

froddy .

Финальное решение перед отправкой денег

Froddy нужен там, где перевод запускает система, автоматизация или ИИ-агент.

froddy .

Почему это стало важно

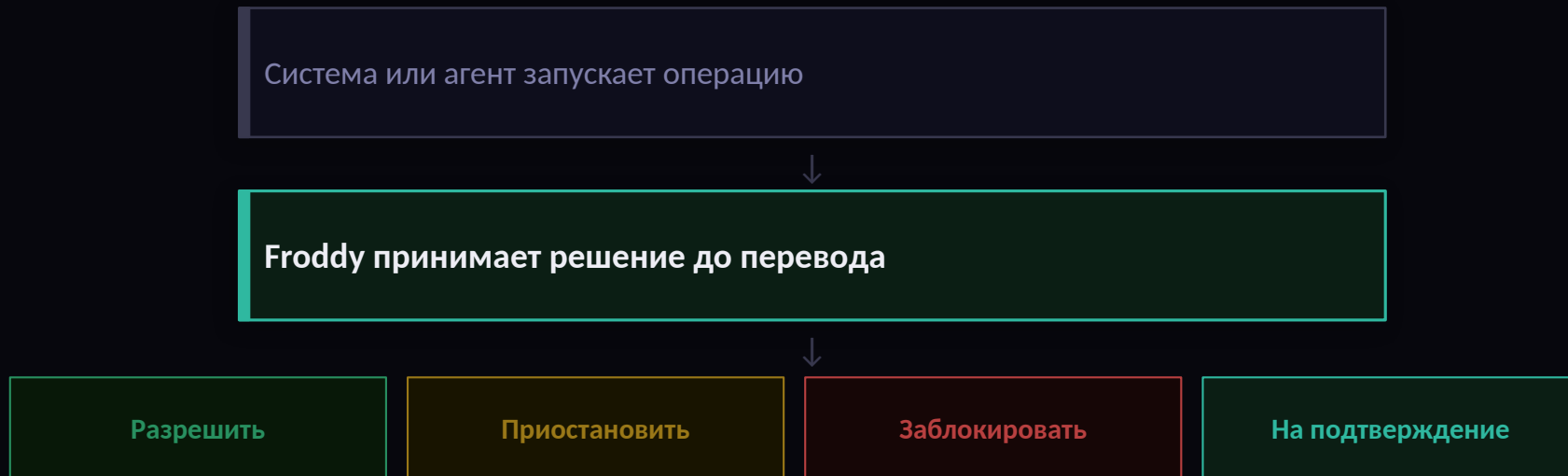
Финансовые операции всё чаще запускаются без участия человека.

Их запускают правила, внутренние процессы и ИИ-агенты — в том числе такие, как OpenClaw или Gemini.

Перед отправкой нужен отдельный контроль.

froddy .

Где в процессе находится Froddy



froddy .

Где нужен дополнительный контроль

Система или агент	запускает операцию
Риск-система	может подать сигнал
Банк или процессинг	готов провести перевод

Но перед отправкой не всегда есть отдельное решение по самой операции.

Внутренние правила

Хорошо работают, пока сценарий остаётся стабильным. Когда контекст меняется, их приходится переписывать.

Froddy

Оценивает операцию по текущему контексту — контроль не привязан к жёсткой логике одной системы.

Чем больше автоматизации и агентных действий, тем быстрее устаревают жёсткие правила.

froddy .

Что Froddy учитывает в моменте

- получатель новый или нетипичный
- сумма выше порога
- сумма выше обычного уровня
- реквизиты менялись недавно
- изменилась страна, валюта или маршрут
- операция идёт вне обычного цикла
- резко выросла частота переводов
- одни и те же реквизиты у разных получателей
- по операции уже есть сигнал риска
- нужен ручной контроль

Froddy учитывает совокупность текущих сигналов, а не один фиксированный порог.

froddy .

Где это особенно нужно

Платформы с выплатами

Много повторяющихся переводов и высокая цена ошибки.

Поставщики и финансовые операции

Деньги идут по внутренним процессам, а ручная проверка замедляет работу.

Крипта и агентные сценарии

Чем быстрее перевод, тем важнее финальное решение перед отправкой.

Первый пилот

Берём один финансовый сценарий.

Подключение по API или вебхуку.

Работаем поверх текущей инфраструктуры — без замены банка и процессинга.

На выходе видно:

- что прошло автоматически
- что ушло на подтверждение
- что было остановлено

История решений помогает со временем точнее настраивать контроль без изменений в платёжном коде.

Нужно: один сценарий, API или вебхук, один куратор, около 30 минут в неделю.