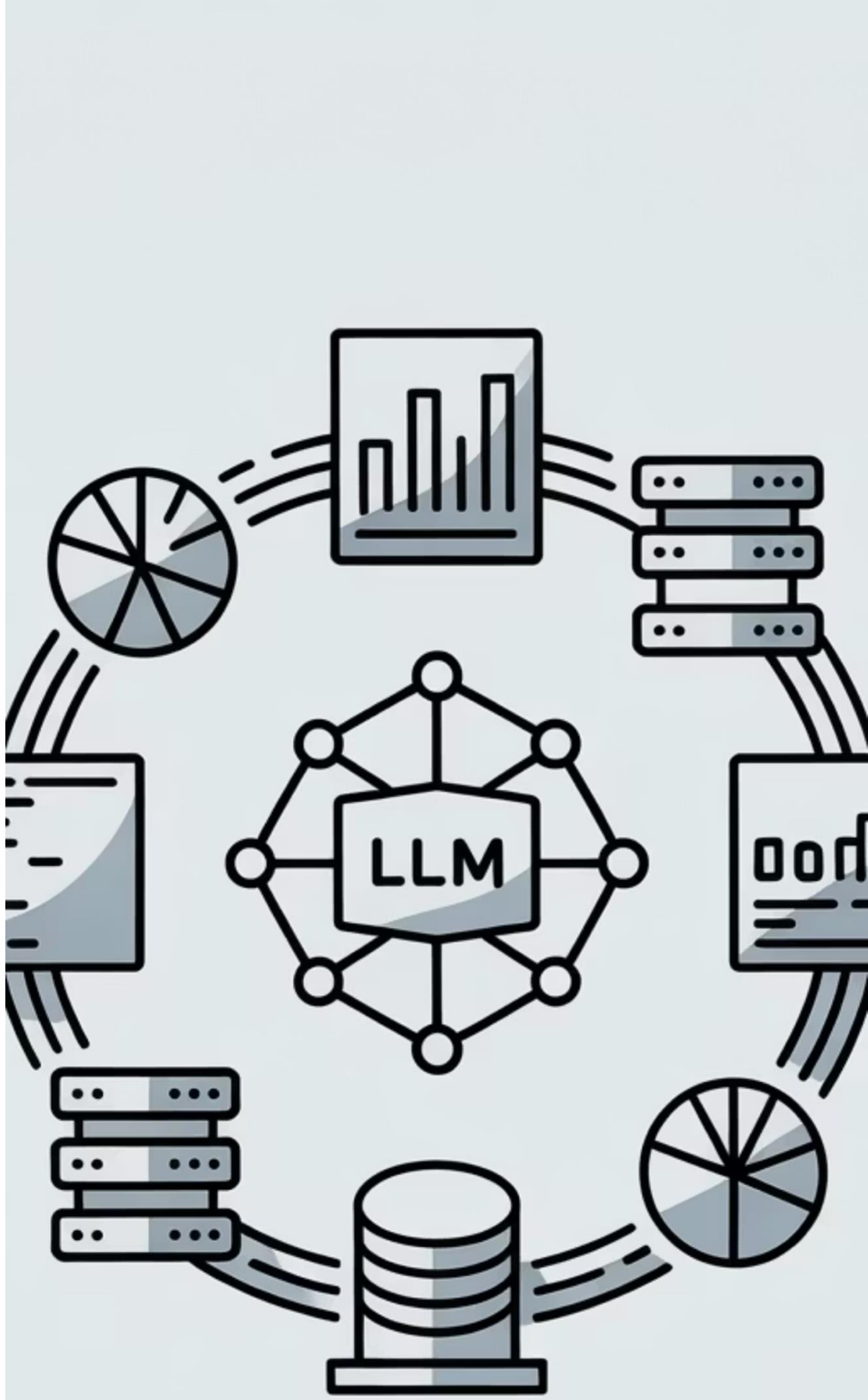
Валидация применения LLM в индустриальных кейсах

LLM в продакшне: не игрушка, а критический компонент инфраструктуры



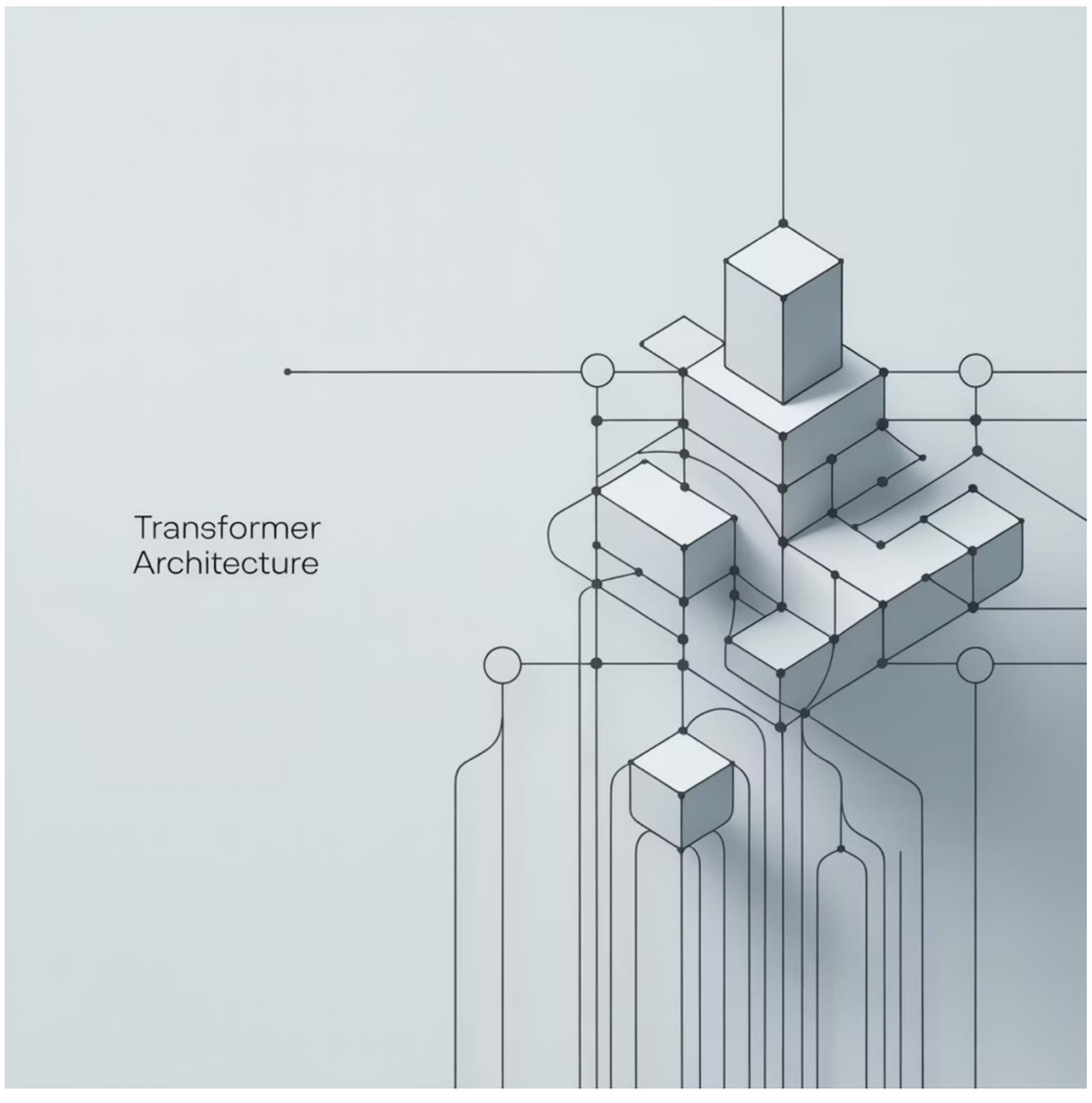
Что такое LLM в современном контексте

Основы и характеристики больших языковых моделей

Большие языковые модели

LLM (Large Language Models) — это не энциклопедии с готовыми ответами, а генераторы правдоподобных текстов на основе вероятностных распределений. Они обучаются на огромных объемах текстовых данных и способны создавать человекоподобные ответы.

- GPT-4, Claude, Gemini — лидеры рынка
- Миллиарды параметров
- Генеративный характер работы



Применение LLM в финансовых технологиях

Как большие языковые модели преобразуют финансовый сектор

Чат-боты поддержки

Автоматизация ответов на типовые вопросы клиентов о банковских продуктах, тарифах и услугах

- Консультации по вкладам
- Информация о кредитах
- Техническая поддержка

Аналитическая отчетность

Генерация финансовых отчетов и аналитических обзоров на основе структурированных данных

- Обзоры рынков
- Риск-анализ
- Инвестиционные рекомендации

Антифрод системы

Анализ транзакций и выявление подозрительных операций через обработку естественного языка

- Детекция аномалий
- Классификация угроз
- Анализ поведения

Валидация как основа надежности

Обеспечение надежности и безопасности LLM-систем в финансах

Валидация LLM — это совокупность методов, которые позволяют измерять, контролировать и гарантировать корректность, стабильность и безопасность ответов модели.

> **"Validation is not bureaucracy — it's survival."**
>
> В финансовой индустрии каждая ошибка может стоить миллионы. Валидация — это не бюрократическая процедура, а жизненно важный процесс защиты бизнеса.

ПРОБЛЕМЫ LLM В ПРОДАКШНЕ

Четыре критические угрозы для финансовых систем

Прежде чем говорить о решениях, важно понять, с какими проблемами мы сталкиваемся при использовании LLM в критически важных системах.

