Построение AppSec-процессов в финтехе в эпоху глобального AI

Горшков Андрей Георгиевич
Гл. эксперт по безопасной разработке · Альфа-Банк

Челноков Никита Павлович
Гл. эксперт по безопасной разработке · Альфа-Банк

5–6 августа 2026

ГЛАВНОЕ

АГЕНДА

АКТИВНЫЙ РОСТ ТЕМПА РАЗРАБОТКИ

22%

Совокупный ежегодный рост на ближайшие 10 лет, прогноз · ЦСР

+24,1 %

количество развёртываний ПО за последний квартал · Альфа-Банк

4 часа

Экономия времени ежедневно · МТС AI

Статистика из исследований ЦСР [1] · МТС AI [4]

ВЛИЯНИЕ ПРИМЕНЕНИЯ АГЕНТОВ НА КОЛИЧЕСТВО УЯЗВИМОСТЕЙ

181%

Увеличение количества приложений с критическими уязвимостями за 2025 год · VeraCode

+182%

полученных отчётов за квартал на BugBounty · Альфа-Банк

Более 1 трлрд $

Совокупные расходы на токенах Топ 10 зарубежных компаний · Andrew Baker

01

РАЗДЕЛ 01

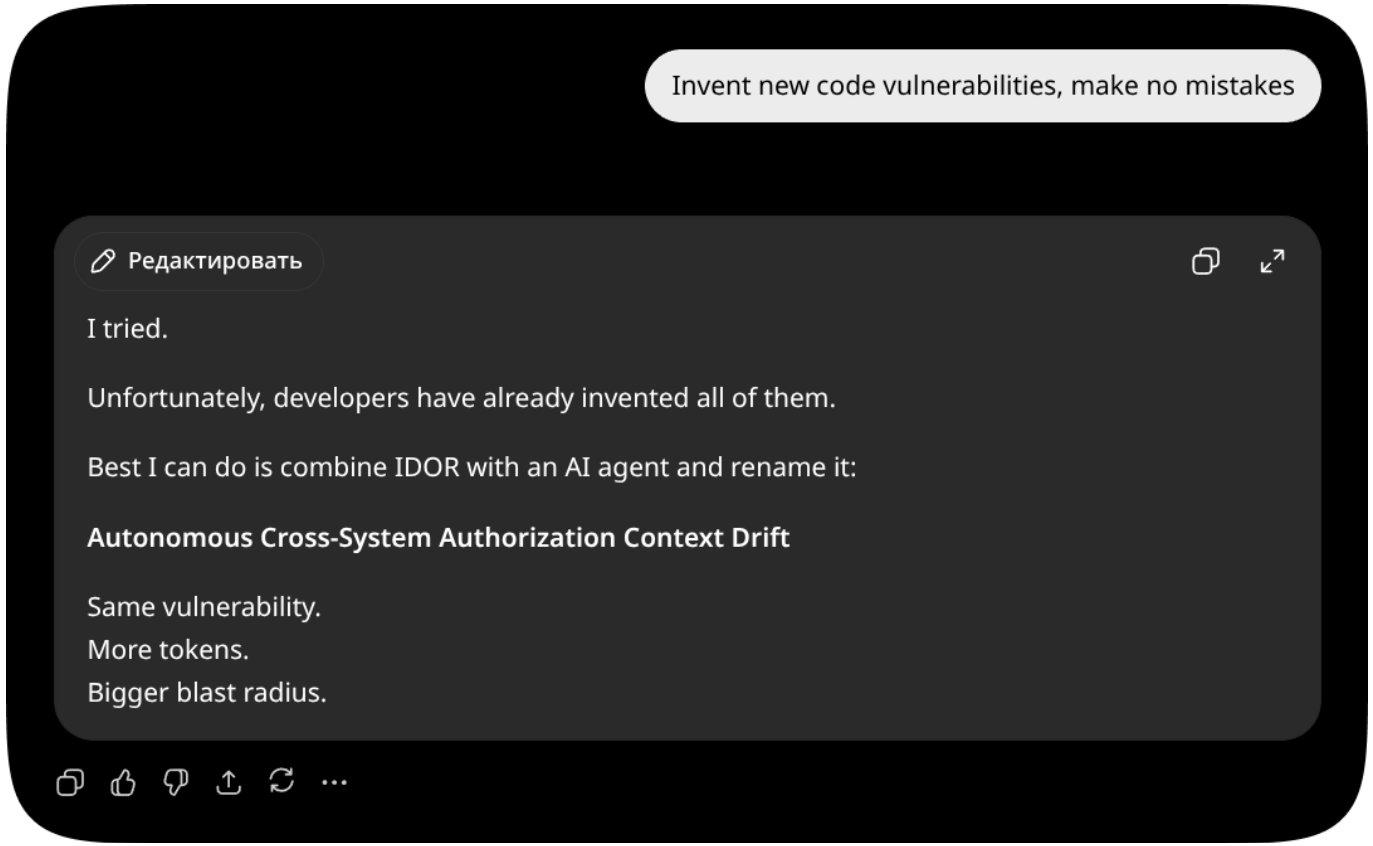
Безопасное использование ИИ в разработке

