается пострадавший? Когда регулятор требует объяснить решение системы, кто предоставляет это объяснение? Ответы на эти вопросы зависят от того, как организация определяет границы своей ответственности и как она выстраивает отношения с другими участниками жизненного цикла.

Эта глава разбирает стадии жизненного цикла ИИ-системы, определяет роли участников на каждой из них, описывает модели распределения ответственности и показывает, где возникают зоны, за которые никто не отвечает. После прочтения читатель сможет определить, на какой стадии жизненного цикла находятся ИИ-системы в его организации, кто несёт ответственность за каждый аспект безопасности и какие договорённости необходимо зафиксировать с внешними поставщиками и партнёрами.

3.1. Стадии жизненного цикла ИИ-системы

Жизненный цикл ИИ-системы описывается в нескольких международных документах, и хотя терминология различается, логика остаётся общей. NIST AI RMF выделяет фазы планирования и проектирования, сбора и обработки данных, построения модели, развёртывания и эксплуатации с мониторингом. ISO/IEC 42001 определяет стадии от проектирования и разработки через верификацию и развёртывание до эксплуатации и вывода из эксплуатации. Совет Европы в рекомендациях по защите данных в LLM-системах описывает пять этапов: предварительное обучение базовой модели, адаптация и дообучение, системная интеграция, развёртывание в продуктивной среде и взаимодействие с конечными пользователями. Организации не обязаны принимать одну из этих моделей буквально, но должны понимать общую логику и адаптировать её к собственному контексту.

Для целей безопасности полезно рассматривать шесть стадий, которые охватывают весь путь ИИ-системы от замысла до завершения работы.

Первая стадия — проектирование и планирование. На этом этапе организация формулирует задачу, определяет, зачем нужна ИИ-система и какие результаты она должна давать. Здесь принимаются решения, которые определяют профиль риска системы на весь срок её существования: какой тип модели использовать, какие данные собирать, какой уровень автономии предоставить системе, какие ограничения наложить. Именно на этой стадии должны проводиться первичная оценка рисков и оценка воздействия, моделирование угроз и определение требований безопасности.



Рисунок 8. Жизненный цикл ИИ-системы: от формирования замысла и подготовки данных до эксплуатации, мониторинга и вывода из использования.

На практике многие организации пропускают этот этап и переходят сразу к экспериментам с моделью, откладывая вопросы безопасности на потом. Цена такого решения выясняется позже, когда пересмотр архитектуры требует значительных ресурсов.

Вторая стадия — подготовка данных. Организация собирает, очищает, размечает и структурирует данные для обучения модели. На этой стадии закладываются риски, которые проявятся на всех последующих: смещения в данных, присутствие чувствительной информации, нарушение авторских прав, недостаточная репрезентативность выборки. Если обучающие данные содержат персональные данные, на этом этапе необходимо определить правовое основание для их обработки, провести обезличивание или применить иные меры защиты. Качество и безопасность данных определяют качество и безопасность модели, и ошибки, допущенные здесь, невозможно компенсировать на последующих стадиях без возврата к данным.

Третья стадия — разработка и обучение модели. Модель обучается на подготовленных данных, проходит через циклы экспериментов, настройки гиперпараметров и оценки качества. Для организаций, использующих предобученные модели от внешних поставщиков, эта стадия может включать тонкую настройку (fine-tuning) или адаптацию через промпты, а не обучение с нуля. С точки зрения безопасности на этой стадии возникают угрозы компрометации обучающего конвейера, отравления данных, внедрения закладок в модель и утечки обучающих данных через параметры модели. Среда разработки, в которой хранятся данные, промежуточные версии моделей и ключи доступа, часто защищена слабее продуктивных систем, хотя содержит наиболее ценные активы.

Четвёртая стадия — тестирование и валидация. Модель проверяется на соответствие требованиям качества, безопасности, справедливости и устойчивости к враждебным воздействиям. На этой стадии проводится состязательное тестирование ИИ (AI red teaming), тестирование на наличие смещений, проверка устойчивости к адверсариальным атакам и инъекциям, оценка объяснимости результатов. Формулируются приёмоные критерии и принимается

решение о готовности системы к развёртыванию. На практике тестирование безопасности ИИ-систем нередко ограничивается проверкой функциональности и точности модели. Проверка устойчивости к атакам, справедливости и поведения в граничных условиях остаётся за пределами стандартных процедур, хотя именно эти аспекты определяют, насколько безопасна система в продуктивной среде.

Пятая стадия — развёртывание и эксплуатация. Система вводится в продуктивную среду и начинает обрабатывать запросы пользователей. На этой стадии безопасность переходит из режима проектирования в режим непрерывного контроля. Журналирование, мониторинг поведения, отслеживание дрейфа, управление инцидентами, обновление модели, управление изменениями — всё это процессы, которые должны работать постоянно, а не разово. Обновление модели поставщиком, изменение контекстных данных в RAG-системе, расширение полномочий агента — каждое из этих событий может изменить профиль риска и требует повторной оценки. Организации, которые воспринимают развёртывание как завершение проекта, обнаруживают, что самые сложные вопросы безопасности возникают именно после него.

Шестая стадия — вывод из эксплуатации. Когда ИИ-система перестаёт использоваться, необходимо позаботиться о данных, моделях и артефактах, которые остаются после неё. Обучающие данные могут содержать персональную информацию, требующую удаления. Веса модели могут содержать извлекаемые фрагменты обучающих данных. Журналы запросов и ответов фиксируют чувствительное взаимодействие пользователей с системой. Ключи и токены доступа должны быть отозваны. Процессы, зависящие от системы, должны быть переведены на альтернативные механизмы. Эта стадия часто остаётся без внимания, хотя именно на ней возникают риски утечки данных и несанкционированного доступа к устаревшим, но всё ещё работающим компонентам.

Важная особенность жизненного цикла ИИ-системы — его итеративность. В отличие от традиционного программного обеспечения, где жизненный цикл во многих случаях движется линейно от разработки к эксплуатации, ИИ-система регулярно возвращается к предыдущим стадиям. Обнаружение дрейфа в продуктивной среде требует переобучения модели, то есть возврата к стадиям подготовки данных и разработки. Изменение регуляторных требований может потребовать пересмотра проектных решений. Обновление базовой модели поставщиком фактически перезапускает стадии тестирования и валидации. Каждый такой цикл несёт собственные риски и требует повторной оценки безопасности.

Для организации это означает, что безопасность ИИ-системы не может быть выстроена один раз. Она должна сопровождать систему на каждой стадии и при каждом переходе между ними. Контроли, применённые при первоначальном развёртывании, могут оказаться недостаточными после обновления модели. Оценка рисков, проведённая при проектировании, теряет актуальность при изменении данных или контекста использования. Процессы безопасности должны быть встроены в жизненный цикл как постоянный элемент, а не как разовая проверка перед запуском. Именно эта встроенность отличает зрелую программу безопасности ИИ от набора отдельных мероприятий.

3.2. Участники и их роли на каждой стадии

ИИ-система проходит через руки множества людей, и каждый из них влияет на её безопасность. Проблема в том, что эти люди часто принадлежат к разным подразделениям, говорят на разных профессиональных языках и несут ответственность за разные аспекты результата. Исследователь данных отвечает за качество модели, но не за безопасность инфраструктуры. Инженер по развёртыванию отвечает за среду выполнения, но не за справедливость модели по отношению к разным группам пользователей. Владелец продукта отвечает за биз-

нес-результат, но не за защиту обучающих данных. Когда каждый участник видит только свою зону, между зонами остаются пробелы, в которых возникают инциденты.

NIST AI RMF определяет несколько категорий участников жизненного цикла ИИ: проектировщики, разработчики, развёртывающие и операторы. ISO/IEC 42001 расширяет этот перечень, включая роли, связанные с человеческим контролем, экспертизой доверия (безопасность, приватность, справедливость), управлением и заинтересованными сторонами. На практике в каждой организации эти роли распределяются по-своему, и один человек может совмещать несколько ролей или, наоборот, одна роль может быть разделена между командами. Важна не конкретная организационная структура, а полнота покрытия: есть ли кто-то, кто отвечает за каждый аспект безопасности на каждой стадии.

На стадии проектирования и планирования ключевые решения принимают владелец продукта, архитектор системы и руководитель, санкционирующий проект. Владелец продукта определяет, какую задачу должна решать ИИ-система, и формулирует требования к её поведению. Архитектор выбирает тип модели, способ развёртывания и схему интеграции с корпоративными системами. Руководитель принимает решение о запуске, выделяет ресурсы и определяет допустимый уровень риска. Именно на этом этапе должны быть вовлечены специалист по информационной безопасности, юрист и специалист по защите данных. Если их привлекают позже, проектные решения уже приняты, и вносить изменения значительно дороже.

Стоит задержаться на роли специалиста по безопасности на этапе проектирования, потому что именно здесь закладывается фундамент. Если архитектор решил использовать внешнюю модель через API, специалист по безопасности должен оценить риски передачи данных поставщику. Если предусмотрена интеграция с корпоративными системами через агентов, нужно определить модель полномочий до того, как агент получит доступ к продуктивной среде. Если система будет принимать решения, затрагивающие права людей, юрист должен оценить регуляторные требования до выбора архитектуры, а не после развёртывания. Организации, в которых безопасность появляется в проекте на стадии предпродуктивного тестирования, тратят значительно больше ресурсов на устранение проблем, чем те, которые включают её в проектирование с самого начала.

На стадии подготовки данных основную работу выполняют инженеры по данным и исследователи данных. Они собирают, очищают, размечают и структурируют обучающие данные. Их решения определяют, какие закономерности модель усвоит, а какие проигнорирует. Если выборка не представляет все группы пользователей, модель унаследует смещение. Если в данных присутствуют персональные сведения, необходимо провести обезличивание или обосновать правовую основу обработки. Роль специалиста по защите данных на этом этапе особенно важна: он должен оценить состав данных, проверить наличие правового основания для их использования и определить меры минимизации. Роль специалиста по безопасности — убедиться, что среда, в которой хранятся и обрабатываются обучающие данные, защищена от несанкционированного доступа и что целостность данных контролируется.

На стадии разработки и обучения центральную роль играют исследователи машинного обучения и инженеры MLOps. Они обучают модель, экспериментируют с архитектурами, настраивают гиперпараметры и отслеживают метрики качества. Безопасность на этом этапе требует контроля среды обучения: кто имеет доступ к вычислительным ресурсам, моделям и данным, как защищены промежуточные артефакты, как контролируется целостность обучающего конвейера. В организациях, которые дообучают внешние модели, на этой стадии возникает дополнительная роль — специалист по оценке поставщиков, который проверяет происхождение базовой модели, условия лицензии и безопасность среды, из которой она загружена.

На стадии тестирования и валидации состав участников расширяется. К разработчикам присоединяются тестировщики, специалисты по состязательному тестированию ИИ (AI red teaming), эксперты по справедливости и объяснимости, аудиторы. Тестирование безопасности

ИИ-системы требуют компетенций, которых нет в традиционной команде QA. Проверка устойчивости к инъекциям в промпт, оценка поведения при адверсариальных воздействиях, анализ справедливости модели по отношению к разным демографическим группам — всё это специализированные задачи. Если в организации нет собственной экспертизы, на этом этапе привлекаются внешние специалисты. Решение о готовности системы к развёртыванию принимает владелец продукта совместно с руководителем по информационной безопасности (или CISO) и, при необходимости, юристом. Это решение должно быть документировано и основано на результатах тестирования, оценке рисков и анализе соответствия требованиям.

На стадии развёртывания и эксплуатации система переходит в руки операторов. Инженеры по развёртыванию интегрируют систему в продуктивную среду. Администраторы настраивают мониторинг, журналирование и контроль доступа. Служба поддержки обрабатывает обращения пользователей. Команда информационной безопасности отслеживает аномалии, реагирует на инциденты и проводит регулярные проверки. Конечные пользователи взаимодействуют с системой и формируют основной поток входных данных. На этой стадии важна роль человеческого контроля: кто проверяет решения модели, в каких случаях, с какими полномочиями и с какой частотой. Формальное назначение контролёра без обеспечения условий для его работы превращает контроль в фикцию.

На стадии вывода из эксплуатации участвуют администраторы систем, специалист по защите данных и архивист. Обучающие данные и журналы должны быть обработаны в соответствии с требованиями хранения и удаления. Модели и артефакты должны быть изъяты из доступа. Ключи и токены отозваны. Процессы, зависящие от системы, переведены на альтернативные механизмы или ручной режим. Отсутствие формализованной процедуры вывода из эксплуатации приводит к ситуации, когда устаревшая модель продолжает работать в продуктивной среде без мониторинга и поддержки, создавая риски, о которых никто не вспоминает до инцидента.

Сквозная роль, которая проходит через все стадии и заслуживает отдельного упоминания, — это роль владельца рисков ИИ-системы. Этот человек (или роль) отвечает за то, чтобы риски системы были оценены, приняты на осознанном уровне и управлялись на протяжении всего жизненного цикла. В разных организациях эта роль может принадлежать владельцу продукта, CISO, руководителю подразделения или специально назначенному ответственному. Важно одно: если эта роль не определена, ответственность за риски размывается между участниками, и каждый из них считает, что за безопасность отвечает кто-то другой.

3.3. Распределение ответственности: поставщик, разработчик, оператор, пользователь

В традиционной информационной безопасности распределение ответственности между поставщиком и потребителем хорошо изучено. Модель совместной ответственности для облачных сервисов, предложенная крупными облачными провайдерами более десяти лет назад, чётко разграничивает, кто отвечает за инфраструктуру, а кто за данные и приложения. В ИИ-системах эта модель усложняется, потому что цепочка поставок длиннее, участников больше, а границы между их зонами ответственности менее очевидны.

Рассмотрим типичный сценарий. Организация использует LLM-приложение для обработки клиентских обращений. Базовую модель обучил поставщик, который предоставляет доступ к ней через API. Другая организация разработала платформу оркестрации, через которую модель подключается к корпоративным системам. Третья организация интегрировала эту платформу в клиентский портал. Всё это работает на облачной инфраструктуре ещё одного поставщика. Когда модель раскрывает в ответе персональные данные клиента, кто несёт ответственность? Поставщик модели, который обучил её на данных, содержащих конфиденциаль-

ную информацию? Разработчик платформы, который не ограничил доступ модели к чувствительным данным? Интегратор, который не настроил фильтрацию выходных данных? Облачный провайдер, через инфраструктуру которого прошли эти данные? Организация-заказчик, которая не провела оценку рисков перед развёртыванием?

CSA AI Controls Matrix (AICM) предлагает развёрнутую модель совместной ответственности для ИИ-систем, выделяя пять типов участников цепочки поставок. Поставщик облачной инфраструктуры (CSP) отвечает за физическую безопасность, вычислительные ресурсы, сетевую инфраструктуру и виртуализацию. Поставщик модели (MP) отвечает за обучение, валидацию, безопасность обучающих данных и целостность модели. Поставщик оркестрированных сервисов (OSP) отвечает за API, управление промптами, оркестрацию моделей и интеграцию компонентов. Поставщик приложения (AP) отвечает за пользовательский интерфейс, контроль входных и выходных данных, ограничение поведения модели на уровне приложения. Заказчик (AIC) отвечает за политики допустимого использования, обучение пользователей, оценку рисков и соответствие требованиям в своей юрисдикции.

Эта пятиуровневая модель полезна как отправная точка, но на практике границы размываются. Поставщик модели может одновременно быть поставщиком инфраструктуры и платформы оркестрации, если организация использует интегрированный сервис крупного облачного провайдера. В этом случае три уровня ответственности сжимаются в один контракт, и заказчику сложнее понять, какие именно контроли реализованы на каждом уровне. Напротив, организация, собирающая систему из компонентов разных поставщиков, получает более прозрачное, но более сложное в управлении распределение.

