

УДК 004.02, 004.04, 004.6

ТЕХНОЛОГИИ И ИНСТРУМЕНТЫ РАЗРАБОТКИ DAPP

О. Д. Окладникова¹, С. В. Окладникова²¹Национальный исследовательский университет ИТМО, Санкт-Петербург, Россия²Астраханский государственный архитектурно-строительный университет, Астрахань, Россия¹ORCID: 0009-0009-0831-7452²ORCID: 0000-0002-0513-9664Статья **поступила** 26.06.2023, **принята** к публикации 27.09.2023. Опубликована онлайн.

Аннотация. Появление Интернет привело к глобальным изменениям в жизни общества, способствовало развитию цифровых технологий и внедрению их во все области экономической и общественной жизни. В свою очередь научное и технологическое развитие трансформировало и сам Интернет. На современном этапе особую значимость в его развитии играет блокчейн, как инновационная технология хранения и передачи данных, в которой пользователь будет являться владельцем своих данных в цифровой экосистеме и осуществлять контроль над своими персональными данными. Изменение философии Интернет требует от разработчиков новых технологических идей и подходов при создании современных веб-приложений, следовательно, возрастает и запрос на новые инструменты их разработки.

В статье авторы рассматривают технологические особенности блокчейн, а также проводят обзор основных инструментов, используемых для разработки децентрализованных приложений (DApp), в частности при решении задач интеграции клиентской части с блокчейн сетями и разработки Front-end DApp. На примере авторы с помощью данных инструментов демонстрируют реализацию отдельных функций, выполняемых на стороне клиента приложения. Авторами рассматриваются особенности одной из популярных у блокчейн-разработчиков инфраструктурной блокчейн-платформы Alchemy и ее инструментальные возможности.

Ключевые слова: блокчейн, DApp, фреймворк, смарт-контракт, Metamask, децентрализация, токен, алгоритм консенсуса, Alchemy, API

TECHNOLOGIES AND TOOLS FOR DAPP DEVELOPMENT

Olga D. Okladnikova¹, Svetlana V. Okladnikova²¹ITMO National Research University, Russia, Saint Petersburg²Astrakhan State University of Architecture and Civil Engineering, Russia, Astrakhan

Abstract. The emergence of the Internet has led to global changes in the life of society, contributed to the development of digital technologies and their implementation in all areas of economic and social life. In turn, scientific and technological development has transformed the

Internet itself. At the present stage, blockchain plays a special role in its development, as an innovative technology for storing and transmitting data, in which the user will be the owner of his data in the digital ecosystem and exercise control over his personal data.

Changing the philosophy of the Internet requires developers to develop new technological ideas and approaches when creating modern web applications, therefore, the demand for new tools for their development is also increasing.

In the article, the authors consider the technological features of blockchain, as well as review the main tools used for the development of decentralized applications (DApp), in particular when solving the problems of integrating the client part with blockchain networks and

developing Front-end DApp. By example, the authors use these tools to demonstrate the implementation of individual functions performed on the client side of the application. The authors consider the features of one of the most popular blockchain developer's infrastructure blockchain platform Alchemy and its instrumental capabilities.

Keywords: blockchain, DApp, framework, smart contract, Metamask, decentralization, token, consensus algorithm, Alchemy, API

Статья **поступила** 26.06.2023, **принята** к публикации 27.09.2023. Опубликовано онлайн.

Введение

Блокчейн — это развивающаяся технология, не имеющая в настоящее время универсальных стандартов и общих принципов взаимодействия между разными сетями. Существующие инструментальные средства используют различные блокчейн-платформы, соответственно выбор стека технологий для разработки DApp (децентрализованных приложений) ограничен возможностями этих платформ. В качестве инструментов интеграции клиентской части с блокчейн сетями и разработки Front-end DApp, как правило, используются существующие языки программирования (Java, Python, JavaScript, C++, C#, Go и др.) и фреймворки (React, Angular, Vue.js и др.), функциональные возможности которых могут решать задачи блокчейн.

Методы

Новым технологическим развитием Интернет является Web 3.0, основанный на технологии искусственного интеллекта, включая машинное обучение и блокчейн. В основе концепции Web 3.0 заложена идея создания «умных» сайтов и приложений на

основе технологии блокчейн, которая позволит обеспечить децентрализацию, конфиденциальность и безопасность информации [1].

В архитектуре современных Web приложений обязательными элементами являются база данных (database), серверная (back-end) и клиентская (front-end) части приложения. Пользователь взаимодействует с клиентской частью приложения, которая определяет логику пользовательского интерфейса и написана на JavaScript, HTML и CSS.

Серверная часть приложения отвечает за бизнес-логику приложения и может быть написана на GO, Java, Python или др. База данных может быть разработана на PostgreSQL, MySQL, MongoDB и др.

В архитектуре приложений Web 3.0 нет централизованной базы данных, в которой хранится состояние приложения, и нет централизованного веб-сервера, на котором находится внутренняя логика. Одной из ключевых особенностей Web 3.0 является децентрализованность.

На рисунке 1 приведен пример архитектурного решения приложения, построенного на базе блокчейн Ethereum.

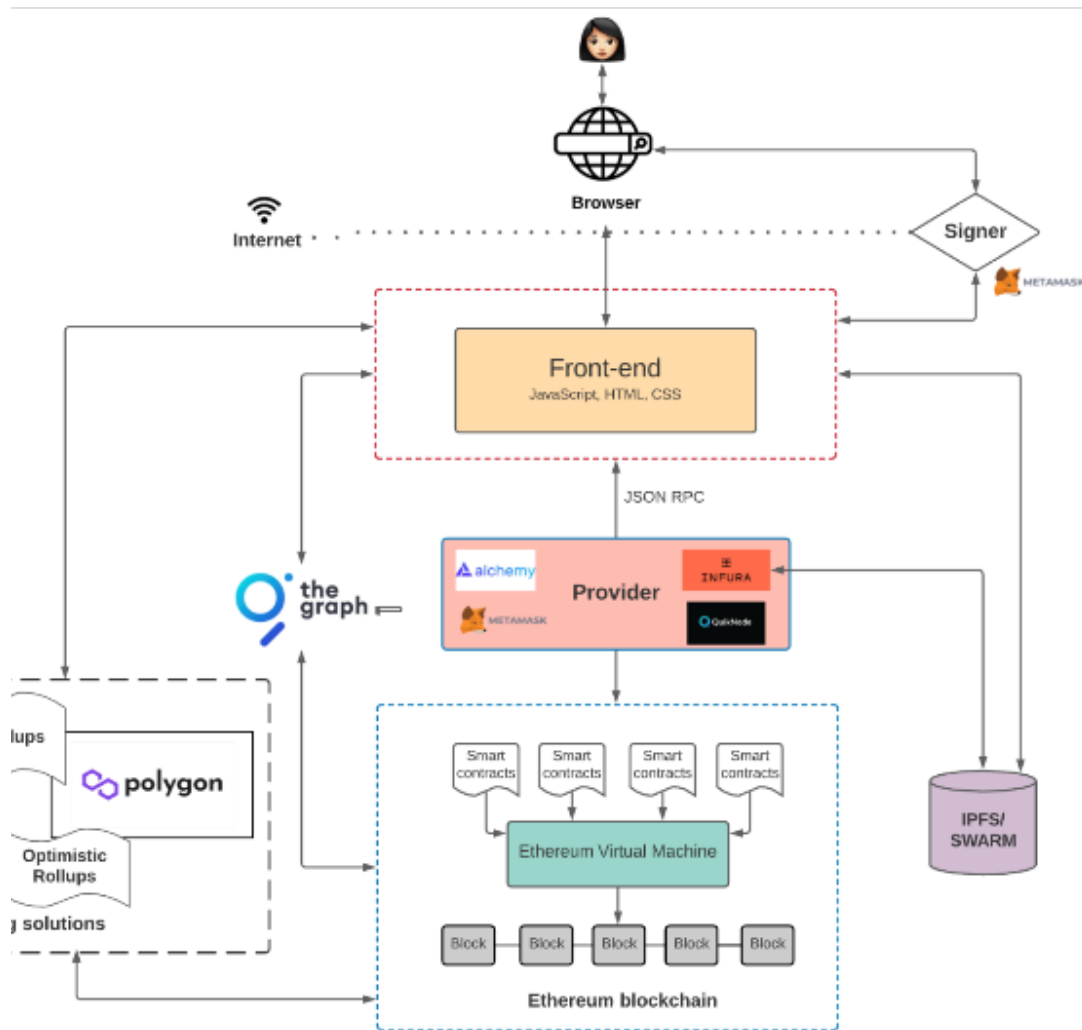


Рисунок 1. Типовая архитектура приложения Web 3.0 [2]

Front-end у децентрализованных, как и у обычных приложений практически не отличается. Интерфейс может быть написан на любом языке программирования и вне зависимости от блокчейн Front-end должен: 1) обеспечивать пользователю понятное для него взаимодействие с приложением; 2) поддерживать работу со смарт-контрактами (smarts contracts).

Смарт-контракты представляют собой программы, написанные на языках высокого уровня (например, Solidity, Vyper и др.), которые отвечают за логику изменения состояний, происходящих в блокчейн. Код смарт-контракта хранится в блокчейне Ethereum.

Back-end в децентрализованных приложениях представляет собой

непосредственно блокчейн, т.е. базу данных, в которой записан программный код и происходит учет всех действий внутри приложения.

Физически эта база данных находится на устройствах множества пользователей. Единое хранилище информации (database) отсутствует.

Каждый узел в сети Ethereum хранит копию всех состояний, включая код и данные, связанные с каждым смарт-контрактом. Взаимодействие с данными и кодом в блокчейн предусматривает взаимодействие с одним из этих узлов. Это связано с тем, что любой узел может транслировать запрос на выполнение транзакции на виртуальную машину EVM Ethereum (EMV).

Затем майнер выполнит транзакцию и распространит полученное изменение состояния на остальную часть сети. EVM выполняет логику, определенную в смарт-контрактах, и обрабатывает изменения состояния, которые на ней происходят. EVM не понимает языки высокого уровня, такие как Solidity и Vyper, которые используются для написания смарт-контрактов. Вместо них необходимо скомпилировать язык высокого уровня в байткод, который затем может выполнить EVM.

Узлы, через которые можно подключиться и взаимодействовать с блокчейн называют «провайдерами». Они реализуют спецификацию JSON-RPC — упрощенный протокол удаленного вызова процедур (RPC) без сохранения состояния, который определяет несколько структур данных и правила их обработки. Он не зависит от транспорта, поэтому концепции могут использоваться в одном и том же процессе через сокеты, HTTP или во многих различных средах передачи сообщений. Подключение к блокчейн через провайдера позволяет прочитать состояние, хранящееся в блокчейн. Если требуется писать состояние то, прежде чем отправить транзакцию в блокчейн ее необходимо подписать, используя уникальный закрытый ключ.

Для подписания транзакций может использоваться, например, Metamask, который вызывается в случае, когда внешнему интерфейсу требуется, чтобы пользователь подписал транзакцию. Metamask хранит закрытые ключи пользователя в браузере, и всякий раз, когда внешнему интерфейсу требуется, чтобы пользователь подписал транзакцию, он вызывает Metamask. Metamask одновременно является поставщиком и подписантом, т.к. обеспечивает подключение к блокчейн (в качестве «провайдера»).

Для хранения и доступа к данным в архитектуре Web 3.0 используется IPFS — распределенная файловая система, которая распределяет и хранит данные в одноранговой сети. IPFS имеет пооператорный слой Filecoin, который «стимулирует» узлы по

всему миру хранить и извлекать эти данные. В качестве провайдера можно использовать, например, Infura, предоставляющий узел IPFS, или Pinata, предоставляющий простой в использовании сервис, где можно «прикрепить» файлы к IPFS, взять хэш IPFS и сохранить его в блокчейн.

Swarm — это встроенная децентрализованная сеть, которая обеспечивается с помощью смарт-контрактов на блокчейн Ethereum для хранения и извлечения данных. Код внешнего интерфейса не хранится в блокчейн, а размещается в децентрализованном решении для хранения, таком как IPFS или Swarm [3].

Результаты

DApp (Decentralized Application) — это децентрализованные приложения, которые состоят из front-end и взаимодействуют со смарт-контрактами, т.е. они работают без back-end, под управлением алгоритмов, записанных в смарт-контрактах. Ключевое отличие DApp от приложений Web 2.0 заключается в том, что смарт-контракт и блокчейн заменяют типичный сервер и базу данных. DApp работают на сетях блокчейн, а не централизованных серверах. Существует три типа решений DApp [4]:

- DApp имеют собственный блокчейн, например, Ethereum, Tron, EOS и др.
- DApp разработаны и работают на основе существующих платформ блокчейн.

Представляют из себя протоколы и обладают своим механизмом выпуска токенов. Пример — протокол MakerDAO, который основан на Ethereum. MakerDAO конвертирует криптовалюту ETH в стейблкоины DAI.

– DApp основаны на приложениях, основанных на блокчейн. Работают на протоколе приложений Type II, обладают механизмом выпуска токенов. Примером такого типа DApps является Augur, приложение. Которое в целях уменьшения волатильности использует блокчейн Ethereum на использовании протокола MakerDAO.

В качестве основных критериев, характеризующих DApp, выделяют:

- Открытый код. Исходный код программы должен быть полностью открытым и работать автономно. Это позволит любому пользователю выполнить его проверку, найти и исправить ошибки, улучшить приложение. Внесение изменений возможно только на основе консенсуса пользователей. Для работы с кодом DApp можно использовать сервисы для хостинга и совместной разработки IT-проектов — например, GitHub или Netlify.
- Децентрализация. Данные должны храниться в криптографически защищенном виде, в общедоступном, децентрализованном блокчейн.
- Токены. В сети приложения должны ходить в обращении токены (цифровые монеты), которые позволяют осуществлять доступ к некоторым функциям программы, быть разменной валютой для майнеров, пользователей.
- Алгоритм консенсуса. В DApp используется стандартный криптографический алгоритм для запуска интеллектуального анализа данных в сети узлов (узлов). Алгоритмы консенсуса (например, Proof of Work (PoW), Proof of Stake (PoS)) принимают решение о добавлении новой информации в блок [5].

Для работы с аккаунтами в DApp может использоваться любой из браузеров (например, Chrome, FireFox и др.) с установленным дополнением, например, Metamask. С помощью браузера объект Web 3.0 внедряется в сеть блокчейн. Для обеспечения взаимодействия веб-приложения с блокчейн Ethereum используется Web3.js.

Данная библиотека позволяет разработчикам взаимодействовать со смарт-контрактами, извлекать данные из блокчейн и выполнять транзакции в сети Ethereum. Web3.js выполняет важную функцию подключения вебсайтов к блокчейн. Без неё вебсайты были бы ограничены параметрами соединений Web 2.0.

Клиентская часть DApp представляет собой desktop приложение, которое может быть разработано на одном из фреймворков, например, React. В задачи desktop приложения входит установка и настройка Go Ethereum (Geth) — клиент Ethereum, написанный на языке Go. Запуск Geth превращает компьютер в узел Ethereum.

Межпланетная файловая система (Interplanetary Filing System, IPFS) — это блокчейн-сеть, используемая для хранения всех типов файлов децентрализованным, одноранговым (P2P) способом без доверия. IPFS необходим для хранения таких файлов как картинки, документы и т.д.

Для обеспечения надежности взаимодействия клиента и блокчейн используется технология IPC (механизм, который позволяет процессам взаимодействовать друг с другом и синхронизировать свои действия), которая обеспечивает стабильный обмен данными между потоками информации одного или разных процессоров.

Технология блокчейн принципиально меняет архитектуру приложений Web 3.0, в частности, принцип хранения, распространения и использования данных, обеспечивая нативный уровень транзакций. Ключевыми особенностями приложений Web 3.0 выступают децентрализация, безопасность, масштабируемость и конфиденциальность.

Ключевое отличие приложений Web 3.0 заключается в том, что смарт-контракт и блокчейн заменяют типичный сервер и базу данных, принадлежащие одному человеку или компании, которые их обслуживают [6,7].

Компоненты DApp можно разделить на три различные категории: смарт-контракты, фронтенд-логика и пользовательский интерфейс, хранилище данных. Для разработки перечисленных компонентов DApp могут использоваться различные инструменты: блокчейн-платформы, языки программирования, фреймворки, библиотеки, расширения для браузеров и др.

Блокчейн-платформа — это устоявшаяся технология, состоящая из непрерывной цепочки информационных блоков, функционирующая по определенным протоколам. В настоящее время существует пять поколений блокчейн-платформ. Один из способов разработки DApp заключается в использовании существующей блокчейн-платформы, имеющей открытый исходный код, что позволяет форкнуть их репозитории и развернуть код на собственных серверах. Наибольшее количество DApp размещено на Ethereum, Solana и Tron.

Для сравнения блокчейн-платформы используют следующие критерии [8,9,10]:

- Создание и подтверждение блока — это метрики скорости транзакций. Чем быстрее генерируются блоки, тем меньше времени пользователям приходится ждать денежных переводов или выполнения смарт-контракта.

- Производительность показывает, способна ли платформа обрабатывать смарт-контракты в достаточном объеме, чтобы работать с более сложными продуктами на блокчейн, такими как DeFi, GameFi и DAO.

- Масштабируемость напрямую связана с количеством пользователей и их взаимодействиями в сети (транзакциями, выполнением смарт-контрактов, инфраструктурными запросами).

Одним из актуальных инструментов DApp, которые упрощают разработку и поддержку сложных и высоконагруженных веб-приложений, являются фреймворки. Основными критериями, определяющими выбор фреймворка, являются: развертывание в локальной блокчейн-сети, наличие быстрой и эффективной системы проверки, способность разворачивать смарт-контракт в тестовой сети. К наиболее используемым фреймворкам для разработки смарт-контрактов относятся:

- Hardhat (JavaScript) — среда разработки с открытым исходным кодом для создания и тестирования смарт-контрактов на Ethereum, имеет встроенный компилятор

Solidity, среду тестирования, инструмент отладки, инструмент развертывания и систему плагинов, простую интеграцию. Для тестирования используется Waffle и Ethers.js.

- Truffle (JavaScript) — среда разработки, среда тестирования и канал управления ресурсами Ethereum, легко интегрируется со своими родственными инструментами Drizzle и Ganache, доступна прямая интеграция с плагином для обновления смарт-контрактов от OpenZeppelin:

- Brownie (Python) — аналог фреймворка Truffle для разработки полного цикла умных контрактов для Ethereum-based платформ.

Логика Back-end DApp требует от разработчика написания кода смарт-контракта для развертывания на блокчейн. Front-end DApp может использовать стандартные веб-технологии, такие как HTML и JavaScript. Это позволяет разработчикам использовать знакомые инструменты, библиотеки и фреймворки (например, React, Angular, Vue.js и др.) вместе с библиотеками для интеграции с блокчейн сетью.

Пользовательский интерфейс на стороне клиента DApp обычно связан со смарт-контрактами через клиентские библиотеки, такие как Web3.js, Ethers.js и др., которые поставляются в комплекте с ресурсами Front-end и отправляются в браузер вместе с пользовательским интерфейсом.

Взаимодействие со смарт-контрактами, например, подписание сообщений и отправка транзакций смарт-контрактам, обычно осуществляется через браузерный Web3-кошелек, например, MetaMask [11].

Обсуждение

MetaMask — это браузерный инструмент, разработанный для Ethereum и доступный в виде расширения для браузера, которое превращает все основные браузеры (Chrome, Firefox и Opera) в браузер Ethereum, MetaMask позволяет вам без проблем взаимодействовать с платформой Ethereum.

Его можно использовать для извлечения данных из блокчейн и позволяет пользователям безопасно отправлять или получать подписанные транзакции.

Расширение включает Ethereum WEB3 API в контекст JavaScript каждого веб-сайта, чтобы DApp могли считывать данные прямо из блокчейн.

Предоставляемый MetaMask инструмент не строго описывает Ethereum API, что затрудняет процесс разработки. Недостаток функционала API блокчейн сети Ethereum не позволяет предоставить

пользователю полную информационную картину о его активности. Для получения более подробной инфографики отсутствует базовый функционал, например, просмотр истории транзакций конкретного пользователя.

Пример простой реализации взаимодействия с кошельком в Ethereum представлен на рисунках 1 и 2.

На рисунке 2 выводится базовая информация о кошельке: адрес, баланс, количество транзакций, на рисунке 3 — использование Ethereum API через клиент MetaMask.

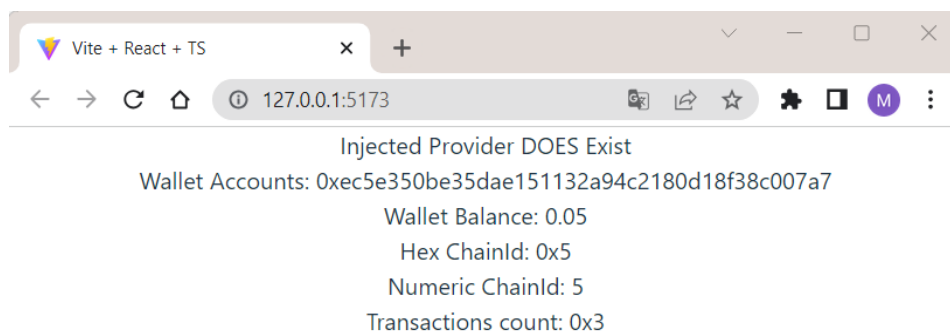


Рисунок 2. Вывод данных кошелька

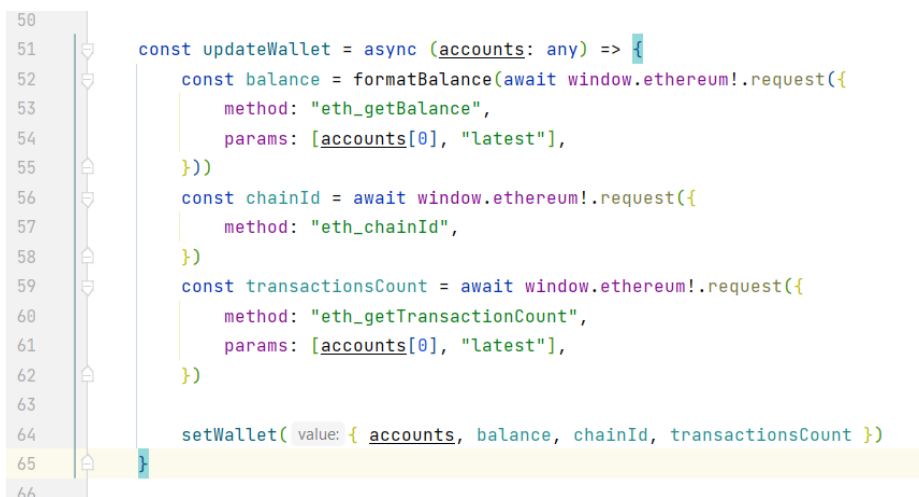


Рисунок 3. Программный код для обновления данных кошелька

В настоящее время наибольшей популярностью у блокчейн-разработчиков пользуются инфраструктурные блокчейн-платформы, обладающие необходимыми

инструментами для создания DApp и упрощающие процесс их разработки (например, Alchemy, Moralis).

Особенность таких платформ заключается в отсутствии необходимости самостоятельно управлять инфраструктурой блокчейн.

Alchemy поддерживает разработку известных сетей блокчейн, таких как Ethereum, Solana, Polygon и Optimism. Программные инструменты Alchemy предоставляют готовый набор, помогающий масштабировать криптографические приложения, в первую очередь на блокчейн Ethereum. Это сокращает время выполнения сложных и дорогостоящих задач, таких как отладка и запись в узлы. Основные инструменты: Supernode, Build, Monitor, Notify и Transact [12].

Supernode — один из наиболее часто используемых API-интерфейсов Ethereum, позволяющий масштабировать функциональность каждого узла с помощью выделенных распределенных систем, обеспечивая при этом корректность данных в режиме реального времени за счет использования собственной службы координатора. Supernode предлагает однолинейный инфраструктурный API для выполнения всех функций одного узла с более высокой доступностью и временем безотказной работы. API-интерфейс однолинейной инфраструктуры также может обеспечить наличие необходимых ресурсов, необходимых для запуска приложения в рабочей среде.

Расширение Alchemy Web3 для библиотеки Web3.js и обновленные Smart WebSockets обеспечивают автоматическую обработку повторного подключения, наряду с обратной засыпкой для пропущенных событий.

Build — это собственный набор инструментов разработчика, который не требует настройки, позволяет ускорить разработку прототипов, отладку и поставку блокчейн-продукта. Одной из проблем разработчика приложений Web 3.0 является отсутствие инструментов для отладки ошибок клиентской части DApp. Build предлагает решение для выявления ошибок в решениях Web3 с

помощью уникальных функций: Explorer, Composer и Mempool Visualizer.

Explorer помогает мгновенно просматривать миллионы запросов за прошлые периоды для выявления конкретных ошибок, шаблонов ошибок и оптимизации производительности.

Composer помогает в выполнении прямых вызовов JSON-RPC с панели управления, что позволяет разрабатывать прототипы, что также помогает в разрешении неудачных запросов наряду с поддержкой разработчиков в изучении поведения с помощью новых методов.

Mempool Visualizer позволяет разработчикам визуализировать просмотр состояния транзакций в режиме реального времени для выявления отложенных, отброшенных или неразрешенных транзакций.

Monitor — это набор информационных панелей и оповещений для мониторинга работоспособности и производительности приложений и поведения пользователей. