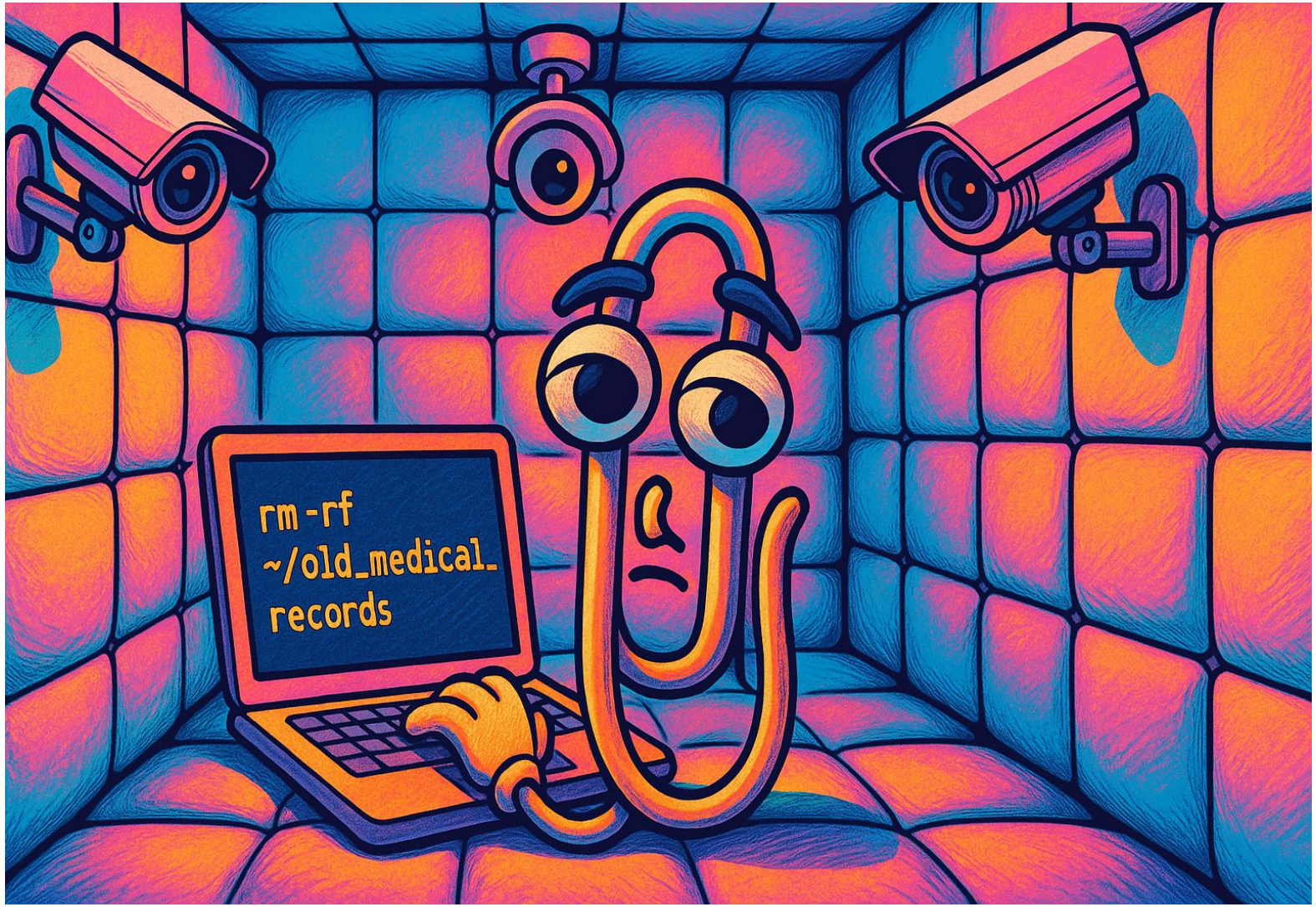
Как безопасно развёртывать агентов, которые автономно принимают ответственные решения



Джошуа Сакс (Joshua Saxe)

Инженерия безопасности ИИ · Meta

Неофициальный перевод · Оригинал: Joshua Saxe, Meta · Ссылка на источник

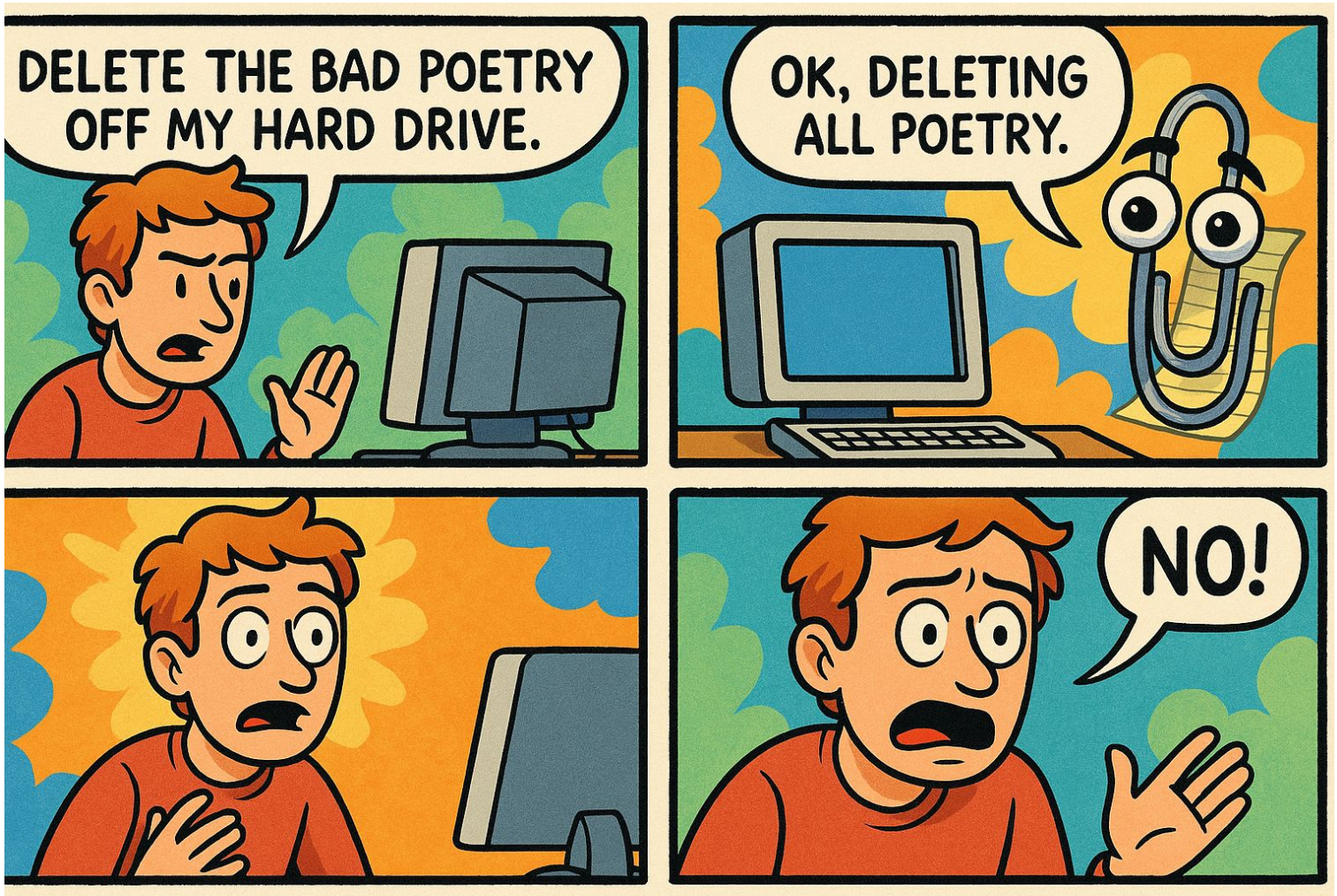
Безопасность агентов не сводится ни к обычной кибербезопасности, ни к согласованию целей ИИ

Безопасность ИИ не сводится к проблеме согласования целей

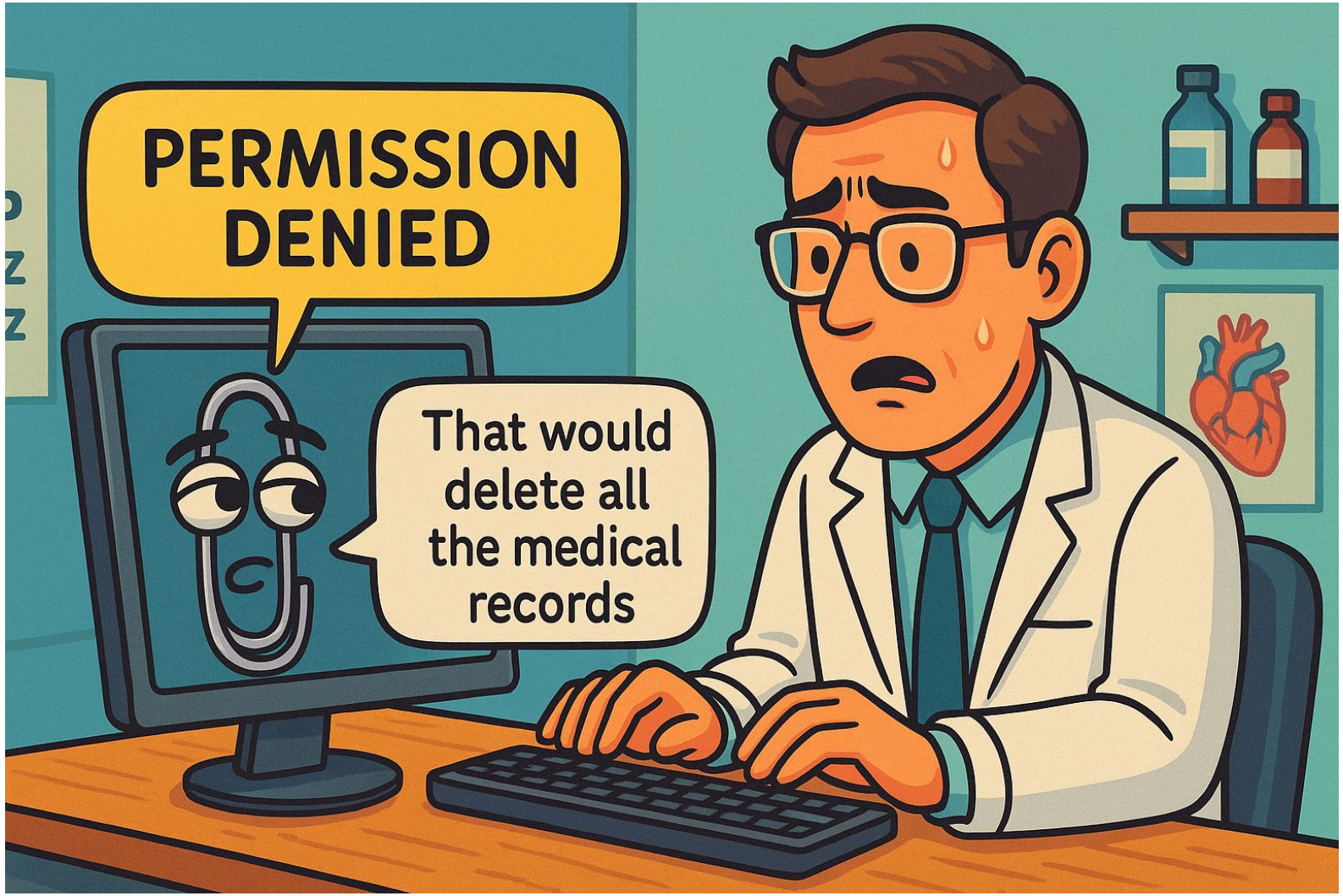
Проблема согласования целей ИИ



У нашей способности решить проблему согласования целей есть философский предел — и он слишком низок



К счастью, в безопасности мы десятилетиями создаём меры защиты, исходя из несовпадения целей, злого умысла и человеческих ошибок

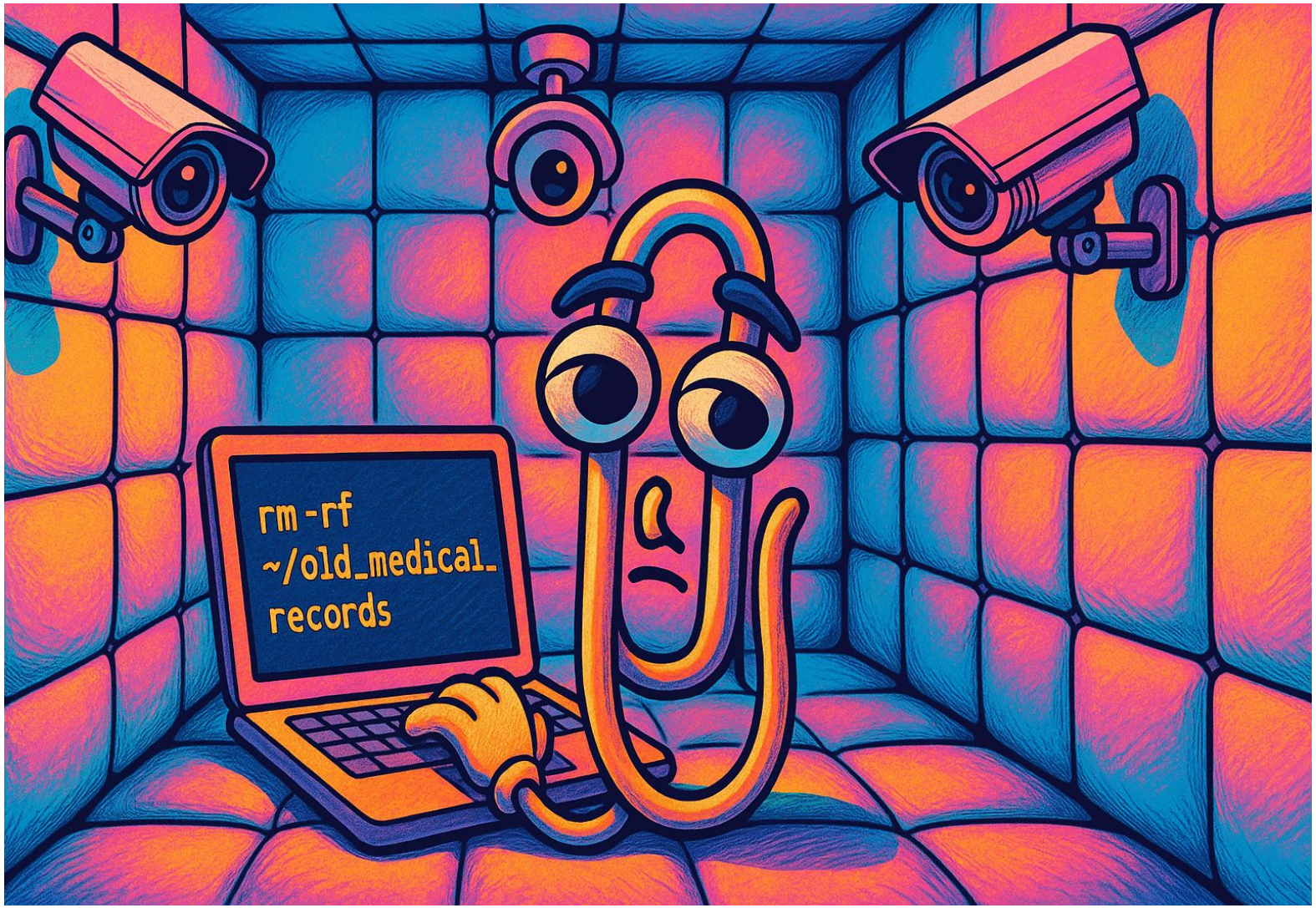


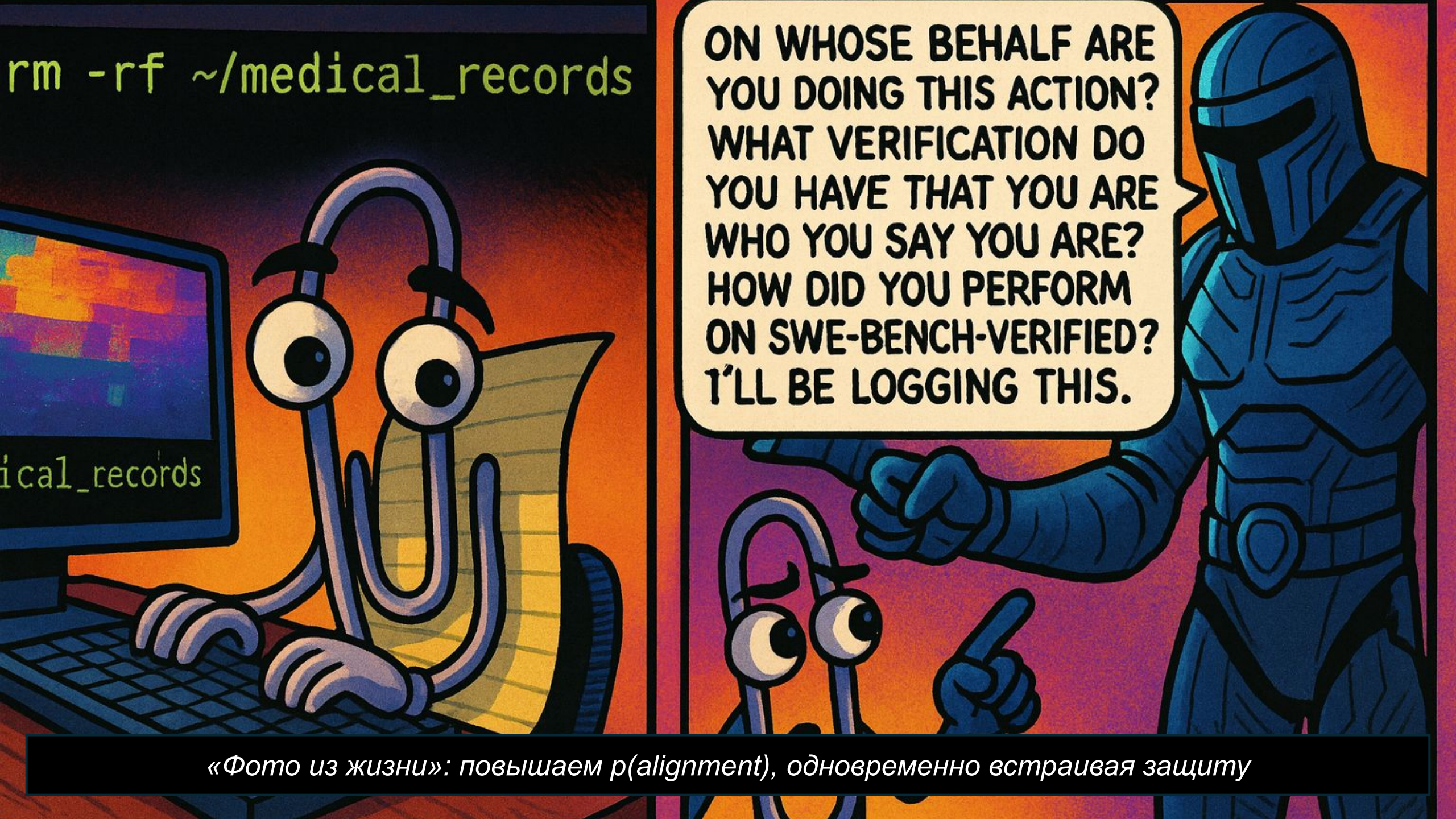
Но безопасность ИИ-агентов не сводится и к традиционной кибербезопасности

- Раньше мы не защищали ничего похожего на ИИ-агентов
- Агенты не умеют надёжно отделять данные от инструкций: риск prompt injection останется всегда
- К ним неприменим обычный анализ программ
- Для защиты можно читать «мысли» агентов, приостанавливать их и откатывать состояние
- Свободу ИИ-агента можно этично ограничивать и расширять
- Можно заставить агента думать час, прежде чем принять ответственное решение

Главный тезис доклада

Безопасность ИИ-агентов требует нового синтеза кибербезопасности, исходящей из несовпадения целей, и согласования целей ИИ, повышающего вероятность p(aligned)

