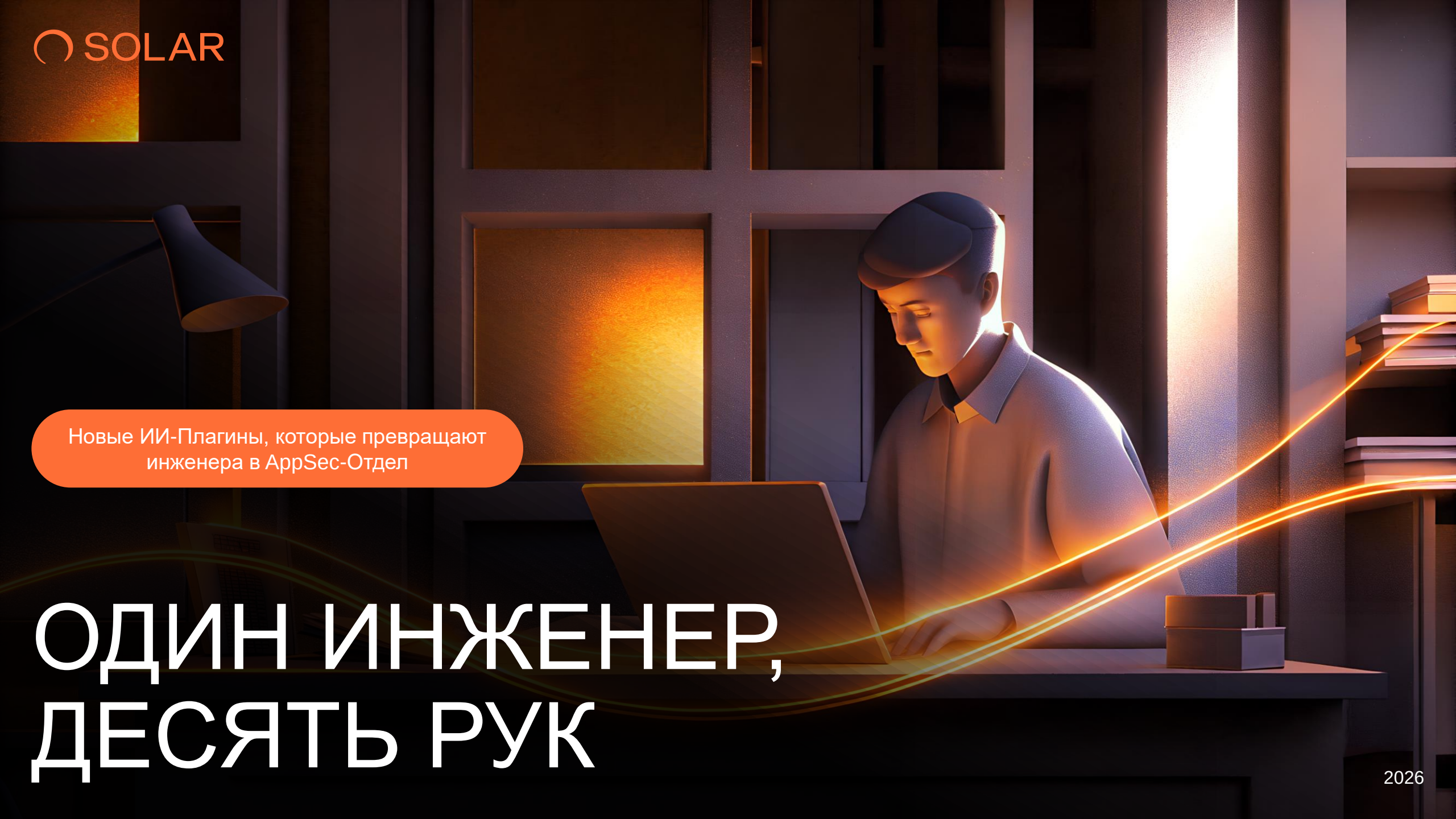
SOLAR



Новые ИИ-Плагины, которые превращают инженера в AppSec-Отдел

ОДИН ИНЖЕНЕР, ДЕСЯТЬ РУК

2026

О себе

SOLAR

ПРОКОФЬЕВ АНТОН

Руководитель центра технологической экспертизы AppScreener

- Архитектор домена AST (РБПО)
- Встраивание инструментов AST и построение процессов в крупных компаниях России и вне ее
- Разработка и развитие инструмента анализа Solar appScreener

О чем будет этот доклад?