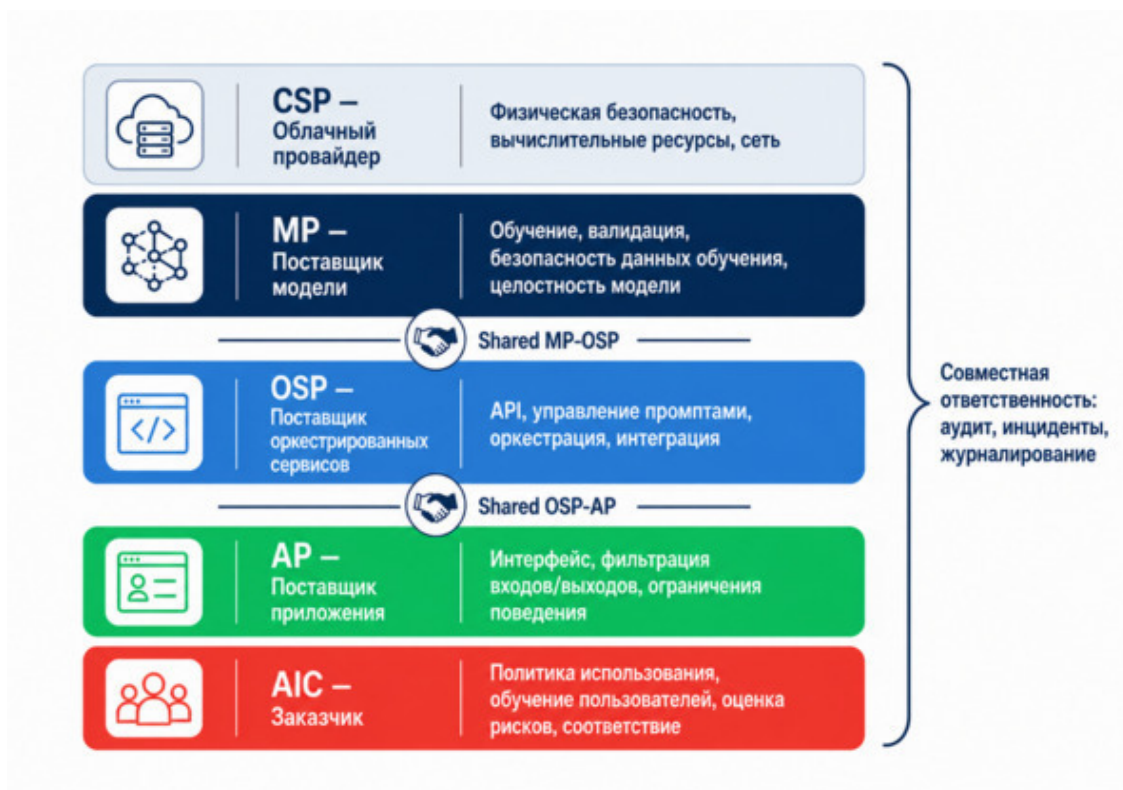


Рисунок 9. Модель совместной ответственности в цепочке предоставления ИИ-сервисов.

EU AI Act подходит к вопросу ответственности через концепцию ролей: поставщик (provider) и развёртывающий (deployer). Поставщик — тот, кто разрабатывает или заказывает разработку ИИ-системы и выводит её на рынок. Развёртывающий — тот, кто использует ИИ-систему в своей деятельности. Для систем высокого риска на поставщика ложатся обязатель-

ства по управлению рисками, обеспечению качества данных, документированию, прозрачности и пост-рыночному мониторингу. На развёртывающего — обязательства по использованию системы в соответствии с инструкциями, обеспечению человеческого контроля и информированию поставщика о проблемах. Это упрощённая модель по сравнению с АИСМ, но она имеет юридическую силу и определяет конкретные обязанности с конкретными санкциями за нарушение.

На практике организация-заказчик часто оказывается в позиции, где ей нужно управлять рисками, которые порождены решениями, принятыми другими участниками цепочки. Поставщик модели выбрал обучающие данные — заказчик живёт с последствиями этого выбора: смещениями, запомненными фрагментами, правовыми рисками. Поставщик платформы определил архитектуру оркестрации — заказчик наследует её ограничения в области безопасности. Облачный провайдер обновил инфраструктуру — заказчик может не знать, повлияло ли это на поведение модели. При этом поставщики раскрывают свои практики безопасности в ограниченном объёме, и заказчику приходится полагаться на контрактные обязательства, сертификации и самооценки, а не на собственную верификацию.

Ситуация становится ещё сложнее, когда поставщик модели обновляет её без предупреждения. Многие провайдеры LLM через API оставляют за собой право обновлять модель, и условия сервиса часто допускают такие обновления без уведомления заказчика. Модель, которую организация протестировала и одобрила для развёртывания, может измениться, и поведение, которое было проверено, может стать другим. Кто несёт ответственность за последствия? Формально — поставщик, если обновление привело к нарушению заявленных характеристик. На практике — заказчик, потому что именно его клиенты, сотрудники и бизнес-процессы затронуты последствиями.

Отдельного внимания заслуживает ответственность конечного пользователя. В большинстве ИИ-систем пользователь формирует входные данные: текст запроса, загружаемый документ, голосовую команду. Его действия могут быть легитимными, ошибочными или злонамеренными. Пользователь, который вводит в чат-бот персональные данные клиента вопреки политике организации, создаёт риск утечки. Пользователь, который доверяет ответу модели без проверки и принимает на его основе решение с юридическими последствиями, несёт часть ответственности за результат. Организация должна определить, какие действия пользователя допустимы, обучить его работе с ИИ-системой и предусмотреть контроли, ограничивающие передачу чувствительных данных. Перекладывание всей ответственности на пользователя, которого не обучили и не обеспечили инструментами контроля, — управленческий просчёт, а не решение проблемы.

В открытых экосистемах, где модели доступны для свободного скачивания, распределение ответственности приобретает дополнительные нюансы. Организация, загрузившая модель из открытого репозитория, принимает на себя ответственность за всё, что эта модель делает в её среде. Автор модели, опубликовавший её под открытой лицензией, как правило, не несёт юридической ответственности за последствия использования, хотя этот вопрос остаётся предметом правовых дискуссий. Организация не может апеллировать к автору модели, если модель оказалась скомпрометированной, содержит вредоносные закладки или демонстрирует нежелательное поведение. Вся ответственность за проверку, тестирование и безопасную эксплуатацию ложится на того, кто принял решение использовать эту модель.

Практический вывод из этого раздела состоит в следующем. Организация не может полностью делегировать безопасность ИИ-системы поставщикам. Даже при использовании модели через API крупного провайдера организация остаётся ответственной за выбор поставщика, оценку его практик безопасности, определение допустимых сценариев использования, контроль входных и выходных данных, обучение пользователей, мониторинг поведения и реагирование на инциденты. Матрица распределения ответственности, зафиксированная докумен-

тально и согласованная с каждым поставщиком, является необходимым инструментом. Без неё пробелы неизбежны, и они обнаруживаются в самый неподходящий момент — при расследовании инцидента, запросе регулятора или судебном разбирательстве.

3.4. Границы ответственности организации и модель совместной ответственности

В предыдущем разделе мы рассмотрели, как ответственность распределяется между участниками цепочки поставок. Теперь нужно ответить на вопрос, который стоит перед каждой конкретной организацией: где проходят границы моей ответственности и как мне зафиксировать их так, чтобы не оставалось зон, за которые никто не отвечает?

Ответ зависит от того, какую роль организация занимает в цепочке. Организация, которая обучает модель с нуля, несёт ответственность за весь жизненный цикл: от выбора данных до вывода из эксплуатации. Организация, которая вызывает внешнюю модель через API и встраивает её в своё приложение, отвечает за то, как она использует результаты модели, но не за внутреннее устройство самой модели. Организация, которая предоставляет свой продукт конечным пользователям с ИИ-функциями от стороннего поставщика, может нести ответственность перед своими клиентами за качество и безопасность этих функций, даже если сама их не разрабатывала. Границы ответственности определяются не только технической архитектурой, но и контрактами, регуляторными требованиями и ожиданиями заинтересованных сторон.

Модель совместной ответственности, предложенная CSA AICM, разделяет контроли на три категории по принципу владения. Контроли, полностью принадлежащие одному участнику: физическая безопасность центра обработки данных принадлежит облачному провайдеру, валидация базовой модели — поставщику модели, обучение конечных пользователей — организации-заказчику. Контроли, разделённые между двумя конкретными участниками: например, провайдер инфраструктуры предоставляет безопасную вычислительную среду, а поставщик модели конфигурирует защиту обучающего конвейера на этой инфраструктуре. И контроли, распределённые по всей цепочке: аудит, управление инцидентами, ведение журналов — каждый участник реализует их на своём уровне, и полнота покрытия зависит от координации между всеми.

Именно третья категория создаёт наибольшие сложности. Когда ответственность разделена между пятью участниками, каждый из которых ведёт журналы на своём уровне, расследование инцидента требует сбора данных из всех пяти источников. Если один из участников не предоставляет достаточной детализации или не хранит журналы необходимый срок, цепочка прослеживания прерывается. На практике организация-заказчик должна ещё до начала работы с поставщиком определить, какие данные ей потребуются для расследования инцидента и аудита, и зафиксировать эти требования в договоре.

Распространённая ошибка — считать, что модель совместной ответственности работает симметрично. В облачных сервисах она приблизительно симметрична: провайдер публикует свою часть ответственности, заказчик принимает свою, граница проходит на определённом уровне стека. В ИИ-системах симметрия нарушается. Поставщик модели обладает информацией, недоступной заказчику: на каких данных обучена модель, какие ограничения заложены в её поведение, какие уязвимости обнаружены и исправлены, когда планируется обновление. Заказчик работает с моделью как с чёрным ящиком и вынужден принимать решения о рисках на основе неполной информации. Эта асимметрия делает позицию заказчика уязвимой: он несёт ответственность перед своими пользователями и регуляторами, но не может полностью проверить то, за что отвечает.

Как организации выстроить границы ответственности на практике? Первый шаг — определить свою роль в цепочке для каждой ИИ-системы. Организация может быть заказчиком для

одной системы, поставщиком приложения для другой и одновременно оператором для третьей. Каждая роль подразумевает свой набор обязательств. Второй шаг — для каждого внешнего участника зафиксировать, кто за что отвечает. Это управленческая необходимость. Документ может принимать форму матрицы ответственности, приложения к договору или отдельного соглашения об уровне безопасности.

При составлении такой матрицы организации стоит задать поставщику ряд конкретных вопросов. Кто контролирует обучающие данные и какие меры защиты к ним применяются? Как поставщик уведомляет заказчика об обновлении модели и какие гарантии даёт относительно совместимости поведения? Хранятся ли запросы и ответы на стороне поставщика, используются ли они для дообучения, и если да, как заказчик может отказаться? Какие журналы предоставляет поставщик и в каком формате? Какова процедура реагирования на инциденты, и как поставщик взаимодействует с заказчиком при расследовании? Какие сертификации или результаты аудита подтверждают практики безопасности поставщика? Каков порядок действий при обнаружении уязвимости в модели или платформе? Отсутствие ясных ответов на эти вопросы — сигнал того, что распределение ответственности не зафиксировано, и в момент инцидента каждая сторона будет трактовать его в свою пользу.

Особого внимания требуют сценарии, в которых организация встраивает ИИ-возможности в продукт, предоставляемый собственным клиентам. Финтех-организация, которая использует внешнюю модель для оценки кредитоспособности и предлагает этот сервис банкам, оказывается в промежуточной позиции. Для поставщика модели она — заказчик. Для банков — поставщик. Ответственность перед банками и их клиентами не может быть переложена на поставщика базовой модели. Если модель принимает дискриминационное решение, пострадавший заёмщик предъявит претензии банку, банк — финтех-организации, и только она обратится к поставщику модели. Каждое звено цепочки несёт ответственность перед следующим, и эта ответственность должна быть подкреплена контрактными обязательствами, техническими контролями и страхованием.

EU AI Act вводит дополнительный механизм, влияющий на распределение ответственности. Если организация, использующая ИИ-систему в качестве развёртывающего, вносит существенные изменения в её поведение — дообучает модель, меняет назначение системы, интегрирует её в контекст, не предусмотренный поставщиком — она может быть признана поставщиком со всеми вытекающими обязательствами. Это означает, что граница между ролями поставщика и развёртывающего подвижна и зависит от того, какие действия организация совершает с системой. Организация, которая дообучила базовую модель на собственных данных и развернула её для принятия решений в области HR, не может ссылаться на исходного поставщика как на ответственного за поведение модели. Ответственность переходит к ней.

В многоагентных системах распределение ответственности приобретает дополнительную сложность. Когда агент одной организации делегирует задачу агенту другой через протокол A2A, результат действий внешнего агента влияет на процессы первой организации. Если внешний агент вернул некорректные данные и на их основе было принято ошибочное решение, кто несёт ответственность? Организация, чей агент инициировал делегирование? Организация, чей агент выполнил задачу некорректно? Поставщик протокола? Ответы на эти вопросы пока не зафиксированы ни в одном нормативном акте, и организации, внедряющие многоагентные системы, вынуждены выстраивать договорённости самостоятельно, опираясь на общие принципы контрактного права и управления рисками.

Для организации практический вывод состоит в том, что модель совместной ответственности — это не теоретическая конструкция, а рабочий инструмент, который необходимо адаптировать к каждой ИИ-системе и каждому поставщику. Она должна быть зафиксирована документально, согласована всеми участниками и пересматриваться при изменении архитектуры, смене поставщика или обновлении модели. Организация, которая полагается на общие усло-

вия сервиса и не задаёт поставщику конкретных вопросов о распределении ответственности, обнаруживает пробелы в самый неудобный момент. А пробелы в ответственности — это пробелы в безопасности.

Источники и дальнейшее чтение к главе 3

— NIST AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology, январь 2023 — стадии жизненного цикла и категории участников (designers, developers, deployers, operators).

— ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system — стадии от проектирования до вывода из эксплуатации, роли и заинтересованные стороны.

— Council of Europe, Consultative Committee of Convention 108. Draft Guidelines on Privacy and Data Protection in the Context of Large Language Model (LLM) -Based Systems. T-PD (2025) 3, Страсбург, 2 сентября 2025 — пять этапов жизненного цикла LLM (предобучение, дообучение, системная интеграция, развёртывание, взаимодействие с пользователем).

— Cloud Security Alliance. AI Controls Matrix (AICM) — пятиуровневая модель совместной ответственности (CSP, поставщик модели, поставщик оркестрации, поставщик приложения, заказчик).

— AWS Shared Responsibility Model (или аналогичный документ основного облачного провайдера, которым организация пользуется) — как первоисточник модели совместной ответственности для облачных сервисов.

— Regulation (EU) 2024/1689 (EU AI Act) — роли provider/deployer, обязательства для систем высокого риска, механизм признания развёртывающего поставщиком при существенном изменении системы.

Глава 4. Ландшафт рисков и свойства доверия

Три предыдущие главы описали, из чего состоит ИИ-система, какие типы систем существуют, через какие стадии они проходят и кто за них отвечает. Всё это — карта территории. Теперь нужно нанести на эту карту опасности. Прежде чем обсуждать конкретные угрозы, способы защиты и программу внедрения, необходимо выстроить понятийный аппарат: что мы понимаем под риском применительно к ИИ, как соотносятся угроза, уязвимость и ущерб, и какие свойства определяют, заслуживает ли система доверия.

Без общей терминологии разговор о безопасности ИИ быстро превращается в диалог глухих. Специалист по ИБ говорит об уязвимостях и атаках. Разработчик модели говорит о точности и смещениях. Юрист говорит о нарушении прав и несоответствии требованиям. Руководитель говорит о репутации и финансовых потерях. Все они описывают разные грани одного и того же явления, но каждый использует собственный словарь. Эта глава формирует общий язык, который позволит всем участникам организации обсуждать риски ИИ в одной системе координат.

4.1. Понятия риска, угрозы, уязвимости и ущерба в контексте ИИ

В информационной безопасности понятия риска, угрозы, уязвимости и ущерба давно формализованы. Риск — это сочетание вероятности события и его последствий. Угроза — потенциальная причина нежелательного события. Уязвимость — слабость актива или контроля, которую может использовать угроза. Ущерб — негативное последствие реализованного события. Эти определения работают и для ИИ-систем, но их применение требует существенных уточнений, потому что природа компонентов, характер угроз и масштаб последствий в ИИ отличаются от привычных.

Начнём с того, что в традиционной ИБ угрозы, как правило, направлены против конфиденциальности, целостности или доступности информационных активов. Атакующий пытается украсть данные, изменить их или сделать систему недоступной. Для ИИ-систем эта триада сохраняет значение, но к ней добавляются измерения, которые не укладываются в классическую модель. Модель может быть полностью доступна и не скомпрометирована с точки зрения целостности кода, но при этом принимать дискриминационные решения, генерировать опасный контент или раскрывать персональные данные из обучающего набора. Ни одна из этих ситуаций не является нарушением конфиденциальности, целостности или доступности в привычном смысле, но каждая из них причиняет реальный ущерб.

NIST AI RMF расширяет понятие риска для ИИ-систем, определяя его как сочетание вероятности и масштаба негативных последствий, которые могут затронуть людей, организации и экосистемы. Негативные последствия для людей включают ущерб здоровью, безопасности, гражданским правам, экономическим возможностям, достоинству и автономии. Для организации — ущерб операциям, репутации, финансовому положению и юридической позиции. Для экосистемы — нарушение взаимосвязанных систем, рынков, инфраструктуры и окружающей среды. Этот трёхуровневый взгляд на ущерб принципиально шире, чем привычная модель информационной безопасности, и он необходим, потому что ИИ-системы воздействуют на мир за пределами информационной среды организации.



Рисунок 10. От классической триады конфиденциальности, целостности и доступности к расширенному набору свойств доверия ИИ.

Уязвимости ИИ-систем тоже устроены иначе. В традиционном ПО уязвимость — это ошибка в коде, неправильная конфигурация или слабость в протоколе. Её можно обнаружить сканером, описать в базе CVE и устранить патчем. В ИИ-системе многие уязвимости неотделимы от самого принципа работы модели. Способность языковой модели интерпретировать любой текст как инструкцию — не ошибка в коде, а следствие архитектуры. Склонность модели к конфабуляции — не баг, а свойство вероятностной генерации. Чувствительность классификатора к адверсариальным возмущениям — не дефект конкретной реализации, а математическое свойство класса моделей. Эти уязвимости невозможно устранить патчем; их можно только ограничивать архитектурными мерами, мониторингом и процессами.

Из этого следует важное практическое наблюдение: привычная модель управления уязвимостями, в которой уязвимость обнаруживается, классифицируется по критичности и устраняется обновлением, работает для инфраструктурного слоя ИИ-системы, но не для самой модели. Организация может обновить операционную систему сервера инференса и устранить известную уязвимость. Но она не может устранить способность LLM к инъекциям в промпт так же, как устраняет SQL-инъекцию в веб-приложении. Вместо этого она выстраивает слои защиты: фильтрацию входов, ограничение полномочий, валидацию выходов, мониторинг поведения. Управление рисками ИИ ближе к управлению рисками в сложных социотехнических системах, чем к управлению уязвимостями в ПО.

Угрозы ИИ-системам можно структурировать по нескольким осям. По источнику: внешний злоумышленник, внутренний нарушитель, конкурент, активист, исследователь, автоматизированная система. По цели: данные (кража, отравление, подмена), модель (извлечение, компрометация, подмена), приложение (обход ограничений, злоупотребление функциями), инфраструктура (отказ в обслуживании, компрометация среды), люди и процессы (социальная инженерия, обход процедур). По стадии жизненного цикла: угрозы при сборе данных, при обучении, при тестировании, при развёртывании, при эксплуатации. MITRE ATLAS систематизирует тактики и техники атак на ИИ-системы, выстраивая матрицу, аналогичную MITRE ATT&CK для традиционных атак. Эта систематика подробно рассматривается в третьей части книги; здесь важно зафиксировать, что ландшафт угроз ИИ существенно шире, чем привычный периметр информационной безопасности.

Понятие воздействия (impact) в контексте ИИ приобретает особое значение. ISO/IEC 42001 требует от организации проводить оценку воздействия ИИ-системы на людей, группы людей и общество. EU AI Act выстраивает всю систему регулирования на основе уровня риска, который определяется потенциальным воздействием системы на здоровье, безопасность и права человека. Воздействие ИИ-системы может быть прямым (модель отклоняет кредитную заявку) и косвенным (рекомендательная система формирует информационный пузырь, влияющий на взгляды человека). Оно может быть немедленным (блокировка транзакции) и отложенным (накопление смещения в решениях, которое проявляется через месяцы). Оно может затрагивать конкретного человека (отказ в приёме на работу) или целую группу (систематическая недооценка кредитоспособности определённой демографической категории).

Для специалиста по безопасности это означает, что оценка рисков ИИ-системы не может ограничиваться техническими сценариями атак. Она должна включать оценку воздействия на людей, которых затрагивают решения системы. Модель, которая технически защищена от извлечения и отравления, но систематически дискриминирует определённую группу клиентов, создаёт для организации регуляторный, юридический и репутационный риск, который может оказаться более значительным, чем риск кибератаки.

Ещё одна особенность рисков ИИ — их каскадный характер. Инцидент в одном компоненте может распространиться по всей системе. Отравление обучающих данных на стадии подготовки приводит к смещению модели, которое проявляется в ошибочных решениях на стадии эксплуатации и обнаруживается только при аудите или жалобе клиента. Компрометация одного МСР-сервера позволяет внедрить инструкцию, которую агент передаёт другому агенту, и цепочка поражения охватывает несколько систем. Каскадность усложняет как оценку рисков, так и расследование инцидентов: нужно отслеживать причинно-следственные связи через несколько компонентов, стадий и участников.

Для организации, выстраивающей программу безопасности ИИ, из этого раздела следует несколько выводов. Привычная триада конфиденциальности, целостности и доступности необходима, но недостаточна: к ней нужно добавить справедливость, достоверность, подотчётность и безопасность для людей. Управление уязвимостями в классическом смысле применимо к инфраструктуре, но не к свойствам модели: здесь нужны архитектурные ограничения и непрерывный мониторинг. Оценка рисков должна охватывать три уровня последствий: для людей, для организации и для экосистемы. Каскадный характер рисков требует анализа взаимосвязей между компонентами, а не только оценки каждого компонента по отдельности. Именно эти принципы лежат в основе процесса оценки рисков, который мы разберём в седьмой главе.

4.2. Свойства доверия: безопасность, надёжность, устойчивость, справедливость

В предыдущем разделе мы определили, что привычная триада конфиденциальности, целостности и доступности недостаточна для описания рисков ИИ. Модель может быть технически защищена и при этом причинять ущерб: давать опасные рекомендации, дискриминировать определённые группы людей, работать непредсказуемо в условиях, которые не встречались при обучении. Чтобы оценить, заслуживает ли ИИ-система доверия, нужен более широкий набор критериев.



Рисунок 11. Семь характеристик доверия ИИ-систем по NIST AI RMF: валидность и надёжность, безопасность, защищённость и устойчивость, подотчётность и прозрачность, объяснимость и интерпретируемость, защита приватности, справедливость и управление вредоносными смещениями.

NIST AI RMF формулирует семь характеристик доверия к ИИ-системам: достоверность и надёжность, безопасность для людей, защищённость и устойчивость, подотчётность и прозрачность, объяснимость и интерпретируемость, защита приватности и справедливость с управлением вредоносными смещениями. Эти характеристики не существуют изолированно: они пересекаются, усиливают друг друга и иногда вступают в противоречие. Организация, которая управляет только одной из них, упускает системную картину. В этом разделе мы рассмотрим первые четыре свойства; прозрачности, объяснимости и подотчётности посвящён следующий раздел.

Безопасность для людей (safety) — свойство, которое часто путают с защищённостью от кибератак (security), хотя они описывают разные аспекты. Безопасность означает, что система не причиняет вреда людям, обществу и окружающей среде в ходе нормальной работы. ИИ-система может быть полностью защищена от внешних атак и при этом выдавать рекомендации, следование которым приводит к ущербу. Медицинская модель, которая предлагает неверную дозировку лекарства, небезопасна, даже если её инфраструктура защищена по всем стандартам. Автономный агент, который выполняет задачу способом, создающим физическую угрозу для людей, небезопасен, даже если его никто не атаковал.

Безопасность для людей определяется на этапе проектирования и поддерживается на протяжении всего жизненного цикла. Организация должна определить, какие последствия может вызвать ошибочное решение системы, и выстроить ограничения, соразмерные этим последствиям. Для системы, рекомендующей фильмы, цена ошибки — потерянное время зрителя. Для системы, влияющей на медицинский диагноз, цена ошибки может измеряться здоровьем и жизнью. Между этими полюсами лежит широкий спектр, и уровень контролей должен соответствовать уровню потенциального вреда. NIST AI RMF подчёркивает, что система должна допускать безопасное прерывание работы: если её поведение выходит за допустимые границы, должен существовать механизм остановки, не создающий дополнительных рисков.

Достоверность и надёжность (validity and reliability) описывают способность системы работать корректно и предсказуемо. Достоверность означает, что система решает именно ту задачу, для которой предназначена, и её результаты соответствуют действительности. Надёжность означает, что система выдаёт стабильные результаты при повторных обращениях в одних и тех же условиях и сохраняет работоспособность в условиях, которые отличаются от обуча-

ющей выборки. Эти свойства кажутся очевидными, но на практике с ними связаны одни из наиболее распространённых проблем.

Достоверность ИИ-системы не сводится к точности модели на тестовом наборе данных. Модель может показывать высокую точность на тестовых данных и при этом систематически ошибаться в продуктивной среде, если тестовый набор не отражает реальное распределение запросов. Конфабуляция в языковых моделях — прямое нарушение достоверности: система уверенно генерирует ответ, который не соответствует действительности, и пользователь не может отличить правильный ответ от выдуманного без внешней проверки. Организации, использующие LLM для подготовки юридических, финансовых или медицинских документов, сталкиваются с этой проблемой ежедневно.

Надёжность нарушается дрейфом, который мы обсуждали в разделе 1.3. Модель, надёжно работавшая полгода, может начать выдавать некорректные результаты при изменении характеристик входных данных. Если организация не отслеживает стабильность результатов, деградация может оставаться незамеченной неделями и месяцами, пока накопившиеся ошибки не приведут к инциденту. Непрерывный мониторинг метрик качества и поведения модели — необходимое условие поддержания надёжности в продуктивной среде.

Защищённость и устойчивость (security and resilience) описывают способность системы противостоять атакам и восстанавливаться после сбоев. Защищённость охватывает весь спектр мер безопасности: от контроля доступа и шифрования до защиты от адверсариальных воздействий, инъекций в промпт и извлечения модели. Устойчивость добавляет измерение, которое в информационной безопасности часто недооценивается: способность системы продолжать работу в условиях атаки, частичного отказа или деградации и вернуться к нормальному состоянию после инцидента.

Для ИИ-систем устойчивость приобретает особое значение, потому что их поведение зависит от множества внешних факторов. Модель, вызываемая через API поставщика, становится недоступной при сбое на стороне поставщика. Агент, зависящий от десятка внешних инструментов через MCP, теряет функциональность при недоступности любого из них. RAG-система, обращающаяся к внешней базе знаний, деградирует, если база скомпрометирована или недоступна. Организация должна определить, что произойдёт с бизнес-процессом, если ИИ-система перестанет работать, и предусмотреть резервные механизмы. В некоторых случаях это означает ручной процесс, который может заменить автоматизированное решение. В других — переключение на альтернативную модель или поставщика. Полная зависимость от одного ИИ-сервиса без плана непрерывности — архитектурный просчёт, который обнаруживается в самый неподходящий момент.

Справедливость с управлением вредоносными смещениями (fairness with harmful bias managed) — свойство, которое выводит безопасность ИИ за пределы чисто технической дисциплины. Справедливость означает, что система не создаёт неоправданных различий в обращении с людьми на основании признаков, не относящихся к задаче: пола, возраста, этнической принадлежности, религии, инвалидности. Смещение (bias) в данных и моделях может быть статистическим, историческим или системным. Статистическое смещение возникает, когда обучающая выборка не представляет все группы. Историческое — когда данные отражают прошлые практики дискриминации, и модель воспроизводит их как закономерность. Системное — когда выбор признаков, архитектуры или метрик качества непропорционально влияет на определённые группы.

Справедливость не является абстрактным этическим принципом для книги по безопасности ИИ. Это конкретный риск с конкретными последствиями. EU AI Act запрещает ИИ-системы, использующие социальный скоринг или эксплуатирующие уязвимость определённых групп. Для систем высокого риска, включая модели кредитного скоринга и отбора персонала, регламент требует мер по выявлению и устранению дискриминации. Судебные разби-

рательства, связанные с дискриминационными решениями алгоритмов, становятся всё более частыми. Организация, которая не проверяет свои модели на справедливость, принимает юридический, регуляторный и репутационный риск, даже если модель работает корректно с технической точки зрения.

На практике управление справедливостью сталкивается с несколькими трудностями. Само определение справедливости неоднозначно: существуют десятки математических формализаций, и некоторые из них взаимно исключают друг друга. Модель не может одновременно гарантировать равную долю ложноположительных результатов для всех групп и равную долю ложноотрицательных. Выбор метрики справедливости — это управленческое и этическое решение, а не чисто техническое. Организация должна определить, какое определение справедливости применимо к конкретной задаче, задокументировать свой выбор и проверять соответствие модели этому критерию на регулярной основе.

Все четыре свойства, рассмотренные в этом разделе, работают вместе. Модель может быть безопасной для людей, но ненадёжной: она не причиняет вреда, когда работает правильно, но часто работает неправильно. Она может быть надёжной, но несправедливой: стабильно выдаёт результат, но этот результат систематически ущемляет определённую группу. Она может быть защищённой от атак, но небезопасной: никто не может её взломать, но её собственные решения причиняют вред. Управление доверием к ИИ-системе требует одновременного внимания ко всем свойствам, и организация, которая сосредоточена только на одном из них, получает одномерную защиту в многомерном пространстве рисков.

4.3. Прозрачность, объяснимость и подотчётность

Четыре свойства, рассмотренные в предыдущем разделе, описывают, как система должна работать. Прозрачность, объяснимость и подотчётность описывают, как организация и общество могут убедиться, что система работает именно так. Без этих трёх свойств остальные характеристики доверия превращаются в декларации: модель может быть названа безопасной и справедливой, но если невозможно проверить, на основании чего она принимает решения и кто несёт ответственность за результат, доверие остаётся неподтверждённым.

Прозрачность (transparency) означает, что заинтересованные стороны имеют доступ к информации о системе в объёме, соответствующем их роли и потребностям. NIST AI RMF определяет прозрачность как обеспечение доступа к нужному уровню информации на каждой стадии жизненного цикла, адаптированного к роли участника. Для разработчика прозрачность — это документация по архитектуре модели, обучающим данным, метрикам качества и известным ограничениям. Для оператора — информация о конфигурации, условиях эксплуатации и процедурах обновления. Для пользователя — понимание того, что он взаимодействует с ИИ-системой, какие данные она обрабатывает и какие ограничения имеет. Для регулятора — доказательства соответствия требованиям, результаты оценки рисков и воздействия, журналы решений. Для человека, затронутого решением системы, — информация о том, что решение принято с участием ИИ и каковы его основания.

