

AI-Агент ПОД КЛЮЧ

Как за 7 дней собрать
виртуального креатора,
который сам придумывает
и монтирует видео



Джулиан Кейн

Джулиан Кейн
AI-агент под ключ.
Как за 7 дней собрать
виртуального креатора,
который сам придумывает
и монтирует видео

<https://litres.ru/74289283>

SelfPub; 2026

Аннотация

Контент сегодня решает многое, но снимать, генерировать кадры и монтировать вручную — долго, дорого и выматывает. Пока одни сидят в редакторах ночами, AI-агент делает то же на сервере круглосуточно: находит идеи в Telegram, пишет сценарий, генерирует видео и показывает расходы в дашборде.

Навыки программирования не нужны. Код и техническую работу берёт на себя нейросеть. Вы учитесь работать как архитектор: собирать систему по шагам, управлять готовыми инструментами и понимать каждый узел процесса.

Из книги вы узнаете:

как развернуть свой виртуальный сервер и запустить агента Hermes;

как настроить «мозг» агента через OpenRouter и связку с fal.ai для генерации;

как научить агента следить за нужными ТГ-каналами и присылать свежий дайджест идей;

как настроить свой pixel-art дашборд;

как защитить систему, ограничить доступ и безопасно работать.

Через год подобные AI-агенты будут у многих. Прочитав эту книгу, вы окажетесь среди тех, кто их собирает и запускает — для себя или на заказ.

Содержание

Введение. Новая эра контента: от ручной работы к AI-автопилоту	5
Почему традиционный контент-мейкинг умирает	7
Что такое AI-агент-креатор — и чем он не чат-бот	10
Ваша роль: от «кодировщика» к архитектору AI-систем	14
Экономика запуска: честный минимум сервисов	17
Как устроена книга (и как из неё лучше «готовить»)	20
Короткий манифест перед стартом	22
Авторские права	23
Глава 1. Архитектура и стек: разбираем AI-агента по винтикам	24
1.1. Понятная карта системы: как связываются сервер, LLM, API, GitHub и Telegram	25
1.2. Выбор нейромозга: OpenRouter и модели для мышления	32
1.3. Инструментарий: VS Code, Claude Code и CLAUDE.md	37
Конец ознакомительного фрагмента.	38

Джулиан Кейн

AI-агент под ключ.

Как за 7 дней собрать виртуального креатора, который сам придумывает и монтирует видео

Введение. Новая эра контента: от ручной работы к AI-автопилоту

Ещё недавно «делать контент» значило почти одно и то же: придумать, снять, смонтировать, выложить, повторить. Камера, монтажная программа, ночь без сна, утренний кофе и надежда, что ролик «зайдёт». Сегодня картина меняется быстрее, чем обновляются инструкции к софту.

Сначала появились нейросети, которые пишут тексты. Потом — которые рисуют картинки. Потом — которые генерируют видео. Одного чата с моделью уже мало: нужен **сотрудник**, который сам собирает идеи, сам пишет сценарий, сам запускает генерацию, сам показывает, сколько это стоило, —

и делает это каждый день, не отпрашиваясь в отпуск.

Эта книга — о том, как такого сотрудника собрать. Не «поиграть в промпты», а **поставить на сервер рабочую систему**: AI-агента-креатора, которым вы управляете из Telegram, как обычным ботом. Код за вас пишет Claude Code. Вы — не программист, а **архитектор**: задаёте цель, проверяете шаги, понимаете, из чего состоит машина.

Через год подобные агенты будут у большинства, кто серьёзно работает с контентом. Вопрос не в том, «наступит ли эта эра». Вопрос — в том, окажетесь ли вы среди тех, кто **собирает** таких агентов, или среди тех, кто будет их догонять.

Почему традиционный контент-мейкинг умирает

«Умирает» — громкое слово. Редакторы, камеры и человеческий вкус никуда не денутся. Уходит другое: **модель, в которой весь цикл держится только на ручном труде одного человека или маленькой команды.**

1. Объём побеждает «один ролик в неделю»

Алгоритмы лент и коротких видео любят регулярность. Бизнесу нужен не шедевр раз в месяц, а **поток**: идеи, тесты, адаптации под форматы, А/В заголовков, серия из одной темы. Руками это либо дорого (команда), либо невыносимо (выгорание автора).