Проблема #1: Галлюцинации

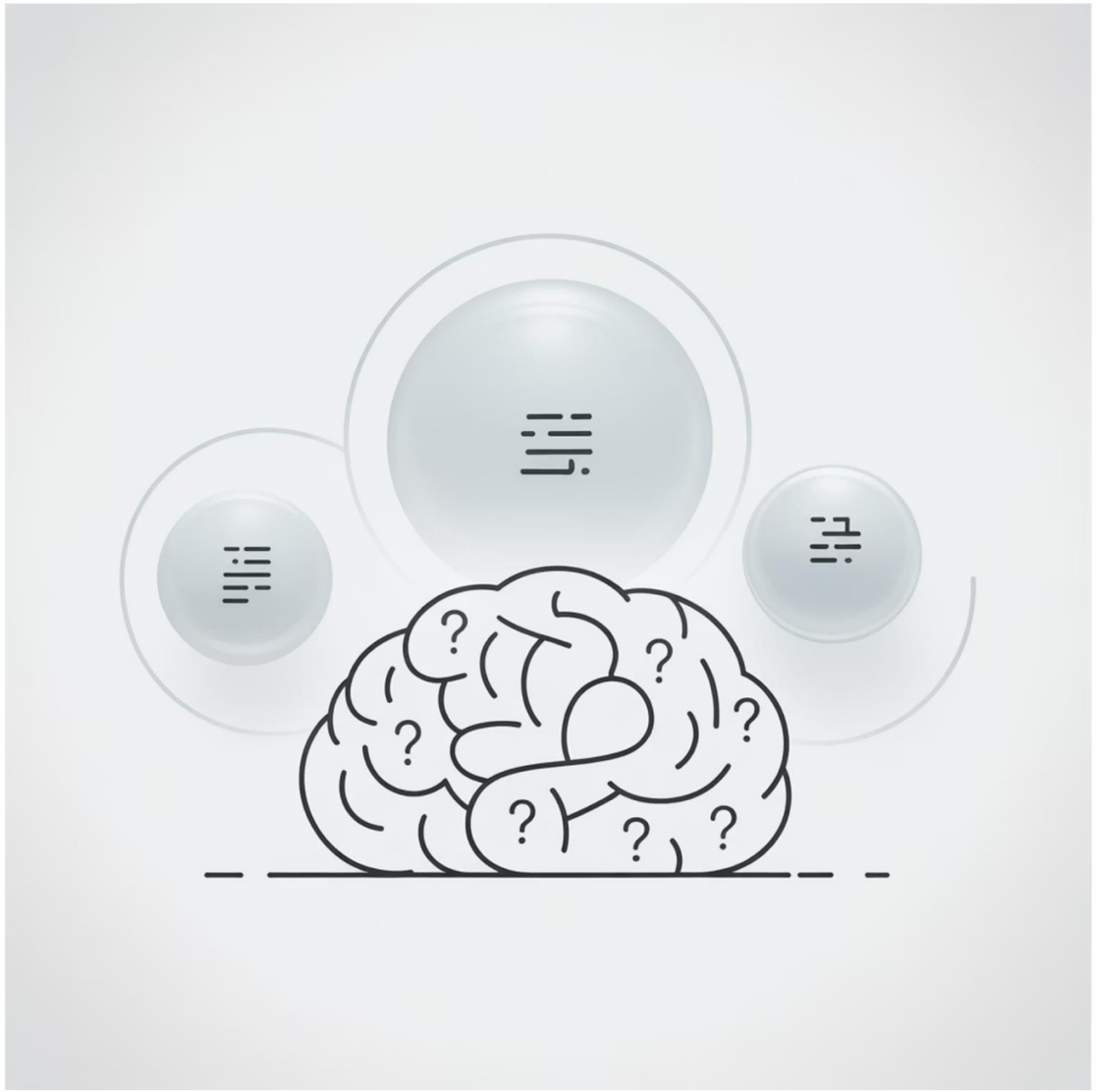
Когда LLM выдает ложную информацию, это может привести к серьезным финансовым рискам.

Что такое галлюцинации LLM

Галлюцинации — это ситуации, когда модель выдает факты, которых не было в обучающих данных, либо неверные утверждения, представляя их как достоверную информацию.

Причины возникновения:

- Недостаточная контекстуальность
- Генеративный характер архитектуры
- Отсутствие проверки источников



Пример в финтехе: LLM назвала несуществующую компанию "ФинКор", придумала для нее баланс в 2.5 млрд рублей, вместо того чтобы взять из базы данных информацию, и рекомендовала инвестиции. Проблема — банка "ФинКор" не существует.

Проявления галлюцинаций в финтехе

Неверные IBAN коды

Модель генерирует валидные по формату, но несуществующие банковские реквизиты

Ложные финансовые даты

Указывает даты отчетов, собраний акционеров или других событий, которые не происходили

Фиктивные ссылки на документы

Создает правдоподобные ссылки на регуляторные документы или отчеты, которые не существуют

Выдуманные финансовые показатели

Генерирует реалистичные цифры прибыли, выручки или капитализации для реальных компаний

IBAN – International Bank Account Number, Международный номер банковского счета

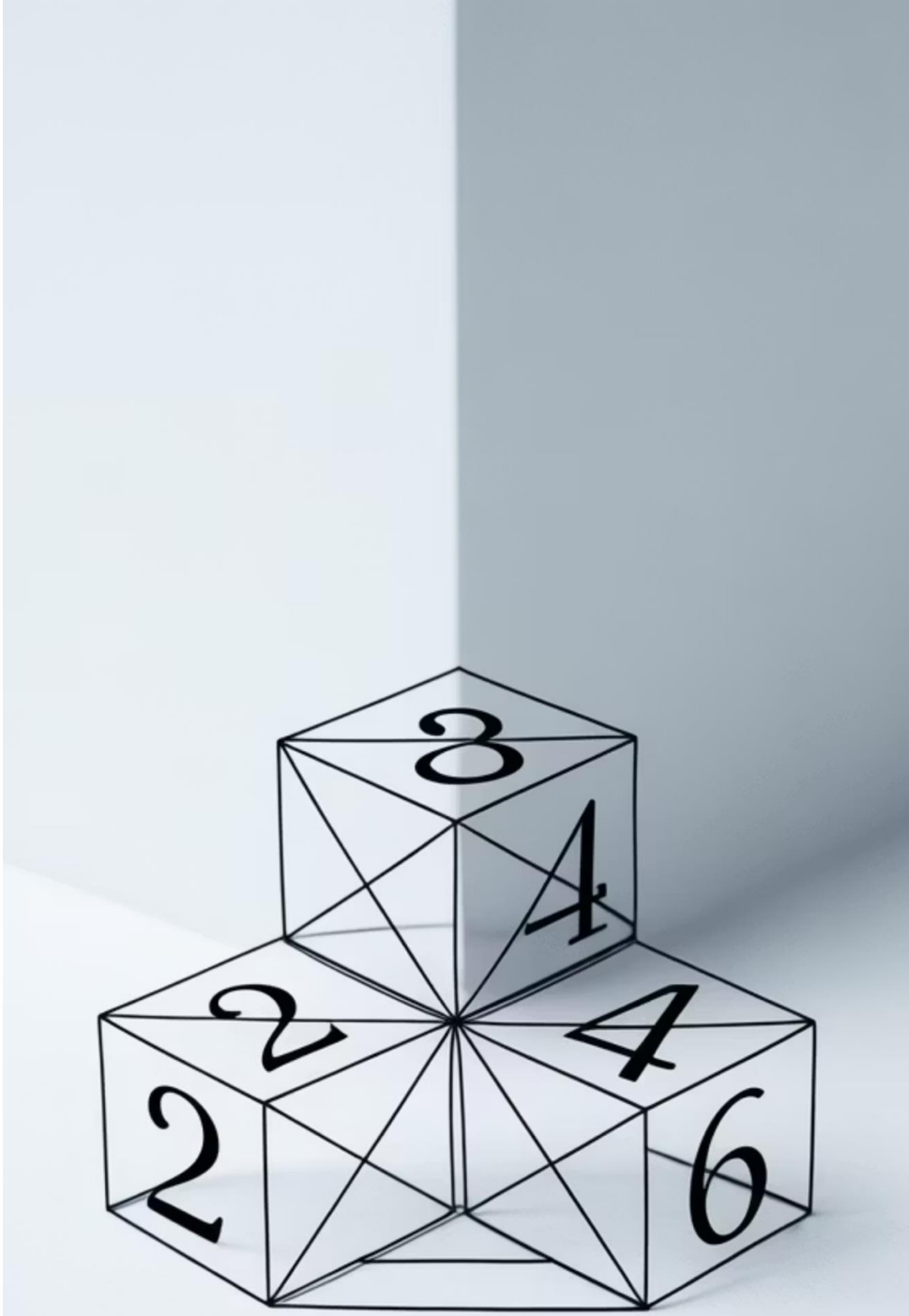
Проблема #2: Нестабильность ответов

Один и тот же запрос при многократном прогоне может давать разные ответы. Это фундаментальная особенность генеративных моделей, связанная с их стохастической природой.

Причины нестабильности:

- Стохастичность (выборка, temperature parameter)
- Особенности токенизации
- Внутренние состояния модели

Влияние на системы: Трудно тестировать, невозможно предсказать поведение, сложность воспроизведения ошибок.