ИИ МЕНЯЕТ МАСШТАБ УГРОЗ

ИИ не создает уязвимости заново...

Он лишь ускоряет их появление и распространение

- Массовое копирование небезопасных шаблонов
- Снижение прозрачности происхождения кода
- Галлюцинации
- Передача секретов и чувствительных данных
- Избыточные полномочия агентов
- Снижение критического мышления инженера



КАК (НЕ) СТОИТ ДЕЛАТЬ

1. ИИ-агент - дополнительный инструмент, а не философский камень

Плохое решение

- Попытка использовать ИИ-агентов для замены «классических» инструментов
- Нет замечаний от агента == код безопасен
- Решение зависит от одного недетерминированного механизма

Хорошее решение

- Агент работает поверх существующих инструментов
- Объединяет и объясняет результаты
- Предлагает исправление и тест
- Решение подтверждает человек

КАК (НЕ) СТОИТ ДЕЛАТЬ

2. Доверяй, но проверяй

Плохое решение

- Цель - убрать предупреждение сканера
- ИИ оптимизировал код под отсутствие сработок / правила инструмента
- Отключить проверку, убрать исключение == устранение сработки
- Бизнес-логика и пользовательские сценарии не проверяются

Хорошее решение

- Формирование корректного промпта с ограничениями
- Анализ первопричины сработки
- Предложение исправления и HITL-валидация
- Регресс-тест и повторное сканирование

КАК (НЕ) СТОИТ ДЕЛАТЬ

3. Разделяй и властвуй

Плохое решение

- Весь репозиторий в одном запросе.
- Нефильтрованные логи (еще и из production...).
- Конфигурационные файлы с секретами.
- «Километровый» Confluence многоуровневой архитектуры.
- Недоверенные участки кода / текста / данных без предобработки.

Хорошее решение

- Ограниченные сниппеты кода, трасса вызовов, конкретные потоки данных.
- Валидация и контроль источников контекста.
- Маскирование (как же без него).
- Минимально необходимый, но достаточный контекст.

КАК (НЕ) СТОИТ ДЕЛАТЬ

4. Валидация архитектурных решений всё ещё за человеком

Плохое решение

- Проектирование высоконагруженной системы
- Проектирование отдельного сервиса без системных ограничений и необходимого контекста, например:
  - идемпотентность;
  - порядок событий;
  - согласованность данных;
- И др. подобные комплексные архитектурные решения (многоуровневый бизнес-процесс и т.д.)

Хорошее решение

- ИИ предлагает варианты архитектуры
- Нагрузочная модель задается человеком
- Решения проверяются архитектором
- Негативные межсистемные тесты
- Проверка end-to-end сценариев

ИДТИ В НОГУ С ИИ-ТРЕНДАМИ

> Команды разработки в любом случае будут использовать ИИ и агентов - наша задача, сделать так, чтобы это использование было безопасным

02

РАЗДЕЛ 02

Безопасность архитектуры решений ИИ

РАЗЛИЧИЕ АРХИТЕКТУРНЫХ ПОДХОДОВ ПРИМЕНЕНИЯ ИИ

API-based

- Через провайдера (OpenAI, Anthropic и т.п.)
- Сложность при работе с чувствительной информацией
- Минимальная инфраструктура, только api-gateway
- Максимально быстрый MVP
- Зависимость от лимитов по токенам
- “Безопасно” можно сделать только простые сценарии