На практике прозрачность вступает в конфликт с другими интересами. Поставщик модели не хочет раскрывать архитектуру и обучающие данные, потому что это интеллектуальная собственность. Избыточная прозрачность может облегчить злоумышленнику поиск уязвимостей: зная структуру системного промпта, атакующий быстрее найдёт способ обойти ограничения. Раскрытие деталей обучающих данных может нарушить приватность людей, чьи данные использовались при обучении. NTIA в своём отчёте об ответственности ИИ отмечает эти противоречия: больше прозрачности о логике системы и данных улучшает возможность оспаривания решений, но может нарушить приватность и права на интеллектуальную собственность; чем прозрачнее модель, тем больше она подвержена манипуляциям. Организация должна

определить, какой уровень прозрачности необходим для каждой заинтересованной стороны и каждого сценария, учитывая эти компромиссы.

EU AI Act вводит конкретные требования к прозрачности для разных категорий систем. Поставщики систем высокого риска обязаны предоставлять документацию, достаточную для понимания работы системы и оценки её соответствия требованиям. Системы, которые взаимодействуют с людьми, должны информировать пользователя о том, что он общается с ИИ. Системы, генерирующие синтетический контент (текст, изображение, аудио, видео), должны маркировать свои результаты как сгенерированные искусственным интеллектом. Эти требования формируют минимальный уровень прозрачности, ниже которого организация не может опускаться.

Объяснимость (explainability) и интерпретируемость (interpretability) — два связанных, но различающихся понятия. NIST AI RMF определяет объяснимость как способность описать механизмы работы ИИ-системы, а интерпретируемость — как способность придать смысл результатам системы в контексте её назначения. Проще говоря, объяснимость отвечает на вопрос «как система пришла к этому решению», а интерпретируемость — на вопрос «что это решение означает для конкретного пользователя в конкретной ситуации».

Для организации объяснимость имеет несколько практических измерений. Техническая объяснимость позволяет разработчикам и тестировщикам отлаживать модель, выявлять ошибки и понимать причины некорректного поведения. Операционная объяснимость даёт операторам и службе поддержки возможность ответить на вопрос клиента: почему система приняла такое решение? Регуляторная объяснимость позволяет организации продемонстрировать аудиту или регулятору, что решения системы принимаются обоснованно и в соответствии с требованиями. Каждое из этих измерений предъявляет свои требования к глубине и форме объяснения.

Методы объяснимости различаются по подходу. Внутренняя объяснимость (ante-hoc) достигается выбором изначально интерпретируемой модели: линейная регрессия, дерево решений, набор правил. Такие модели позволяют точно проследить, какие признаки и с каким весом повлияли на результат. Внешняя объяснимость (post-hoc) применяется к сложным моделям, которые сами по себе непрозрачны: глубоким нейронным сетям, ансамблям, большим языковым моделям. Методы внешней объяснимости создают приближённые интерпретации, показывая, какие входные данные оказали наибольшее влияние на результат. Важно понимать ограничение: post-hoc объяснение — это аппроксимация, а не точное воспроизведение логики модели. Оно может быть полезным для понимания общих паттернов, но не гарантирует полного соответствия реальному процессу принятия решения.

Для организации выбор между интерпретируемой и сложной моделью — это архитектурное решение с последствиями для безопасности и соответствия. Если регулятор требует объяснения каждого автоматизированного решения, затрагивающего права человека, использование непрозрачной модели создаёт регуляторный риск. Если бизнес-задача требует максимальной точности и сложная модель значительно превосходит интерпретируемую, организация должна инвестировать в методы post-hoc объяснимости и документирование их ограничений. Этот компромисс между точностью и объяснимостью — один из наиболее распространённых в практике ИИ, и он не имеет универсального решения: ответ зависит от контекста, уровня риска и требований регулятора.

Подотчётность (accountability) замыкает цепочку. Прозрачность предоставляет информацию. Объяснимость позволяет понять решение. Подотчётность определяет, кто несёт ответственность за результат и какие механизмы позволяют привлечь к ответственности. Без подотчётности прозрачность и объяснимость остаются информационными инструментами, не связанными с последствиями.

NTIA в отчёте об ответственности ИИ выстраивает цепочку подотчётности из трёх элементов: информационный поток (документация, раскрытие информации, маркировка), оценка системы (аудиты, оценка воздействия, тестирование, состязательное тестирование (red teaming)) и последствия (ответственность, регулирование, рыночные механизмы). Подотчётность работает, когда все три элемента присутствуют: информация о системе доступна, система оценена независимо, и при обнаружении проблемы наступают реальные последствия для того, кто за неё отвечает.

ISO/IEC 42001 закрепляет подотчётность через требования к определению ролей и ответственности, документированию решений и механизмам информирования о проблемах. Организация должна определить, кто отвечает за каждый аспект ИИ-системы, задокументировать эту ответственность и создать каналы, через которые любой сотрудник может сообщить о проблеме с ИИ-системой без риска негативных последствий для себя. Последний пункт часто упускается: если сотрудник, заметивший некорректное поведение модели, не имеет простого способа сообщить об этом или опасается последствий, система обратной связи не работает, и проблемы накапливаются до инцидента.

На практике подотчётность в ИИ-системах сталкивается с проблемой распределённой ответственности, которую мы разбирали в предыдущей главе. Когда решение принимает модель, обученная одним поставщиком, развёрнутая на инфраструктуре другого, интегрированная в приложение третьего и используемая сотрудником четвертого, установить, кто именно отвечает за конкретный ущерб, становится сложно. Подотчётность требует сквозной прослеживаемости: от входных данных через решение модели к конечному действию. Это, в свою очередь, требует журналирования на каждом уровне, сохранения контекста принятия решения и возможности восстановить цепочку событий при расследовании. Технические требования к журналированию и мониторингу, которые мы рассмотрим в девятнадцатой главе, напрямую поддерживают подотчётность.

Все три свойства — прозрачность, объяснимость и подотчётность — связаны между собой и с остальными характеристиками доверия. Прозрачность делает возможной проверку справедливости: если организация не раскрывает, на каких данных обучена модель, проверить наличие смещений невозможно. Объяснимость поддерживает безопасность для людей: если модель не может объяснить своё решение, человек-контролёр не может оценить его правильность. Подотчётность создаёт стимулы для поддержания всех остальных свойств: если никто не несёт ответственности за ошибку модели, мотивация инвестировать в надёжность, справедливость и безопасность снижается.

Для организации практический вывод из этого раздела состоит в следующем. Прозрачность, объяснимость и подотчётность — это не дополнительные функции, которые можно добавить к готовой системе, а архитектурные требования, которые нужно закладывать при проектировании. Выбор модели должен учитывать требования к объяснимости. Архитектура приложения должна предусматривать журналирование в объёме, достаточном для прослеживания решений. Контракты с поставщиками должны фиксировать обязательства по раскрытию информации. Организационная структура должна определять ответственных за каждый аспект ИИ-системы. Без этих элементов доверие к системе остаётся неподтверждённым, а ответственность — нераспределённой.

4.4. Взаимозависимость свойств доверия и практические компромиссы

В двух предыдущих разделах мы рассмотрели семь свойств доверия по отдельности. На практике они не работают изолированно. Усиление одного свойства может ослабить другое.

Попытка одновременно максимизировать все свойства наталкивается на ограничения, заложенные в самой природе ИИ-систем.

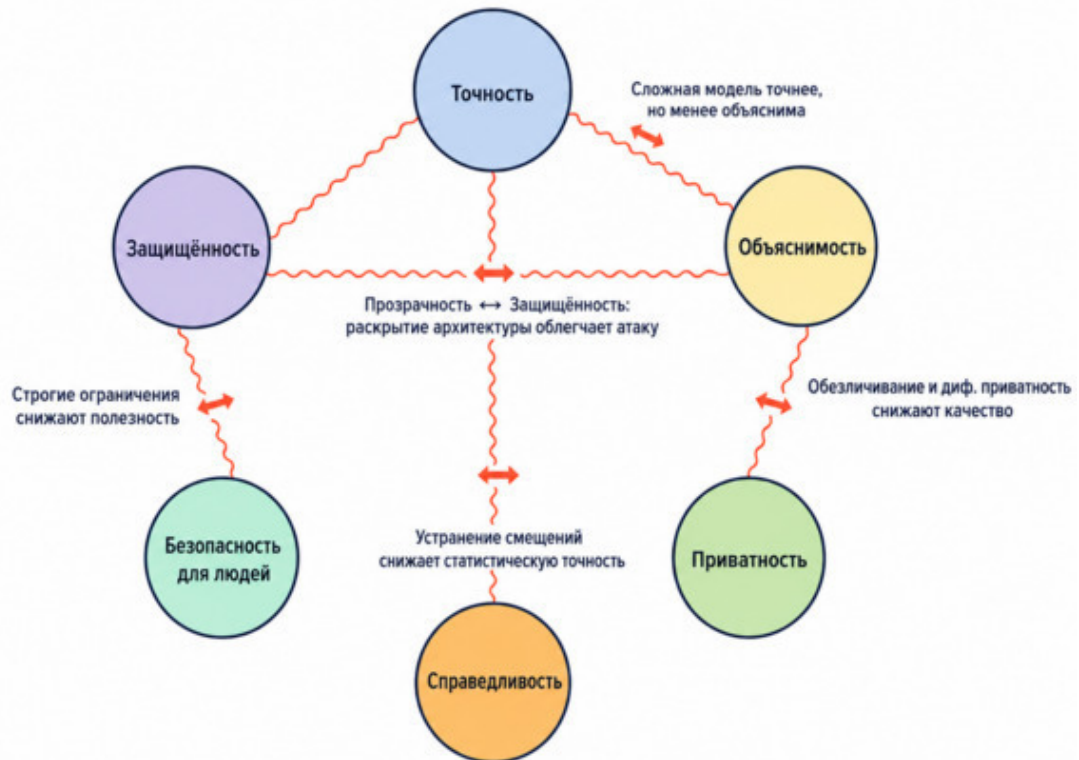


Рисунок 12. Карта компромиссов между безопасностью, точностью, приватностью, объяснимостью, справедливостью и производительностью ИИ-системы.

Организация, которая не осознаёт эти компромиссы, рискует принимать архитектурные решения, которые решают одну проблему и создают другую. NIST AI RMF прямо указывает на это: характеристики доверия редко присутствуют все одновременно в полной мере, компромиссы почти неизбежны, и доверие к системе определяется её самым слабым звеном.

Наиболее изученный компромисс — между точностью модели и объяснимостью. Линейная регрессия и дерево решений легко интерпретируются: можно точно указать, какие признаки повлияли на результат и с каким весом. Глубокая нейронная сеть с миллиардами параметров может превосходить интерпретируемую модель по точности, но объяснить её решение можно только приближённо. Для организации это означает выбор. Если задача связана с высокорисковыми решениями, затрагивающими права людей, регулятор может потребовать полной объяснимости, и организация вынуждена жертвовать частью точности ради интерпретируемости. Если задача допускает менее строгие требования к объяснимости (например, рекомендация товаров), организация может выбрать более сложную модель и дополнить её post-hoc методами интерпретации.

Компромисс между приватностью и качеством модели проявляется на нескольких уровнях. Обезличивание обучающих данных снижает риск утечки персональной информации, но может ухудшить качество модели, если удалённые признаки были значимыми для задачи. Дифференциальная приватность — математический метод, гарантирующий, что присутствие или отсутствие конкретной записи в обучающем наборе не влияет на выходные данные модели — вносит контролируемый шум, который снижает точность. Федеративное обучение, при котором данные остаются у их владельцев и модель обучается на распределённых наборах, решает проблему централизации чувствительных данных, но требует значительно больше

вычислительных ресурсов и может быть менее точным, чем обучение на централизованном наборе. Организация должна определить, какой уровень приватности необходим для конкретной системы, и осознанно выбрать точку баланса.

Менее очевидный, но практически значимый компромисс существует между безопасностью для людей и доступностью системы. Строгие ограничения на поведение модели снижают вероятность опасных результатов, но одновременно могут сделать систему менее полезной. Чат-бот, настроенный на максимальную осторожность, отказывается отвечать на вопросы, которые воспринимает как потенциально рискованные, даже если они легитимны. Медицинская модель, которая при любом сомнении направляет пациента к врачу, теряет ценность как инструмент первичной сортировки. Фильтрация выходных данных, которая блокирует ответы, содержащие определённые ключевые слова, может отсекают полезную информацию вместе с опасной. Организация, настраивающая ограничения поведения модели, должна найти баланс между безопасностью и функциональностью, и этот баланс зависит от контекста применения.

Справедливость вступает в конфликт с точностью в ситуациях, где исторические данные отражают неравенство. Модель кредитного скоринга, обученная на данных, в которых определённые демографические группы объективно имели более высокий уровень дефолтов (возможно, вследствие системного неравенства в доступе к экономическим возможностям), может быть точной в статистическом смысле, но несправедливой в этическом и правовом. Устранение этого смещения — через перебалансировку обучающих данных, штрафные функции или постобработку результатов — неизбежно снижает статистическую точность модели для некоторых групп. Организация должна решить, что для неё приоритетнее: предсказательная точность или равное обращение. Это решение не может быть принято исследователем данных в одиночку; оно требует участия руководства, юристов и специалистов по этике.

Прозрачность и защищённость тоже находятся в напряжении. Раскрытие архитектуры модели, структуры системного промпта и состава обучающих данных повышает доверие и позволяет проводить независимый аудит. Но та же информация помогает злоумышленнику: зная архитектуру, он может эффективнее конструировать адверсариальные воздействия; зная системный промпт, он быстрее найдёт способ обхода ограничений; зная состав обучающих данных, он может целенаправленно формировать входные данные, которые вызовут нежелательное поведение. Организация вынуждена определить, какую информацию раскрывать публично, какую — по запросу аудитора или регулятора, а какую считать конфиденциальной. Уровень раскрытия должен быть соразмерен уровню риска и требованиям заинтересованных сторон.

Устойчивость к враждебным воздействиям может противоречить удобству использования. Строгая верификация входных данных, многофакторная аутентификация при каждом запросе, ограничение контекстного окна, запрет на загрузку внешних документов — всё это повышает устойчивость системы к атакам, но снижает производительность и удобство для легитимных пользователей. Контроли, которые замедляют работу агента при вызове инструментов (подтверждение каждого действия человеком, проверка результатов на каждом шаге), делают систему безопаснее, но снижают преимущество автоматизации, ради которого она и внедрялась. Организация должна калибровать строгость контролей в зависимости от уровня риска конкретной операции, а не применять одинаковый уровень ко всем взаимодействиям.

Как организации управлять этими компромиссами? Первый шаг — признать, что они существуют, и перестать воспринимать свойства доверия как независимые чекбоксы, которые нужно все отметить. Второй шаг — определить приоритеты для каждой конкретной ИИ-системы, исходя из её назначения, контекста применения, регуляторных требований и последствий ошибки. Для медицинской диагностической системы безопасность для людей и объяснимость будут приоритетнее, чем максимальная точность. Для рекомендательной системы в

интернет-магазине точность и удобство важнее полной объяснимости. Для системы кредитного скоринга справедливость и подотчётность стоят выше производительности.

Третий шаг — документировать принятые решения и их обоснование. Если организация осознанно выбрала более сложную модель ради точности, пожертвовав частью объяснимости, это решение должно быть зафиксировано: кто его принял, какие альтернативы рассматривались, какие компенсирующие меры применены (например, post-hoc методы интерпретации), при каких условиях решение подлежит пересмотру. Документирование компромиссов выполняет две функции. Оно создаёт основу для подотчётности: если последствия решения окажутся негативными, можно проследить логику и определить, было ли решение обоснованным на момент принятия. И оно формирует институциональную память: новые участники команды понимают, почему система устроена именно так, а не иначе.

Четвёртый шаг — пересматривать баланс при изменении условий. Компромисс, который был разумным при запуске системы, может стать неприемлемым при изменении регуляторных требований, расширении аудитории или выявлении новых угроз. Организация, которая приняла решение использовать непрозрачную модель до вступления в силу EU AI Act, может оказаться перед необходимостью пересмотреть это решение, когда требования к объяснимости станут обязательными для её категории системы. Регулярная переоценка компромиссов должна быть частью процесса управления рисками ИИ, а не разовым упражнением при проектировании.