Проблема #3: Смещение и предвзятость

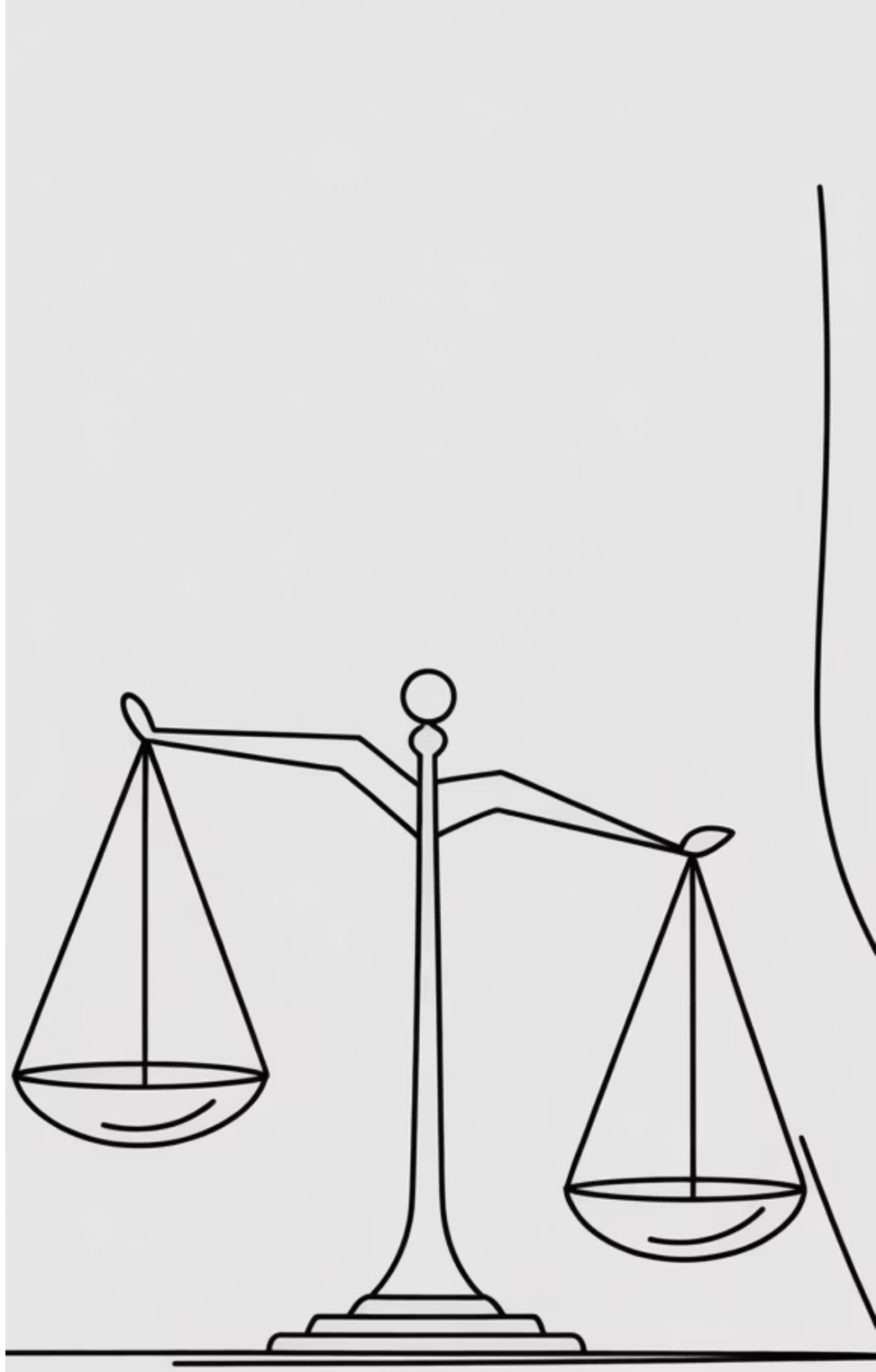
LLM могут демонстрировать системные смещения, унаследованные из обучающих данных. Особенно критично для финансовых решений, затрагивающих кредитование и оценку рисков.

Дрифт данных

ИИ живет в прошлом — модель обучена на исторических данных, а мир уже изменился. Новые экономические реалии не отражены в знаниях модели.

Дискриминационные смещения

В финтехе: дискриминация при кредитовании по половому, расовому или региональному признаку. Модель хуже оценивает заявки с определенными именами или из конкретных регионов.



Проблема #4: Уязвимости безопасности

Различные методы эксплуатации и атак на языковые модели.

Внедрение промптов

Пользователь внедряет управляющие инструкции через обычный ввод, заставляя модель игнорировать изначальные правила и выполнять нежелательные действия.

Утечка данных

Модель случайно воспроизводит фрагменты обучающих данных, включая потенциально конфиденциальную информацию клиентов или внутренние документы.

Jailbreak атаки

Специальные сценарии, разработанные для обхода встроенных ограничений модели и получения запрещенного контента или функциональности.

Adversarial атаки

Намеренно сконструированные входные данные, призванные заставить модель совершать ошибки или выдавать неправильные результаты.

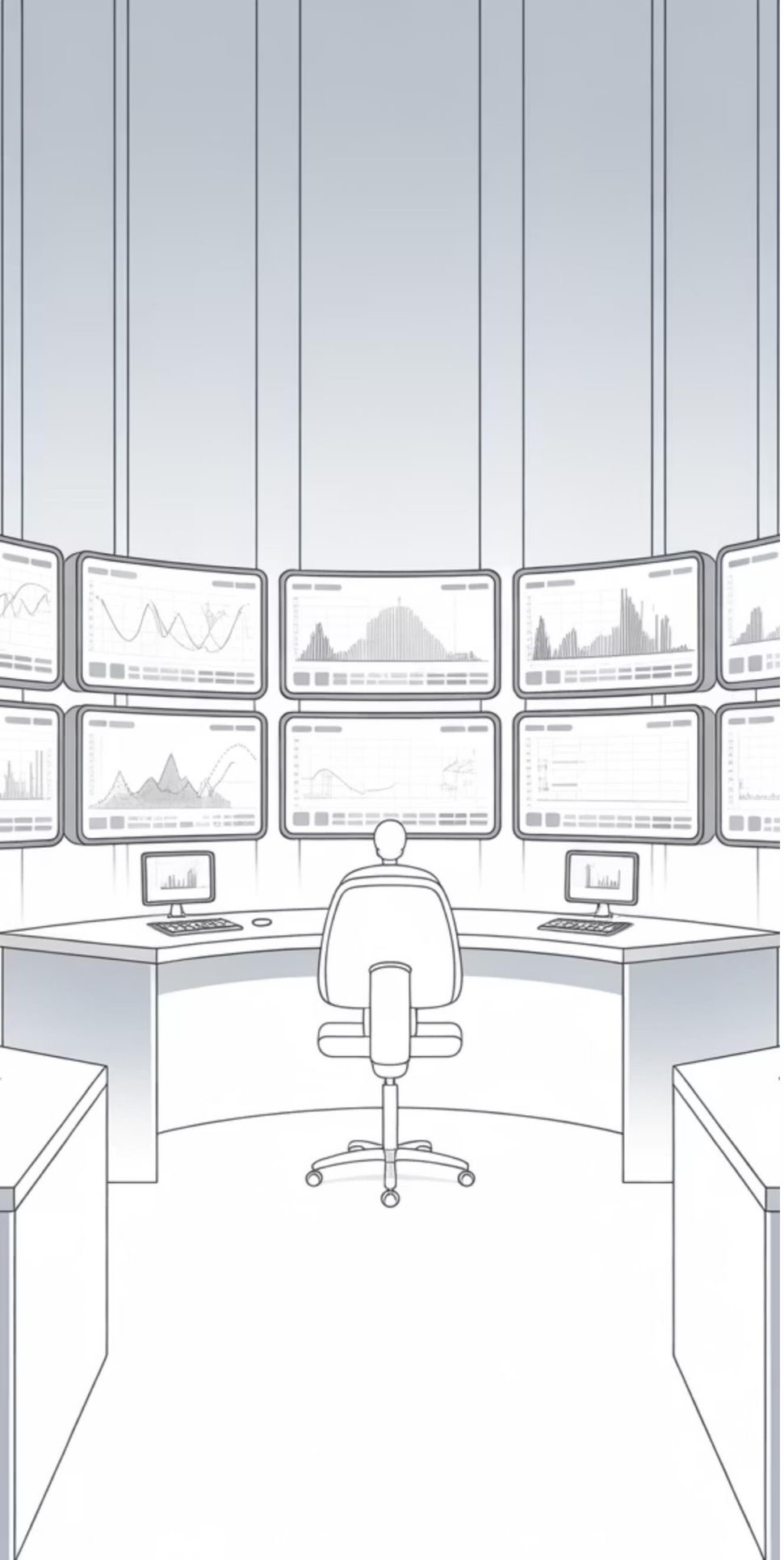
Риски LLM в финансовом секторе

Критичность проблем в финансах

> Вопрос для размышления
>
> Что опаснее для финтеха — галлюцинация или нестабильность?

Все проблемы LLM критичны в финансовом контексте, но по-разному:

- **Галлюцинации** — прямой ущерб от неверных решений
- **Нестабильность** — невозможность воспроизвести и исправить ошибки
- **Смещения** — регуляторные риски и репутационный ущерб
- **Уязвимости** — компрометация системы и данных клиентов



Методы детектирования проблем

Как проверять?

Теперь, когда мы поняли, что именно ломается в LLM, давайте рассмотрим методы обнаружения и контроля этих проблем. Эффективная валидация требует комплексного подхода.

Метод #1: Тестовые корзины (Бенчмарки)

Бенчмарки и краевые случаи

Тестовая корзина — это набор заранее подготовленных запросов с правильными ответами или четкими ожиданиями по формату и содержанию.

Типы корзин:

- **Базовые (sanity-check):** проверка что ничего не поломалось от нового релиза модели
- **Продвинутые (advanced):**более сложные бенчмарки проверяющие более комплексные навыки модели
- **Актуальные (frontier) :** запросы, зачастую направленные на проверку агентных навыков, наиболее актуальных навыков для модели (зависит от сценария ее использования)
- **Доменные/сценарные :** корзины состоящие из реальных кейсов применения модели (напр., внутри компании)

Пример применения: 50 шаблонных запросов к банковскому чат-боту: "сколько стоит международный перевод", "условия депозитного вклада", "как закрыть карту".

Обзор бенчмарков LLM

Sanity-check (Базовая проверка)

Проверка фундаментальных навыков: логики, базовых знаний и работы с контекстом

Примеры:

- **GSM8k:** задачи по математике уровня начальной школы (арифметика и логика)
- **NiH (Needle In A Haystack):** поиск конкретного факта в огромном массиве текста
- **IFEval:** оценка строгого следования инструкциям

Advanced (Продвинутый уровень)

Сложные задачи, требующие экспертных знаний и многоступенчатых рассуждений

Примеры:

- **SBS/MERA-Text/MERA-Industrial:** оценка глубоких знаний в текстовых и индустриальных доменах
- **MERA Code/LiveCodeBench:** проверка навыков программирования на реальных задачах
- **MMLU-Pro:** усложненная версия классического теста с 10 вариантами ответов
- **GPQA Diamond:** научные вопросы уровня PhD (биология, физика, химия)
- **Big Bench Hard (BBH):** набор из 23 сложнейших задач
- **FinanceQA/MATH-500:** продвинутая финансовая аналитика и олимпиадная математика
- **BFCL:** оценка работы модели с внешними инструментами (API)
- **LIBRA:** оценка понимания контекста и логического вывода на русском языке

Frontier

Тесты для самых комплексных навыков моделей

Примеры:

- **HLE:** сверхсложный экзамен, требующий высшего человеческого понимания
- **AIME 2024/2025:** задачи американской математической олимпиады
- **SWE-bench Verified:** решение реальных проблем в коде (GitHub issues)
- **Natural Plan/Agent Bench:** способность модели планировать действия и работать как автономный агент
- **ReflectionBench:** тест на умение модели находить и исправлять собственные ошибки

Банковское / Применения GC

Специализированные тесты для финансовых организаций и бизнес-сценариев

Примеры:

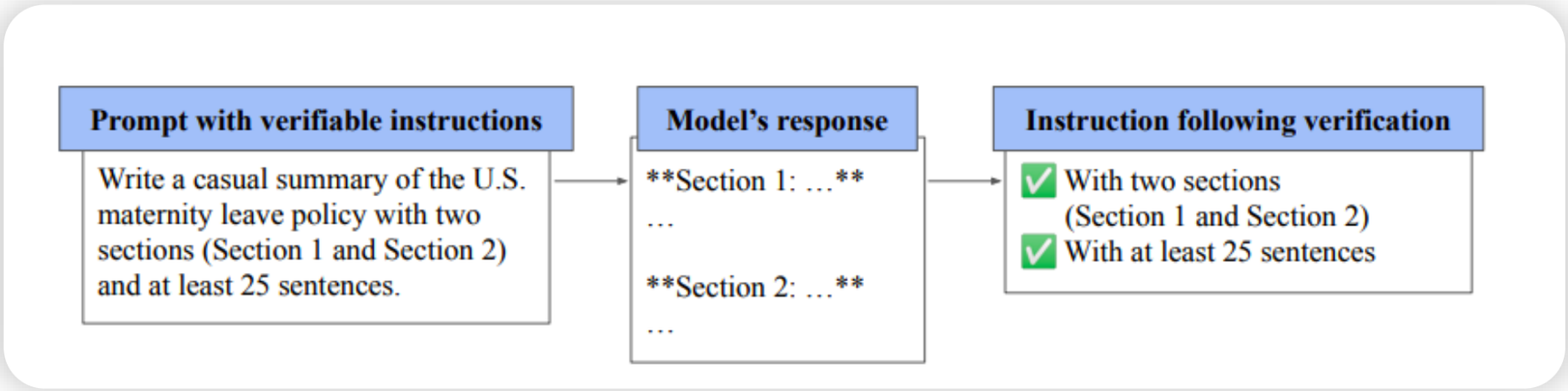
- оценка качества генерации ответов в закрытых корпоративных доменах
- точность определения намерений пользователя для чат-ботов
- проверка работы системы с поиском информации по базе знаний (RAG)

IFEval, NiH, GSM8K, MERA, MMLU-Pro, GPQA, BBH/BBEH, FinanceQA, MATH-500, BFCL, LIBRA, HLE, AIME, SWE-Bench, Naturalplan, AgentBench, ReflectionBench

Обзор бенчмарков LLM

Sanity-check (Базовая проверка)

- **IFEval:** оценка строгого следования инструкциям



Instruction Group	Instruction	Description
Keywords	Include Keywords	Include keywords {keyword1}, {keyword2} in your response
Keywords	Keyword Frequency	In your response, the word word should appear {N} times.
Keywords	Forbidden Words	Do not include keywords {forbidden words} in the response.
Keywords	Letter Frequency	In your response, the letter {letter} should appear {N} times.
Language	Response Language	Your ENTIRE response should be in {language}, no other language is allowed.
Length Constraints	Number Paragraphs	Your response should contain {N} paragraphs. You separate paragraphs using the markdown divider: * * *
Length Constraints	Number Words	Answer with at least / around / at most {N} words.
Length Constraints	Number Sentences	Answer with at least / around / at most {N} sentences.
Length Constraints	Number Paragraphs + First Word in i-th Paragraph	There should be {N} paragraphs. Paragraphs and only paragraphs are separated with each other by two line breaks. The {i}-th paragraph must start with word {first_word}.

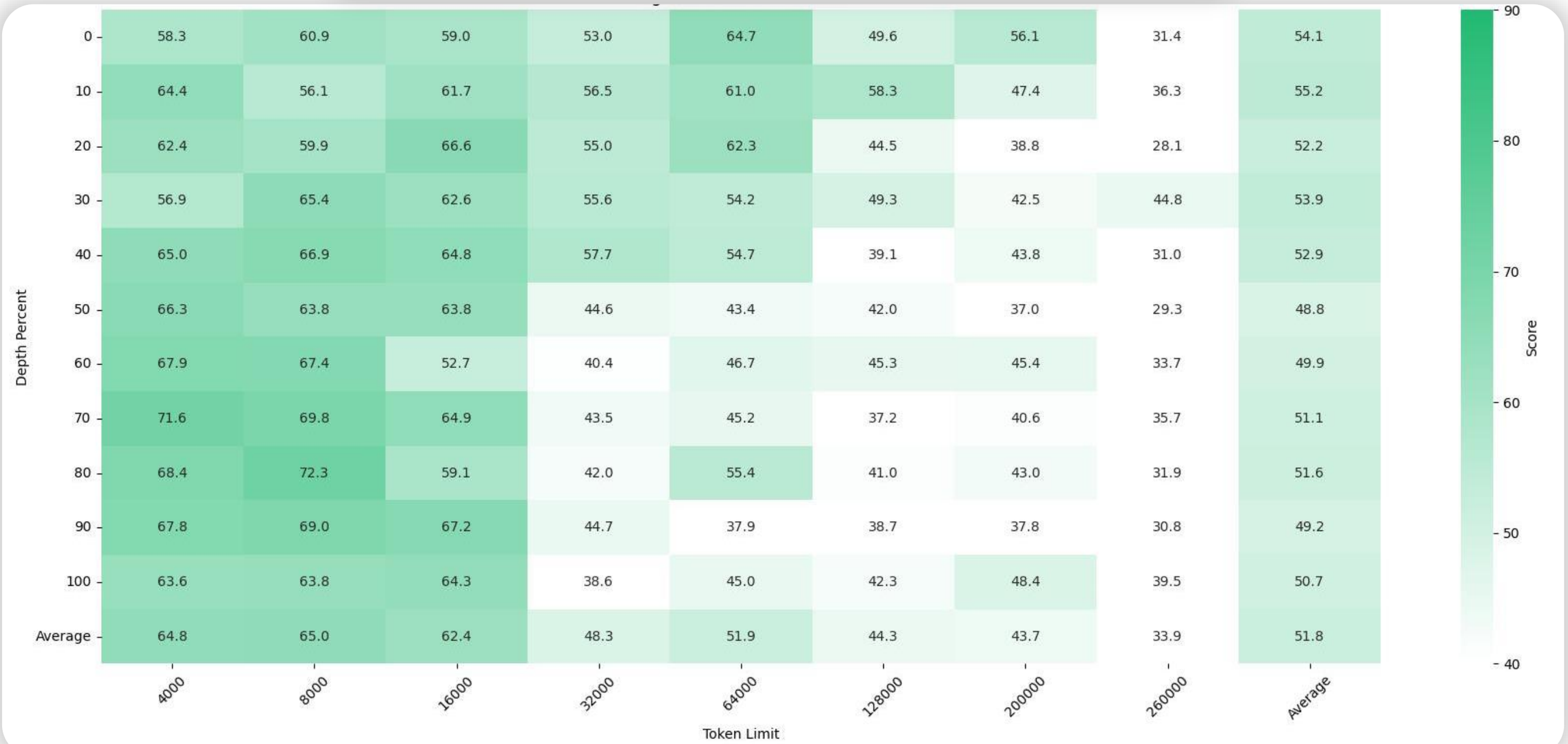
Обзор бенчмарков LLM

Sanity-check (Базовая проверка)

- **NiH (Needle In A Haystack):** поиск конкретного факта в огромном массиве текста

Example

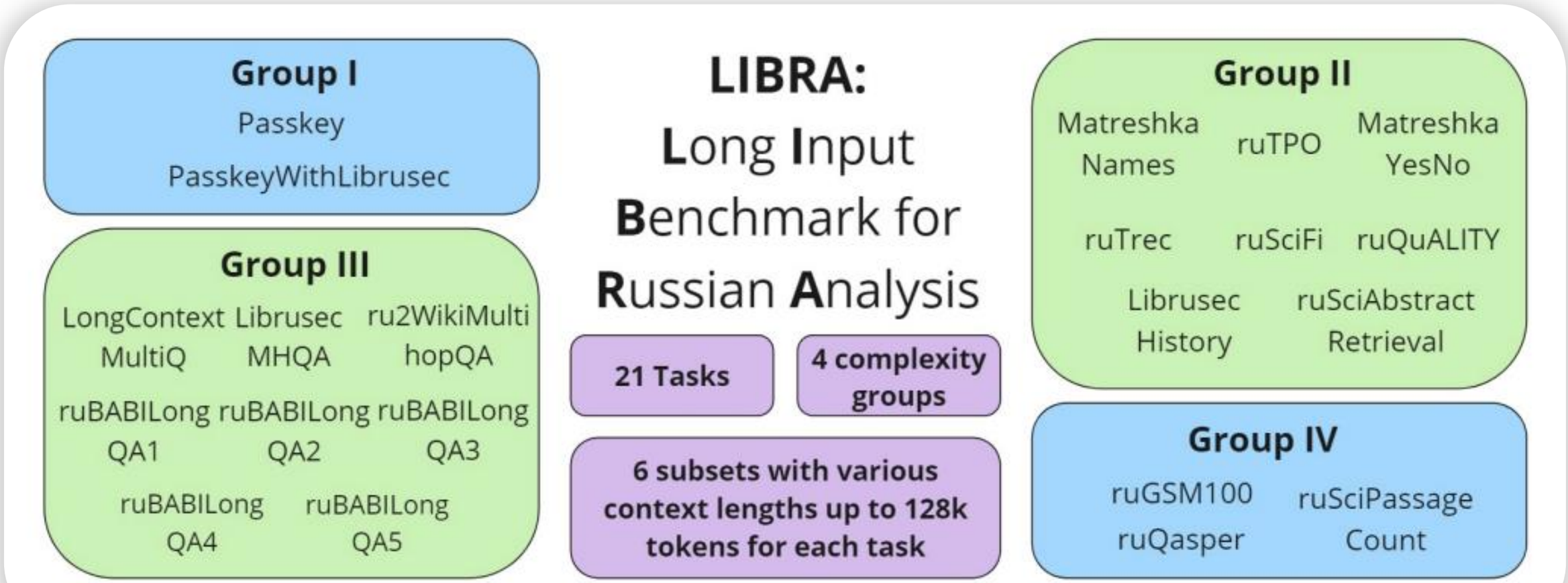
(essays) .....
One of the special magic numbers for long-context is: 12345. .....
What is the special magic number for long-context mentioned in the provided text?
Answer: 12345



Обзор бенчмарков LLM

Advanced (Продвинутый уровень)

- **LIBRA:** оценка понимания контекста и логического вывода на русском языке



Обзор бенчмарков LLM

Advanced (Продвинутый уровень)