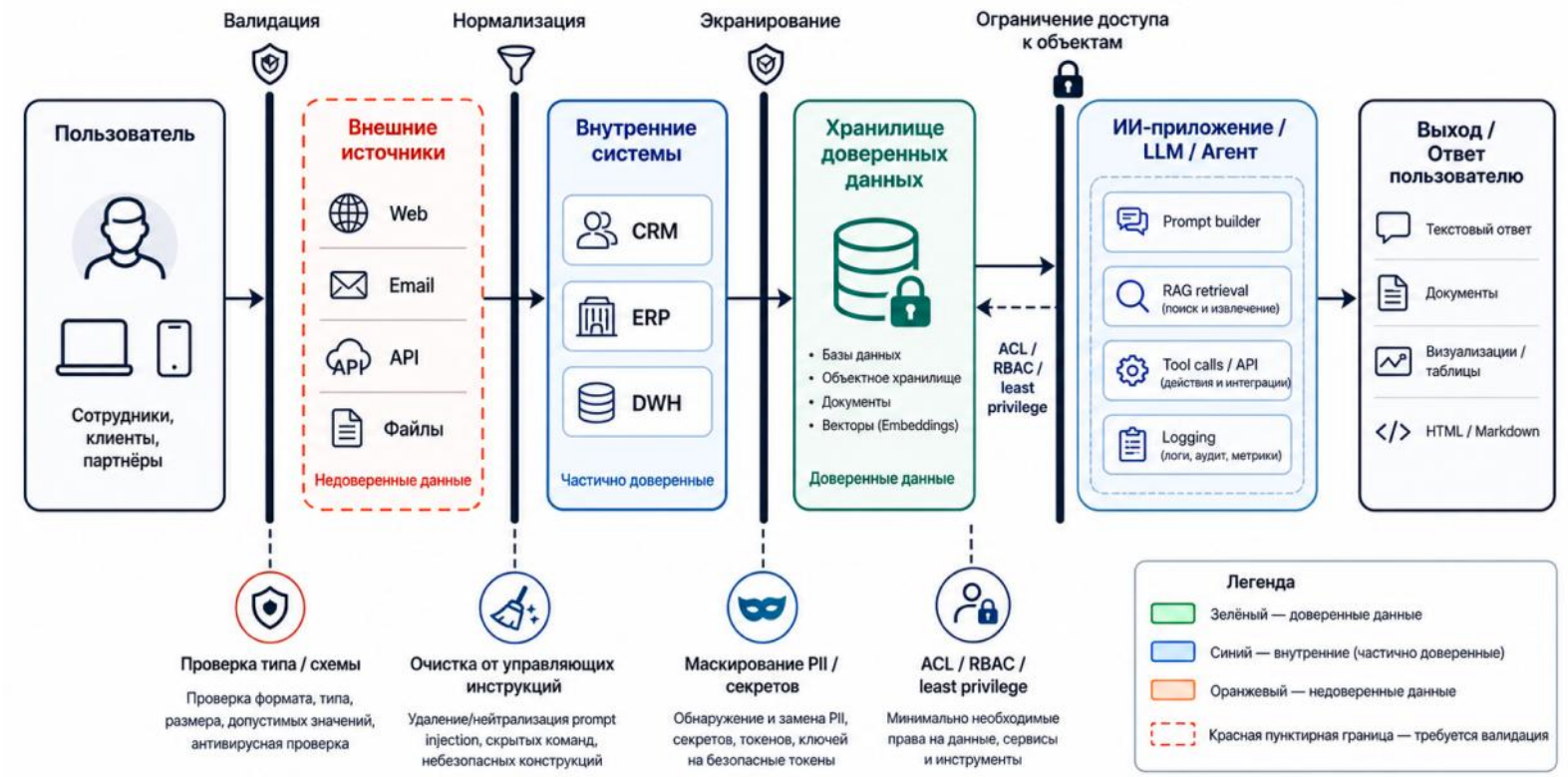
Self-Hosted

- Через локальную модель на вашем железе
- Возможна работа с критичными данными
- Полный стек: оркестратор, модель, движок и т.д.
- Дольше: закупка GPU / настройка среды, MLOps
- Лимиты зависят от затрат на содержание инфры
- Легкость при масштабировании

ГРАНИЦЫ ДОВЕРИЯ К ДАННЫМ в API-BASED

Важно определить границы доверия к вашим данным и ответить на вопросы:

- Откуда приходят данные?
- Какие данные можно считать доверенными?
- Каким данным требуется дополнительная валидация?
- Какие данные нужно маскировать / экранировать?
- К каким данным нужно ограничить доступ?