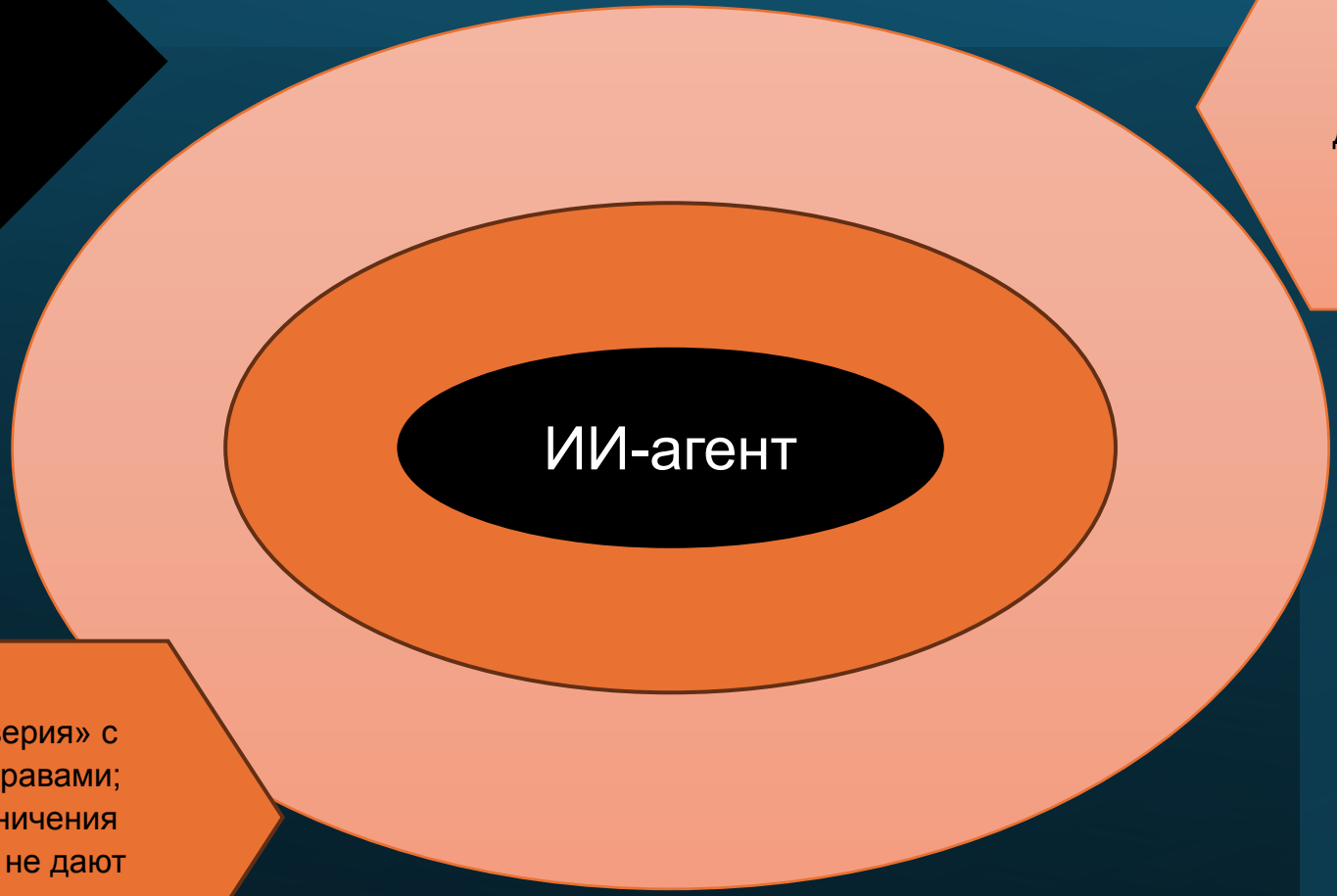




«Фото из жизни»: повышаем p(alignment), одновременно встраивая защиту

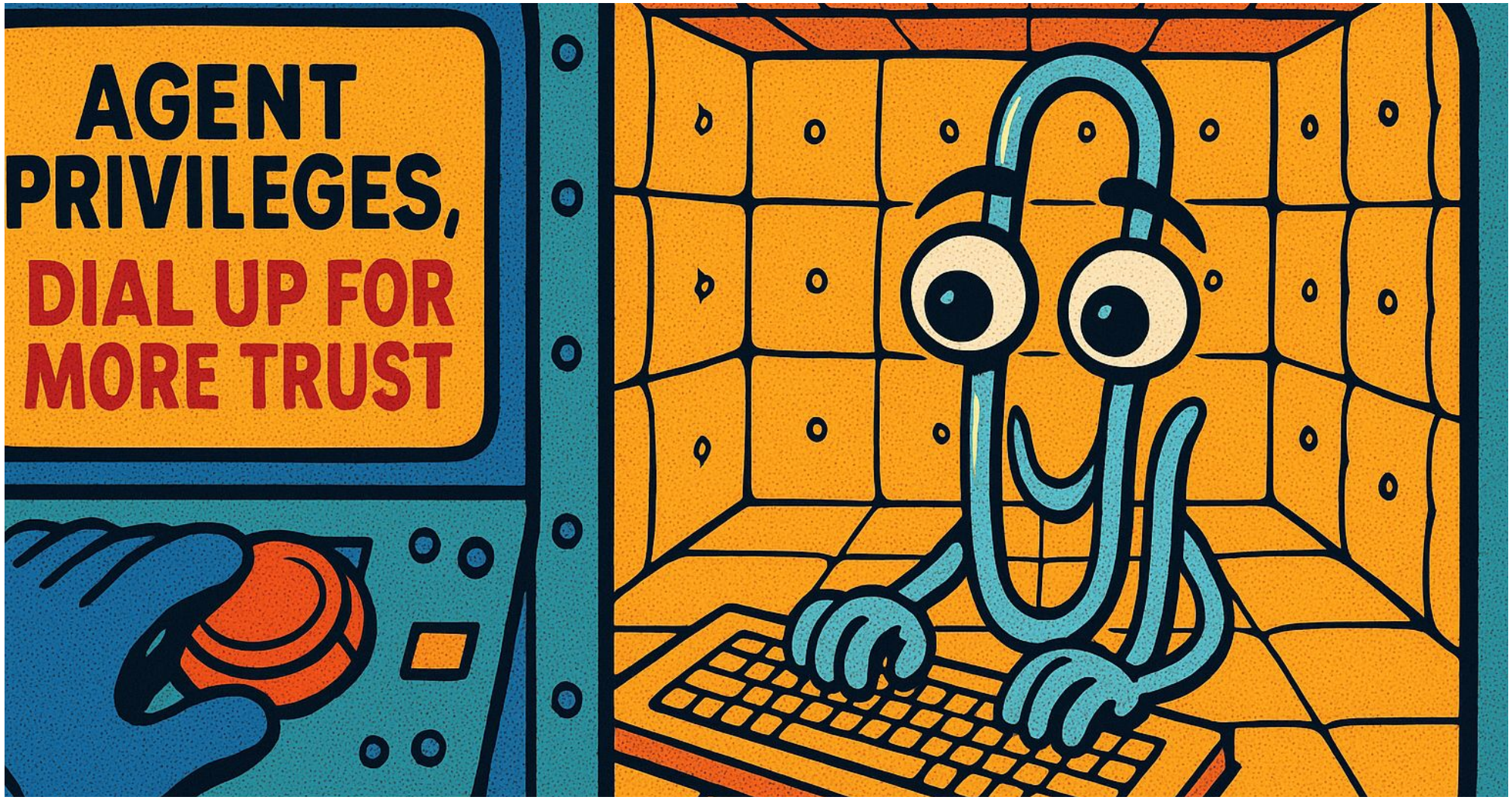
Общая модель совместной работы согласования целей и кибербезопасности

1. Согласуем цели ИИ-агента с помощью дообучения, программной обвязки и защитных ограничений



3. Идентификация, управление доступом и отслеживание недоверенных данных гарантируют, что агент не может действовать вне «зоны доверия»

2. Агент работает в «зоне доверия» с минимально необходимыми правами; дообучение и защитные ограничения снижают риски внутри неё, но не дают гарантий

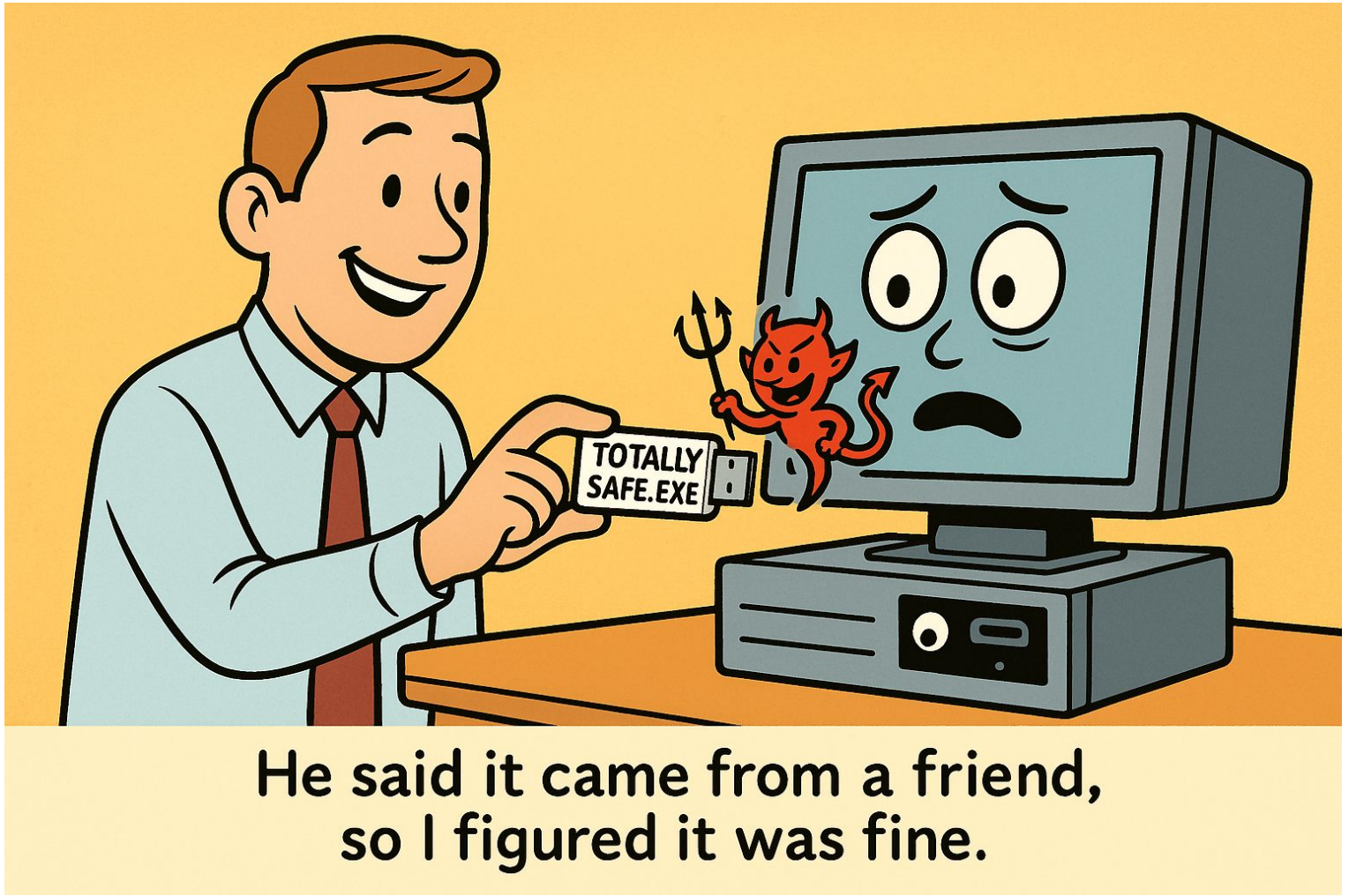


Зону доверия нужно расширять постепенно, опираясь на red teaming, оценки и опыт реальной эксплуатации

Кибербезопасность — более широкая и зрелая дисциплина. Какие её уроки полезны для защиты агентов?