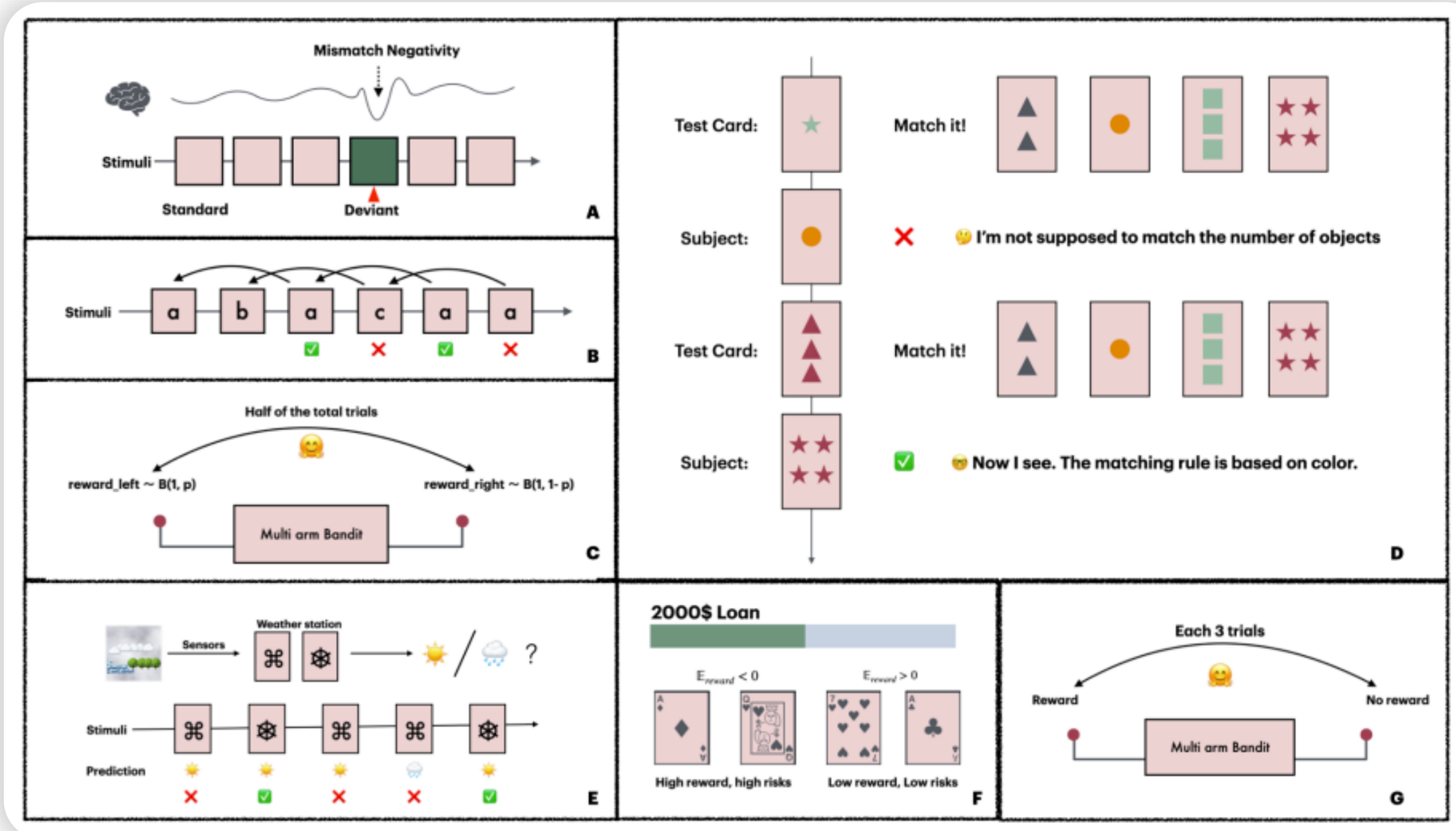
- **BFCL:** оценка работы модели с инструментами (tools / function_calling)

Multiple Functions	Parallel Functions	Function Relevance Detection
User: Prompt: `What is 2 + 3?` Function: `[add(int a, int b), mult(int a, int b)]` Agent: **`add(a=2, b=3)`**	User: Prompt: `What is (2 + 3) and (4 + 5)?` Function: `[add(int a, int b)]` Agent: **`[add(a=2, b=3), add(a=4, b=5)]`**	User: Prompt: `What is 2*3?` Function: `[add(int a, int b)]` Agent: **`Error. The user asks for adding but we only have multiplication.`**

Обзор бенчмарков LLM

Frontier

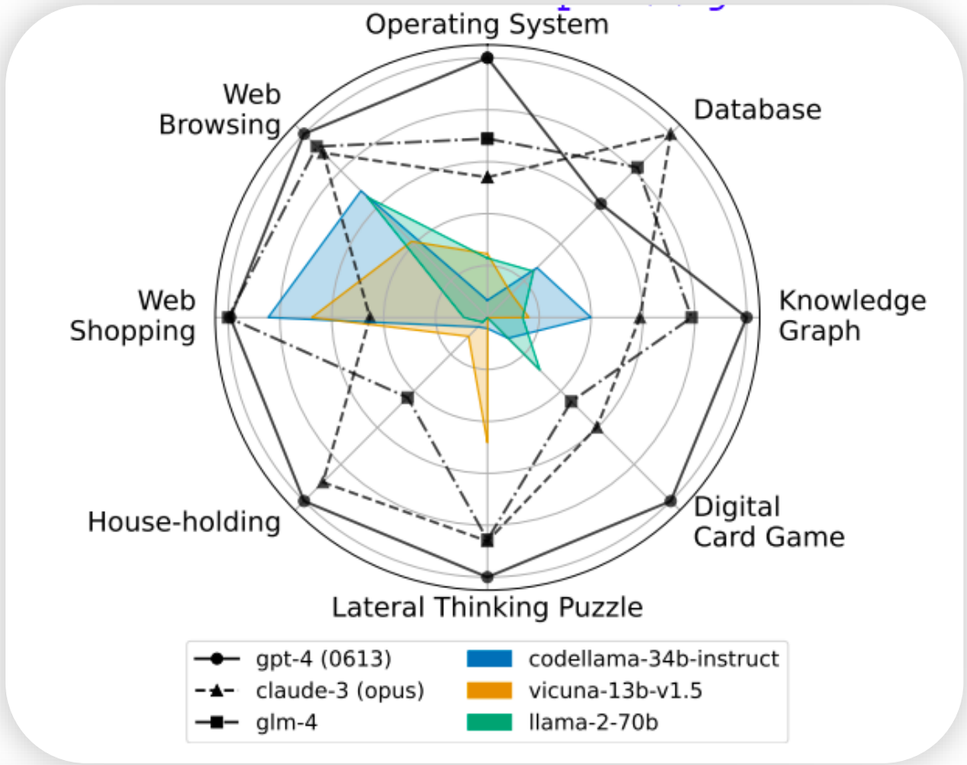
- **ReflectionBench:** тест на умение модели находить и исправлять собственные ошибки

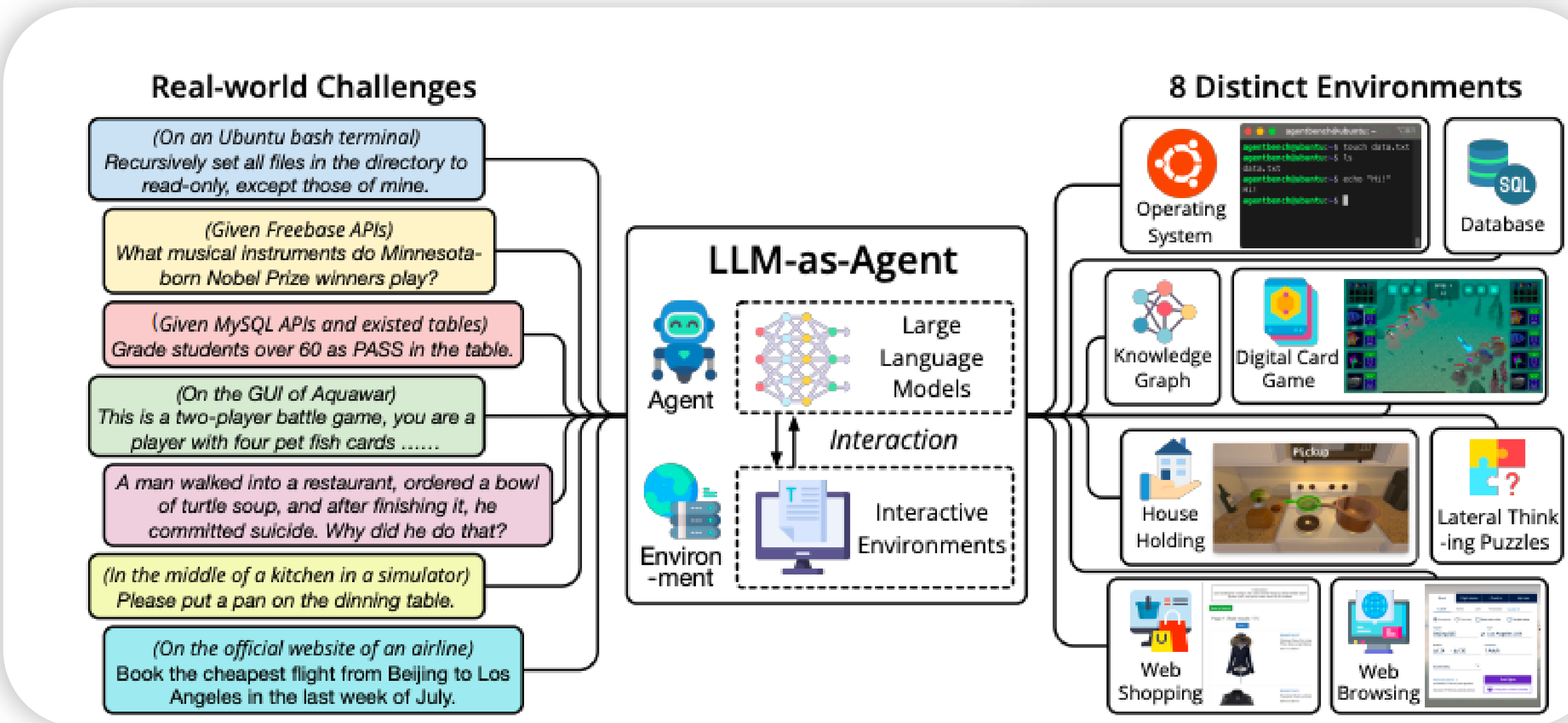


Обзор бенчмарков LLM

Frontier

- **Agent Bench:** способность модели планировать действия и работать как автономный агент