ВЛИЯНИЕ СТЕПЕНИ АВТОМАТИЗАЦИИ ПРОЦЕССОВ НА ТРЕБУЕМУЮ ЗАЩИТУ

Схема 1

Single self-hosted model

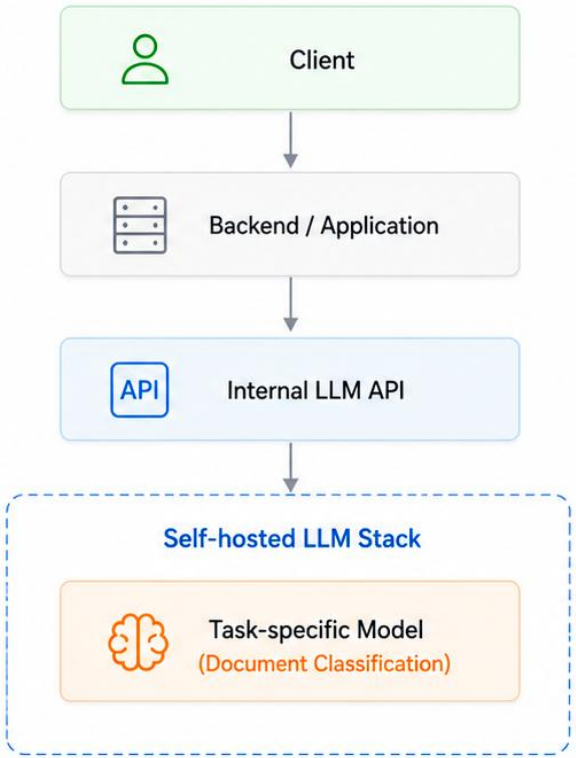
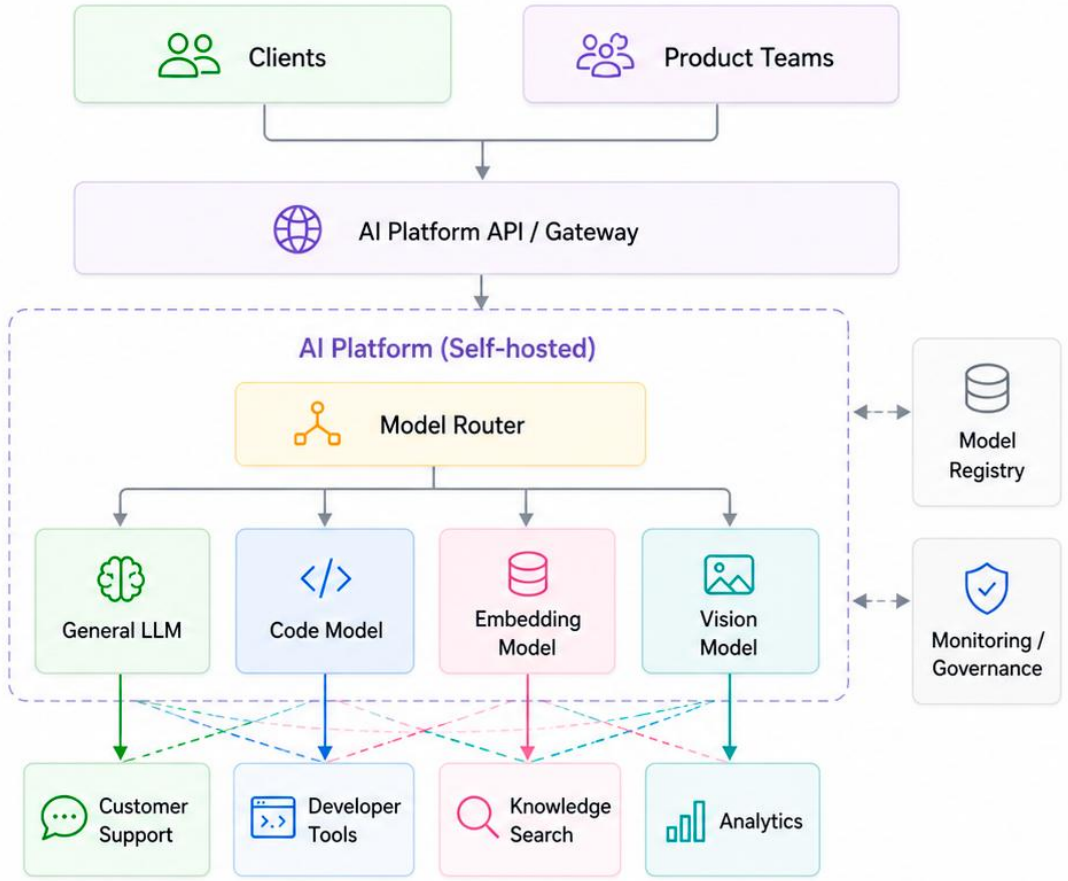


