

Технология блокчейн

#10

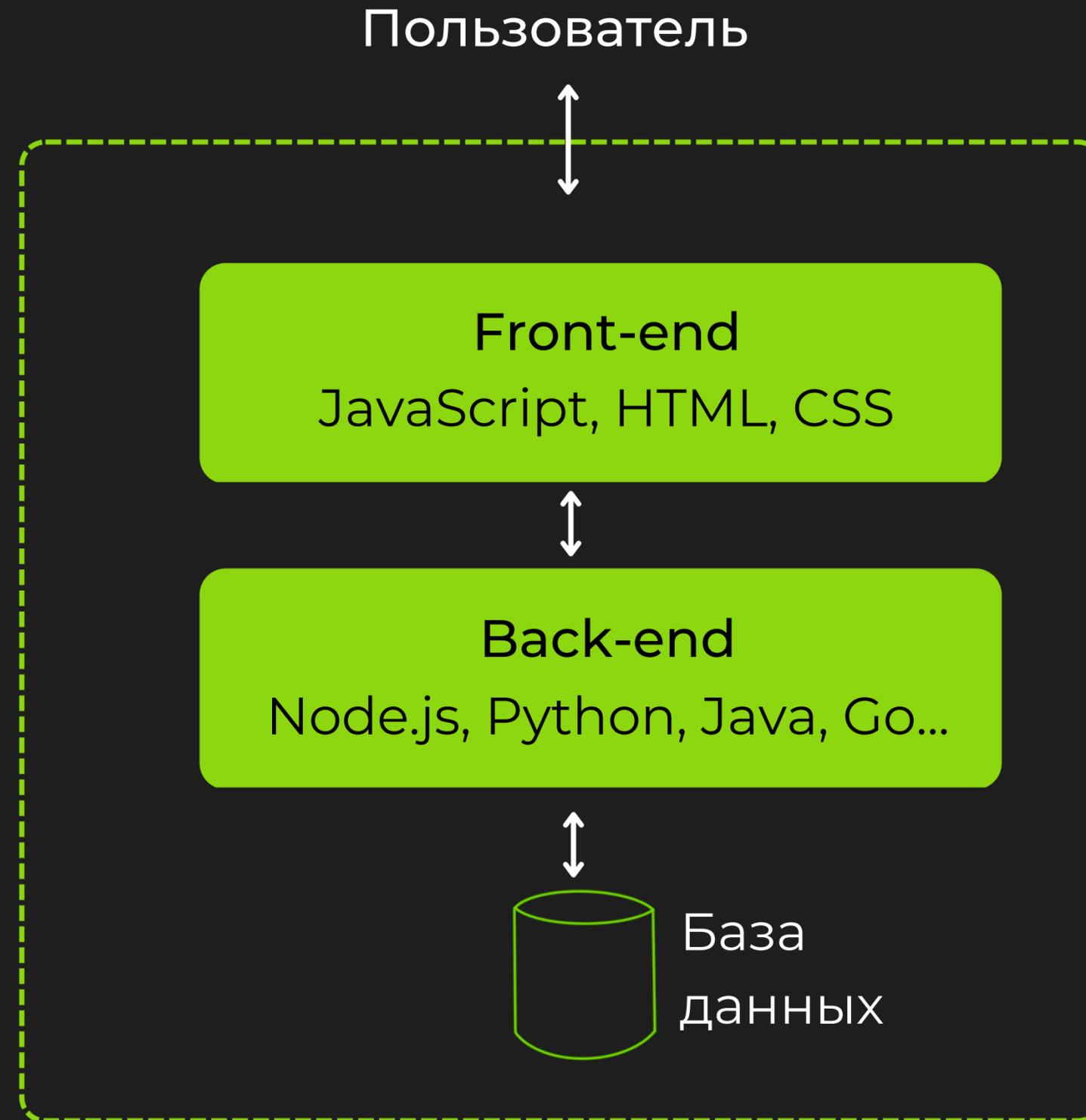
Децентрализованные

приложения

Обзор

- Архетиктура web2 приложений
- Архетиктура web3 приложений
- Провайдеры
- Подписанты
- Хранение данных в блокчейне

Архитектура приложений web2



Архитектура приложений web2

Фулл-стэк Web2

приложения могут

иметь:

Базу Данных (данные)

Бэкенд (логика)

Фронтенд (интерфейс)

Когда вы пользуетесь сайтом:

Вы взаимодействуете с

фронтендом,

который взаимодействует с

бэкендом,

который взаимодействует с

базой данных.