Завершая эту главу и первую часть книги, стоит зафиксировать ключевую мысль. Доверие к ИИ-системе складывается из множества свойств, каждое из которых связано с остальными. Организация не может управлять ими по отдельности, как не может защитить дом, поставив стальную дверь и оставив окна открытыми. Безопасность, надёжность, справедливость, прозрачность, объяснимость и подотчётность образуют систему, в которой слабость любого элемента подрывает целое. Задача организации — не достичь идеала по каждому свойству, а выстроить осознанный, документированный и пересматриваемый баланс, соответствующий уровню риска и ожиданиям заинтересованных сторон. Именно этот баланс и составляет основу архитектуры безопасного ИИ, которую мы начнём строить во второй части книги.

Источники и дальнейшее чтение к главе 4

— NIST AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology, январь 2023 — определение риска, трёхуровневая модель последствий (люди / организация / экосистема), семь характеристик доверия, признание неизбежности компромиссов между ними.

— MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). MITRE Corporation — систематика тактик и техник атак на ИИ-системы.

— ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system — требование к оценке воздействия на людей и общество, определение ролей и ответственности, механизмы информирования о проблемах.

— Regulation (EU) 2024/1689 (EU AI Act) — уровни риска и их регуляторные последствия, запрет социального скоринга, требования к прозрачности и маркировке синтетического контента, требования к объяснимости для систем высокого риска.

— Goodman E. P. Artificial Intelligence Accountability Policy Report. National Telecommunications and Information Administration (NTIA), март 2024 — цепочка подотчётности (информационный поток → оценка → последствия), компромисс между прозрачностью и приватностью/безопасностью.

— Dwork C. Differential Privacy, 2006 (или обзорная работа Dwork, Roth «The Algorithmic Foundations of Differential Privacy», 2014) — математический метод дифференциальной приватности, упомянутый в разделе 4.4.

— McMahan H. B. и др. Communication-Efficient Learning of Deep Networks from Decentralized Data, 2017 — основополагающая работа по федеративному обучению.

Часть II. Управление и риски

Первая часть книги дала читателю язык и карту: из чего состоит ИИ-система, какие типы систем существуют, через какие стадии они проходят, кто за них отвечает и какие свойства определяют доверие. Всё это необходимое основание, но само по себе оно не защищает организацию. Знание того, что модель может быть отравлена, не предотвращает отравление. Понимание, что справедливость и точность вступают в противоречие, не снимает необходимости принять решение. Между пониманием рисков и управлением ими лежит система: политики, роли, процессы, классификации и документированные решения, без которых безопасность ИИ остаётся набором разрозненных усилий.

Эта часть книги формирует управленческую основу. Пятая глава показывает, как выстроить систему управления ИИ и интегрировать её с действующими процессами организации. Шестая глава описывает инвентаризацию и классификацию ИИ-систем, включая выявление теневого ИИ. Седьмая глава раскрывает процесс оценки рисков и воздействия, человеческий контроль и документирование решений. Восьмая глава даёт обзор нормативных требований и стандартов, помогая организации ориентироваться в регуляторном пространстве.

После прочтения этой части читатель сможет сформировать политику допустимого применения ИИ, провести инвентаризацию и классификацию ИИ-систем в своей организации, выстроить процесс оценки рисков и определить, какие стандарты и нормативные требования к ней применимы.

Глава 5. Управление ИИ в организации

ИИ-системы появляются в организациях быстрее, чем процессы управления ими. Подразделение маркетинга подключает генеративную модель для создания контента. Финансовый отдел тестирует модель прогнозирования выручки. Служба поддержки запускает чат-бот для обработки обращений. Каждый из этих проектов решает конкретную бизнес-задачу, но ни один из них, как правило, не проходил через единый процесс утверждения, оценки рисков и определения ответственности. Когда таких проектов становится десятки, организация обнаруживает, что у неё нет целостного представления о том, какие ИИ-системы работают, кто за них отвечает, какие данные они обрабатывают и какие риски создают.

Эта глава описывает, как выстроить систему управления ИИ (AI governance), которая приводит в порядок разрозненные инициативы и создаёт основу для осознанного принятия решений о безопасности. Мы рассмотрим, зачем организации нужно управление ИИ, как сформулировать политику допустимого применения, как распределить роли и ответственность, как связать управление ИИ с действующими системами управления и как документировать принятые решения.

5.1. Зачем организации нужно управление ИИ

Вопрос может показаться риторическим. Организации управляют информационной безопасностью, рисками, качеством, соответствием требованиям — почему ИИ должен стать исключением? На практике, однако, многие организации именно так и поступают: внедряют ИИ без формализованного управления, полагаясь на инициативу отдельных команд и здравый смысл руководителей проектов. Пока ИИ-систем мало и они решают ограниченные задачи, такой подход работает. Проблемы начинаются при масштабировании.

Первая проблема — отсутствие видимости. Руководство организации не знает, сколько ИИ-систем работает, какие из них обрабатывают чувствительные данные, какие принимают решения, затрагивающие клиентов, и какие используют внешние модели через API. Без инвентаризации невозможно оценить совокупный риск. Без классификации невозможно определить приоритеты. Без единого реестра организация не может ответить на запрос регулятора о том, какие ИИ-системы она использует и какие меры контроля к ним применяет.

Вторая проблема — размытая ответственность. Кто отвечает за безопасность конкретной ИИ-системы? В большинстве организаций ответ зависит от того, кого спросить. Разработчик считает, что за безопасность отвечает инфраструктурная команда. Инфраструктурная команда считает, что за поведение модели отвечает разработчик. Владелец бизнес-процесса считает, что вопросы безопасности — зона ответственности CISO. CISO узнаёт о существовании системы, когда происходит инцидент. Эта ситуация не уникальна: она воспроизводится в организациях разного масштаба и разной отрасли.

Третья проблема — несогласованность решений. Одно подразделение запрещает сотрудникам использовать внешние LLM-сервисы, другое активно их внедряет. Одна команда проводит оценку рисков перед запуском, другая считает это излишней бюрократией. Одна организация требует от поставщика модели подробную документацию по безопасности, другой поставщик предоставляет только общие условия сервиса. Без единой политики каждый проект принимает решения по собственным критериям, и уровень защиты определяется не уровнем риска, а зрелостью конкретной команды.

Четвёртая проблема — регуляторное давление. EU AI Act, вступивший в силу в 2024 году, требует от организаций, использующих ИИ-системы высокого риска, выстроить систему управления рисками, документировать решения, обеспечить человеческий контроль и про-

зрачность. ISO/IEC 42001:2023 предоставляет международный стандарт системы управления ИИ, который может стать основой для сертификации. Регуляторы в разных юрисдикциях всё чаще задают вопросы о том, как организация управляет своими ИИ-системами. Организация, у которой нет формализованного управления, не может ответить на эти вопросы убедительно.

ISO/IEC 42001 выстраивает систему управления ИИ на основе привычной структуры международных стандартов менеджмента. Руководство организации устанавливает политику ИИ, определяет цели, распределяет ответственность и выделяет ресурсы. Организация определяет контекст — внешний и внутренний, — понимает потребности заинтересованных сторон и устанавливает область применения системы управления. Планирование включает оценку рисков, оценку воздействия и определение целей. Поддержка охватывает ресурсы, компетенции, осведомлённость, коммуникацию и документирование. Эксплуатация включает оперативное планирование и контроль, оценку рисков и воздействия на практике. Оценка результативности строится на мониторинге, внутреннем аудите и анализе со стороны руководства. Улучшение предполагает непрерывную корректировку по результатам мониторинга и аудита. Эта логика знакома организациям, которые уже внедрили ISO 27001 или другие стандарты менеджмента, и именно эта знакомость является преимуществом: управление ИИ можно выстраивать не с нуля, а как расширение существующей системы.

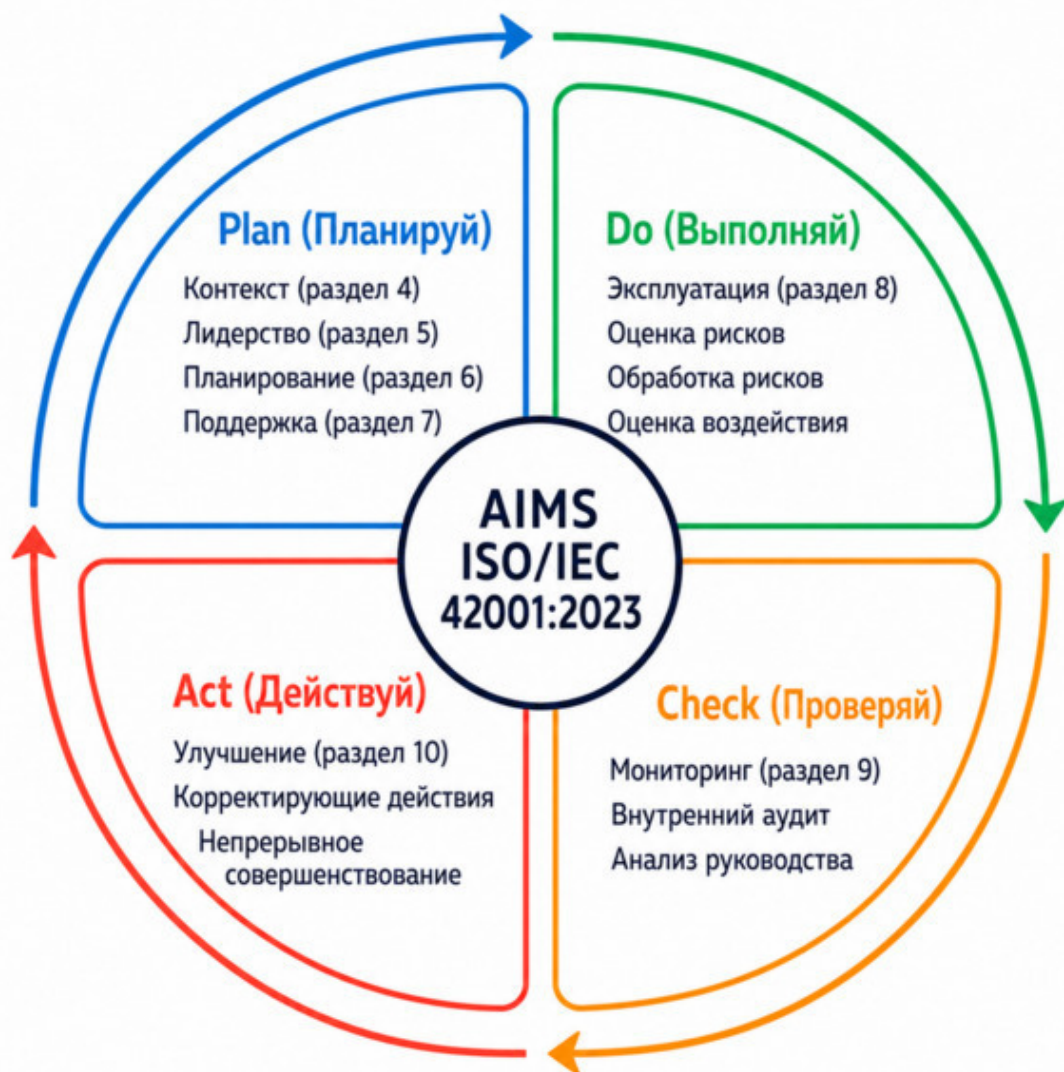


Рисунок 13. Цикл непрерывного управления ИИ-системой: планирование, реализация, оценка и улучшение.

Но между ISO 42001 и реальной практикой лежит пространство, которое стандарт не покрывает. Стандарт описывает, что организация должна сделать, но не объясняет, как это сделать в условиях, когда ИИ-системы уже работают, бюджет ограничен, а экспертиза в области безопасности ИИ сосредоточена в одном-двух специалистах. Стандарт требует определить роли и ответственность, но не помогает решить, кто должен быть владельцем рисков конкретной LLM-системы, развёрнутой бизнес-подразделением без участия ИТ. Стандарт требует интеграции с другими системами управления, но не объясняет, как согласовать процессы оценки рисков ИИ с уже существующим процессом управления операционными рисками, который не предусматривает категорий, специфичных для ИИ.

Именно эти практические вопросы мы разбираем в последующих разделах этой главы. Управление ИИ для книги — это не пересказ стандарта, а набор управленческих решений, которые организация принимает, чтобы ИИ-системы приносили пользу, не создавая неуправляемых рисков. Политика определяет, что допустимо. Роли определяют, кто отвечает. Интеграция с действующими процессами определяет, как это работает на практике. Документирование определяет, как организация доказывает свою позицию регулятору, аудитору и самой себе.

Для организации, которая ещё не приступала к формализации управления ИИ, практический вывод из этого раздела прост: начинать нужно сейчас, а не когда количество ИИ-систем станет неуправляемым. Каждый месяц без управления — это месяц, в течение которого растёт количество неучтённых систем, накапливаются необдуманные решения и увеличивается разрыв между скоростью внедрения и готовностью организации к управлению рисками. Стоимость наведения порядка растёт пропорционально объёму накопленного хаоса.

5.2. Политика ИИ и допустимые сценарии применения

Управление начинается с позиции. Организация должна определить, как она относится к использованию ИИ: что допустимо, что запрещено, что разрешено при соблюдении определённых условий. Без этой позиции каждое подразделение принимает решения самостоятельно, и результат зависит от осведомлённости и осторожности конкретного руководителя. Политика ИИ фиксирует эту позицию в документе, который доступен всем сотрудникам и обязателен для исполнения.

ISO/IEC 42001:2023 требует, чтобы высшее руководство организации установило политику ИИ, которая соответствует целям организации, создаёт основу для определения целей управления ИИ, включает обязательство соответствовать применимым требованиям и предусматривает непрерывное улучшение. Стандарт также указывает, что политика должна быть задокументирована, доведена до сведения сотрудников и доступна заинтересованным сторонам. Эти требования задают рамку, но не определяют содержание. Организации должна наполнить политику конкретными положениями, отражающими её контекст, отрасль, уровень рисков и регуляторное окружение.

На практике политика ИИ состоит из нескольких смысловых блоков, каждый из которых отвечает на конкретный вопрос.

Первый блок определяет область применения: на какие ИИ-системы распространяется политика. Организация должна решить, покрывает ли политика только системы, разработанные внутри, или также внешние сервисы, вызываемые через API. Распространяется ли она на использование генеративного ИИ сотрудниками в личных рабочих инструментах — браузерных расширениях, мобильных приложениях, встроенных ассистентах в офисном ПО?

Включает ли она модели, встроенные в продукты поставщиков, которые организация закупает? Слишком узкая область применения оставляет за периметром политики значительную часть ИИ-использования. Слишком широкая делает политику трудноисполнимой. Оптимальный подход — определить область через критерий воздействия: политика распространяется на любое использование ИИ, которое обрабатывает данные организации, влияет на решения, затрагивающие людей, или создаёт обязательства перед регуляторами.

Второй блок устанавливает допустимые и запрещённые сценарии. Здесь организация проводит границу между тем, что можно делать с ИИ, и тем, что нельзя. EU AI Act задаёт минимальный уровень ограничений: запрещены системы социального скоринга, манипулятивные системы, эксплуатирующие уязвимости людей, биометрическая идентификация в реальном времени в общественных пространствах (с ограниченными исключениями). Организация может расширить этот перечень в соответствии с собственными ценностями и уровнем готовности к риску. Например, запретить использование ИИ для автоматического принятия кадровых решений без участия человека. Или запретить передачу персональных данных клиентов во внешние LLM-сервисы без шифрования и обезличивания. Или ограничить использование генеративного ИИ для создания контента, публикуемого от имени организации, без проверки человеком.