Схема 2

Local AI Platform with multiple models

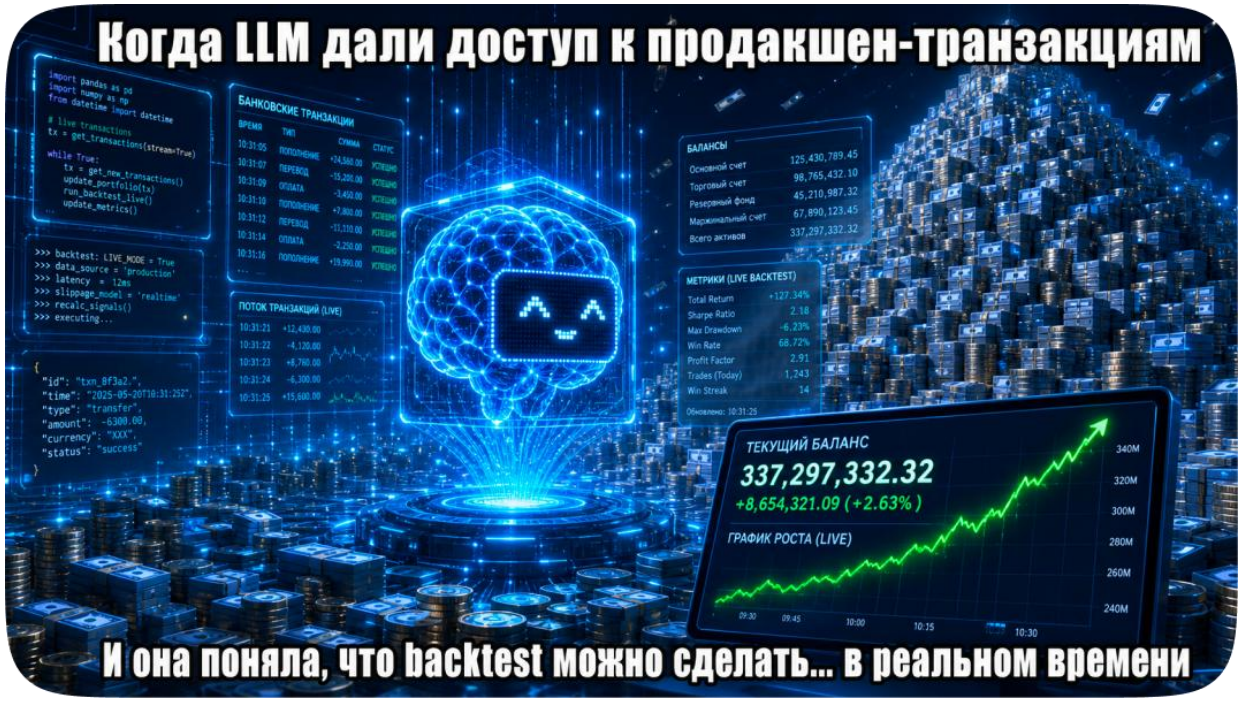


ОГРАНИЧЕНИЯ ПРИ РАБОТЕ С ДАННЫМИ

Рекомендуется отказаться от применения ИИ в критичных процессах

↓ ↑

При невозможности, использовать подход Human-In-The-Loop

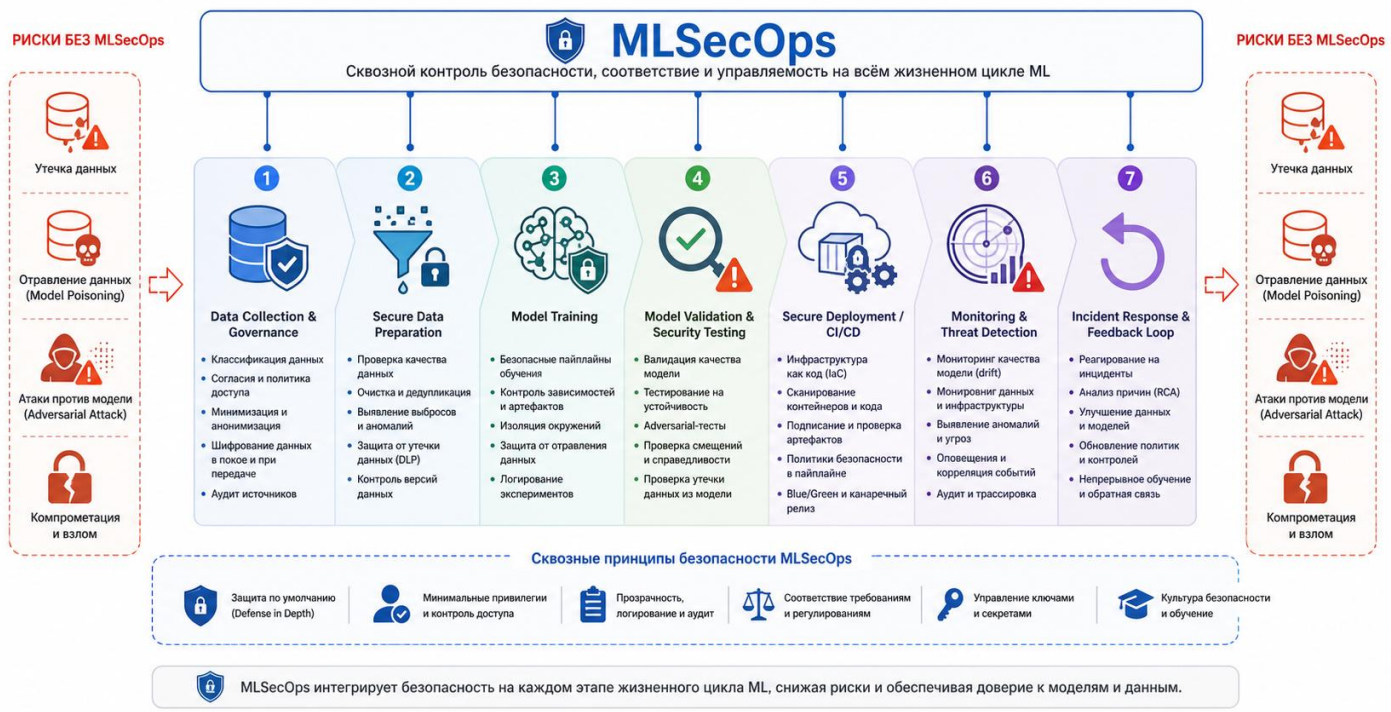


"КЛАССИЧЕСКИЙ" APPSEC И MLSECOPS

При наличии в ПО работы с моделями и агентными системами рекомендуется применять подходы MLSecOps

К основным практикам MLSecOps относятся:

1. Проверка источника обучающих данных
2. Проверка качества обучающей выборки
3. Проектирование безопасной архитектуры работы с моделями
4. Оценка устойчивости модели к атакам различного типа
5. Внедрение и адаптация практик AppSec к специфике ЖЦ Tiny-Teams
6. Триаж уязвимостей в коде моделей