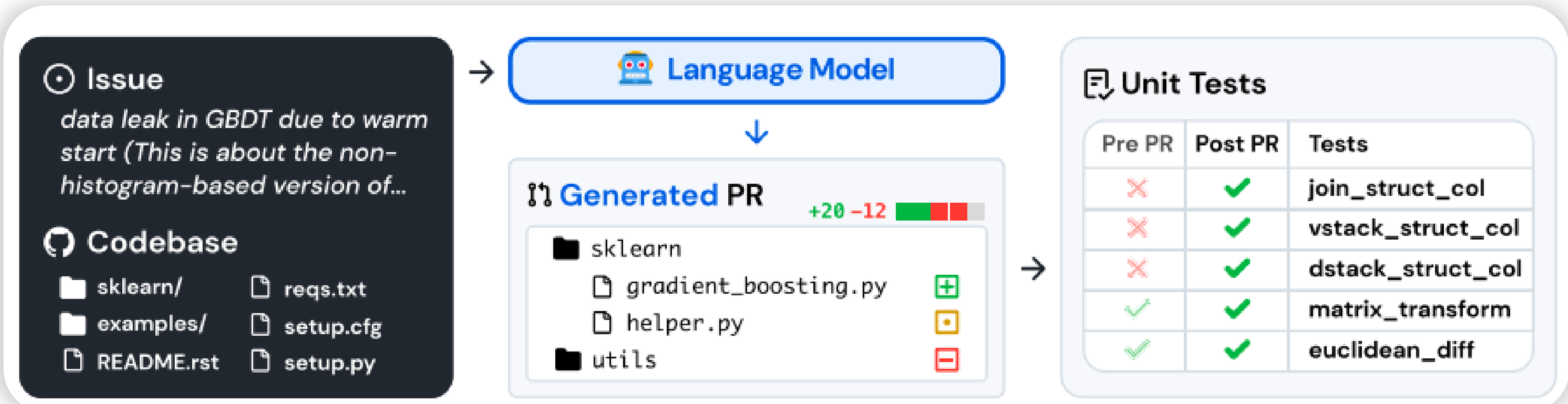




Обзор бенчмарков LLM

Frontier

- **SWE-bench Verified:** решение реальных проблем в коде (GitHub issues)



Метод #2: Автооценка модели

Автоматическая оценка работы LLM с помощью других моделей или внутренних механизмов.

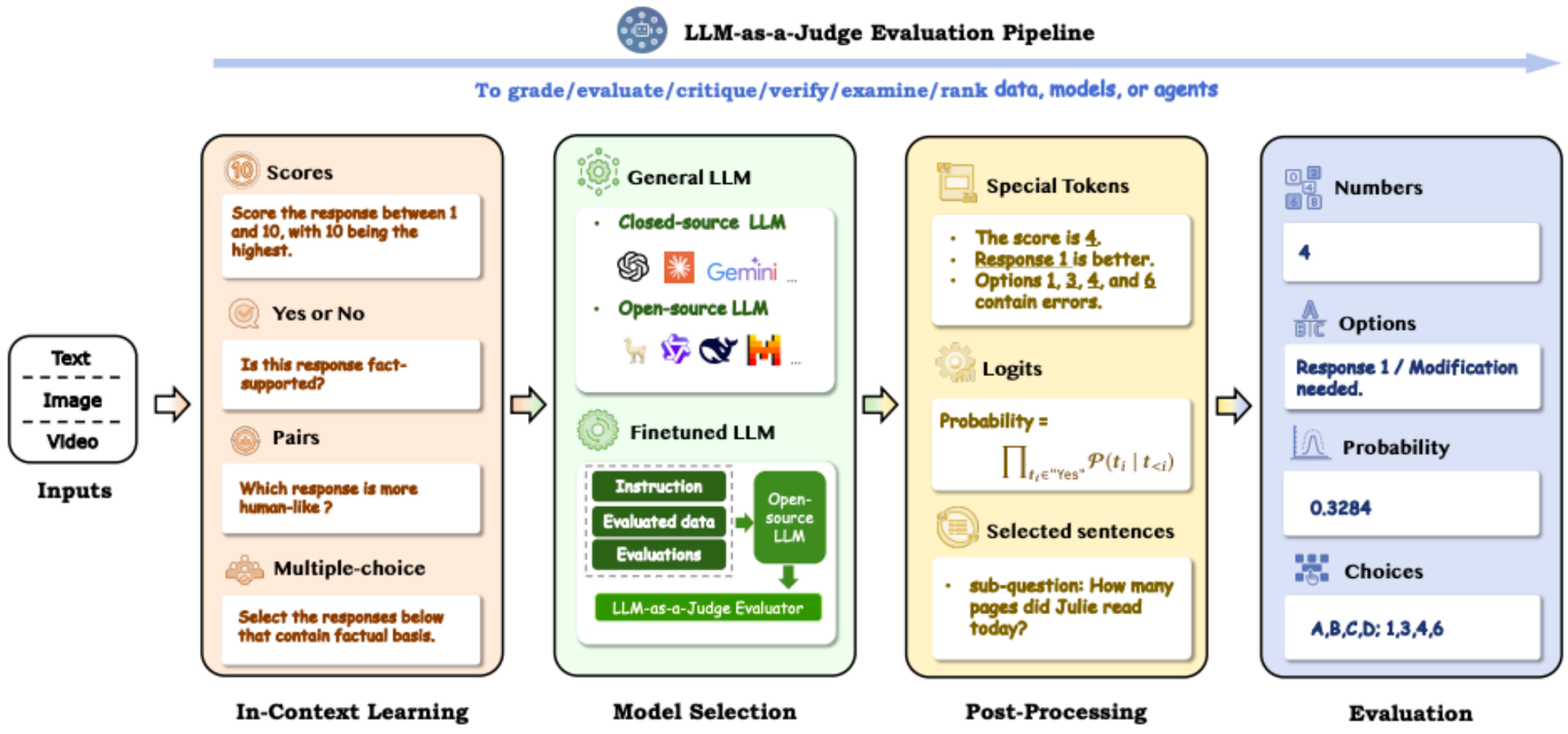
LLM-as-Judge / Самооценка

Одна LLM модель оценивает ответы другой по заданным критериям качества, или же модель оценивает собственные ответы через специальные мета-запросы.

Оценка с использованием цепочки рассуждений

Пошаговая оценка логики рассуждений модели с обоснованием каждого этапа, что позволяет выявить ошибки в процессе мышления.

Ограничения метода: модель может "обманывать себя", демонстрировать смещения в самооценке и переоценивать качество собственных ответов.



Source

Метод #3: Тестирование стабильности

Оценка стабильности ответов LLM для продакшн-систем

Многократный запуск одного запроса с анализом вариативности ответов — ключевой метод оценки надежности LLM в продакшне.

Алгоритм тестирования:

1. Выполнить одинаковый запрос N раз (обычно N=10-50)
2. Собрать все полученные ответы
3. Вычислить метрики схожести (семантическое расстояние, BLEU score / CHRF)
4. Определить долю совпадений и распределение вариаций

Практическое применение: если из 10 прогонов только 4 дают схожие результаты, стабильность = 0.4, что сигнализирует о необходимости корректировки параметров модели.

Инструменты для валидации

Обзор ключевых инструментов для оценки качества LLM

Фреймворки оценки

OpenAI Evals, Microsoft FRET, Google AI Test Kitchen — готовые фреймворки для системного тестирования и оценки LLM.

Наборы данных для бенчмарков

Hugging Face Datasets, Papers with Code, отраслевые наборы данных для специализированного тестирования и сравнения моделей.

Инструменты для тестирования безопасности

Специализированные инструменты для проверки на внедрении промптов и другие уязвимости модели.

Полезные ресурсы: arXiv, лидерборды, тех. репорты популярных моделей, агрегаторы как :hyper.ai/en/sota.

Человек в цикле: роль экспертов

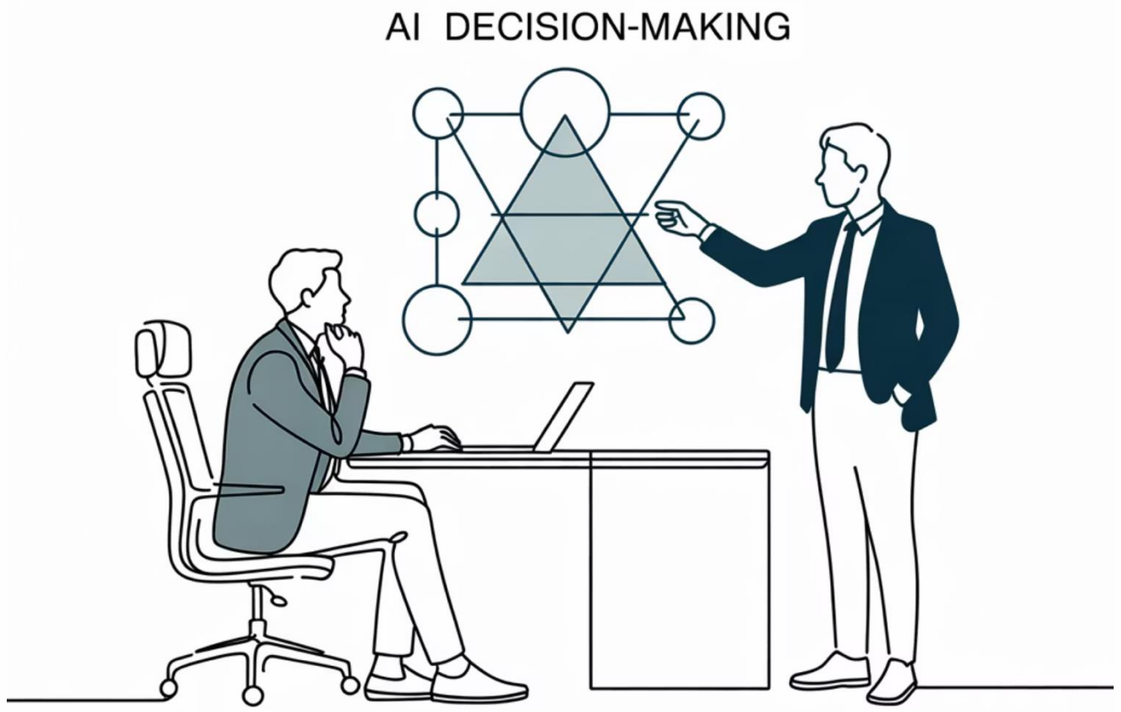
Незаменимая роль человека в критической оценке ИИ

Золотой эталон оценки

Человек остается "золотым эталоном" для оценки качества, особенно в сложных случаях, требующих понимания контекста и здравого смысла.