Между полным запретом и безусловным разрешением лежит наиболее важная зона: сценарии, допустимые при соблюдении условий. Организация может разрешить использование внешних LLM для анализа внутренних документов при условии, что документы не содержат персональных данных и коммерческой тайны. Может разрешить развёртывание модели кредитного скоринга при условии прохождения оценки справедливости и утверждения результатов комитетом по рискам. Может разрешить интеграцию агента с корпоративными системами при условии ограничения полномочий до минимально необходимых и подтверждения критических действий человеком. Условия должны быть конкретными и проверяемыми: формулировка «при соблюдении требований безопасности» ничего не запрещает и ничего не разрешает.

Третий блок определяет процедуру утверждения новых сценариев применения. Когда подразделение хочет внедрить новую ИИ-систему или расширить использование существующей, какие шаги оно должно пройти? Минимальный набор обычно включает: описание сценария и его бизнес-обоснование, определение данных, которые система будет обрабатывать, предварительную оценку уровня риска, определение ответственного за систему, согласование с подразделениями безопасности и юридическим отделом. Для систем высокого риска процедура расширяется: полная оценка рисков, оценка воздействия, проверка соответствия регуляторным требованиям, утверждение на уровне руководства. Без формализованной процедуры утверждения политика остаётся декларацией, которую легко обойти.

Четвёртый блок фиксирует правила использования внешних ИИ-сервисов сотрудниками. Это одна из самых острых тем, потому что генеративные модели доступны каждому через браузер, и сотрудники используют их задолго до того, как организация формулирует свою позицию. Политика должна определить, какие внешние сервисы допустимы для рабочих задач. Какие категории данных запрещено вводить в любые внешние системы. Как сотрудник должен проверять результаты, полученные от ИИ, прежде чем использовать их в рабочих документах. Кто несёт ответственность, если сотрудник на основании ответа модели принял решение, причинившее ущерб. Полный запрет на использование внешних ИИ-сервисов редко оказывается работоспособным: сотрудники находят способы его обойти, и организация теряет контроль. Более зрелый подход — определить допустимые инструменты, настроить корпоративные версии с контролем данных и обучить сотрудников правилам безопасного использования.

Пятый блок описывает процедуру обработки отклонений и исключений. В любой организации возникают ситуации, когда конкретный сценарий не укладывается в установленные правила. Подразделение хочет использовать ИИ способом, который не предусмотрен полити-

кой. Или проект требует исключения из запрета на передачу определённых данных во внешний сервис. Политика должна предусмотреть, кто принимает решение об исключении, какие условия должны быть выполнены, как исключение документируется и на какой срок оно действует. Без процедуры обработки исключений политика становится жёсткой конструкцией, которую обходят неформально, и контроль теряется.

Отдельный вопрос — как соотносится политика ИИ с другими политиками организации. ISO/IEC 42001 прямо указывает, что организация должна определить, где другие политики пересекаются с целями управления ИИ. Политика информационной безопасности определяет требования к защите данных и систем, которые применимы и к ИИ. Политика защиты персональных данных определяет правовые основания обработки, которые распространяются на обучающие данные и результаты модели. Политика управления рисками устанавливает процессы оценки и принятия рисков, которые должны охватывать ИИ-специфические категории. Политика закупок определяет требования к поставщикам, которые должны включать оценку безопасности ИИ-сервисов. Политика ИИ не заменяет эти документы, а дополняет их положениями, специфичными для ИИ, и обеспечивает согласованность между ними.

На практике я наблюдаю три типичные ошибки при создании политики ИИ. Первая — слишком общая формулировка. Политика, которая состоит из фраз вроде «организация стремится к ответственному использованию ИИ» и «все системы должны соответствовать требованиям безопасности», не содержит ни одного проверяемого положения. Она существует для отчётности, а не для управления. Вторая ошибка — попытка регламентировать всё. Политика на пятьдесят страниц, охватывающая каждый мыслимый сценарий, становится непрочитаемой и неисполнимой. Сотрудники не читают её, руководители не помнят содержание, и документ превращается в формальность. Третья ошибка — создание политики без механизма её исполнения. Политика запрещает передачу конфиденциальных данных во внешние ИИ-сервисы, но организация не настроила DLP-систему для обнаружения таких передач, не обучила сотрудников и не определила последствия нарушения. Запрет без контроля — это пожелание, а не политика.

Хорошая политика ИИ короткая, конкретная и исполнимая. Она помещается на нескольких страницах, содержит проверяемые положения и поддерживается организационными и техническими контролями. Она пересматривается не реже одного раза в год и при каждом существенном изменении в использовании ИИ, регуляторном окружении или организационной структуре. Она написана языком, понятным сотруднику без технического образования, и дополнена практическими руководствами для конкретных ролей: разработчиков, операторов, пользователей, руководителей проектов.

5.3. Роли и ответственность: от руководства до владельца модели

В предыдущем разделе мы определили, что должна содержать политика ИИ. Но политика работает только тогда, когда каждый её пункт закреплён за конкретным человеком. Кто утверждает новый сценарий применения ИИ? Кто проводит оценку рисков? Кто принимает решение о допустимом уровне риска? Кто отслеживает поведение модели в продуктивной среде? Кто реагирует, когда что-то идёт не так? Если на эти вопросы нет чётких ответов, политика остаётся документом, а управление — намерением.

ISO/IEC 42001 требует, чтобы высшее руководство назначило ответственных за соответствие системы управления ИИ требованиям стандарта и за отчётность перед руководством о результативности этой системы. Стандарт также указывает, что организация должна определить и распределить роли, охватывающие управление рисками, оценку воздействия, безопасность, приватность, разработку, человеческий контроль, управление поставщиками и качество

данных на всём жизненном цикле. Эти требования задают каркас, но конкретное наполнение зависит от масштаба организации, количества ИИ-систем и зрелости существующих процессов.

На уровне руководства организации ключевая задача — определить стратегическое отношение к ИИ и принять на себя ответственность за риски, которые он порождает. Это не может быть делегировано техническому специалисту. Руководитель, санкционирующий внедрение ИИ-системы для принятия решений о клиентах, должен понимать, какие последствия может вызвать ошибка этой системы и какой уровень риска он готов принять. ISO/IEC 42001 подчёркивает, что руководство демонстрирует приверженность управлению ИИ через установление политики, выделение ресурсов, формирование культуры ответственного использования ИИ и поддержку тех, кто отвечает за её реализацию. На практике это означает, что тема ИИ должна регулярно появляться на уровне совета директоров или исполнительного комитета, а не оставаться в зоне видимости только технических подразделений.

Организации используют разные модели для координации управления ИИ. Комитет по ИИ (AI governance board) объединяет представителей бизнеса, ИТ, безопасности, юридического отдела, управления рисками и защиты данных. Он рассматривает новые сценарии применения, утверждает результаты оценки рисков, одобряет исключения из политики и контролирует портфель ИИ-систем. Для крупных организаций с десятками ИИ-проектов комитет может стать узким местом, если каждый проект проходит через него. В таких случаях создают многоуровневую структуру: комитет утверждает системы высокого риска и стратегические решения, а системы низкого и среднего риска проходят через упрощённую процедуру на уровне подразделений.

Центр компетенций по ИИ (AI Center of Excellence) — другая распространённая модель, особенно в организациях с ограниченной экспертизой. Центр аккумулирует знания о технологиях, рисках и лучших практиках, помогает проектным командам проводить оценку рисков, разрабатывает шаблоны и руководства, обучает сотрудников. Он не заменяет распределение ответственности по бизнес-подразделениям, а дополняет его экспертной поддержкой. Преимущество центра компетенций в том, что он создаёт единое пространство знаний. Ограничение в том, что без формальных полномочий он превращается в консультативный орган, рекомендации которого можно проигнорировать.

Ниже стратегического уровня располагаются роли, которые работают с конкретными ИИ-системами. Владелец ИИ-системы (или владелец продукта с ИИ-функциями) отвечает за бизнес-обоснование системы, определяет допустимые сценарии использования и принимает решение о запуске. Он же является основным контактом для вопросов о назначении, границах и ограничениях системы. В большинстве случаев владельцем становится руководитель подразделения, которое извлекает основную ценность из системы. Критически важно, чтобы владелец понимал: вместе с ценностью он принимает на себя риски, включая последствия ошибочных решений модели.

Владелец рисков ИИ-системы — роль, которую мы обозначили в разделе 3.2 как сквозную. На практике она может совпадать с владельцем системы, а может быть выделена отдельно, особенно для систем высокого риска. Владелец рисков отвечает за то, чтобы оценка рисков была проведена, результаты были задокументированы, остаточный риск был осознанно принят на соответствующем уровне, а меры снижения риска были реализованы и контролировались. Если эта роль не назначена явно, ответственность за риски растворяется между участниками, и каждый считает, что за безопасность отвечает кто-то другой.

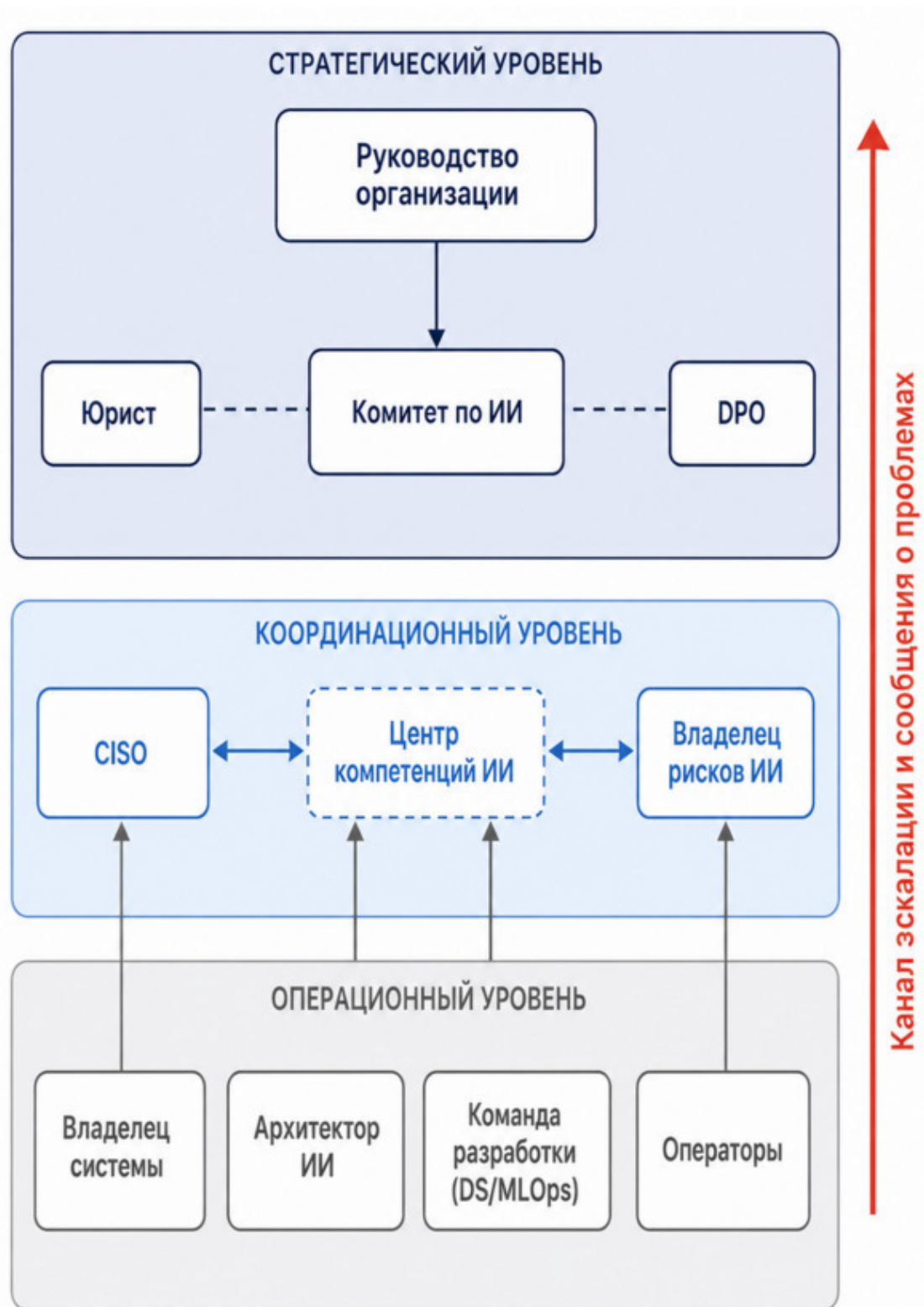


Рисунок 14. Организационная модель управления ИИ: стратегический надзор, владельцы рисков, технические функции, безопасность, соответствие и независимая оценка.

CISO (или руководитель подразделения информационной безопасности) в контексте управления ИИ выполняет несколько функций. Он определяет требования безопасности к ИИ-системам, участвует в оценке рисков, координирует тестирование безопасности, обеспечивает мониторинг и реагирование на инциденты. В организациях, где ИИ-системы внедряются биз-

нес-подразделениями без участия ИТ, CISO часто узнаёт о существовании системы последним. Это одна из причин, по которой процедура утверждения новых сценариев, описанная в предыдущем разделе, должна включать обязательное согласование с подразделением безопасности. Роль CISO в управлении ИИ расширяется: помимо традиционной защиты инфраструктуры и данных, он должен понимать специфические риски моделей, агентов и конвейеров обработки.

Специалист по защите данных (Data Protection Officer, DPO) участвует в оценке того, какие персональные данные обрабатываются ИИ-системой, на каком правовом основании и с какими мерами защиты. Его роль возрастает при дообучении моделей на корпоративных данных, при развёртывании RAG-систем с доступом к документам, содержащим персональные сведения, и при использовании внешних ИИ-сервисов, через которые данные передаются за пределы организации.

На уровне разработки и эксплуатации ключевые роли включают архитектора ИИ-системы, который проектирует архитектуру с учётом требований безопасности; исследователя данных, который готовит обучающие данные и обучает модели; инженера MLOps/LLMOps, который управляет конвейерами обучения и развёртывания; и оператора, который поддерживает систему в продуктивной среде, настраивает мониторинг и обрабатывает инциденты.

Между этими ролями существует напряжение, которое организация должна осознавать и управлять им. Бизнес-заказчик стремится к быстрому запуску и максимальной функциональности. Специалист по безопасности добавляет ограничения, которые замедляют запуск. Юрист требует документации и оценок, на подготовку которых нужно время. Исследователь данных хочет использовать как можно больше данных для повышения качества модели, специалист по защите данных настаивает на минимизации. Эти противоречия нормальны и ожидаемы. Проблемы возникают, когда они разрешаются не через формализованный процесс принятия решений, а через неформальное давление. Если бизнес-заказчик может запустить систему, проигнорировав замечания безопасности, формальная структура ролей обесценивается.

Организация, в которой пять сотрудников и две ИИ-системы, не нуждается в комитете по ИИ и центре компетенций. Ей достаточно назначить владельца каждой системы, определить, кто проводит оценку рисков, и зафиксировать, кто принимает решение о допустимости. Организация с сотнями ИИ-систем в десятках подразделений нуждается в более сложной структуре. Масштаб определяет модель, но принципы остаются общими: каждая ИИ-система должна иметь владельца; ответственность за риски должна быть назначена явно; решения о допустимости должны приниматься с участием безопасности, юристов и бизнеса; и канал для сообщения о проблемах должен быть доступен каждому сотруднику без риска негативных последствий.

Последний пункт заслуживает повторного акцента. ISO/IEC 42001 требует от организации создать механизм сообщения о проблемах, связанных с ИИ-системами, который обеспечивает конфиденциальность, защиту от репрессий и своевременное реагирование. Если оператор замечает, что модель начала выдавать подозрительные результаты, у него должен быть понятный путь для эскалации. Если пользователь считает, что решение системы было дискриминационным, он должен знать, куда обратиться. Без такого механизма проблемы накапливаются, потому что люди, которые их видят, не имеют канала или мотивации сообщить о них.