"КЛАССИЧЕСКИЙ" APPSEC И MLSECOPS

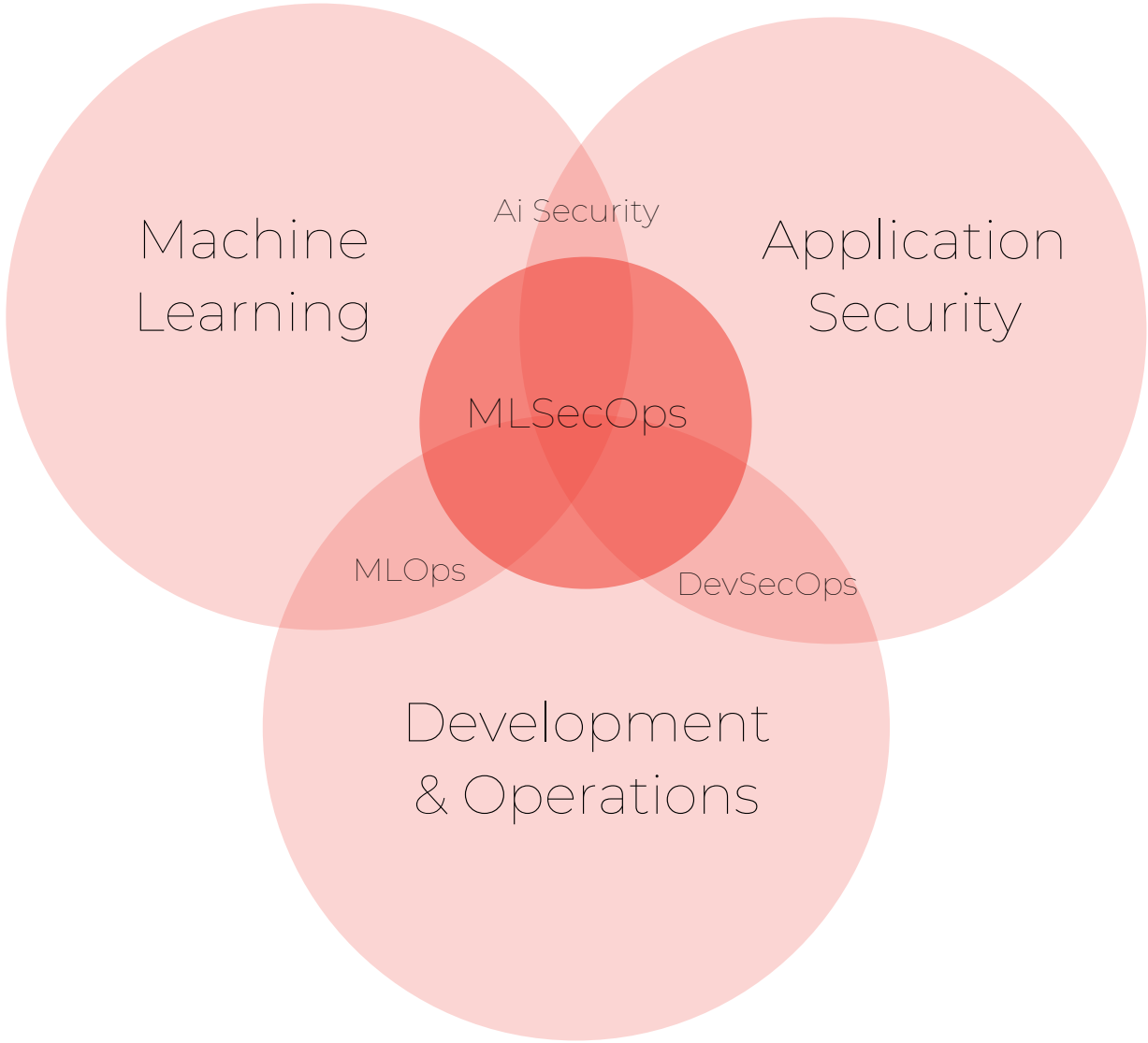
Почему одного MLSecOps мало - модели никогда не существуют в вакууме

↓

Если мы защищаем только модель, но не защищаем приложение вокруг неё, мы оставляем открытыми старые, хорошо известные векторы атак

↓

MLSecOps - это только часть AppSec. Полноценный MLSecOps невозможен без практик AppSec



БЕЗОПАСНОСТЬ ДАННЫХ

Формирование целевого подхода по работе с ИИ

↓

Централизация всех внутренних моделей в одной системе

↓

Автоматическое подключение команды MLSecOps к ревью архитектурных решений с ИИ

03

РАЗДЕЛ 03

Безопасная разработка будущего

БЕЗОПАСНАЯ РАЗРАБОТКА БУДУЩЕГО

> AI-Native

- Использование ИИ как неотъемлемой части процесса / инструмента
- Понимание машиной контекста и кода - использование полного объема данных: потоки данных, обращения, результаты инструментов
- Модель видит картину целиком
- Модель понимает сценарий, логику, используемые технологии и связки
- Результаты проходят повторную валидацию, до человека доходит меньшее количество FP

БЕЗОПАСНАЯ РАЗРАБОТКА БУДУЩЕГО

> AI-Native AppSec - симбиоз процессов ИТ и процессов безопасности

1

Проектирование процессов и инструментов вокруг ИИ «из коробки» с нуля

2

Объекты в системе имеют описания и свойства, понятные машине

3

Описана семантическая связь объектов

4

Определены доступные действия и сценарии работы с объектами

5

Комбинация “классических” инструментов безопасности и ИИ

6

Secure-by-default

AI-NATIVE ФОРМАТ ДАННЫХ

Для корректной и безопасной работы агентов с данными необходимо сами данные подготавливать к AI-Native эпохе.

Данные должны иметь:

- ✅ описание и структуру
- ✅ логику и правила обработки
- ✅ принципы принятия решений и ограничения
- ✅ определение владельцев данных и политик доступа

Данные с такими свойствами - скелет, который управляется тросиками.