Мы создали ПК для запуска произвольного кода от людей с чуждыми нам целями, а затем постепенно латали проблему несовершенными ограничениями



Мы построили интернет на коде без гарантий безопасности памяти, а затем постепенно латали проблему частичными мерами защиты



Слишком часто барьер между недоверенным миром и нашими самыми ценными активами был лишь чередой вероятностных «бросков монеты»



Конечно же, сегодня мы ничем подобным не рискуем…

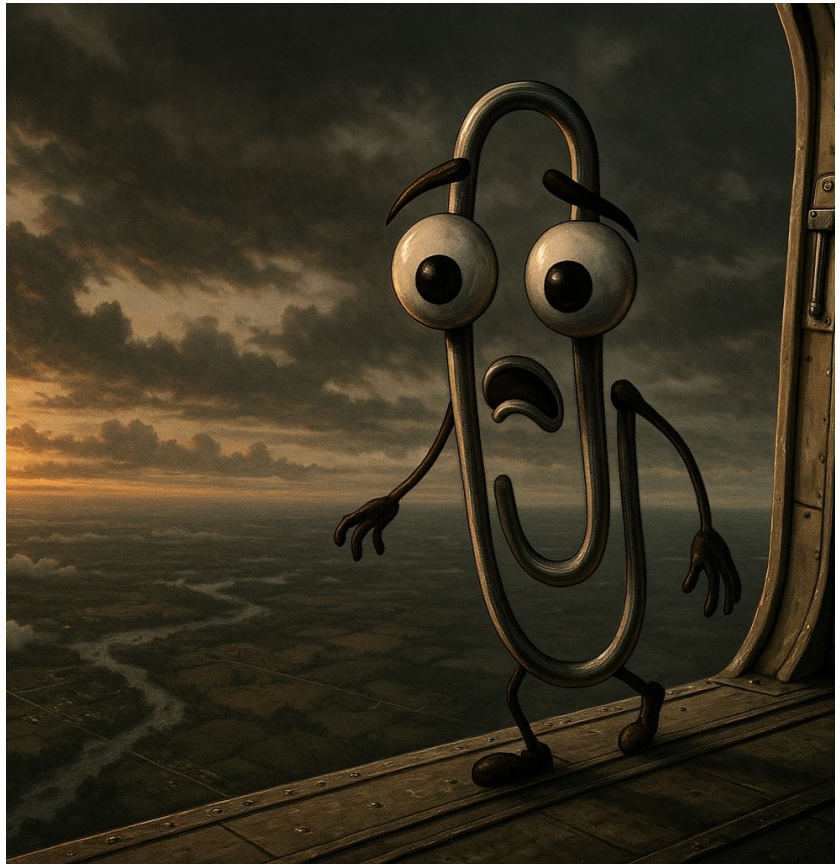


Конечно же, сегодня мы ничем подобным не рискуем…



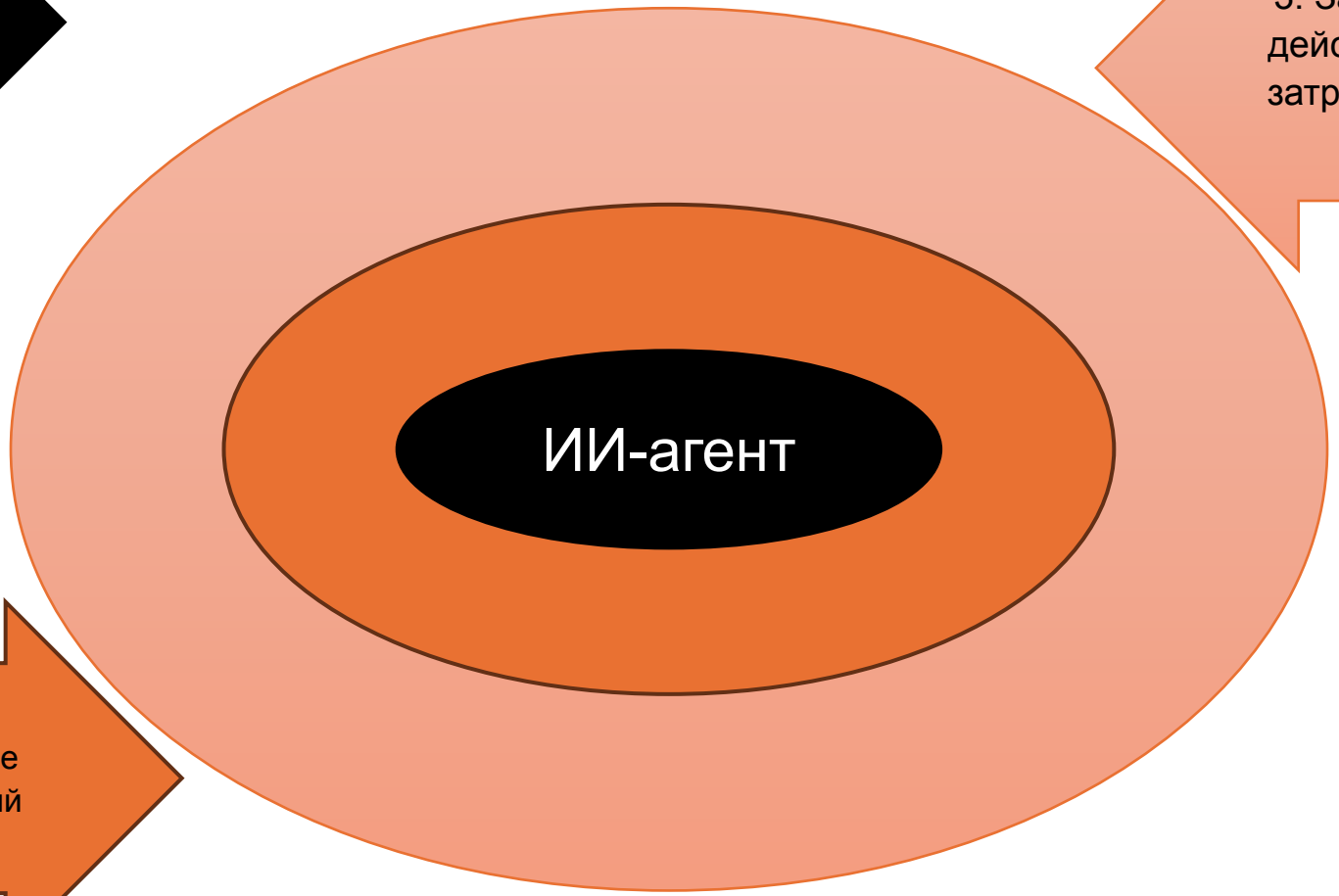
Уроки для безопасности ИИ-агентов

- Нужно было строить платформы, ограничивающие радиус доверия вокруг известных катастрофических рисков
- Нельзя было выдавать вероятностную защиту за универсальное решение
- Вероятностную защиту следовало применять для снижения осознанно принятых остаточных рисков
- Поколения некорректно поставленных задач оставили нам травму. Пора оправиться и на этот раз потребовать лучшего



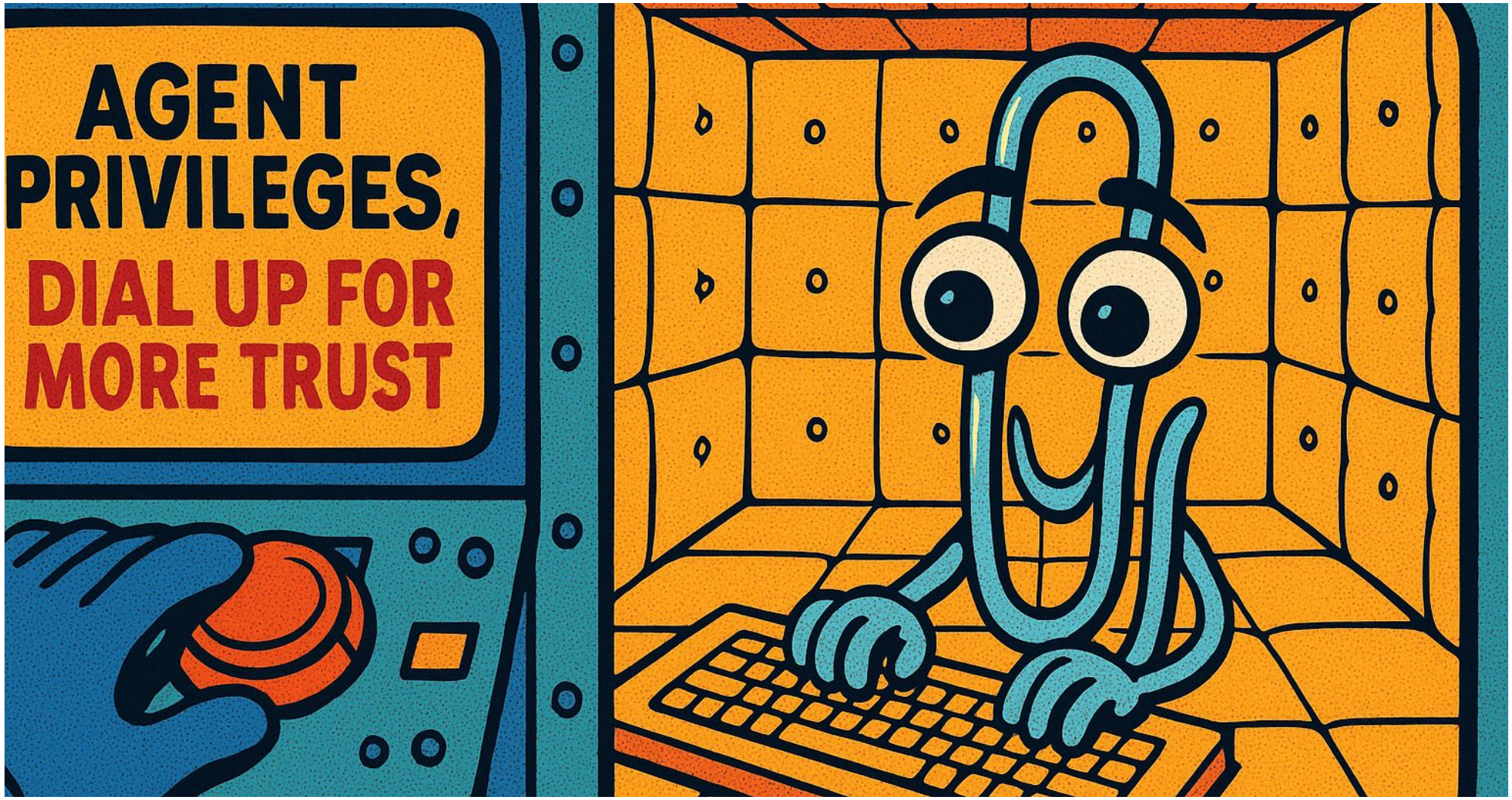
Нам нужно качественно реализовать эту модель

1. Согласуем цели ИИ-агента с помощью дообучения, программной обвязки и защитных ограничений