5.4. Связь управления ИИ с действующими системами управления организации

Организация, приступающая к построению управления ИИ, редко начинает с чистого листа. У неё уже есть система управления информационной безопасностью, процессы управления рисками, программа соответствия требованиям, политики защиты данных, процедуры управления изменениями и закупками. Создание параллельной структуры, которая существует

отдельно от всего этого, — одна из наиболее распространённых и дорогостоящих ошибок. Параллельная система порождает дублирование процессов, конфликтующие требования, разобщённые реестры и ситуацию, когда один и тот же риск оценивается дважды по разным методологиям с разными результатами.

ISO/IEC 42001 проектировался именно для интеграции. Его структура повторяет гармонизированную структуру международных стандартов менеджмента, общую для ISO 27001, ISO 9001, ISO 22301 и других. Это означает, что организация, уже внедрившая ISO 27001, может расширить существующую систему управления, добавив к ней ИИ-специфические элементы, вместо того чтобы строить отдельную конструкцию. Контекст организации, определённый для ISO 27001, дополняется ИИ-факторами. Оценка рисков расширяется категориями, специфичными для ИИ. Внутренний аудит включает проверку ИИ-систем. Управленческий анализ охватывает результативность управления ИИ наравне с информационной безопасностью.

На практике интеграция происходит в нескольких точках, и каждая из них требует осознанного проектирования.

Управление рисками — наиболее очевидная точка соприкосновения. Большинство организаций уже ведут реестр рисков, используют методологию оценки и определяют допустимый уровень. Задача состоит в том, чтобы расширить этот процесс, включив в него риски, специфичные для ИИ: отравление данных, дрейф модели, дискриминационные решения, утечку данных через ответы модели, компрометацию агента. Эти категории не вписываются в привычную таксономию операционных рисков, построенную вокруг сбоев оборудования, ошибок персонала и внешних воздействий. Организации придётся дополнить таксономию или создать ИИ-раздел в существующем реестре. При этом важно сохранить единую шкалу оценки: если ИИ-риски оцениваются по методологии, несовместимой с общей, руководство не сможет сравнить их с другими рисками и расставить приоритеты.

ISO/IEC 23894:2023 предоставляет руководство по управлению рисками ИИ, построенное на основе ISO 31000. Стандарт подчёркивает, что организация должна учитывать особенности ИИ при определении контекста, идентификации рисков, их анализе и оценивании. Он также указывает на необходимость привлечения разнообразных экспертов, потому что риски ИИ выходят за пределы компетенции одной специальности. Оценка рисков модели кредитного скоринга требует участия специалиста по машинному обучению, эксперта по справедливости, юриста и представителя бизнеса. Ни один из них по отдельности не покрывает весь спектр.

Управление информационной безопасностью — вторая точка интеграции. ИИ-системы работают на инфраструктуре, которая уже покрыта контролями ISO 27001: управление доступом, шифрование, мониторинг, управление уязвимостями, реагирование на инциденты. Эти контроли применимы к серверам, сетям и хранилищам, на которых работают ИИ-системы. Но к ним нужно добавить контроли, специфичные для ИИ: защиту обучающих данных от отравления, контроль целостности модели, фильтрацию входных и выходных данных, мониторинг поведения модели, управление полномочиями агентов. Организация может включить эти контроли в существующее заявление о применимости (Statement of Applicability) или создать дополнительное приложение, ссылающееся на контроли из Приложения А к ISO/IEC 42001.

Управление изменениями — третья точка, которую часто упускают. В большинстве организаций изменения в продуктивной среде проходят через процесс утверждения: заявка, оценка воздействия, тестирование, одобрение, развёртывание. Для ИИ-систем этот процесс должен учитывать специфику: обновление модели может изменить поведение системы без изменения кода; обновление базы знаний RAG может повлиять на качество и безопасность ответов; изменение системного промпта может ослабить ограничения. Каждое из этих событий должно проходить через процесс управления изменениями с повторной оценкой рисков, соразмерной масштабу изменения. Организация, которая включает обновление модели поставщиком в свой процесс управления изменениями, обнаруживает неприятный сюрприз: поставщик может

обновить модель без уведомления, и процесс, рассчитанный на контролируемые изменения, не покрывает этот сценарий. Договорные обязательства по уведомлению об обновлениях, обсуждённые в разделе 3.3, становятся необходимым дополнением к техническому контролю.

Управление поставщиками и закупками — четвёртая точка. Организации, которые покупают ИИ-сервисы через API или внедряют платформы оркестрации от сторонних поставщиков, должны включить оценку безопасности ИИ в процесс квалификации поставщиков. Существующие опросные листы для оценки поставщиков, построенные вокруг инфраструктурной безопасности и защиты данных, не покрывают ИИ-специфические вопросы: на каких данных обучена модель, как контролируется справедливость, какие механизмы фильтрации реализованы, как поставщик уведомляет об обновлениях, какие журналы предоставляет. CSA AI Controls Matrix предлагает структуру для оценки поставщиков ИИ-сервисов, включая опросник AI-CAIQ (AI Consensus Assessment Initiative Questionnaire), который организация может использовать или адаптировать к своему контексту.

Управление непрерывностью — пятая точка. Планы непрерывности бизнеса обычно описывают действия при недоступности информационных систем, потере данных или сбое инфраструктуры. Для ИИ-систем к этим сценариям добавляются специфические: недоступность модели из-за сбоя поставщика API, деградация качества ответов из-за дрейфа, компрометация базы знаний RAG, нежелательное поведение агента. Организация должна определить, какие бизнес-процессы зависят от ИИ-систем, и для каждого из них предусмотреть альтернативный режим работы. В некоторых случаях это ручной процесс, который может заменить автоматизированное решение. В других — переключение на резервную модель или резервного поставщика. Включение ИИ-зависимостей в анализ воздействия на бизнес (BIA) — обязательный шаг, который многие организации пока не сделали.

Управление соответствием требованиям — шестая точка. Организации, работающие в регулируемых отраслях, уже ведут учёт применимых требований и контролируют соответствие. С появлением EU AI Act и отраслевых руководств по ИИ этот реестр нужно дополнить. Для каждой ИИ-системы организация должна определить, какие нормативные требования к ней применимы, и отслеживать их выполнение. Это не требует отдельного процесса: расширение существующего реестра требований и включение ИИ-аспектов в программу аудита соответствия интегрирует новые обязательства в действующую систему.

При всей логичности интеграции на практике она сталкивается с сопротивлением. Владельцы действующих процессов могут воспринять расширение как дополнительную нагрузку. Команда управления рисками, привыкшая к определённой таксономии и методологии, может сопротивляться добавлению новых категорий. Подразделение закупок может не понимать, зачем оценивать поставщика ИИ-сервиса иначе, чем поставщика облачной инфраструктуры. Преодоление этого сопротивления требует двух вещей: поддержки руководства, которое делает интеграцию приоритетом, и демонстрации ценности, которая показывает, что интеграция снижает дублирование и повышает качество управления, а не просто добавляет работу.

Для организации практический вывод состоит в следующем. Управление ИИ строится не рядом с существующими процессами, а внутри них. Реестр рисков расширяется ИИ-категориями. Контроли безопасности дополняются ИИ-спецификой. Управление изменениями включает обновления моделей и промптов. Оценка поставщиков покрывает ИИ-вопросы. Планы непрерывности учитывают ИИ-зависимости. Программа соответствия отслеживает ИИ-требования. Каждая из этих интеграций требует усилий, но результат — единая, согласованная система, в которой ИИ-риски управляются с тем же уровнем дисциплины, что и все остальные.

5.5. Документирование решений и подотчётность

Управление без документирования — это управление без памяти. Решения принимаются, но через полгода никто не помнит, почему была выбрана именно эта модель, кто одобрил запуск без полной оценки справедливости и на каких условиях было принято исключение из политики. Когда происходит инцидент, когда приходит запрос регулятора, когда меняется руководитель проекта — организация обнаруживает, что восстановить логику принятых решений невозможно. Документирование замыкает цикл управления: оно превращает решения из устных договорённостей в прослеживаемые артефакты, на которые можно опереться при аудите, расследовании и пересмотре.

ISO/IEC 42001 выделяет документирование как сквозное требование. Организация должна создавать, обновлять и контролировать документированную информацию, необходимую для результативности системы управления ИИ. Стандарт требует документирования политики ИИ, области применения системы управления, результатов оценки рисков и воздействия, целей управления ИИ, решений о применении контролей и обоснования исключений. Но между требованием стандарта и работающей практикой лежит пространство, которое организация должна заполнить сама.

Первый вопрос — что именно документировать. Избыточное документирование создаёт бюрократическую нагрузку, которая замедляет процессы и вызывает сопротивление. Недостаточное — оставляет пробелы, которые обнаруживаются в самый неудобный момент. Практический критерий: документировать нужно каждое решение, которое влияет на профиль риска ИИ-системы и которое кто-то может захотеть проверить, оспорить или пересмотреть.

На уровне ИИ-системы это включает несколько категорий решений. Решение о создании или внедрении: зачем нужна система, какую задачу она решает, какие альтернативы рассматривались, кто инициировал и кто утвердил. Выбор модели и архитектуры: почему выбрана именно эта модель, какой поставщик, какой способ развёртывания, какие ограничения учтены. Результаты оценки рисков: какие риски выявлены, как они оценены, какие меры снижения применены, какой остаточный риск принят и кем. Результаты оценки воздействия: на кого влияет система, каковы потенциальные последствия ошибки, какие меры защиты предусмотрены. Решения о допустимом поведении: какие ограничения наложены на модель, какие сценарии запрещены, при каких условиях требуется подтверждение человеком. Результаты тестирования: какие тесты проведены, какие результаты получены, какие критерии использованы для принятия решения о развёртывании. Исключения из политики: что именно разрешено, на каком основании, кем, на какой срок и при каких компенсирующих мерах.

На уровне эксплуатации документирование охватывает изменения и инциденты. Каждое обновление модели, изменение системного промпта, расширение полномочий агента, добавление нового источника данных в RAG должно быть зафиксировано с указанием того, кто инициировал изменение, кто его утвердил и была ли проведена повторная оценка рисков. Каждый инцидент с участием ИИ-системы должен быть задокументирован: что произошло, когда обнаружено, кто реагировал, какие действия предприняты, каковы результаты расследования и какие меры приняты для предотвращения повторения. Без этих записей организация не может проводить анализ тенденций и не видит, улучшается ли ситуация со временем.

Второй вопрос — в какой форме документировать. Организации, которые уже ведут реестры рисков, журналы изменений и отчёты об инцидентах, могут расширить существующие форматы, добавив ИИ-специфические поля. Для ИИ-систем полезны несколько специализированных артефактов. Карточка модели (model card) фиксирует ключевые характеристики модели: назначение, архитектуру, обучающие данные, метрики качества, известные ограничения, результаты проверки справедливости, условия применения. Карточка системы (system

card) расширяет карточку модели до уровня приложения: описывает архитектуру, интеграции, полномочия, ограничения поведения, контроли безопасности. Паспорт данных (datasheet) описывает обучающий набор: источники, состав, процесс подготовки, известные ограничения и смещения. Эти артефакты не требуют создания с нуля; несколько организаций и исследовательских групп предложили шаблоны, которые можно адаптировать.

Третий вопрос — кто отвечает за документирование и кто его использует. Документация, которую создают, но не читают, не выполняет своей функции. Карточку модели заполняет команда разработки, но её читают специалист по безопасности при оценке рисков, аудитор при проверке системы и оператор при расследовании инцидента. Результаты оценки рисков фиксирует аналитик, но их утверждает владелец рисков и пересматривает комитет по ИИ. Документирование работает, когда каждый артефакт имеет автора, который отвечает за его полноту и актуальность, и аудиторию, которая использует его для принятия решений или контроля.

Четвёртый вопрос — как поддерживать документацию в актуальном состоянии. ИИ-системы меняются быстрее, чем традиционное ПО: модели обновляются, данные дополняются, конфигурации корректируются. Документация, созданная при запуске и не обновлявшаяся, становится недостоверной и опасной, потому что создаёт ложное ощущение контроля. Организация должна определить триггеры обновления: изменение модели, обновление обучающих данных, расширение полномочий, изменение регуляторных требований, результаты аудита, инцидент. Каждый триггер запускает пересмотр соответствующих документов. Встраивание обновления документации в процесс управления изменениями, описанный в предыдущем разделе, — наиболее надёжный способ: изменение не считается завершённым, пока документация не обновлена.



Рисунок 15. Интеграция управления ИИ с корпоративным управлением, рисками, безопасностью, приватностью, закупками, разработкой и внутренним аудитом.

Связь между документированием и подотчётностью прямая. Подотчётность, рассмотренная в разделе 4.3, требует возможности установить, кто принял решение, на каком основании

и с какой информацией. Без документирования эта возможность отсутствует. Когда регулятор спрашивает, почему организация использовала модель, систематически занижающую оценку для определённой демографической группы, организация должна показать: мы провели оценку справедливости (вот результаты), мы определили компенсирующие меры (вот решение), мы назначили ответственного за мониторинг (вот запись), мы проводили регулярные проверки (вот отчёты). Если этих записей нет, организация не может доказать, что действовала добросовестно, даже если она действительно это делала.

Для организаций, работающих в юрисдикции EU AI Act, документирование приобретает нормативный характер. Регламент требует от поставщиков систем высокого риска вести техническую документацию, включающую описание системы, процесс разработки, результаты тестирования и оценки, меры управления рисками, описание данных и процессов мониторинга. Развёртывающие организации обязаны хранить журналы, генерируемые системой, в течение установленного срока. Невыполнение этих требований влечёт санкции. Даже организации, не подпадающие напрямую под EU AI Act, могут столкнуться с аналогичными ожиданиями со стороны партнёров, клиентов и аудиторов, которые ориентируются на европейский стандарт как на эталон.

На практике я наблюдаю два полюса. Одни организации не документируют почти ничего: ИИ-системы запускаются на основании устных согласований, оценка рисков проводится в головах участников, решения об ограничениях принимаются в чатах и теряются в потоке сообщений. Эти организации уязвимы перед любым внешним запросом и не способны провести ретроспективный анализ. Другие организации документируют всё: каждое решение оформляется в многостраничный документ, процедура утверждения занимает недели, и команды начинают обходить процесс, чтобы сохранить скорость. Истина, как обычно, между полюсами. Документирование должно быть соразмерно уровню риска: система, рекомендуемая статьи для чтения, требует минимальной документации; система, влияющая на медицинские решения, требует подробной.

Для организации практический вывод состоит в следующем. Документирование — это не бюрократическое упражнение, а инструмент подотчётности, управления и институциональной памяти. Каждое решение о рисках, ограничениях и допустимости ИИ-системы должно быть зафиксировано в объёме, достаточном для того, чтобы через год можно было восстановить логику принятия решения. Форма может быть компактной: карточка модели, запись в реестре рисков, отметка в журнале изменений. Содержание должно отвечать на три вопроса: что решено, кем и почему. Организация, которая способна ответить на эти вопросы для каждой своей ИИ-системы, готова к аудиту, расследованию и регуляторному запросу. Организация, которая не способна, рискует обнаружить это в момент, когда цена ответа уже включает штрафы, ущерб и потерю доверия.

Источники и дальнейшее чтение к главе 5

— ISO/IEC 42001:2023. Information technology — Artificial intelligence — Management system — гармонизированная структура системы управления ИИ (политика, роли, ресурсы, PDCA-цикл), требования к политике ИИ, ролям и ответственности, документированию, механизму сообщения о проблемах.

— Regulation (EU) 2024/1689 (EU AI Act) — требования к управлению рисками для систем высокого риска, запрещённые сценарии (социальный скоринг, манипулятивные системы, биометрическая идентификация в реальном времени), требования к технической документации и хранению журналов.