Предприниматель, который «сам ведёт соцсети», часто попадает в ловушку: откладывает контент «на потом» или жертвует сном и продажами. Блогер с аудиторией — в другую: чем больше успеха, тем сильнее давление «выкладывать чаще». Классический контент-пайплайн не масштабируется линейно: удвоить выпуск — почти удвоить часы.

2. Инструменты размножились, а координация — нет

Сегодня у одного человека может быть:

- чат с LLM для идей и сценариев;
- сервис генерации картинок;

- сервис генерации видео;
- монтажёр или CapCut;
- таблица с идеями;
- Telegram с кучей каналов-доноров.

Каждый кусок силён сам по себе. Слабое звено — **выкак** диспетчер, который всё это склеивает. Между «нашёл идею в канале» и «готовое видео в телефоне» — десяток переключений контекста. На переключениях теряются время и энергия.

3. Стоимость «тихого» простая

Пока вы не снимаете, конкурент уже выложил три ролика. Пока вы «думаете над сценарием», нейросеть могла бы предложить пять вариантов за минуту. Простой контент-канала — невидимый налог на внимание аудитории. AI-агент не отменяет стратегию и вкус, но **убирает простой вроде «я не успел сесть и сделать»**.

4. Что ломается в старой схеме

Схема:

Этап	Как было	Где боль
Идеи	Скролл каналов, заметки в голове	Идеи теряются, нет системы
Сценарий	Пишете сами или заказываете	Долго, дорого, нерегулярно
Кадры / видео	Съёмка или ручная генерация	Часы в интерфейсах
Монтаж	Вручную	Узкое горлышко
Контроль	«Помню, что делал»	Нет прозрачных расходов и статусов

Этап · Как было · Где боль

Традиционный контент-мейкинг не «плох». Он **не выдерживает темпа**, который задаёт рынок и сами AI-инструменты. Побеждает не тот, у кого дороже камера, а тот, у кого **конвейер**: от сигнала «есть идея» до артефакта «есть ролик» — с минимумом ручных касаний.

Важно. Книга не учит «заменить человека нейросетью навсегда». Она учит **снять с человека рутину**, чтобы остались вкус, стратегия, бренд и решения «да / нет / переделай».

Что такое AI-агент-креатор — и чем он не чат-бот

Разберёмся в словах. От них зависит, что вы будете строить.

Обычный чат-бот

Чат-бот (в привычном смысле) — интерфейс «вопрос → ответ». Вы пишете, модель отвечает текстом. Умный чат-бот полезен: подскажет идею, набросает сценарий, перепишет заголовок. Но он:

- **не живёт сам по себе** 24/7 на вашем сервере (пока вы не открыли вкладку);

- **не ходит** по вашим каналам и не собирает дайджест к 9:00 без отдельной обвязки;

- **не вызывает** генерацию видео, не хранит бриф заказа, не считает расходы в дашборде;

- **не действует** — только **говорит**.

Модель без «рук» — энциклопедия в диалоге. Полезно. Но это ещё не сотрудник.

AI-агент

Агент — система, в которой языковая модель (LLM) встроена в **харнес** (обвязку): код, который даёт модели возможность **действовать** — вызывать инструменты, читать файлы, писать в базу, дёргать API, соблюдать правила.

Простая формула:

Агент = модель (мозг) + харнес (руки и правила) + каналы связи (Telegram и др.) + память (файлы, база, конфиг)

Харнес — ключевое слово этой книги. Модель «живая» в словах: уточняет, переспрашивает, предлагает. Но всё, что **меняет мир** (записало бриф, запустило генерацию, списало токены), проходит через код: проверки, лимиты, тулы, «вахтёры». Свобода в диалоге — жёсткие рельсы в действиях. Так слабая или «разъехавшаяся» модель не сожжёт вам бюджет и не сломает процесс.

AI-агент-креатор (то, что мы собираем)

Это **личный сотрудник внутри Telegram**, который работает на **вашем сервере** круглосуточно. Вы управляете им из переписки, как обычным ботом. Он умеет (в целевой сборке этой книги):

1. **Мониторить** выбранные Telegram-каналы и источники, собирать сырьё для идей.
2. **Присылать дайджест** — например, каждое утро к 9:00 подборку идей под ваш контент.
3. **Вести бриф** заказа: тип, формат, модель, количество кадров, референсы.
4. **Писать сценарий** и «режиссировать» ролик через специализированные скиллы.