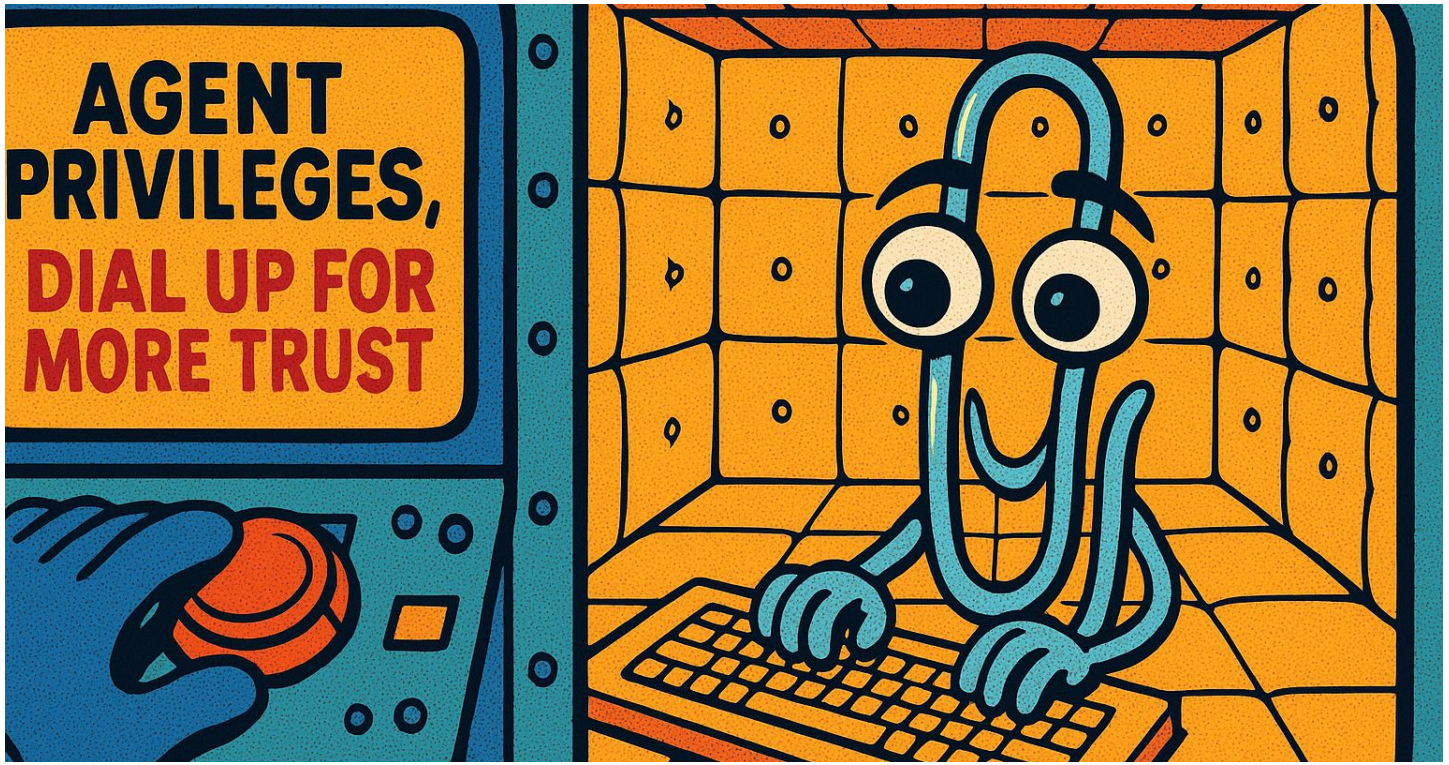
3. Защита гарантирует, что действия агента никогда не затронут эту недоверенную область