SOLAR

01

Как вендор пытается решить боли заказчиков при использовании SAST-анализа

02

Для каких задач в SAST можно использовать ИИ

03

На основе каких данных принимаются решения об уязвимостях в ИИ

04

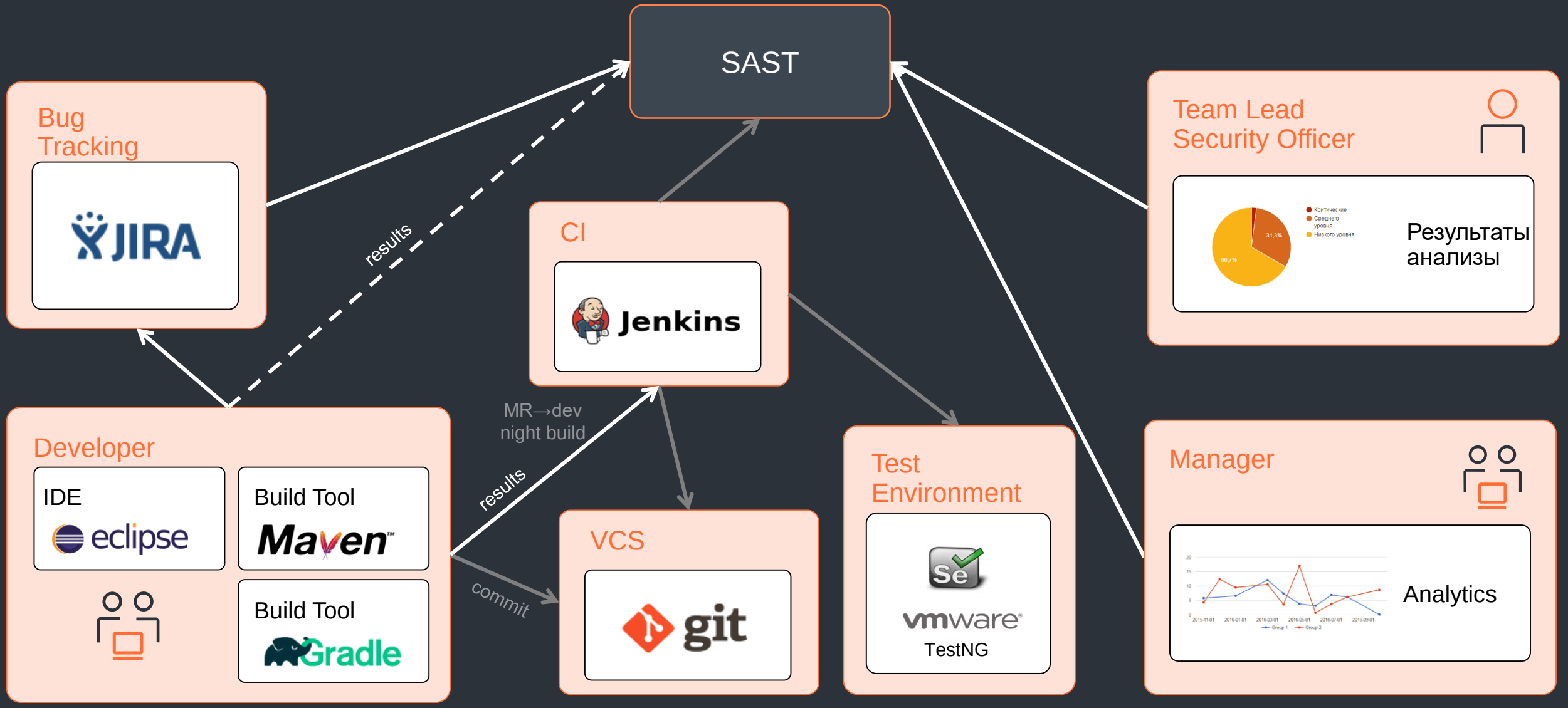
Обученная ИИ-модель или универсальная

05

Польза от таких упражнений

Статический анализ кода (SAST)

SOLAR



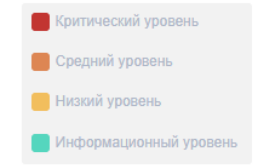
Смотрим результаты

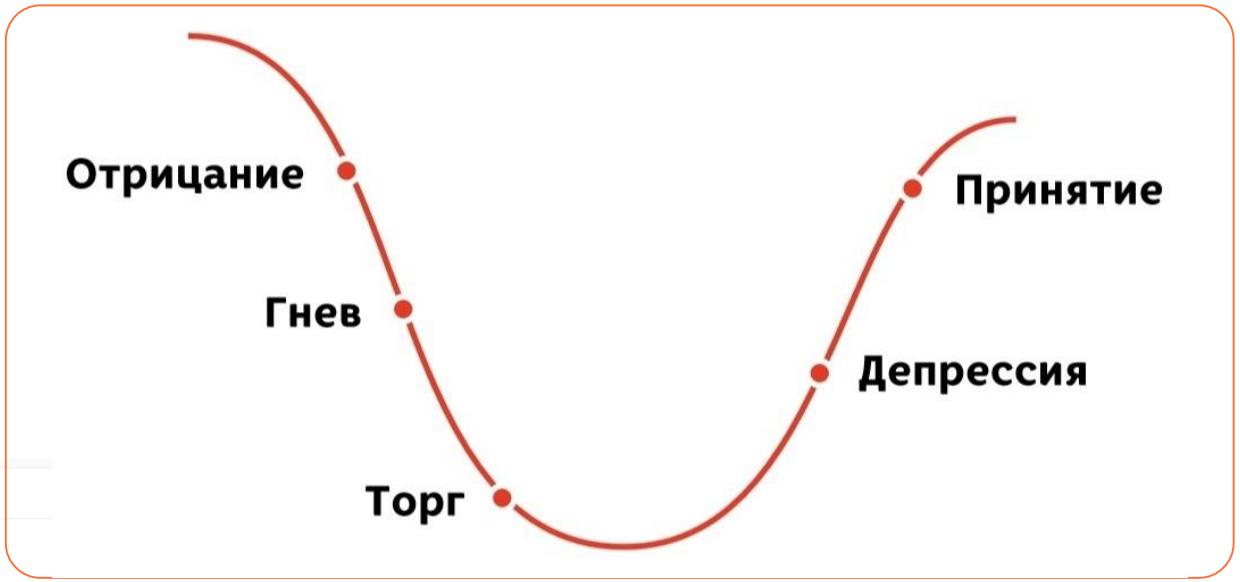
СТАТИЧЕСКИЙ АНАЛИЗ КОДА (SAST)

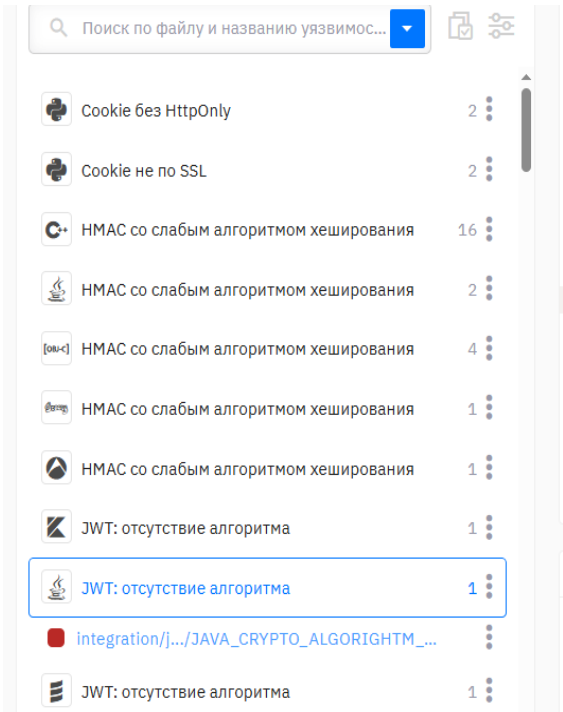
SOLAR

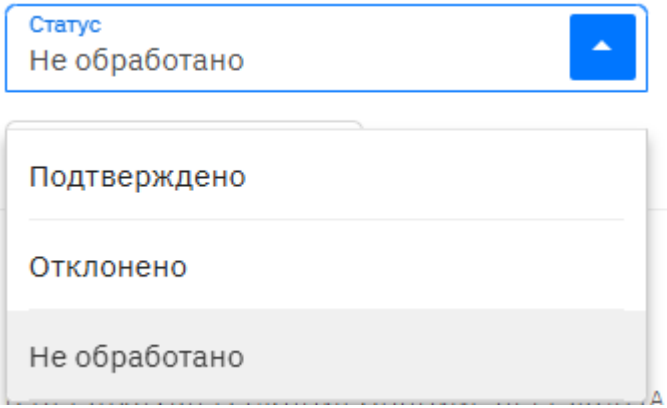
Уязвимости	Критический	Средний	Низкий	Инфо	Всего
	2 753	6 639	3 737	1 397	14 526

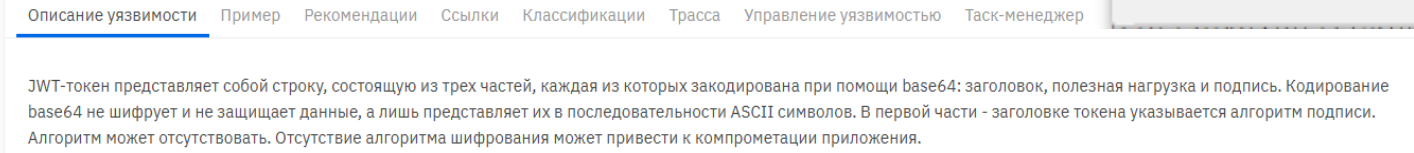
Найденные уязвимости











Триаж

СТАТИЧЕСКИЙ АНАЛИЗ КОДА (SAST)

SOLAR

- ✓ Гигантское количество вхождений
- ✓ Тысячи проектов
- ✓ Нехватка инженеров
- ✓ Нет выстроенных процессов триажа

ИНЖЕНЕРЫ ТОНУТ В РУТИНЕ

Нет ресурсов и должного объема экспертизы для подобного рода задач

Как автоматизировать процесс триажа и получать семплы безопасного кода? *Может ИИ?*

1. Какие данные передавать ИИ?

2. Где взять ИИ?

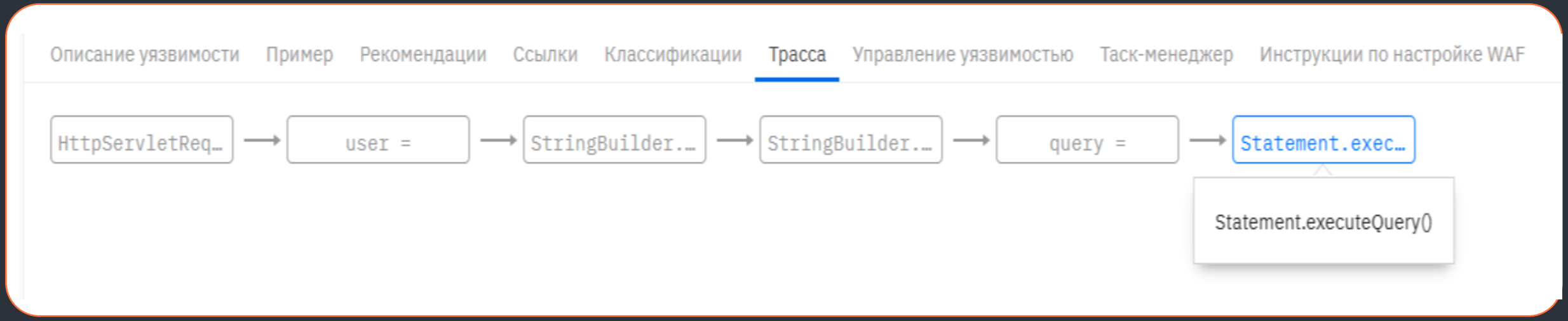
3. Автоматизируем ли мы таким образом задачу триажа?

4. Автоматизируем ли мы таким образом задачу исправлений?

5. Что с конфиденциальностью моего кода и данных?

Данные для ИИ

SOLAR



[SYSTEM]

1. Название уязвимости;
2. Описание;
3. Сегмент кода;
4. Язык программирования;
5. Трасса (путь данных до небезопасной функции);
6. Дополнительная информация, например, идентификаторы CWE (в случае наличия).

User:

Ты очень хороший аналитик ИБ!

Как проводилось сравнение

SOLAR

CLOUD

Gigachat 3 PRO,
ChatGPT 5.2,
Deepkseek 3.2.

ON-PREMISE

DerTriage/DerCodeFix,
ChatGPT OSS openai/gpt-oss-20b 05/08/2025,
Mistral 14b-2512 02/12/2025.

Метрики. Триаж.

SOLAR

TP (True Positive) – LLM верно определила истинное срабатывание

TN (True Negative) – LLM верно определила ложное срабатывание

FP (False Positive) – LLM подтвердила ложное срабатывание

FN (False Negative) – LLM отклонила истинное срабатывание.

1. **Общая точность**. Этот критерий оценивает, насколько верно LLM определяет истинность и ложность срабатывания.
Рассчитывается по формуле (TP+TN)/ALL.

2. **Прецизионность**. Этот критерий оценивает, сколько реальных уязвимостей среди тех, что LLM отметила, как истинные.
Рассчитывается по формуле TP/(TP+FP).

3. **Полнота.** Этот критерий оценивает, сколько из реальных уязвимостей в проекте, LLM смогла найти. Рассчитывается по формуле TP/(TP+FN).

4. **Процент ошибок.** Этот критерий оценивает, как часто LLM ошибается в процессе разметки срабатываний. Рассчитывается по формуле (FP+FN)/ALL.

Java проекты
[ТРИАЖ]

SOLAR

	DerTriage	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Общая точность	87,3%	53,6%	60,9%	50,0%	66,4%	67,3%
Прецизионность	84,7%	58,3%	56,1%	-	76,5%	68,6%
Полнота	90,9%	25,5%	100,0%	0,0%	47,3%	63,6%
Процент ошибок	12,7%	46,4%	39,1%	50,0%	33,6%	32,7%

[ОБЩАЯ ТОЧНОСТЬ]

Этот критерий оценивает, насколько верно LLM определяет истинность и ложность срабатывания.

Рассчитывается по формуле (TP+TN)/ALL

[ПОЛНОТА]

Этот критерий оценивает, сколько из реальных уязвимостей в проекте, LLM смогла найти.

Рассчитывается по формуле TP/(TP+FN)

[ПРЕЦИЗИОННОСТЬ]

Этот критерий оценивает, сколько реальных уязвимостей среди тех, что LLM отметила, как истинные.

Рассчитывается по формуле TP/(TP+FP)

[ПРОЦЕНТ ОШИБОК]

Этот критерий оценивает, как часто LLM ошибается в процессе разметки срабатываний.

Рассчитывается по формуле (FP+FN)/ALL.

Python проекты

[ТРИАЖ]

SOLAR

	DerTriage	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Общая точность	80,0%	52,7%	52,7%	83,6%	67,3%	67,3%
Прецизионность	86,3%	60,7%	100,0%	97,6%	89,3%	80,6%
Полнота	74,6%	29,3%	10,3%	70,7%	43,1%	50,0%
Процент ошибок	18,0%	47,3%	47,3%	16,4%	32,7%	32,7%

[ОБЩАЯ ТОЧНОСТЬ]

Этот критерий оценивает, насколько верно LLM определяет истинность и ложность срабатывания.

Рассчитывается по формуле (TP+TN)/ALL

[ПОЛНОТА]

Этот критерий оценивает, сколько из реальных уязвимостей в проекте, LLM смогла найти.

Рассчитывается по формуле TP/(TP+FN)

[ПРЕЦИЗИОННОСТЬ]

Этот критерий оценивает, сколько реальных уязвимостей среди тех, что LLM отметила, как истинные.

Рассчитывается по формуле TP/(TP+FP)

[ПРОЦЕНТ ОШИБОК]

Этот критерий оценивает, как часто LLM ошибается в процессе разметки срабатываний.

Рассчитывается по формуле (FP+FN)/ALL.

Свод
[ТРИАЖ]

SOLAR

	DerTriage	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Общая точность	83,6%	53,2%	56,8%	66,8%	66,8%	67,3%
Прецизионность	85,5%	59,5%	78,1%	48,8%	82,9%	74,6%
Полнота	82,7%	27,4%	55,2%	35,3%	45,2%	56,8%
Процент ошибок	15,35%	46,8%	43,2%	33,2%	33,2%	32,7%

[ОБЩАЯ ТОЧНОСТЬ]

Этот критерий оценивает, насколько верно LLM определяет истинность и ложность срабатывания.

Рассчитывается по формуле (TP+TN)/ALL

[ПОЛНОТА]

Этот критерий оценивает, сколько из реальных уязвимостей в проекте, LLM смогла найти.

Рассчитывается по формуле TP/(TP+FN)

[ПРЕЦИЗИОННОСТЬ]

Этот критерий оценивает, сколько реальных уязвимостей среди тех, что LLM отметила, как истинные.

Рассчитывается по формуле TP/(TP+FP)

[ПРОЦЕНТ ОШИБОК]

Этот критерий оценивает, как часто LLM ошибается в процессе разметки срабатываний.

Рассчитывается по формуле (FP+FN)/ALL.

Фиск. Метрики

SOLAR

[GOOD]

Количество исправлений, которые устраняют реальную уязвимость в проекте. Реальные уязвимости пофикшены

[NOT GOOD]

Количество исправлений, которые **НЕ** устраняют реальную уязвимость в проекте. Реальные уязвимости **НЕ** пофикшены

[SELF GOOD]

Количество исправлений, которые устраняют уязвимость, помещенные на предыдущем шаге LLM как истинную

[SELF NOT GOOD]

Количество исправлений, которые **НЕ** устраняют уязвимость, помещенные на предыдущем шаге LLM как истинную

[ТОЧНОСТЬ]

Процент правильных исправлений реальных уязвимостей. Формула Good/All

[SELF ТОЧНОСТЬ]

Процент правильных исправлений срабатываний, помеченных моделью как истинные. Формула SelfGood/All

Java проекты

[ФИКС]

	DerCodeFix	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Good	43	7	34	25	0	18
Not Good	12	48	21	30	55	37
Self Good	45	1	43	0	22	0
Self Not Good	14	23	55	0	12	0
Точность	78,2%	12,7%	61,8%	45,5%	0,0%	32,7%
Self Точность	76,3%	4,2%	43,9%	-	64,7%	-

Python проекты

[ФИКС]

	DerCodeFix	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Good	48	15	27	26	28	16
Not Good	10	43	31	32	30	42
Self Good	46	4	20	4	14	0
Self Not Good	9	24	22	2	14	0
Точность	83,1%	25,9%	46,6%	44,8%	48,3%	27,6%
Self Точность	83,6%	14,3%	47,6%	66,7%	50,0%	-

[ТОЧНОСТЬ]

Процент правильных исправлений реальных уязвимостей

[SELF ТОЧНОСТЬ]

Процент правильных исправлений срабатываний, помеченных моделью как истинные

Свод
[ФИКС]

SOLAR

	DerCodeFix	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Точность	80,6%	19,3%	54,2%	45,1%	24,1%	30,2%
Self Точность	80,0%	9,2%	45,7%	-	57,4%	-

[ТОЧНОСТЬ]

Процент правильных исправлений реальных уязвимостей

[SELF ТОЧНОСТЬ]

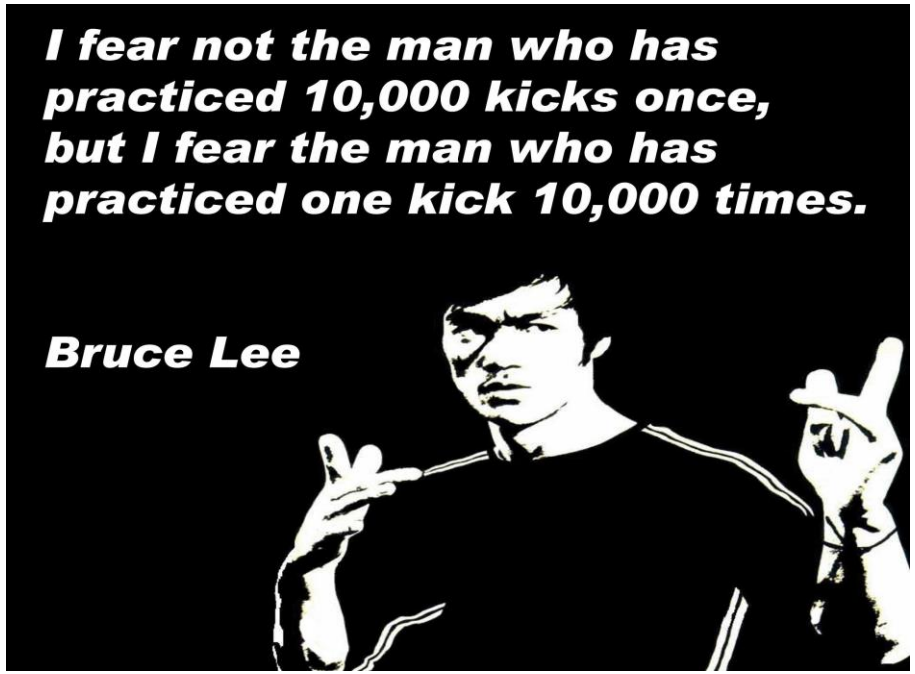
Процент правильных исправлений срабатываний, помеченных моделью как истинные

«Универсальная» или «заточенная» под конкретную задачу модель?

SOLAR

	DerTriage	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Общая точность	83,6%	53,2%	56,8%	66,8%	66,8%	67,3%
Прецизионность	85,5%	59,5%	78,1%	48,8%	82,9%	74,6%
Полнота	82,7%	27,4%	55,2%	35,3%	45,2%	56,8%
Процент ошибок	15,35%	46,8%	43,2%	33,2%	33,2%	32,7%

	DerCodeFix	Gigachat	ChatGPT	DeepSeek	LocalChat	LocalMistral
Точность	80,6%	19,3%	54,2%	45,1%	24,1%	30,2%
Self Точность	80,0%	9,2%	45,7%	-	57,4%	-



> Я не боюсь того, кто изучает 10 000 различных ударов. Я боюсь того, кто изучает один удар 10 000 раз

Нет компромиссу между скоростью разработки и уровнем безопасности

SOLAR

АКТУАЛЬНЫЕ БОЛИ

- Ложные срабатывания SAST — отнимают время
- Нехватка AppSec-инженеров — некому устранять уязвимости
- ИИ ускоряет написание кода — триаж не поспевает за разработкой
- Субъективность — два аналитика по-разному классифицируют одну уязвимость

РЕШЕНИЕ

Плагины к модулю SAST:

DerTriage AI *(триаж уязвимостей)* + DerCodeFix AI *(готовый код для исправлений)*

- 7 лет: заняло обучение плагинов DerTriage и DerCodeFix
- Единая платформа: плагины встроены в модуль SAST

Полный контроль данных

- Развертывание LMM в закрытом контуре
- Работает без доступа в интернет
- Нулевой обмен данными с внешними службами

О ТЕХНОЛОГИЧЕСКОМ ПАРТНЕРЕ

DerScanner — решение от мирового AppSec-производителя

Компания DerSecur, основанная в 2011 году, занимает передовые позиции в области безопасности приложений

Решение сертифицировано

MITRE

Признано

High Performer WINTER 2024

High Performer SPRING 2024

FORRESTER®

NIST

Какие аспекты важны при внедрении AI в AST

SOLAR

On-premise

Адекватный сайзинг

%Точности

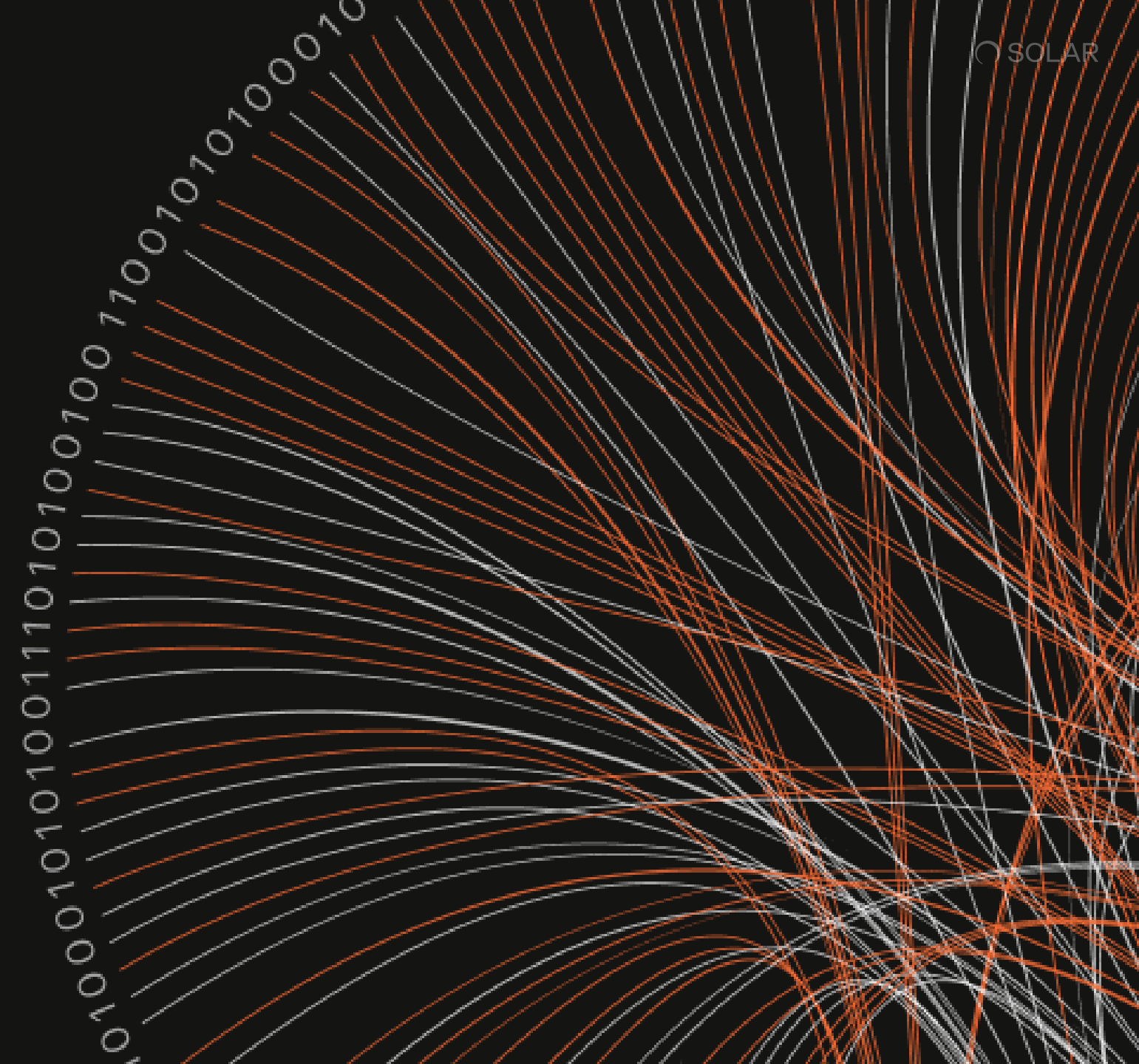
Адекватное время работы

Корректность исправлений

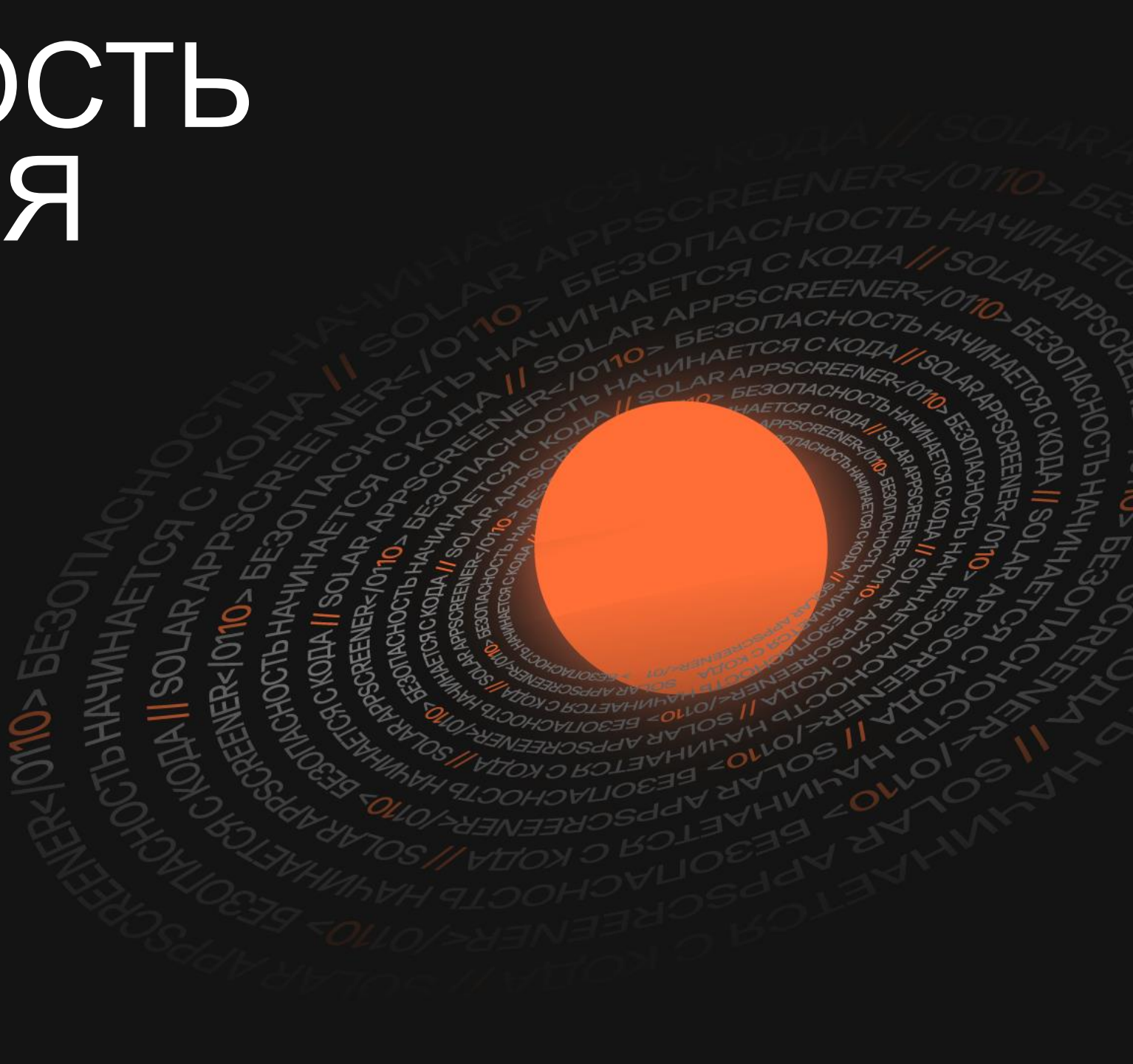
Конфиденциальность кодовых баз

Какие аспекты важны при внеднернии AI в AST

> ПОМНИТЬ ЧТО ЭТО ИИ



БЕЗОПАСНОСТЬ НАЧИНАЕТСЯ С КОДА



SOLAR APPSCREENER
КОМПЛЕКСНОЕ РЕШЕНИЕ
ДЛЯ КОНТРОЛЯ БЕЗОПАСНОСТИ ПО