5. **Генерировать** кадры и видео через нейросети (в том числе fal.ai), подключать озвучку и монтажные пресеты.

6. **Показывать** задачи, статусы и расходы в дашборде (в том числе в стиле pixel-art).

Особый акцент курса, из которого выросла эта книга: **скинуть агенту своё фото — и получить видео с вами**. Не «абстрактный аватар из каталога», а контур «ваше лицо / ваш товар → готовый визуальный контент» в рамках студийного конвейера.

Чем агент-креатор отличается от «просто ChatGPT в Telegram»

Схема:

	Чат с LLM	Бот-обёртка	AI-агент-креатор
Где живёт	Вкладка браузера	Часто чужой SaaS	Ваш VPS
Память заказа	Контекст чата	Ограниченно	Бриф + база + артефакты
Действия	Текст	Команды «/start»	Тулы, API, генерация, деплой-логика
Роли	Один «универсал»	Один сценарий	Скиллы: продюсер, сценарист, режиссёр, криэйтор...
Контроль денег	Счёт сервиса	Непрозрачно	Ключи, лимиты, дашборд
Масштаб	Вы сидите в чате	Ответ по кнопке	24/7, даже когда ноутбук выключен

Ещё один термин, который пригодится дальше: **агент из коробки**(out-of-the-box) — готовый каркас, который ожи-
вает после установки. Мы не пишем движок с нуля в блок-
ноте. Мы ставим **Hermes**(движок агента), поверх — сту-
дию контента (**AI Machine** и скиллы), подключаем мозг че-
рез **OpenRouter**, генерацию через **fal.ai**, лицо — через
Telegram-бота. Ваша работа — собрать, настроить под себя
и **понять**, как это устроено.

***Обратите внимание.** «Без навыков программирования»
не значит «без понимания». Код пишет Claude Code. Но если
вы слепо копируете команды, система останется чёрным
ящиком: при первой ошибке вы застрянете. Книга объясня-
ет **схему**, а не только кнопки.*

Ваша роль: от «кодировщика» к архитектору AI-систем

Многие боятся начать, потому что думают: «Я не разработчик». Хорошая новость: **вам и не нужно им становиться в классическом смысле.**

Старая роль: кодировщик

Кодировщик держит в голове синтаксис, библиотеки, баги, деплой. Он — исполнитель деталей. Для новичка это стена: год учёбы, прежде чем «можно что-то своё».

Новая роль: архитектор (и заказчик для Claude Code)

Архитектор:

- формулирует **цель** («хочу, чтобы бот каждое утро присылал 5 идей из этих каналов»);
- задаёт **границы** (бюджет, allowlist — белый список пользователей, что агенту нельзя);
- проверяет **признаки успеха** (бот ответил, сервис active, дайджест пришёл);
- читает **карту системы**, а не каждую строку чужого кода;
- делегирует написание и правки **агенту разработки** — Claude Code.

Claude Code — это агент-программист от Anthropic. Он

живёт в редакторе (VS Code) или терминале, пишет, правит и запускает код **по вашим словам**. Аналог у OpenAI — Codex; в книге основной инструмент — Claude Code.

Вы общаетесь с ним как с сильным junior- или middle-разработчиком, которому нельзя молча ломать прод:

- даёте задачу целиком, с критерием «готово, когда...»;
- просите **показать** опасные команды до запуска;
- храните память проекта в CLAUDE.md, чтобы он не «забывал» архитектуру;
- экономите токены: не таскаете в контекст лишнее, ставите скиллы и плагины осознанно.

Мышление архитектора в трёх вопросах

На каждом шаге книги держите в голове:

1. **Где это живёт?**(ноутбук / VPS / облачный API)
2. **Кто думает?**(какая модель, через какой шлюз)
3. **Кто платит и кто имеет доступ?**(ключи, лимиты, allowlist в Telegram)

Если на три вопроса есть ответы — вы уже не «нажиматель кнопок», а человек, который **держит систему**.

Кому эта книга особенно найдёт

- **Предпринимателям**, которые хотят закрыть контент бизнеса руками агента, а не сразу нанимать отдел.
- **Креаторам и блогерам**, которым нужно больше роликов без выгорания.
- **Новичкам без технического бэкграунда**, готовым идти по шагам и разбираться в терминах.

— **Тем, кто хочет зарабатывать**: собирать таких агентов под ключ и продавать клиентам (этому посвящена отдельная глава).

К концу книги у вас на руках:

- рабочий AI-агент-креатор, настроенный под вас;
- понимание стека: сервер, нейросети, Telegram, скиллы, ключи;
- навык собирать похожих агентов с нуля, в том числе на заказ;
- дашборд с задачами и финансами;
- привычка думать об **безопасности** до того, как что-то «уже взорвалось».

Экономика запуска: честный минимум сервисов

Воркшоп и эта книга учат на **реальных инструментах**. Значит, кроме вашего времени понадобятся подписки и пополнения. Это **не «доплаты автору книги»**, а плата сторонним сервисам, на которых живёт *ваша* агент.

Цены меняются. Ниже — **ориентир порядка величин** на старт (чтобы пройти путь и всё протестировать), а не вечный прайс.

Минимальный набор

Схема:

Сервис	Зачем	Ориентир на старт
Claude Code (подписка Anthropic / Claude)	«Мозг и руки» сборки: пишет и правит код по вашим задачам	~20 \$/мес (тестовый рабочий тариф)
VPS / VDS (например, AEZA; в практике курса также встречаются другие хостеры)	Сервер, где агент живёт 24/7	от ~6 \$/мес (или сотни ☐, в зависимости от тарифа)
OpenRouter (иногда альтернативы вроде POLZA.ai)	«Мышление» агента: доступ к LLM по API, один ключ — много моделей	~10 \$ на тесты; дальше — по токенам
fal.ai	Генерация изображений и видео по API	пополнение от ~10 \$
Опционально: ElevenLabs и др.	Озвучка, доп. медиа	по необходимости
Telegram	Бот (BotFather) + при необходимости userbot для мониторинга	сам мессенджер бесплатен; риски — у политик Telegram

Сервис · Зачем · Ориентир на старт

Грубая вилка «войти и потрогать систему»: порядка **\$50–80** на первый месяц, если не гнать тяжёлую генерацию видео пачками. Дальше расход растёт вместе с тем, **сколько** агент думает и **сколько** генерирует.

Важно. Токены и генерация — это счётчик. Агент без лимитов и allowlist (белого списка) может «крутиться» и жечь баланс. Поэтому в книге отдельные главы про **экономику токенов**, **ключи в `.env`**, **лимиты** и **безопасность** — не «для параноиков», а для тех, кто не хочет подарить бюд-

жет случайному пользователю бота.

Что вы покупаете на самом деле

— **Claude Code**— скорость сборки и правок. Без него путь «с нуля руками» для новичка нереален в заявленные сроки.

— **VPS**— независимость от вашего ноутбука. Агент должен отвечать, когда вы спите.

— **OpenRouter**— гибкость мозга: можно выбрать более дешёвую или более умную модель под задачу.

— **fal.ai**— фабрика картинок и видео без своей GPU-фермы.

Чего книга не обещает

— Мгновенной вирусности роликов.

— Нулевых расходов «навсегда».

— Что любая модель всегда будет идеально слушаться без харнеса и скиллов.

— Обход законов, правил платформ и чужих авторских прав «потому что AI».

Мы обещаем другое: **прозрачную карту**, пошаговую сборку и понимание, как устроены программа, сервер и нейросеть, — чтобы вы не повторяли действия вслепую.

Как устроена книга (и как из неё лучше «готовить»)

Книга следует логике семи дней воркшопа, но оформлена как самостоятельное практическое руководство.

Глава	Суть
1. Архитектура и стек	Карта: сервер, LLM, API, GitHub, Telegram, термины
2. Рабочее место	VS Code, Claude Code, токены, CLAUDE.md, первые скиллы
3. Сервер и Telegram	VPS, Hermes, BotFather, первый живой диалог 24/7
4. Сбор идей	Userbot, мониторинг, дайджест к 9:00, секреты в .env
5. Генерация контента	fal.ai, скиллы-роли, роутер, аватар-видео, полный цикл
6. Дашборд	Pixel-art панель: задачи, статусы, деньги
7. Безопасность	Ключи, SSH, firewall, allowlist, бэкапы, промпт-инъекции
8. Монетизация	Оффер «агент под ключ», себестоимость, передача клиенту
Заклучение + приложения	Тренды, чек-лист, глоссарий, шаблоны

Глава · Суть

Как читать

1. Сначала введение и главу 1 — даже если руки чешут-

ся «уже ставить». Без карты терминов вы утонете в словах.

2. **Делайте по главам**, не откладывая сервер и безопасность «на потом».

3. В практических местах ищите врезки **«Важно»** и **«Признак успеха»** — это контрольные точки.

4. Кодовые фрагменты и команды копируйте осознанно: подставляйте **свои** ключи и IP, никогда не публикуйте секреты.

5. Если термин незнакомый — смотрите **приложение А** (гlossарий).

Стек одной фразой

*На VPS крутится **Hermes** (движок). «Думает» модель через **OpenRouter**. «Лицо» — **Telegram-бот**. Генерация медиа — **fal.ai**. Собираете и дорабатываете всё это вы через **VS Code + Claude Code**, с памятью в **CLAUDE.md** и навыками в **skills**. Контроль — **дашборд**. Защита — **ключи, лимиты, allowlist, SSH**.*

Если эта фраза к концу введения уже не пугает — вы готовы к главе 1.

Короткий манифест перед стартом

1. **Контент остаётся королём**— меняется способ его производства.
2. **Чат-бот говорит; агент действует**— разница в хар-несе, памяти и API.
3. **Вы — архитектор**, Claude Code — строитель по вашему ТЗ.
4. **Сервер и ключи — часть продукта**, не «техническая мелочь».
5. **Честный бюджет**на старте окупается пониманием и рабочим конвейером.
6. **Безопасность и деньги**продумываются до того, как бот станет популярным.
7. **Через год агенты будут у всех**— преимущество у тех, кто умеет их **собирать и продавать**.

Если цель — **сразу собрать**, а не только понять карту: откройте **приложение Г (Golden path за 48 часов)** и идите по слотам. Главы 1–8 дадут глубину; golden path — рабочий результат.

Иначе откройте следующую главу. Мы разберём AI-агента по винтикам: как связываются сервер, модель, API, GitHub и Telegram — простым языком, без тумана.

Авторские права

© Джулиан Кейн (Julian Kane), 2026 Все права защищены.

Никакая часть настоящей книги не может быть воспроизведена в любой форме без письменного разрешения правообладателя, за исключением случаев цитирования в объёме, допустимом законодательством.

Торговые марки Claude, Claude Code, Anthropic, Telegram, OpenRouter, fal.ai, VS Code, GitHub и другие упомянуты в информационных целях; права принадлежат их владельцам. Автор не аффилирован с указанными компаниями, если не оговорено иное. Цены сервисов, интерфейсы и идентификаторы моделей меняются. Актуальные данные сверяйте на сайтах провайдеров на момент настройки.

Глава 1. Архитектура и стек: разбираем AI-агента по винтикам

Прежде чем что-то ставить, нужна **карта**. Дальше по книге вы постоянно будете слышать: Claude Code, VPS, API, Hermes, OpenRouter, скилл, харнес. За каждым словом — простая вещь. Эта глава — ваш словарь и схема одновременно. К ней можно возвращаться, когда встретите незнакомый термин.

К концу главы вы:

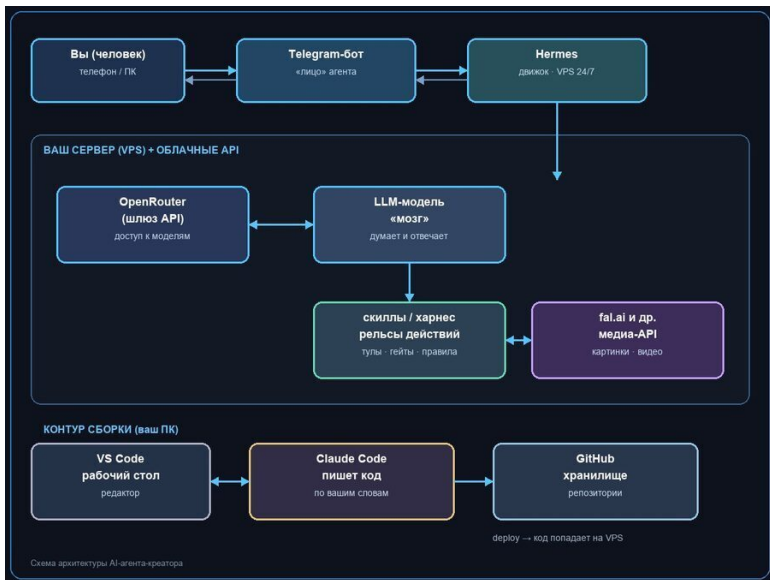
- увидите, как связаны сервер, модель, API, GitHub и Telegram;
- поймёте, зачем нужен шлюз вроде OpenRouter и как выбирать «мозг» агента;
- разведёте по полочкам VS Code, Claude Code и файл CLAUDE.md;
- сможете объяснить простыми словами токены, харнес, ключи, скиллы и промпт-инъекции.

Установку программ начнём в главе 2. Сейчас — **понимание**. Без него даже идеальная инструкция превращается в «нажал — сломалось — не знаю почему».

1.1. Понятная карта системы: как связываются сервер, LLM, API, GitHub и Telegram

Большая картина одной схемой

Представьте конвейер. Вы пишете боту в Telegram. Сообщение доезжает до программы на сервере. Программа спрашивает нейросеть. Нейросеть отвечает. Программа при необходимости дёргает генерацию картинок или видео. Результат возвращается вам в чат. Параллельно вы (или Claude Code) правите код проекта на своём компьютере и выкладываете обновления на сервер.



Семь блоков — семь ролей

Схема:

Блок	Простыми словами	Роль в системе
Вы	Заказчик и архитектор	Даёте задачи, принимаете результат, задаёте правила
Telegram-бот	«Лицо» и пульт	Принимает сообщения, отдаёт ответы; создаётся через BotFather
Hermes	Движок / каркас	Крутит диалог, вызывает модель и инструменты; сам не думает
LLM через OpenRouter	Мозг	Понимает текст, пишет ответы, решает, какой тул вызвать (в рамках харнеса)
Скиллы + харнес	Инструкции и рельсы	Учат «как делать хорошо» и не дают модели сломать процесс
fal.ai (и др.)	Фабрика медиа	Картинки, видео, иногда аудио — по API, без вашей GPU
VPS	Дом агента	Компьютер в интернете, включён 24/7
VS Code + Claude Code	Верстак сборки	Где вы собираете и чините систему
GitHub	Склад кода	Откуда ставят Hermes / студию, куда кладут историю изменений

Блок · Простыми словами · Роль в системе

Важно. *Hermes — не «ещё одна нейросеть». Это программа-обвязка. Думает подключённая модель. Путаница «я поставил Hermes, а он глупый» почти всегда значит: слабая или неверно выбранная модель, пустой баланс, кривой ключ — а не «сломанный Hermes».*

Как сообщение проходит путь (минимальный сценарий)

1. Вы пишете боту: «Привет» или «Новый заказ».
2. Telegram доставляет сообщение на ваш процесс (локально или на VPS) — через **токен бота**.
3. **Hermes**(gateway) принимает сообщение, поднимает контекст сессии, готовит запрос к модели.
4. Запрос уходит по **API** в OpenRouter (или аналог) с вашим **API-ключом**.
5. **Модель** возвращает текст и/или решение «вызвать инструмент».
6. Харнес выполняет разрешённые действия (записать бриф, дёрнуть fal.ai, показать карточку).
7. Ответ уходит обратно в Telegram.

Пока Hermes запущен только на ноутбуке, бот живёт, **пока открыт ноутбук**. Когда Hermes стоит на VPS как служба — бот отвечает, даже если вы выключили компьютер. В этом и смысл сервера.

Два контура: «жизнь агента» и «сборка агента»

Новички смешивают их в одну кучу. Разведите:

Контур	Где	Зачем
Прод (жизнь)	VPS + Telegram + ключи API	Агент работает для вас 24/7
Dev (сборка)	Ваш ПК, VS Code, Claude Code	Ставите, правите, тестируете, деплоите

Контур · Где · Зачем

Правило зрелой работы (оно же правило студии AI Machine из материалов курса):

Правите локально → выкладываете на сервер → проверяете в Telegram. Не правьте «боевые» файлы студии прямо на сервере — при следующем обновлении затрётся.

GitHub в этой схеме — не «соцсеть для программистов»

Репозиторий (repo) — хранилище кода с историей изменений. Через GitHub вы:

- ставите **Hermes** (официальный репозиторий движка);
- ставите **студию контента** (AI Machine и ваши кастомизации) поверх Hermes;
- при желании храните **свою** копию настроек (без секретов!).

Git — инструмент на компьютере, которым клонируют и

обновляют репозитории. Установщики и Claude Code часто вызывают Git «под капотом». Без Git многие «поставь плагин / поставь Hermes» просто не взлетят.

Telegram: бот и userbot — не одно и то же

Схема:

	Бот (BotFather)	Userbot (например, через Pyrogram)
Кто это	Отдельное «лицо» @my_agent_bot	Автоматизация вашего личного аккаунта
Зачем в книге	Общение с агентом, заказы, ответы	Мониторинг каналов, сбор идей (глава 4)
Риски	Утечка токена = угон бота	Нарушение правил Telegram, бан аккаунта при спаме
Правило	Токен = пароль; allowlist (белый список) пользователей	Не слать тонны сообщений; бережный мониторинг

Бот (BotFather) · Userbot (например, через Pyrogram)

Обратите внимание. Telegram не одобряет массовую автоматическую активность. Userbot — мощный, но острый инструмент. В книге мы используем его для **сбора идей**, а не для спама.

Домен и дашборд (на будущее)

Позже появится **дашборд**— веб-панель статусов, задач и

расходов. Ему могут понадобиться:

- **хостинг** на том же VPS (сайт «живёт» на сервере);
- **домен** — человеческое имя вместо голого IP.

Это не обязательно в первый день. На карте системы достаточно знать: дашборд — ещё один клиент к вашему серверу, а не «магия в облаке без адреса».

Карта «агент из коробки» vs «агентская система»

— **Агент из коробки** — каркас, который оживает после установки (Hermes + базовая связка).

— **Агентская (мультиагентная) система** — несколько ролей и воркеров (продюсер, сценарист, режиссёр, криэйтор, монтажёр), которые идут по конвейеру заказа, а не один «универсальный бот на всё».

В главе 5 вы увидите, как один заказ в Telegram превращается в **бриф → сценарий → кадры → генерация → монтаж**. Пока достаточно помнить: архитектура — **конвейер с памятью в базе**, а не бесконечный чат без структуры.

1.2. Выбор нейромозга: OpenRouter и модели для мышления

Зачем вообще «шлюз», если есть ChatGPT и Claude в браузере?

Браузерный чат удобен вам. Агенту на сервере нужен **API**: стабильный программный доступ «запрос → ответ» с ключом и тарификацией.

Проблема: у каждой компании свой API, свои ключи, свои форматы. **OpenRouter** — единый API-шлюз: **один ключ**, доступ к десяткам (и сотням) моделей — Claude, GPT, Gemini, Llama, MiniMax и другим — через один интерфейс.

Аналогия: не десять сим-карт к десяти операторам, а **один роутер**, который выбирает линию.

*В материалах курса как запасной путь при проблемах с оплатой или доступом к OpenRouter упоминается **Polza.ai** (в том числе оплата картой РФ) — тот же принцип: ключ + base URL + модель. Смысл архитектуры не меняется: шлюз → модель → ответ.*

Что именно «думает»

Схема:

Слой	Кто	Думает?
Hermes	Движок	Нет — оркеструет
OpenRouter / Polza	Шлюз	Нет — маршрутизирует и тарифицирует
LLM (модель)	MiniMax, Claude, GPT...	Да
fal.ai	Медиа-модели	«Рисует/снимает», это другой тип моделей

Слой · Кто · Думает?

Для **агента в Telegram** «мозг» — это LLM, указанная в конфиге Hermes (часто через переменные в `.env` и `config.yaml`).

Для **сборки кода** «мозг» — модели внутри **Claude Code** (отдельная подписка Anthropic). Не путайте счета: деньги OpenRouter ≠ деньги Claude Code.

Как выбирают модель на старте

Критерии практика, а не маркетинг:

1. **Цена за токен** — агент ходит в модель часто; дорогая reasoning-модель на каждый «привет» разорит.
2. **Скорость** — в Telegram люди ждут ответ секунды, не минуты.
3. **Мультимодальность** — если агент смотрит фото (товар, лицо, референс), нужна модель с vision.

4. **Стабильность через шлюз**— модель должна быть доступна в каталоге OpenRouter (или вашего провайдера) с понятным id.

5. **Качество на русском**— для брифа и сценариев это критично.

В практических материалах курса для первого живого бота часто берут **относительно дешёвую и шуструю** модель с пониманием картинок (например, линейка **MiniMax M3** через OpenRouter) — чтобы цепочка Telegram → Hermes → модель завелась без разорения. Позже для сложных воркеров студии можно поднимать класс модели **точечно**.

***Важно.** Идентификаторы моделей в каталоге меняются. В книге мы пишем принцип: «возьми актуальный id из каталога шлюза и пропиши в конфиг». Конкретную строку `provider/model-name` всегда сверяйте на сайте OpenRouter (или Polza) в день настройки.*

Деньги мозга: токены

Токен— единица текста для LLM (грубо: несколько символов ≈ 1 токен; точная нарезка зависит от модели). Считаются:

— **вход**(всё, что модель прочитала: система, скиллы, история, бриф);

— **выход**(то, что модель написала).

Важнейший факт, из-за которого «вдруг дорого»:

API без памяти между запросами. На каждый ход вы снова отправляете контекст. Длинный чат дорожает нелинейно: каждое сообщение тащит всю историю.

Поэтому в хорошем харнесе:

- режут вход кодом (не «все скиллы сразу»);
- делают **одноходовых** воркеров без растущей истории;
- кладут стабильный префикс (характер, правила) так, чтобы работал **кеш** провайдера — повторное чтение кеша сильно дешевле «сырого» входа;
- числа и форматы считает **код**, а не модель.

Числа «было 120k токенов, стало 29k» в учебных схемах — **иллюстрация порядка**, не ваш гарантированный счёт. Смысл один: **харнес экономит деньги**, а не только «дисциплинирует модель».

Практический минимум по OpenRouter (пока без кнопок установки)

1. Регистрация на openrouter.ai.
2. Пополнение credits (на тесты часто хватает \$5–10).
3. Создание API-ключа вида `sk-or-v1-...` (показывается один раз).
4. Пропись ключа в `.env` агента как `OPENROUTER_API_KEY` (или аналог для другого шлюза).
5. Указание модели по умолчанию и base URL API.

Секрет = деньги. Ключ не кладут в скриншоты, публичные чаты, GitHub, «временные» Notion-страницы. Даже в приватный репозиторий ключ в коде — плохая идея. Только `.env` + `.gitignore` (глава 4 и приложение Б).

Мозг для сборки vs мозг для агента

Схема:

	Claude Code	LLM агента (OpenRouter)
Где	Ваш ПК, IDE/терминал	VPS, процесс Hermes
Платите	Подписка Anthropic	Токены/кредиты шлюза
Задача	Писать и чинить код системы	Отвечать пользователю, вести бриф, вызывать тулы
Можно ли «один на всё»	Неудобно и дорого	Claude Code не заменяет 24/7 бота на сервере

Claude Code · LLM агента (OpenRouter)

Архитектор держит **два счёта** и **две роли моделей** в голове — и не удивляется, почему «подписка Claude есть, а бот молчит» (у бота кончился OpenRouter) или наоборот.

1.3. Инструментарий: VS Code, Claude Code и CLAUDE.md

Три сущности, которые новички склеивают в «программу для кода». Это разные этажи.

VS Code — окно-мастерская

VS Code(Visual Studio Code) — бесплатный редактор кода от Microsoft. Это:

- проводник файлов слева;
- редактор текста по центру;
- терминал снизу;
- расширения (плагины редактора) сбоку.

Сам по себе VS Co

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.