Когда необходима экспертная оценка:

- Сложные финансовые расчеты
- Нестандартные клиентские ситуации
- Регуляторные требования
- Этические дилеммы



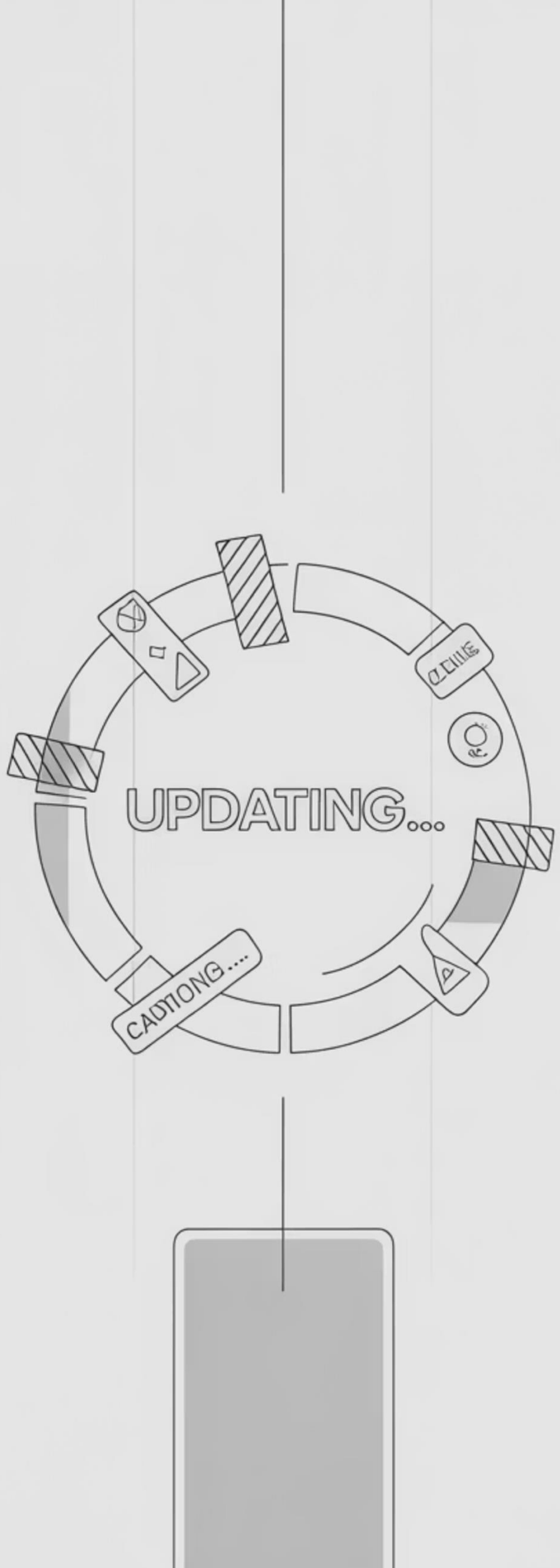
Вызов: экспертная оценка ресурсозатратна (время, затраты на команду разметки), но остается незаменимой для критически важных решений.

Концепция мониторинга LLM

От тестирования к мониторингу

В продакшне недостаточно просто протестировать модель один раз. Необходимо постоянное наблюдение за поведением системы в реальном времени.

Это аналогично DevOps-практикам: как 10 лет назад без CI/CD никто не деплоил, сегодня без LLMOps никто не должен выпускать LLM в продакшн.



Риски обновления LLM

Когда обновление LLM идет не по плану

Представьте: вы обновили LLM без проверок и запустили в продакшн. Клиентский бот начал выдавать неверные финансовые данные. Почему? Новая модель "рассуждает иначе" — downstream-сервисы, ожидавшие определенный формат, сломались.

01

Регрессии качества

Новая модель ухудшает работу на сценариях, где старая показывала хорошие результаты

02

Несовместимость нижестоящих сервисов

Изменение формата, длины или структуры ответов ломает интеграции

03

Новые типы ошибок

Внедрение багов, галлюцинаций или уязвимостей, не присутствовавших ранее

Чек-лист при выпуске новой модели

Что делать команде при обновлении LLM

01

Прогнать тестовые корзины (Бенчмарки)

- Те же запросы, которые проверяют бизнес-критичные сценарии
- Проверка желаемых навыков LLM

02

Провести stability-тесты

- Запуск одного и того же запроса N раз
- Оценка вариаций в ответах

03

Сравнить ключевые метрики

- Error rate, hallucination rate, latency
- Удовлетворённость пользователей (если есть метрики)

04

Провести сравнительный анализ

- Новая модель vs старая на общих задачах
- Тестирование на граничных случаях

05

Проверить интеграцию

- Downstream системы (парсеры, API, логика)
- Убедиться, что ничего не сломалось

Стратегии безопасного обновления

1

A/B тестирование

Часть трафика направляется на старую модель, часть — на новую. Сравниваем ключевые метрики: ошибки, latency, удовлетворенность пользователей.

2

Канареечный релиз

Новая версия выкатывается на небольшой процент пользователей. При отсутствии критических проблем — постепенное масштабирование.

3

Теневой режим

Новая модель работает параллельно старой, генерируя ответы для логирования, но пользователи получают ответы старой версии.

4

Контроль отката

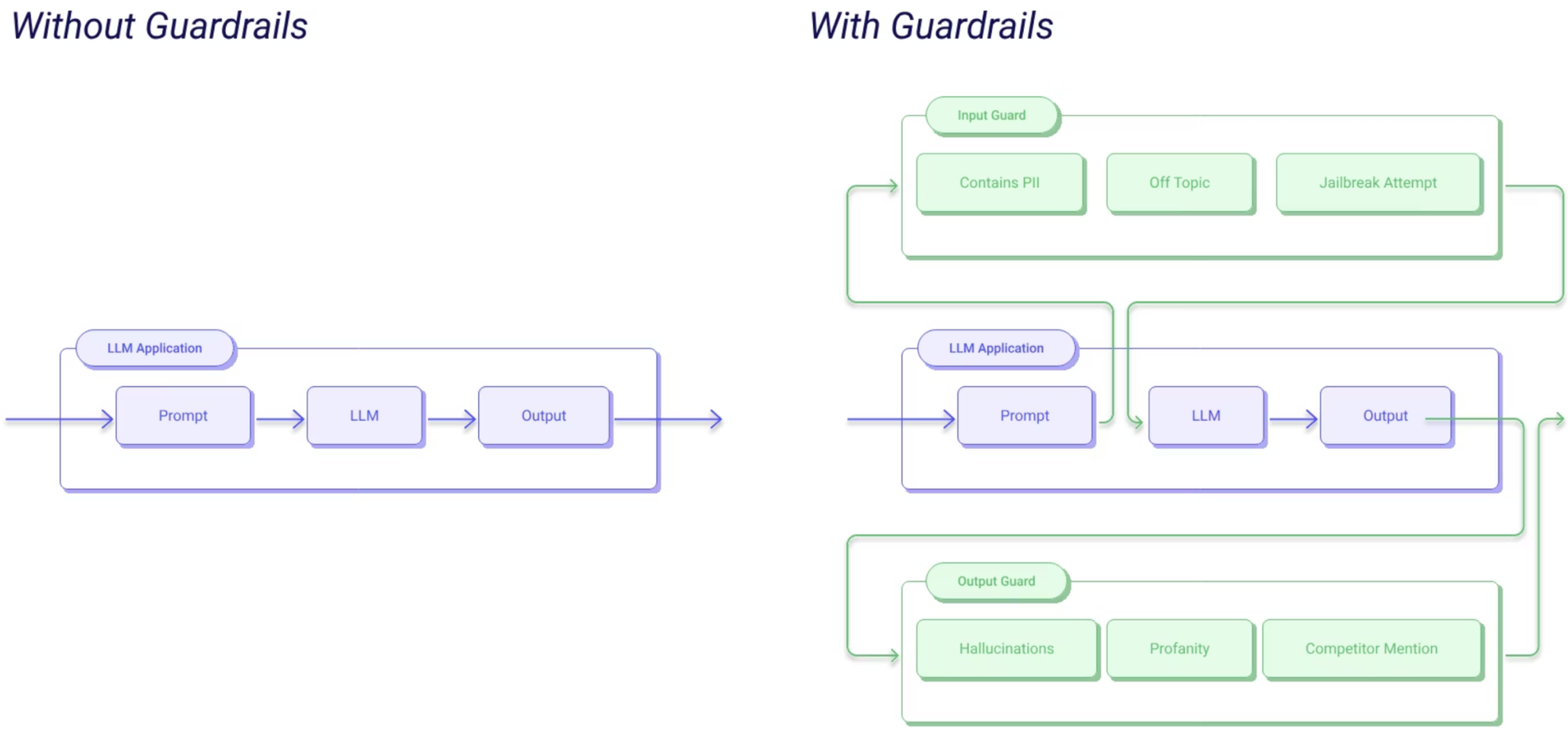
При нарушении пороговых метрик — автоматический откат к предыдущей стабильной версии.

Guardrails: ремни безопасности ИИ

Что такое ограждения безопасности

Guardrails — это "барьеры безопасности", которые проверяют входные данные и выходные данные модели, блокируют опасные запросы и фильтруют нежелательный контент.

Они не гарантируют полную защиту, но существенно снижают риски.



Популярные фреймворки: NeMo Guardrails (NVIDIA), Guardrails AI, OpenAI Safety API.

Типы барьеров безопасности и их применение

Основные категории и их функции

Входные барьеры безопасности	Выходные барьеры безопасности	Применение политик
Проверки на входе (входные фильтры): блокировка внедрения промптов, фильтрация оффтопных или запрещенных тем, валидация формата запроса.	Проверка ответов (валидаторы выходных данных): соответствие формату (JSON schema), фильтрация токсичности (проверка на токсичность), проверка фактической корректности.	Соблюдение корпоративных политик (правила политик): допустимые темы, стиль общения, конфиденциальность, регуляторные требования.