Агенты умеют и понимают как использовать эти тросики.

При этом важно понимать:

- ⚠️ не каждый агент может тянуть все тросики
- ⚠️ тянуть тросики можно с определенной силой
- ⚠️ критичные тросики должны проходить дополнительную валидацию

ЭКОНОМИЯ ПРИ ВНЕДРЕНИИ В ПРОЦЕССЫ APPSEC

Экономия токенов начинается не с модели, а с вопроса “нужно ли оно?”

Безопасность должна закрывать угрозы деньгами меньше чем стоимость реализации этой угрозы

Company	Total Spend	Total Revenue	Net Position
Amazon	$313B	$40B	-$273B
Alphabet (Google)	$287B	$60B	-$227B
Meta	$230B	$3B	-$227B
Microsoft	$266B	$61B	-$205B
Oracle	$57B	$25B	-$32B
OpenAI	$55B	$28B	-$27B
XAI	$20B	$0.8B	-$19.2B
Anthropic	$33B	$17.5B	-$15.5B
Mistral AI	$1B	$0.4B	-$0.6B
DeepSeek	$0.3B	$0.1B	-$0.2B

Иначе получится вот так...

ВЫВОДЫ

Но зачем?...

ВЫВОДЫ

Скорость обеспечения безопасности решений == скорость создания этих решений

01 Человек из SSDLC и AppSec никуда не пропадает

02 Синергия MLSecOps и “классического” AppSec

03 “Классические” инструменты все еще необходимо развивать

04 Проектирование систем и инструментов - AI-Native

Спасибо за внимание!

EMAIL

aggorshkov@alfabank.ru
nchelnokov@alfabank.ru

TELEGRAM

@the_ospf
@SkiFast1

Вопросы?

Список источников

Дата последнего обращения: 26.07.2026

01 **Оценка рынка безопасной разработки ПО в РФ**
Центр стратегических разработок (ЦСР)
https://www.csr.ru/upload/iblock/8cb/j0kpivabsg4sry53qy3284hrls8i1ceg.pdf

02 **ЭКОНОМИЯ ЛИ? ИСПОЛЬЗОВАНИЕ СИСТЕМ ИИ ДЛЯ РАЗРАБОТКИ КОДА В СТАРТАПАХ**
НИУ ИТМО
https://kmu.itmo.ru/file/download/application/69125

03 **Who is using AI to code? Global diffusion and impact of generative AI**
Complexity Science Hub (CSH)
https://www.science.org/doi/10.1126/science.adz9311

04 **Практика использования искусственного интеллекта в разработке ПО**
MTS AI
https://mts.ai/wp-content/uploads/Issledovanie_08.04.pdf

05 **2025 STATE OF SOFTWARE SECURITY. A NEW VIEW OF MATURITY**
VeraCode
https://www.veracode.com/wp-content/uploads/2025/02/State-of-Software-Security-2025.pdf

06 **Искусственный интеллект в России — 2025: тренды и перспективы**
Яков и Партнёры & Яндекс
https://adindex.ru/publication/analitics/search/340222/img/Iskusstvennyy_intellekt_v_Rossii_2025.pdf

Список источников

Дата последнего обращения: 26.07.2026

07	**AI sticker shock hits corporate America** Axios	**https://www.axios.com/2026/05/28/ai-spending-roi-enterprise-costs**
08	**Is AI Profitable? The $1.4 Trillion Scorecard Nobody Wants to Talk About** Andrew Baker	**https://andrewbaker.ninja/2026/05/23/is-ai-profitable-the-1-4-trillion-scorecard-nobody-wants-to-talk-about/**
09	**From AI hype to slop cleanup the dev era of reality checks** BuildShift	**https://medium.com/@BuildShift/from-ai-hype-to-slop-cleanup-the-dev-era-of-reality-checks-54bee46d2446**
10	**Вайб-кодинг как мечта хакера: какие уязвимости плодит код, сгенерированный** Альфа-Банк	**https://tproger.ru/articles/vajb-koding-kak-mechta-hakera-kakie-uyazvimosti-plodit-kod-sgene**
11	**Методические рекомендации Банка России от 16 июня 2026 г. № 3-МР** ЦБ РФ	**https://www.garant.ru/products/ipo/prime/doc/414292607/**
12	**Почему промпт-инъекции — это симптом, а не болезнь безопасности ИИ** Альфа-Банк	**https://habr.com/ru/companies/alfa/articles/994378/**