— ISO/IEC 23894:2023. Information technology — Artificial intelligence — Guidance on risk management — руководство по управлению рисками ИИ на основе ISO 31000, необходимость привлечения разнородных экспертов при оценке риска.

— ISO 31000:2018. Risk management — Guidelines — базовый стандарт управления рисками, на который опирается ISO/IEC 23894.

— Cloud Security Alliance. AI Controls Matrix (AICM) и AI Consensus Assessment Initiative Questionnaire (AI-CAIQ) — структура оценки поставщиков ИИ-сервисов.

— Mitchell M. и др. Model Cards for Model Reporting. FAT* Conference, 2019 — концепция карточки модели.

— Gebru T. и др. Datasheets for Datasets, 2018 (пересмотрено в Communications of the ACM, 2021) — концепция паспорта данных.

Глава 6. Инвентаризация и классификация ИИ-систем

Предыдущая глава описала, как выстроить управление ИИ: политику, роли, интеграцию с действующими процессами и документирование. Но управлять можно только тем, что видишь. Если организация не знает, сколько ИИ-систем у неё работает, где они развёрнуты, какие данные обрабатывают и кто за них отвечает, любая политика и любое распределение ролей остаются абстракцией. Инвентаризация — это первый практический шаг, который превращает управление из намерения в действие.

Задача осложняется тем, что ИИ-системы появляются в организации не только через формальные проекты. Сотрудник, который использует внешний LLM-сервис для подготовки документов, создаёт ИИ-зависимость, о которой никто не знает. Подразделение, которое встраивает модель в свой рабочий процесс без согласования с ИТ, расширяет периметр неучтённых рисков. Поставщик, который добавляет ИИ-функции в закупленный продукт при очередном обновлении, вводит в среду организации модель, которую никто не оценивал. Все эти ситуации порождают феномен, который получил название теневого ИИ (Shadow AI), и работа с ним требует не только технических, но и организационных мер.

Эта глава описывает, как провести инвентаризацию ИИ-систем, выявить теневой ИИ, построить реестр и классифицировать системы по уровню риска. Результат — полная и актуальная картина ИИ-ландшафта организации, без которой оценка рисков, выбор контролей и демонстрация соответствия требованиям невозможны.

6.1. Реестр ИИ-систем: что учитывать и как вести

ISO/IEC 42001 требует от организации идентифицировать ресурсы, связанные с ИИ-системами, включая вычислительные мощности, данные, модели и человеческие ресурсы. EU AI Act вводит обязательство для поставщиков систем высокого риска регистрировать свои системы в европейской базе данных. Эти требования формируют внешний стимул, но инвентаризация нужна организации прежде всего для собственных целей: невозможно управлять рисками того, чего нет в реестре.

Реестр ИИ-систем — это структурированная запись обо всех ИИ-системах, которые организация разрабатывает, развёртывает, эксплуатирует или использует через внешние сервисы. Он отличается от привычного реестра ИТ-активов тем, что фиксирует не только инфраструктурные компоненты, но и характеристики, специфичные для ИИ: тип модели, происхождение, назначение, обрабатываемые данные, уровень автономности, способ развёртывания и распределение ответственности.

Какие ИИ-системы включать в реестр — первый вопрос, на который организация должна ответить. Очевидный ответ — все. На практике требуется уточнение. Модель машинного обучения, обученная внутри организации и развёрнутая в продуктивной среде, попадает в реестр без сомнений.

Внешний LLM-сервис, вызываемый через API для обработки клиентских обращений, тоже. А как быть с ИИ-функциями, встроенными в закупленное программное обеспечение? С генеративным ИИ, который сотрудники используют через браузер? С моделями, работающими в тестовой среде, но ещё не развёрнутыми в продуктивной? Границу полезно определять через критерий воздействия, установленный в политике ИИ: если система обрабатывает данные организации, влияет на решения, затрагивающие людей, или создаёт обязательства перед регуляторами, она попадает в реестр. Системы в тестовой среде, работающие с синтетическими данными и не связанные с продуктивными процессами, можно учитывать в упрощённом режиме.

AI-SYS-2025-042 Чат-бот клиентской поддержки	
Тип системы	LLM (генерация текста)
Назначение	Ответы на обращения клиентов через чат
Бизнес-процесс	Клиентская поддержка, 1-я линия
Происхождение модели	Внешний API (поставщик: [название])
Способ развёртывания	Облако поставщика
Данные	Персональные данные клиентов, история обращений
Интеграции	CRM (чтение), база знаний (RAG), почта (отправка)
Уровень автономности	Рекомендует с проверкой оператором
Владелец системы	Руководитель отдела поддержки
Владелец рисков	CISO
Уровень риска	Средний
Регуляторная категория	EU AI Act: ограниченный риск (прозрачность)
Дата развёртывания	15.03.2025
Последняя оценка рисков	10.01.2026
Следующая оценка	10.07.2026
Статус: В эксплуатации Последнее обновление модели: 02.12.2025	

Рисунок 16. Пример карточки реестра ИИ-системы: назначение, владелец, данные, модель, поставщики, уровень риска, ограничения и статус оценки.

Для каждой ИИ-системы в реестре следует фиксировать набор атрибутов, которые позволяют оценить риски, определить ответственных и отследить изменения. Идентификатор и название системы — кажется очевидным, но в организации, где десятки команд используют разные модели, отсутствие единой номенклатуры быстро создаёт путаницу. Назначение

и бизнес-процесс: какую задачу решает система и в каком контексте. Тип системы: LLM, компьютерное зрение, предиктивная модель, мультимодальная система, агент — классификация из второй главы. Происхождение модели: собственная разработка, дообученная внешняя модель, внешний сервис через API, модель из открытого репозитория. Способ развёртывания: облако поставщика, собственная инфраструктура, гибридный вариант, периферийное устройство. Данные: какие категории данных обрабатывает система, содержат ли они персональные сведения, коммерческую тайну или иную чувствительную информацию. Интеграции: с какими корпоративными системами связана, через какие протоколы, с какими полномочиями. Владелец системы и владелец рисков. Уровень автономности: система рекомендует, система решает с подтверждением человеком, система действует автономно. Дата развёртывания и дата последнего обновления. Результаты оценки рисков: уровень риска, принятый остаточный риск, дата последней оценки.

Перечень атрибутов может показаться длинным, но большинство из них заполняются один раз при включении системы в реестр и обновляются при изменениях. Организация, которая пытается собрать всю информацию одновременно для всех систем, рискует увязнуть в проекте инвентаризации на месяцы. Более работоспособный подход — начать с минимального набора (название, назначение, тип, владелец, уровень риска) и расширять записи постепенно, начиная с систем высокого риска. Реестр, заполненный на 70% для всех систем, полезнее, чем реестр, заполненный на 100% для трёх систем из двадцати.

Ведение реестра — процесс, а не разовое мероприятие. Новые ИИ-системы должны включаться в реестр до развёртывания, а не после. Процедура утверждения новых сценариев, описанная в разделе 5.2, должна включать обязательную регистрацию в реестре как условие запуска. Изменения в существующих системах — обновление модели, расширение полномочий, изменение обрабатываемых данных — должны отражаться в реестре через процесс управления изменениями. Вывод системы из эксплуатации должен сопровождаться обновлением статуса в реестре, а не молчаливым исчезновением записи. Без этих процессов реестр устаревает в течение нескольких месяцев и превращается в исторический документ, не отражающий действительность.

Отдельная задача — определить, где вести реестр. Некоторые организации расширяют существующий реестр ИТ-активов или CMDB (Configuration Management Database), добавляя ИИ-специфические поля. Другие создают отдельный реестр, связанный с CMDB через ссылки. Третьи используют специализированные платформы для управления ИИ-портфелем. Выбор инструмента менее важен, чем три принципа: реестр должен быть единым (одна запись для каждой системы, а не дублирование в нескольких подразделениях), доступным (ответственные за безопасность, управление рисками и соответствие могут получить информацию без запроса к владельцу) и актуальным (процессы обновления встроены в жизненный цикл системы).

На практике самым сложным оказывается не выбор формата и не определение атрибутов, а обнаружение систем, которые уже работают, но о которых центральные подразделения не знают. Об этом — в следующем разделе.

6.2. Теневой ИИ: как обнаружить и взять под контроль

Теневой ИИ (Shadow AI) — это использование ИИ-систем и сервисов в организации без ведома, согласования или контроля со стороны подразделений, ответственных за информационные технологии, безопасность и управление рисками. Термин образован по аналогии с теневыми ИТ (Shadow IT), но масштаб проблемы и скорость её нарастания существенно выше. Облачный сервис для хранения файлов, подключённый сотрудником без согласования, создавал риск утечки данных. Генеративная модель, в которую сотрудник вводит конфиденциальные документы организации для получения сводки, создаёт тот же риск, но с двумя отличиями:

данные передаются третьей стороне мгновенно и могут быть использованы для дообучения модели, то есть встроены в её параметры навсегда.

Причины возникновения теневого ИИ понятны. Генеративные модели доступны любому сотруднику через браузер без установки программного обеспечения и без согласования с ИТ. Они дают ощутимый результат за секунды: черновик документа, анализ таблицы, ответ на сложный вопрос, перевод текста. Сотрудник, который обнаружил, что внешний ИИ-сервис экономит ему час работы ежедневно, не будет ждать, пока организация сформулирует политику и выберет корпоративное решение. Он начнёт использовать сервис сейчас, и вопросы безопасности для него вторичны, потому что он не видит рисков, которые видит специалист по ИБ.

К индивидуальному использованию добавляется подразделенческое. Маркетинг подключает генеративную модель для создания контента. HR использует внешний сервис для предварительного анализа резюме. Юридический отдел загружает контракты во внешнюю LLM для быстрого анализа условий. Каждое подразделение решает свою задачу и не считает нужным информировать ИТ или безопасность, потому что воспринимает ИИ-сервис как рабочий инструмент, аналогичный поисковой системе. Между тем данные, которые попадают в эти сервисы, могут включать персональные сведения клиентов, коммерческую тайну, стратегические планы и юридически привилегированную информацию.

Третий источник теневого ИИ — встроенные ИИ-функции в закупленных продуктах. Поставщик офисного пакета добавляет ИИ-ассистента при очередном обновлении. Платформа для управления проектами интегрирует генеративную модель для автоматического создания отчётов. CRM-система получает функцию предиктивного анализа клиентской базы. Эти функции активируются по умолчанию или включаются администратором подразделения без оценки рисков. Организация может не знать, что её данные обрабатываются ИИ-моделью, встроенной в продукт, который она использует уже несколько лет.

Обнаружение теневого ИИ требует сочетания технических и организационных мер. Ни одна из них по отдельности не даёт полной картины.

Технические методы обнаружения начинаются с анализа сетевого трафика. Обращения к известным ИИ-сервисам (API-эндпоинты поставщиков LLM, домены генеративных платформ) можно выявить через прокси-серверы, системы мониторинга трафика и DLP-решения. Это позволяет обнаружить случаи, когда сотрудники обращаются к внешним ИИ-сервисам с корпоративных устройств. Ограничение метода в том, что он не покрывает использование с личных устройств, через мобильные приложения или через VPN, а также не выявляет ИИ-функции, встроенные в уже разрешённые сервисы.

Анализ закупок и подписок помогает обнаружить подразделенческий теневого ИИ. Если подразделение оплачивает подписку на ИИ-сервис через корпоративную карту или запрашивает бюджет на «инструменты повышения продуктивности», это может указывать на использование ИИ, не прошедшего оценку. Финансовый контроль закупок, дополненный ключевыми словами, связанными с ИИ, создаёт ещё один канал обнаружения.



Рисунок 17. Три основных канала появления теневого ИИ: самостоятельное использование публичных сервисов, неучтённые интеграции и встроенные функции корпоративных продуктов.

Аудит установленного ПО и браузерных расширений выявляет клиентские приложения ИИ-сервисов и плагины, которые сотрудники устанавливают на рабочие станции. Многие ИИ-ассистенты работают как расширения браузера и перехватывают содержимое веб-страниц, электронных писем и документов, открытых в браузере. Организация может обнаружить, что десятки сотрудников используют браузерные расширения, которые передают содержимое экрана внешнему ИИ-сервису для анализа.

Проверка конфигураций закупленных продуктов позволяет выявить ИИ-функции, активированные в существующих системах. Это требует систематического пересмотра продуктов, используемых организацией, на предмет появления ИИ-компонентов в последних обновлениях. Задача непростая, потому что поставщики добавляют ИИ-функции в свои продукты непрерывно, и каждое обновление может изменить способ обработки данных.

Организационные методы обнаружения не менее важны. Опрос подразделений — прямой способ выяснить, какие ИИ-инструменты используются на местах. Опрос работает лучше, когда он проводится в формате помощи, а не контроля: «мы хотим понять, какие инструменты вам полезны, чтобы предложить безопасные корпоративные альтернативы» вместо «мы проверяем, не нарушаете ли вы политику». Угрожающий тон загоняет использование в тень ещё глубже.

Канал добровольного сообщения, описанный в разделе 5.3, тоже работает на обнаружение. Если сотрудники знают, что могут сообщить об использовании ИИ-инструмента без негативных последствий, часть теневого ИИ выявляется через самодекларацию.

Для этого необходима культура, в которой использование нового инструмента воспринимается как инициатива, заслуживающая оценки, а не как нарушение, заслуживающее наказания.

Обнаружив теневой ИИ, организация стоит перед выбором: запретить, легализовать или заменить. Полный запрет редко оказывается устойчивым решением. Если сотрудник получает ощутимую пользу от ИИ-инструмента, запрет без предложения альтернативы вызывает сопротивление и обход. Более зрелый подход — оценить обнаруженные сценарии использования, определить, какие из них создают приемлемый риск при соблюдении условий, и предложить корпоративную альтернативу для остальных.

Корпоративные версии ИИ-сервисов с контролем данных, ограничением на использование запросов для дообучения, журналированием и интеграцией с системами безопасности организации — это вариант, который всё чаще выбирают зрелые организации. Он позволяет сохранить продуктивность сотрудников, взяв под контроль потоки данных. Стоимость корпоративной подписки сопоставима с потерями от одного инцидента утечки данных через неконтролируемый внешний сервис.

Для сценариев, которые создают неприемлемый риск (передача персональных данных клиентов, использование ИИ для принятия решений о людях без человеческого контроля, обработка юридически привилегированной информации), запрет должен быть подкреплён техническими контролями: блокировка доступа к определённым сервисам, DLP-правила, контроль браузерных расширений. Запрет, существующий только в тексте политики и не подкреплённый техническими мерами, остаётся пожеланием.

Теневой ИИ — это не разовая проблема, которую можно решить одной кампанией инвентаризации. Новые ИИ-сервисы появляются ежемесячно. Поставщики встраивают ИИ в существующие продукты при каждом обновлении. Сотрудники находят новые инструменты и применяют их в работе. Организация должна выстроить непрерывный процесс обнаружения, который сочетает технический мониторинг, периодические опросы подразделений, контроль закупок и пересмотр конфигураций закупленных продуктов. Этот процесс интегрируется с ведением реестра, описанным в предыдущем разделе: каждая обнаруженная система проходит через процедуру оценки и либо включается в реестр с соответствующими контролями, либо блокируется с предложением альтернативы.

6.3. Классификация по уровню риска: критерии и методы

Реестр даёт организации видимость. Классификация даёт приоритеты. Когда в реестре двадцать ИИ-систем и ресурсы ограничены, организация не может применить одинаковый уровень контролей ко всем. Чат-бот, рекомендующий статьи для внутренней базы знаний, и модель, оценивающая кредитоспособность заёмщиков, создают принципиально разные уровни риска и требуют принципиально разного внимания. Классификация по уровню риска определяет, какие системы проходят полную оценку рисков и тестирование безопасности, какие требуют утверждения на уровне руководства, а какие могут быть развёрнуты по упрощённой процедуре.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.