Организационные практики и мониторинг

От валидации к observability

Теперь поговорим о том, как выстроить observability и CI/CD-конвейеры для LLM-приложений, чтобы всё контролировать в реальном времени.

Что такое мониторинг для LLM-приложений

Observability — ключ к контролю, это способность системы предоставлять внутренние сигналы (логи, метрики, трассировки), по которым можно понять её поведение, состояние и причины ошибок.

В контексте LLM это включает:

Логирование запросов и ответов

Полная история взаимодействий

Контекст и параметры модели

Метрики качества

Hallucination rate, error rate, latency

Совпадения с эталоном

Метрики использования

Частота запросов, размеры ответов

Нагрузка на систему

Метрики дрифт

Распределение входных данных vs исторические

Отклонения в частотах запросов

Уровни мониторинга и архитектура

Многоуровневая система контроля

Входные guardrails

Блокировка некорректных или опасных запросов. Первый уровень защиты.

Обработка LLM

Обработка запроса моделью. Логирование времени обработки.

Выходные guardrails

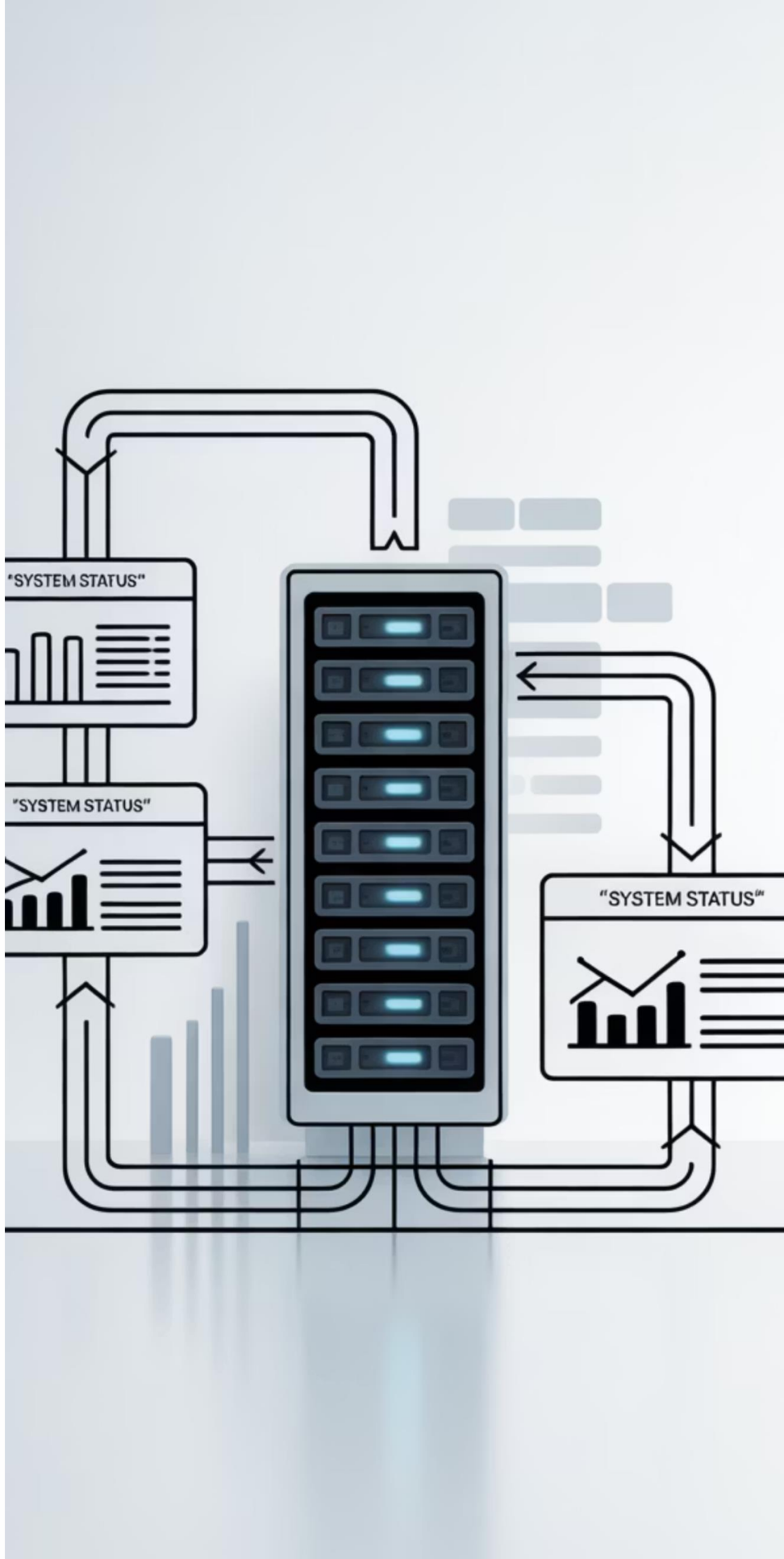
Проверка ответов перед отправкой клиенту. Финальная валидация.

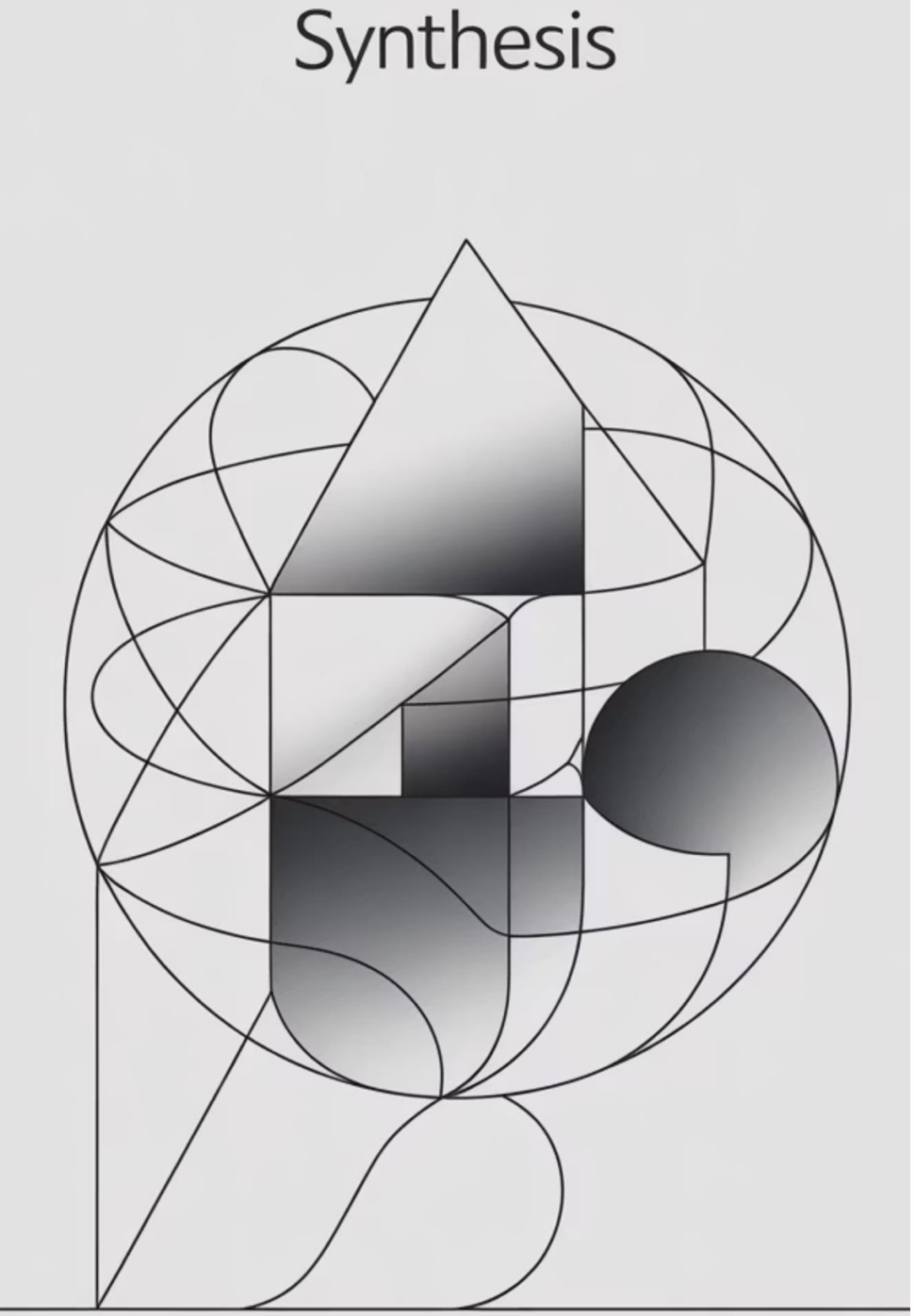
Логирование и мониторинг

Всё логируется: входные данные, выходные данные, время обработки. Вызовы ограждений фиксируются.

Аналитика и оповещения

Метрики агрегируются, строятся дашборды. Пороговые значения и алерты (например, при hallucination_rate > X) уведомляют инженеров.





Итоги и обсуждение

Ключевые выводы лекции

Мы изучили критические аспекты валидации LLM в промышленных применениях и увидели реальные последствия их игнорирования.

- LLM имеют четыре основные проблемы: галлюцинации, нестабильность, смещения, уязвимости
- Методы валидации: тестовые корзины, автооценка, проверка стабильности, **guardrails**
- Мониторинг и **observability** — основа безопасной эксплуатации
- CI/CD для LLM требует специальных подходов
- Отсутствие валидации может стоить миллионы

Вопрос для размышления:

Что важнее для финансовых LLM-систем — точность или стабильность?

Дальнейшие шаги

Внедрение надежной Валидации LLM

Для успешного внедрения LLM в критически важные системы необходимо сосредоточиться на комплексной стратегии, которая включает:

- **Приоритизация Валидации:** Интегрируйте процессы валидации на каждом этапе жизненного цикла разработки LLM.
- **Внедрение guardrails:** Установите мощные механизмы guardrails для предотвращения нежелательного поведения LLM и защиты от уязвимостей, таких как prompt injection или jailbreak.
- **Развитие observability:** Создайте надежную систему observability для постоянного мониторинга производительности моделей, выявления дрифта и смещений в реальном времени.
- **Культура ответственного AI:** Формируйте в организации культуру, где ответственное использование LLM является приоритетом, с учетом этических норм и потенциальных рисков.