Пример web2 приложения

The screenshot displays the CoinMarketCap website interface. At the top, it shows market statistics: Cryptos: 22,335, Exchanges: 536, Market Cap: \$1,050,570,165,613, 24h Vol: \$53,385,362,956, Dominance: BTC: 42.0%, ETH: 19.1%, and ETH Gas: 14 Gwei. The main navigation includes 'Cryptocurrencies', 'Exchanges', 'Community', 'Products', and 'Learn'. A search bar and 'Log In'/'Sign up' buttons are also present.

Today's Cryptocurrency Prices by Market Cap

The global crypto market cap is \$1.05T, a $\uparrow 0.64\%$ increase over the last day. [Read More](#)

Trending

- 1 Bitcoin BTC $\uparrow 0.10\%$
- 2 SingularityNET AGIX $\downarrow 6.55\%$
- 3 TABOO TOKEN TABOO $\downarrow 2.83\%$

Recently Added

- 1 dollarmoon DMOON \$0.0204
- 2 MTG Token MTG \$0.003325
- 3 Veno Finance VNO \$0.3268

Top Community Accounts

- Qtum_Foundation @Qtum_Foundation + Follow
- IOTA @IOTA + Follow
- PlayDapp @PlayDapp_IO + Follow

#	Name	Price	1h %	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 Days
1	Bitcoin BTC	\$22,910.90	$\uparrow 0.63\%$	$\uparrow 0.10\%$	$\uparrow 9.94\%$	\$441,496,525,866	\$24,479,446,914 1,068,473 BTC	19,270,150 BTC	
2	Ethereum ETH	\$1,638.03	$\uparrow 0.23\%$	$\uparrow 0.64\%$	$\uparrow 5.90\%$	\$200,452,282,109	\$6,939,213,342 4,234,362 ETH	122,373,866 ETH	
3	Tether USDT	\$1.00	$\downarrow 0.00\%$	$\uparrow 0.00\%$	$\downarrow 0.02\%$	\$66,756,723,145	\$34,123,179,718 34,116,535,309 USDT	66,744,307,804 USDT	
4	BNB BNB	\$305.86	$\uparrow 0.41\%$	$\uparrow 1.03\%$	$\uparrow 2.31\%$	\$48,296,824,749	\$628,265,759 2,052,796 BNB	157,903,242 BNB	

В чем отличие web3 приложений?

Устранение посредников и централизованных сервисов.

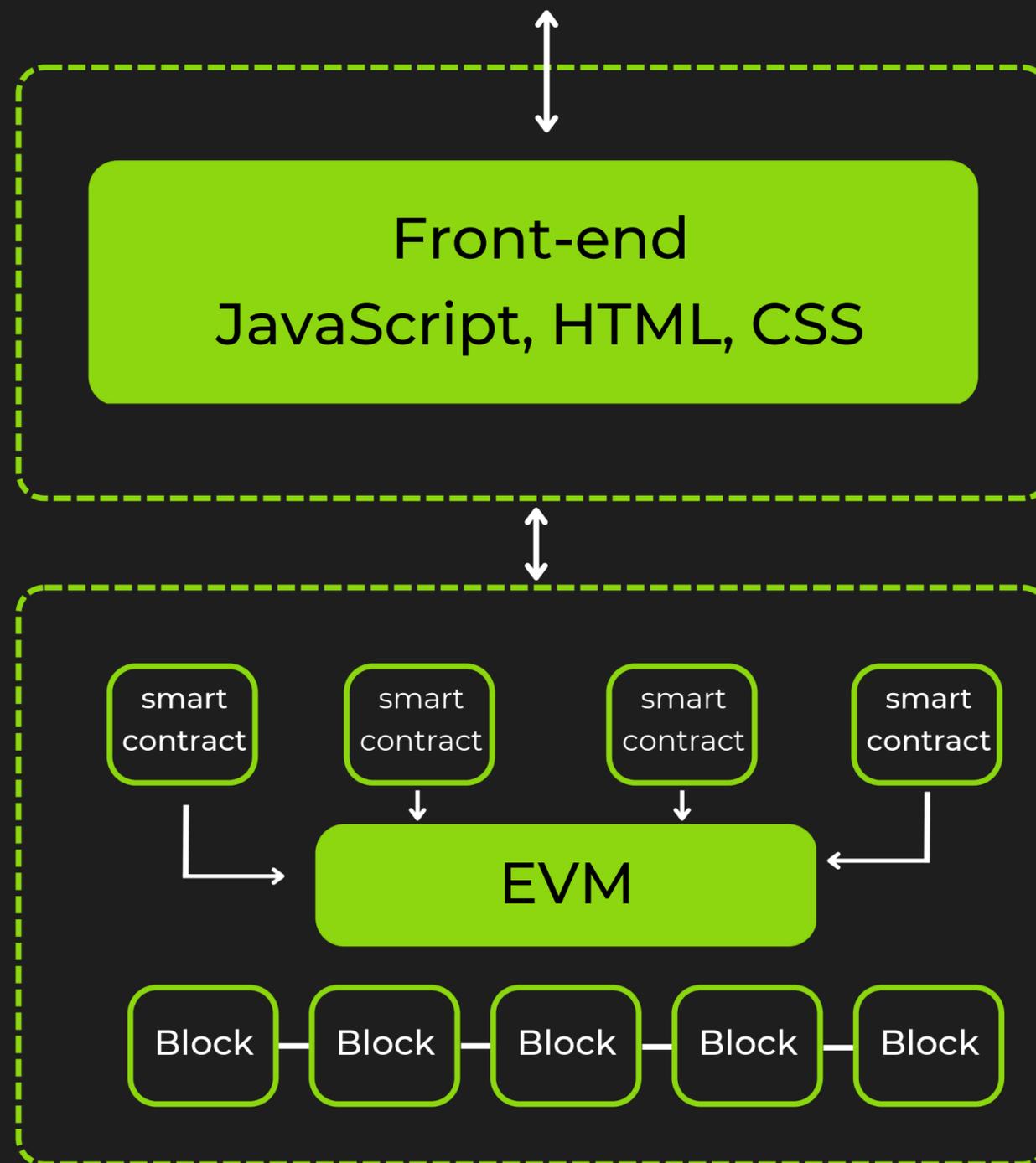


В итоге web3 приложения можно назвать децентрализованными приложениями (dApps)

Системообразующая технология – блокчейн, которая коллективно управляется всеми участниками своей сети, и одновременно никем единолично.

Архитектура приложений web3

Пользователь



В чем отличие web3 приложений?

Отличие web3 приложений

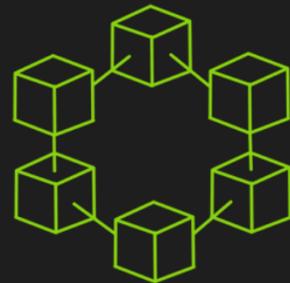
База Данных (данные) → состояния
в смарт-контрактах

Бэкенд (логика) →
смарт-контракты

Фрондэнд (интерфейс) →
добавляется web3 wallet

Компоненты web3 приложений

Блокчейн



Каждый может получить доступ к блокчейну и загрузить в нее свой код.

Блокчейн не принадлежит никому единолично, это собственность всех участников данной сети. в нее свой код.

В блокчейне Эфириума все данные публичны и хранятся в состояниях смарт-контрактов всех участников данной сети.

Смарт-контракты

Программа, работающая на блокчейне и определяющая всю логику изменений, происходящих в блокчейне.

Смарт-контракты написаны на языках высокого уровня, таких как **Solidity** или **Vyper**.



Компоненты web3 приложений

Виртуальная машина Эфириума

Каждый может получить доступ к блокчейну и загрузить в нее свой код.

Чтобы это произошло, код смарт-контрактов компилируется в байт-код и загружается (деплоится) в сеть блокчейна. получить доступ к блокчейну и загрузить в нее свой код.

Фронтенд

Определяет логику веб-интерфейса сайта, который в свою очередь взаимодействует с логикой самого приложения, которое запрограммировано в смарт-контрактах.

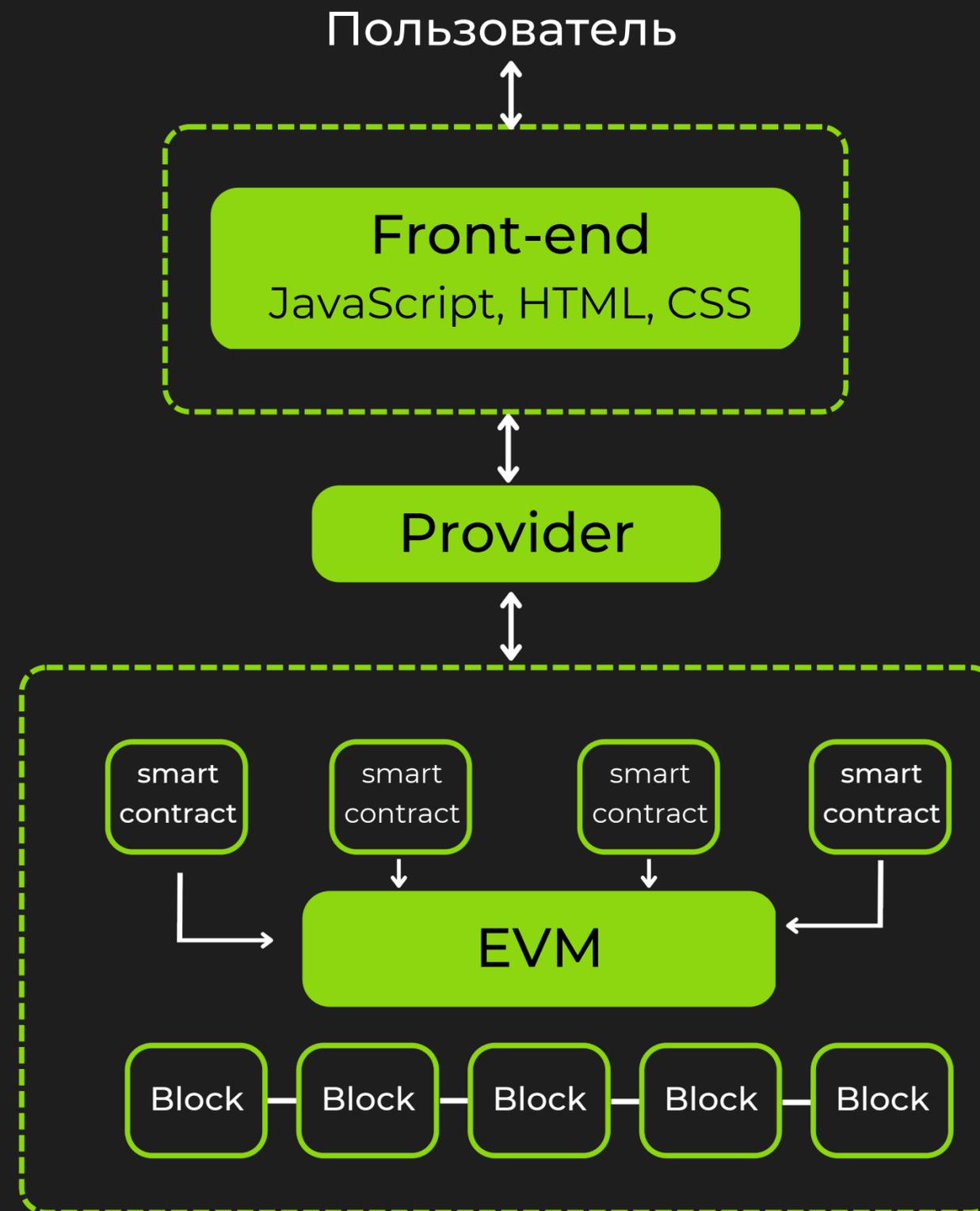
Для создания фронтенда для web3 приложений используются библиотеки как ethers.js или web3.js совместно с тулзами для подключения кошельков

Как фронтэнд взаимодействует со смарт-контрактами на блокчейне?

Блокчейн Эфириума – это есть децентрализованная сеть, где каждый узел сети хранит копию всех состояний «стейт машины», включая код, данные, и другую информацию, связанную с нашим смарт-контрактом.

Чтобы взаимодействовать с данными и кодом, загруженными в блокчейн, достаточно взаимодействовать с одним из этих узлов, а уже этот узел распространит запрос на транзакцию, которую следует выполнить на виртуальной машине Эфириума

Архитектура приложений web3



Есть два способа взаимодействовать с EVM блокчейном

Настроить собственный узел с программным обеспечением Эфириума.

Использовать сторонние сервисы, предоставляющие доступ к существующим узлам сети Эфириума, такие как Infura , Alchemy и Quicknode.

Взаимодействие с EVM Блокчейном

Использование сторонних сервисов ограждает от всех проблем, связанных с самостоятельным запуском собственного узла. К тому же, установка собственного узла сети Эфириум может занять не один день. Очень много данных, которые потребуются синхронизировать.

Поэтому, многие собственники децентрализованных приложений предпочитают использовать сторонние сервисы, такие как Infura или Alchemy, для управления инфраструктурой своих узлов.

Провайдеры

Провайдеры - узлы, к которым вы подключаетесь для взаимодействия с блокчейном (независимо от того, настраиваете ли вы их самостоятельно или используете существующие от сторонних сервисов).

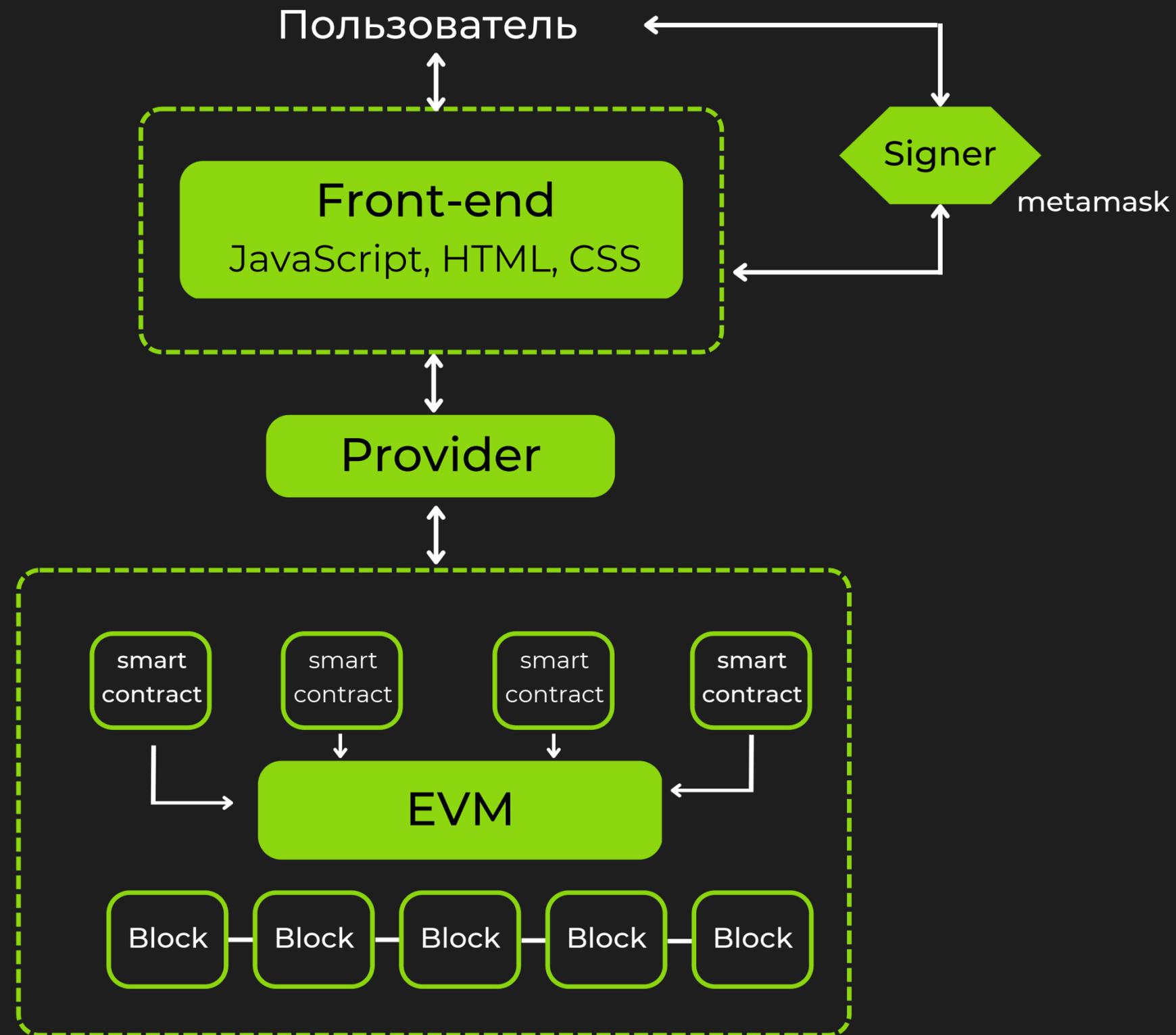
Провайдер реализует спецификацию **JSON-RPC** протокола, что гарантирует наличие единого набора методов взаимодействия внешних приложений и блокчейна. Грубо говоря, это **упрощенный протокол удаленного вызова процедур (RPC)**, который определяет несколько структур данных и правила их обработки.

Подписант (Signer)

С помощью провайдера, вы сможете читать данные из блокчейна, но, если вы хотите сохранять в блокчейн свои данные, нужно сделать еще кое-что. Прежде чем вы отправите свою транзакцию в блокчейн, ее следует «подписать», используя свой закрытый ключ.

Для «подписания» транзакции обычно используется кошелек [MetaMask](#), [Trust wallet](#), [Tinch wallet](#). ИЛИ для проектов с Кастодиальной составляющей - приватный ключ в бэкенде

Архитектура приложений web3



Self-custody Кошельки (Signer)

Кошелек (как Метамаск)— это инструмент, упрощающий управление ключами и подписание транзакций. Metamask хранит закрытые ключи в браузере, и всякий раз, когда потребуется подписание транзакции, интерфейс приложения обращается к Metamask'у.

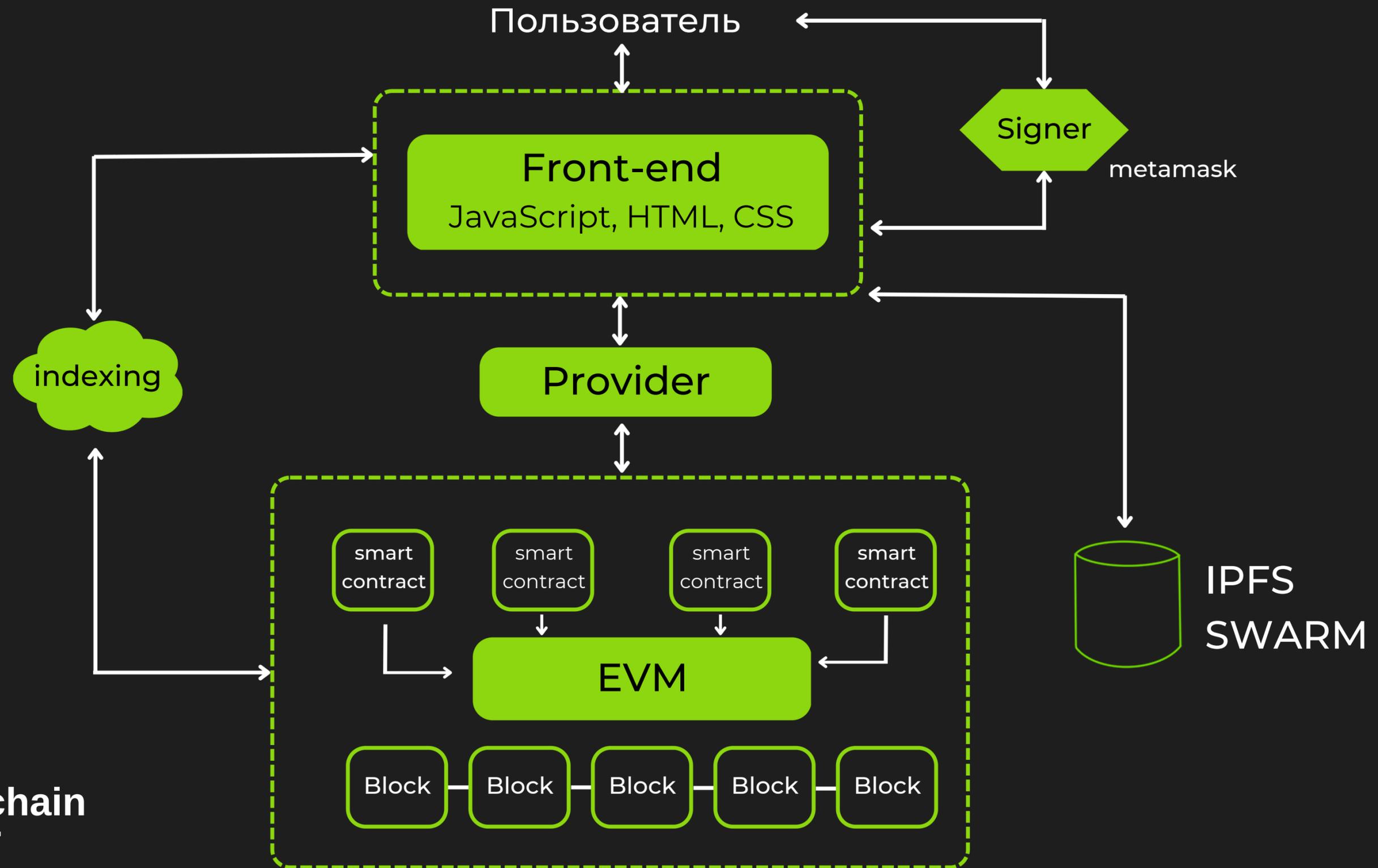
Self-custody Кошелек также обеспечивает подключение к блокчейну в качестве «провайдера», поскольку у него уже есть подключение к узлам через сторонний сервис **как Infura/Alchemy/Quicknode**. Таким образом, Metamask является и провайдером, и сервисом для подписи транзакций.

Хранение данных в блокчейне (1/2)

Описанная архитектура логична, если все данные приложения полностью хранятся в блокчейне. Но каждый, кто создавал приложения на Эфириуме, знает, что такое хранение очень быстро становится очень дорогим.

В блокчейне Эфириум пользователь платит каждый раз, когда добавляет новые данные в сеть. Это связано с тем, что добавление нового состояния (новой транзакции, изменяющей состояние блокчейна) увеличивает затраты на поддержание сети для существующих узлов.

Архитектура приложений web3



Хранение данных в блокчейне (2/2)

IPFS — это сервис, предоставляющий доступ к распределенной файловой системе для хранения и доступа к данным. Таким образом вместо того, чтобы хранить данные в централизованной базе данных, сервис IPFS предоставляет возможность хранить данные в **одноранговой сети** и легко получать доступ к ним, когда это потребуется.

IPFS также взаимодействует с **Filecoin**, протоколом и криптовалютой, стимулирующей хранение данных для других пользователей по всему миру. Такие сервисы есть, например, у **Infura** или **Pinata**, суть которых в том, что вы можете «прикрепить» свои файлы к IPFS, взять хэш IPFS и сохранить его в блокчейне.

Хранение данных в блокчейне (2/2)

Если вы хотите создать действительно децентрализованное приложение, вам следует разместить код веб-интерфейса своего приложения в одном из децентрализованных сервисов для хранения данных, таких как IPFS или Swarm.

Альтернативно можно сделать исходный код фронтенда открытым

Запросы в блокчейне

До сих пор мы говорили о том, как писать в блокчейн, подписывая транзакции и затем отправляя их в блокчейн. Но что насчет чтения данных из смарт-контрактов в блокчейне?

Есть четыре основных способа :

1. Вызывать viewфункций в смарт-контрактах
2. Читать события смарт-контракта
3. Использовать theGraph
4. Использовать сторонние сервисы или API

Спасибо!