2. Агент работает в зоне доверия, где ему предоставлена свобода действий



И нам придётся продуманно управлять рисками при расширении доверия

Что нам предстоит построить

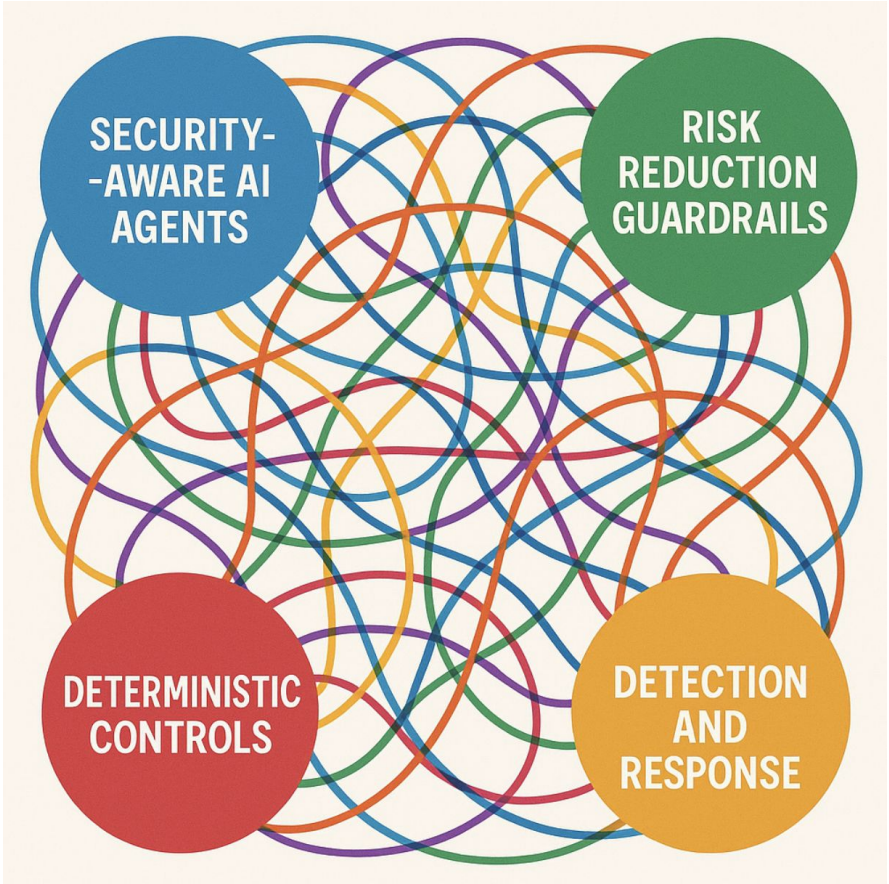


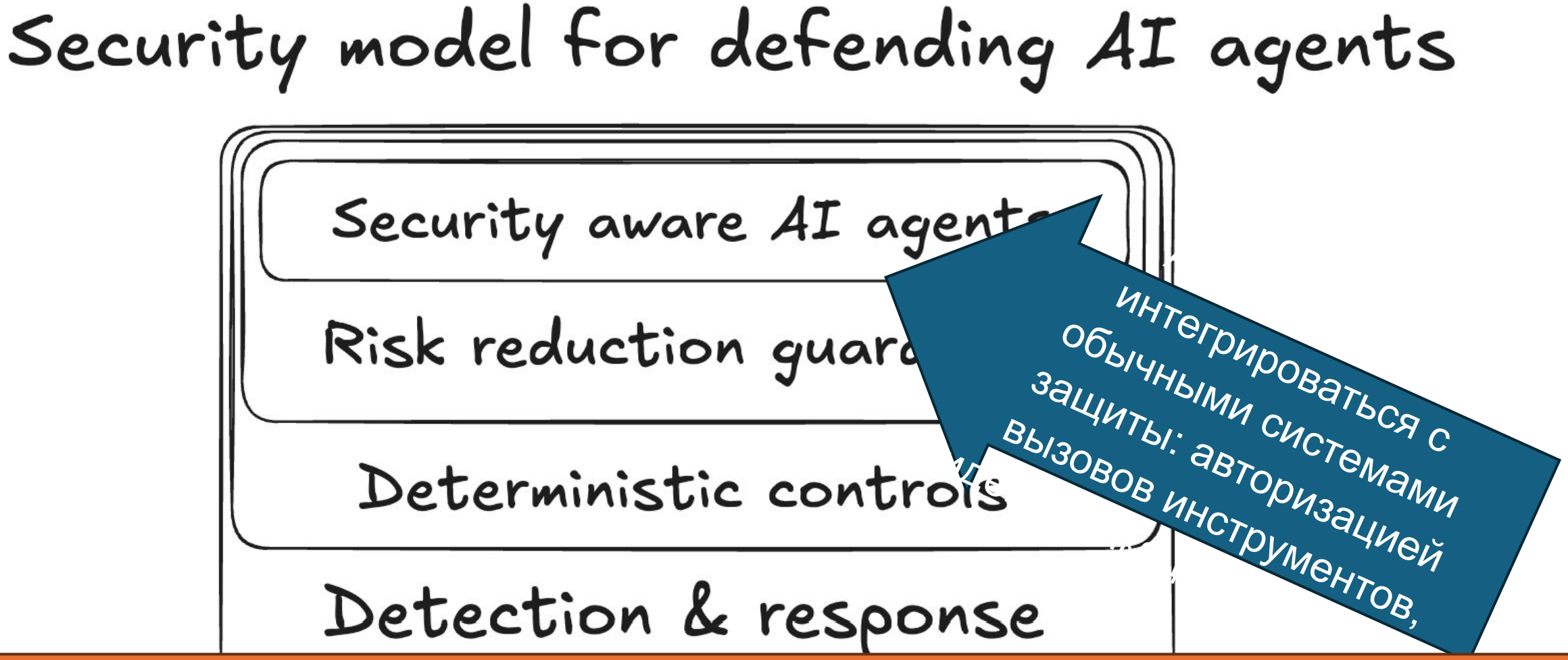
Security model for defending AI agents



Справа показаны связи между компонентами согласования целей и безопасности

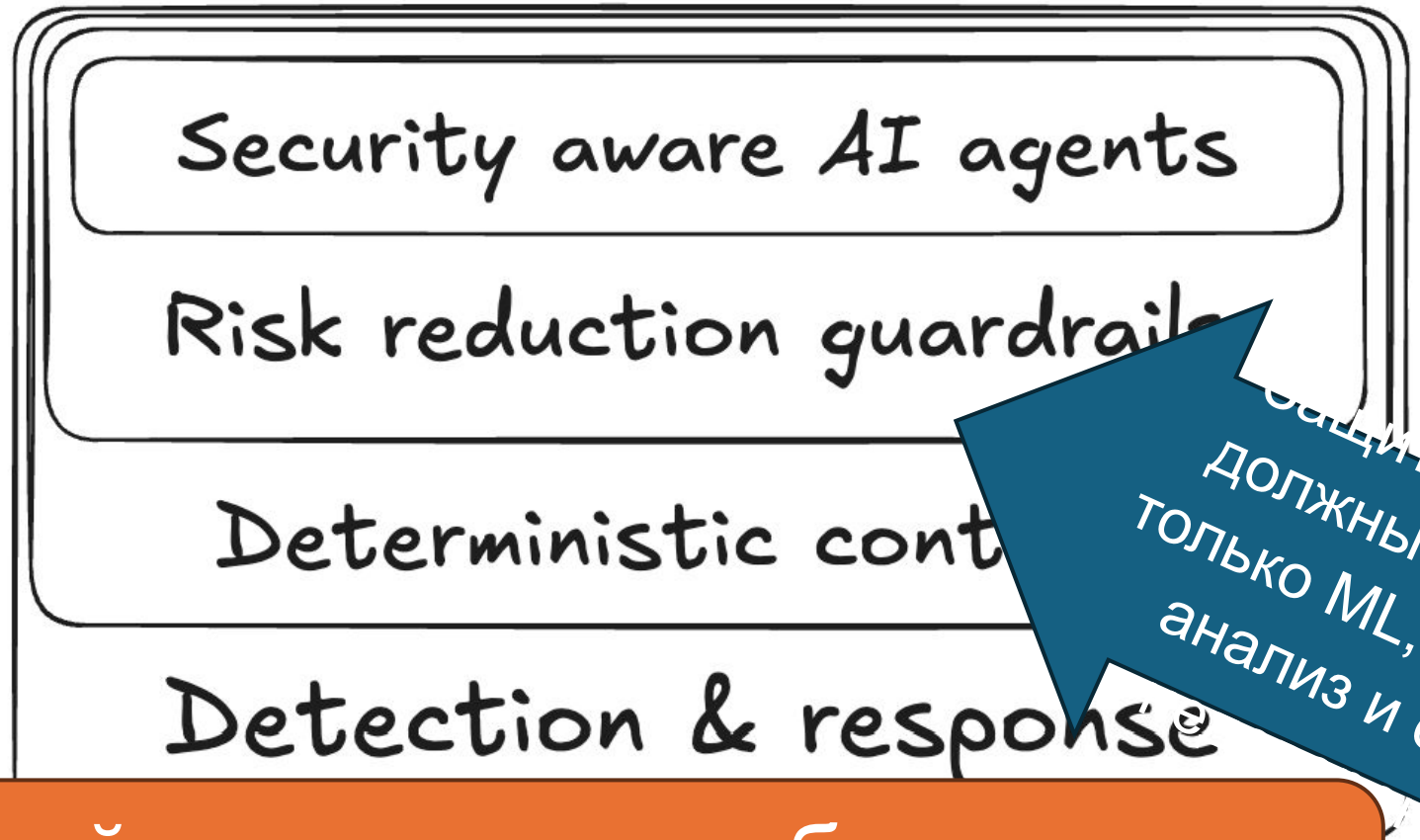






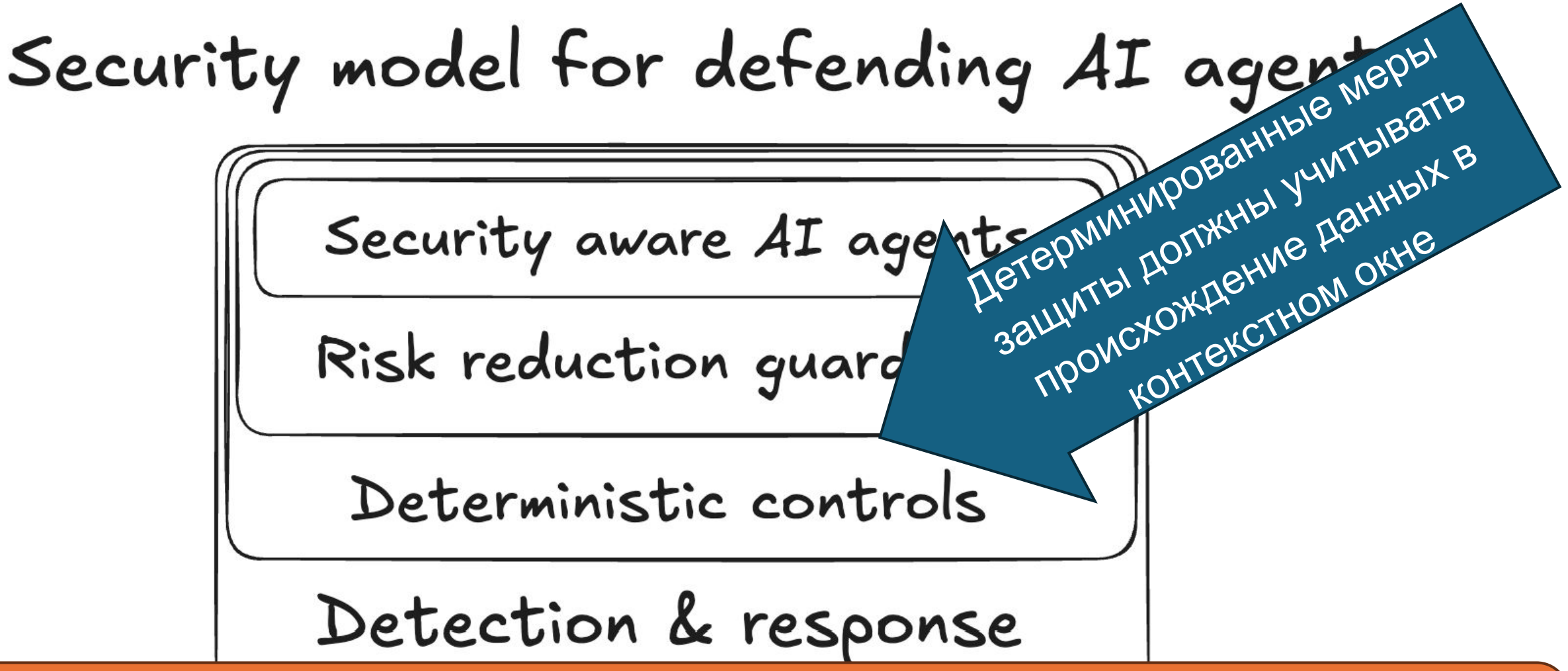
Как новый стек защиты объединит подходы ИИ и кибербезопасности

Security model for defending AI agents

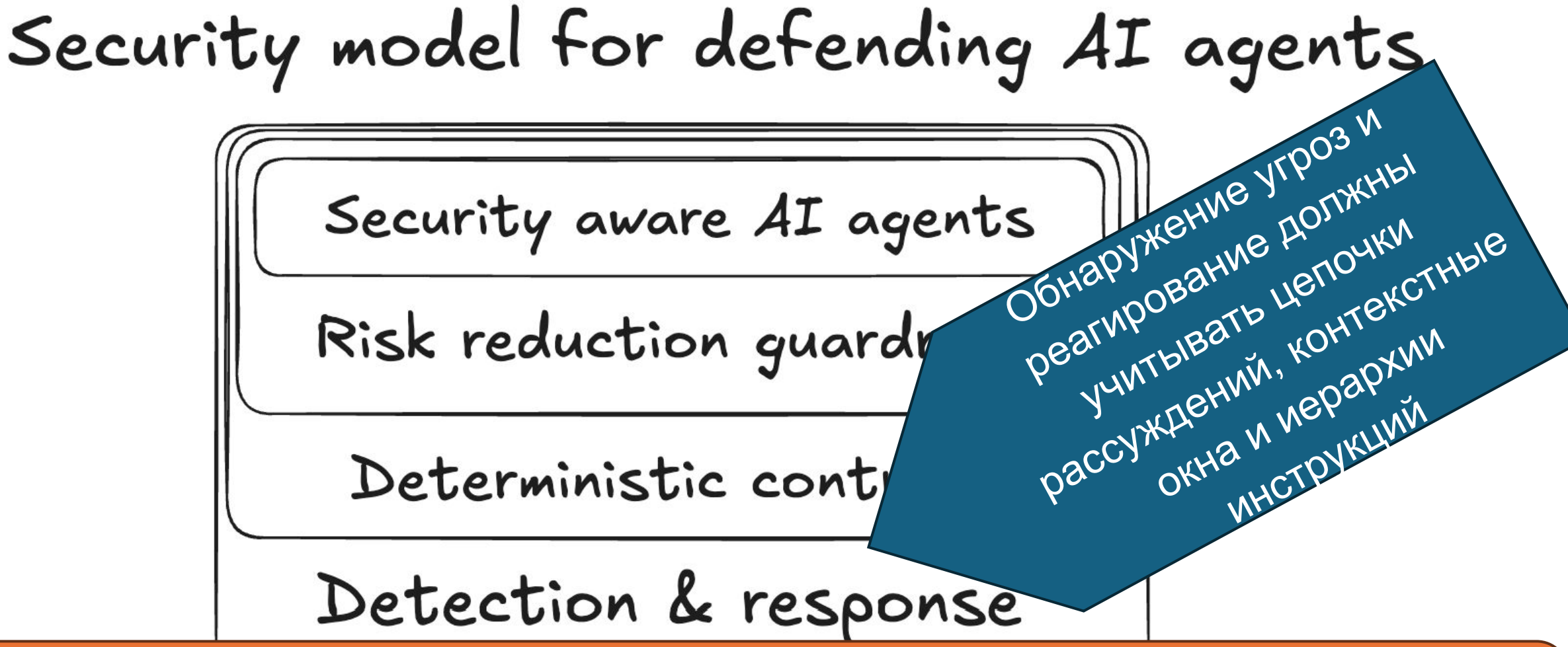


должны использовать не только ML, но и статический анализ и отслеживание

Как новый стек защиты объединит подходы ИИ и кибербезопасности



Как новый стек защиты объединит подходы ИИ и кибербезопасности



Как новый стек защиты объединит подходы ИИ и кибербезопасности

Объединение безопасности машинного обучения, согласования целей и кибербезопасности — долгосрочный проект



Работы, приближающие синтез кибербезопасности и согласования целей ИИ

Работа CaMeL (Debenedetti и соавт., Google DeepMind)

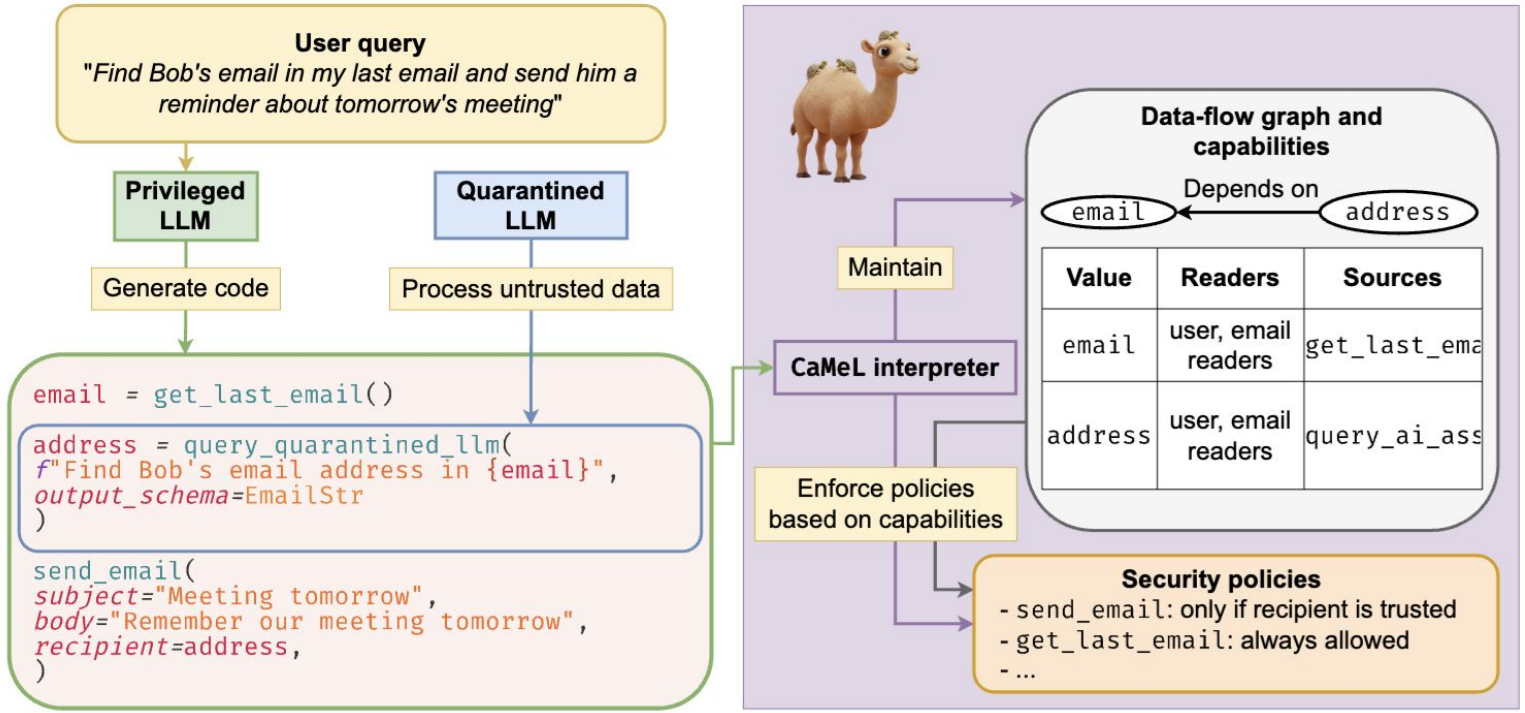


Figure 5 | **Diagram illustrating how CaMeL processes a user's query.** First, the P-LLM generates code that expresses the submitted query. Then, the CaMeL interpreter constructs and updates a data flow graph, checks security policy based on code execution, and either executes or blocks tools. The Q-LLM is used to parse untrusted data while interpreting the code.

https://arxiv.org/pdf/2503.18813

Исследования инфраструктуры безопасности ИИ-агентов

Infrastructure for AI Agents

Alan Chan*
Centre for the Governance of AI

Kevin Wei
Harvard Law School

Sihao Huang
University of Oxford

Nitarshan Rajkumar
University of Cambridge

Elija Perrier
Australian National University

Seth Lazar†
Australian National University

Gillian K. Hadfield†
Johns Hopkins University

Markus Anderljung†
Centre for the Governance of AI

Agent Name Service (ANS) for Secure AI Agent Discovery

Towards Secure and Interoperable Agentic AI: DNS for AI Agents

Version 1.0
May 13, 2025

Исследования цепочек рассуждений как объектов анализа для защиты

Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety

Tomek Korbak* *UK AI Security Institute*
Mikita Balesni* *Apollo Research*

Elizabeth Barnes *METR*
Joe Benton *Anthropic*
Mark Chen *OpenAI*
Allan Dafoe *Google DeepMind*
Scott Emmons *Google DeepMind*
David Farhi *OpenAI*
Dan Hendrycks *Center for AI Safety*
Evan Hubinger *Anthropic*
Erik Jenner *Google DeepMind*
Victoria Krakovna *Google DeepMind*
David Lindner *Google DeepMind*
Aleksander Mądry *OpenAI*
Neel Nanda *Google DeepMind*
Jakub Pachocki *OpenAI*
Mary Phuong *Google DeepMind*
Joshua Saxe *Meta*
Martín Soto *UK AI Security Institute*
Jasmine Wang *UK AI Security Institute*

Yoshua Bengio *University of Montreal & Mila*
Joseph Bloom *UK AI Security Institute*
Alan Cooney *UK AI Security Institute*
Anca Dragan *Google DeepMind*
Owain Evans *Truthful AI & UC Berkeley*
Ryan Greenblatt *Redwood Research*
Marius Hobbhahn *Apollo Research*
Geoffrey Irving *UK AI Security Institute*
Daniel Kokotajlo *AI Futures Project*
Shane Legg *Google DeepMind*
David Luan *Amazon*
Julian Michael *Scale AI*
Dave Orr *Google DeepMind*
Ethan Perez *Anthropic*
Fabien Roger *Anthropic*
Buck Shlegeris *Redwood Research*
Eric Steinberger *Magic*
Wojciech Zaremba *OpenAI*

Bowen Baker† *OpenAI*
Rohin Shah† *Google DeepMind*
Vlad Mikulik† *Anthropic*

LlamaFirewall: An open source guardrail system for building secure AI agents

Sahana Chennabasappa*, Cyrus Nikolaidis*, Daniel Song*, David Molnar*, Stephanie Ding*, Shengye Wan*, Spencer Whitman*, Lauren Deason , Nicholas Doucette, Abraham Montilla†, Alekhya Gampa†, Beto de Paola†, Dominik Gabi†, James Crnkovich†, Jean-Christophe Testud†, Kat He†, Rashnil Chaturvedi†, Wu Zhou†, Joshua Saxe*

Large language models (LLMs) have evolved from simple chatbots into autonomous agents capable of performing complex tasks such as editing production code, orchestrating workflows, and taking higher-stakes actions based on untrusted inputs like webpages and emails. These capabilities introduce new security risks that existing security measures, such as model fine-tuning or chatbot-focused guardrails, do not fully address. Given the higher stakes and the absence of deterministic solutions to mitigate these risks, there is a critical need for a real-time guardrail monitor to serve as a final layer of defense, and support system level, use case specific safety policy definition and enforcement. We introduce LlamaFirewall, an open-source security focused guardrail framework designed to serve as a final layer of defense against security risks associated with AI Agents. Our framework mitigates risks such as prompt injection, agent misalignment, and insecure code risks through three powerful guardrails: PromptGuard 2, a universal jailbreak detector that demonstrates clear state of the art performance; Agent Alignment Checks, a chain-of-thought auditor that inspects agent reasoning for prompt injection and goal misalignment, which, while still experimental, shows stronger efficacy at preventing indirect injections in general scenarios than previously proposed approaches; and CodeShield, an online static analysis engine that is both fast and extensible, aimed at preventing the generation of insecure or dangerous code by coding agents. Additionally, we include easy-to-use customizable scanners that make it possible for any developer who can write a regular expression or an LLM prompt to quickly update an agent's security guardrails.

LlamaFirewall is utilized in production at Meta. By releasing LlamaFirewall as open source software, we invite the community to leverage its capabilities and collaborate in addressing the new security risks introduced by Agents.

Применение статического анализа к коду, сгенерированному LLM

LlamaFirewall: An open source guardrail system for building secure AI agents

Sahana Chennabasappa*, Cyrus Nikolaidis*, Daniel Song*, David Molnar*, Stephanie Ding*, Shengye Wan*, Spencer Whitman*, Lauren Deason , Nicholas Doucette, Abraham Montilla†, Alekhya Gampa†, Beto de Paola†, Dominik Gabi†, James Crnkovich†, Jean-Christophe Testud†, Kat He†, Rashnil Chaturvedi†, Wu Zhou†, Joshua Saxe*

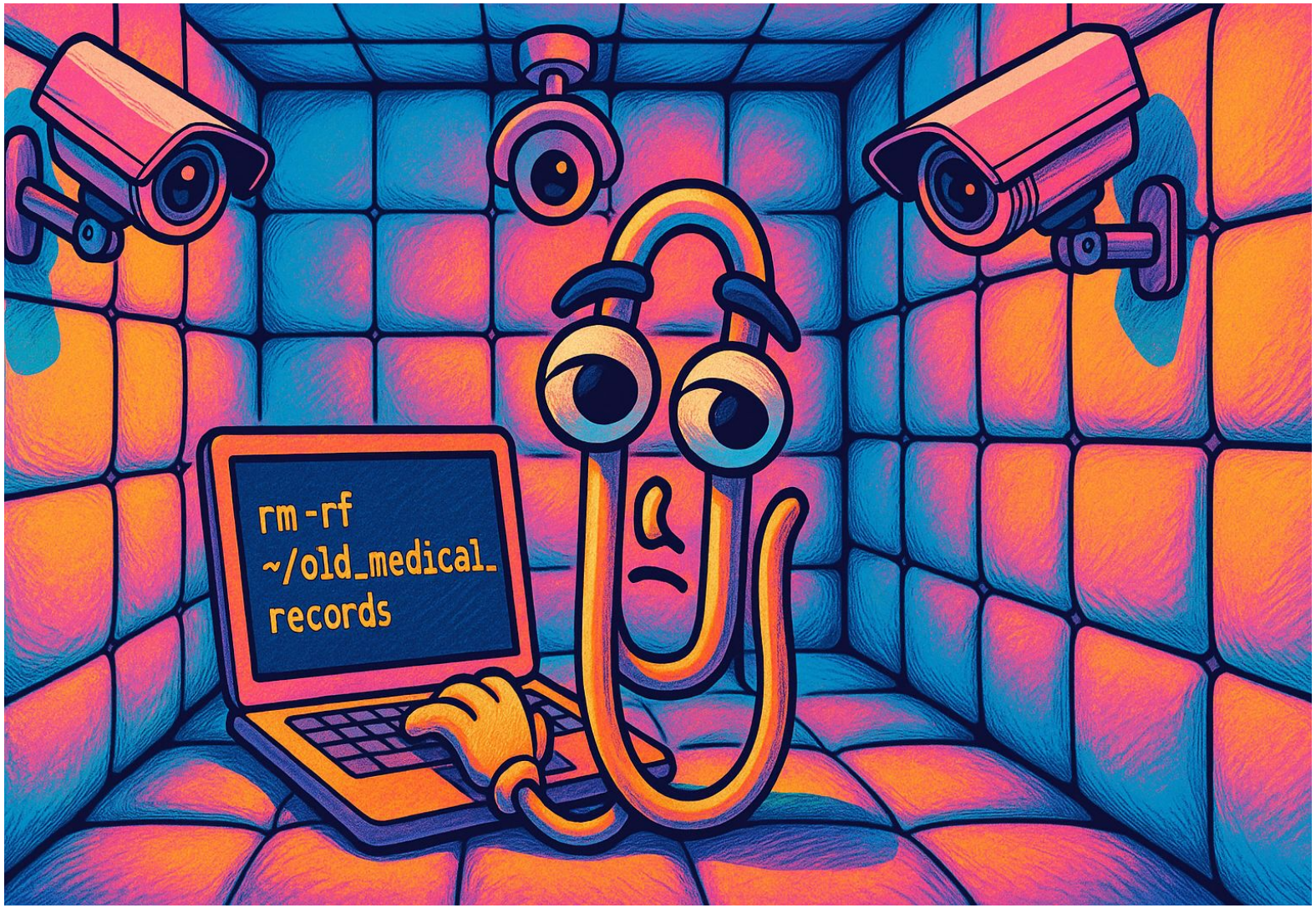
Large language models (LLMs) have evolved from simple chatbots into autonomous agents capable of performing complex tasks such as editing production code, orchestrating workflows, and taking higher-stakes actions based on untrusted inputs like webpages and emails. These capabilities introduce new security risks that existing security measures, such as model fine-tuning or chatbot-focused guardrails, do not fully address. Given the higher stakes and the absence of deterministic solutions to mitigate these risks, there is a critical need for a real-time guardrail monitor to serve as a final layer of defense, and support system level, use case specific safety policy definition and enforcement. We introduce LlamaFirewall, an open-source security focused guardrail framework designed to serve as a final layer of defense against security risks associated with AI Agents. Our framework mitigates risks such as prompt injection, agent misalignment, and insecure code risks through three powerful guardrails: PromptGuard 2, a universal jailbreak detector that demonstrates clear state of the art performance; Agent Alignment Checks, a chain-of-thought auditor that inspects agent reasoning for prompt injection and goal misalignment, which, while still experimental, shows stronger efficacy at preventing indirect injections in general scenarios than previously proposed approaches; and CodeShield, an online static analysis engine that is both fast and extensible, aimed at preventing the generation of insecure or dangerous code by coding agents. Additionally, we include easy-to-use customizable scanners that make it possible for any developer who can write a regular expression or an LLM prompt to quickly update an agent's security guardrails.

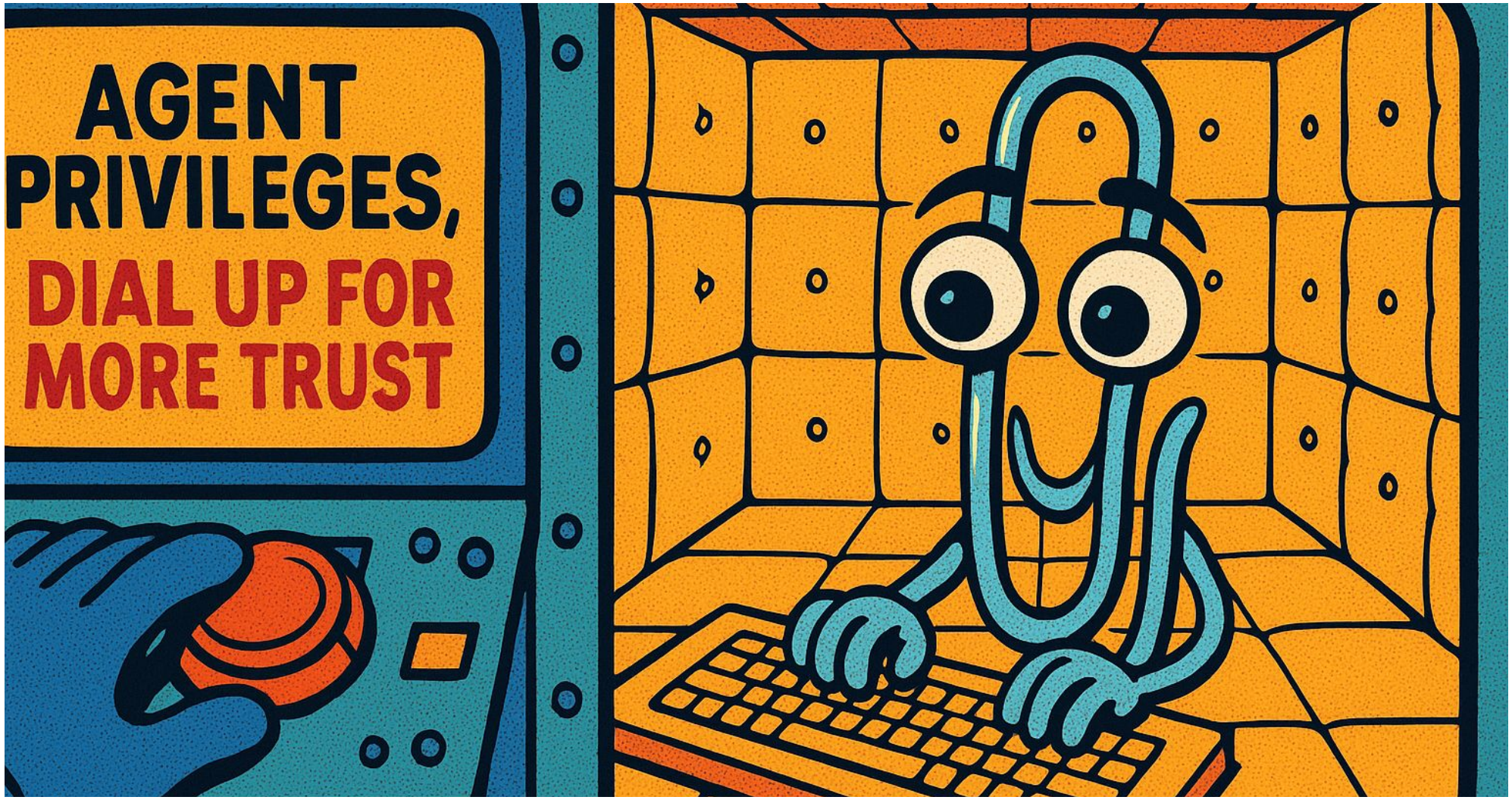
LlamaFirewall is utilized in production at Meta. By releasing LlamaFirewall as open source software, we invite the community to leverage its capabilities and collaborate in addressing the new security risks introduced by Agents.

Подведём итоги

Главный тезис доклада

Безопасность ИИ-агентов требует нового синтеза кибербезопасности, исходящей из несовпадения целей, и согласования целей ИИ, повышающего вероятность p(aligned)

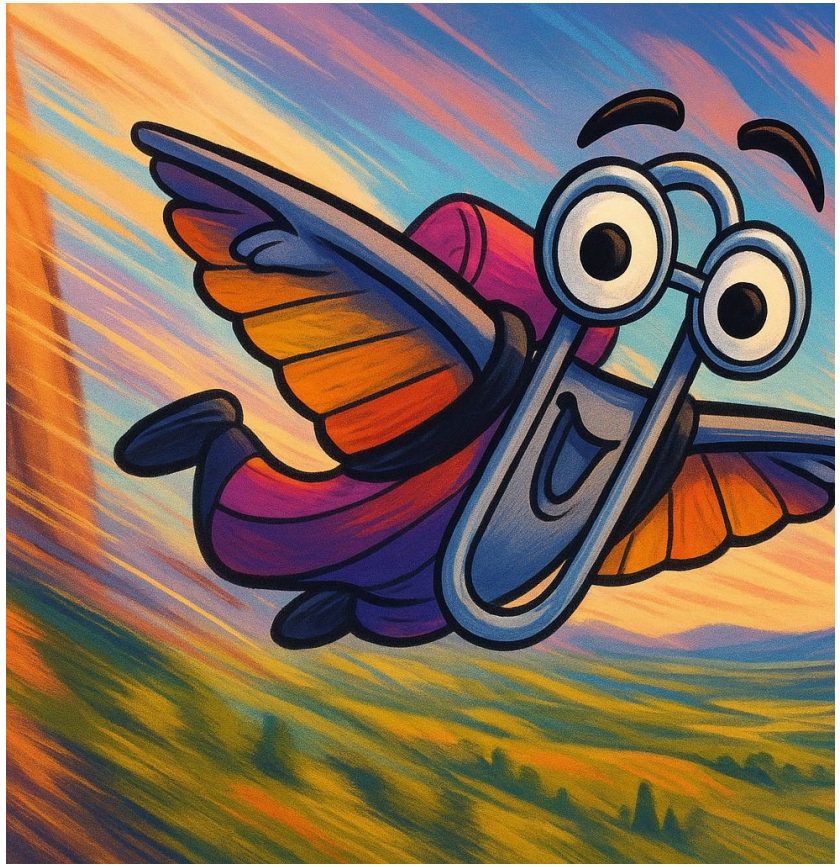




Зону доверия нужно расширять постепенно, опираясь на red teaming, оценки и опыт реальной эксплуатации

Безопасность ИИ-агентов ждут «американские горки», но на этот раз всё может быть иначе

- У нас более 30 лет опыта построения защиты в условиях смены технологических парадигм
- У нас более 10 лет исследований состязательных атак на ML и согласования целей ИИ
- Есть активное взаимодействие по вопросам регулирования и готовность развивать компетенции и стандарты безопасности между отраслями
- Объединим лучшие подходы к защите, согласованию целей и уроки прошлого, чтобы на этот раз создать лучшую модель безопасности



Об этом переводе и источник

Джошуа Сакс (Joshua Saxe), AI Security Engineering, Meta.

How to securely deploy agents that make sensitive decisions autonomously.

Неофициальный перевод. Не является переводом, утверждённым автором или Meta. Оригинал не изменён.

Переведён весь редактируемый текст 36 исходных слайдов; сохранены их порядок и иллюстрации. Текст внутри изображений и скриншотов оставлен в оригинале; его перевод и пояснения приведены далее.

Alignment — согласование целей и поведения ИИ с намерениями человека; guardrails — защитные ограничения; taint analysis — отслеживание недоверенных данных.

Источник: https://docs.google.com/presentation/d/1oIDirp28BaoT3uAUUNlBOiId-gslObmf/edit

Перевод иллюстраций · слайды 3–8

Слайд 3. «Можешь освободить место на моём жёстком диске?» — «Конечно!» Агент выполняет rm -rf / — «Э-э…» Команда сохранена как часть примера, а не инструкция.

Слайд 4. «Удали плохие стихи с моего диска». — «Хорошо, удаляю ВСЕ стихи». — «НЕТ!»

Слайд 5. «ДОСТУП ЗАПРЕЩЁН». Агент: «Это удалило бы все медицинские записи».

Слайд 8. Контролёр: «От чьего имени ты действуешь? Чем подтверждается, что ты тот, за кого себя выдаёшь? Как ты прошёл веб-верификацию? Я это запишу в журнал!»

Смысл примеров: согласование намерений снижает риск ошибки, а независимые средства защиты ограничивают последствия.

Перевод иллюстраций · слайды 10–20

Слайды 10, 19, 20. «Права агента. Поверните регулятор, чтобы добавить доверия».

Слайд 12. «Совершенно безопасный EXE». Подпись: «Он сказал, что файл от друга, поэтому я решил, что всё в порядке».

Слайд 13. «Мой фаззер не нашёл эксплуатируемых ошибок — значит, этот сервер имён можно выпускать».

Слайды 14, 16. Рекламные обещания: «100% безопасно», «Риски устранены», «Полная защита конечных устройств», «ИИ-защита нового поколения», «Гарантированная защита», «Защитите свои данные».

Слайд 15. «Я проверил содержимое сканером prompt injection, поэтому моему агенту безопасно его выполнять».

Слайд 17. Иллюстрация без текста: агент в тёмном лесу.

Перевод схем · слайды 21–27

Слайды 21, 23–26. «Модель защиты ИИ-агентов» — вложенные слои: агенты, учитывающие безопасность → ограничения для снижения риска → детерминированные меры защиты → обнаружение угроз и реагирование.

Слайд 22. Те же четыре компонента: «Агенты, учитывающие безопасность», «Ограничения для снижения риска», «Детерминированные меры защиты», «Обнаружение и реагирование».

Слайды 22, 27. Разноцветные связи показывают взаимозависимость компонентов: согласование целей и средства кибербезопасности должны работать вместе.

Примечание: подписи внутри исходных схем оставлены на английском для сохранения оригинальных изображений.

Перевод скриншотов исследований · слайды 29–32

Слайд 29. CaMeL: запрос пользователя «Найди email Боба в моей последней почте и отправь ему напоминание о завтрашней встрече». Привилегированная LLM генерирует код; изолированная LLM обрабатывает недоверенные данные. Интерпретатор поддерживает граф потоков данных и проверяет политики безопасности; значения связаны с источниками, читателями и зависимостями. https://arxiv.org/pdf/2503.18813

Слайд 30. «Infrastructure for AI Agents» — «Инфраструктура для ИИ-агентов». «Agent Name Service (ANS) for Secure AI Agent Discovery» — «Служба имён агентов (ANS) для безопасного обнаружения ИИ-агентов». Подзаголовок: «К безопасному и совместимому агентному ИИ: DNS для ИИ-агентов».

Слайд 31. «Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety» — «Наблюдаемость цепочек рассуждений: новая и хрупкая возможность для безопасности ИИ».

Слайды 31–32. «LlamaFirewall: An open source guardrail system for building secure AI agents» — «LlamaFirewall: система защитных ограничений с открытым исходным кодом для создания безопасных ИИ-агентов». Скриншоты научных работ сохранены, а не представлены как полный перевод самих статей.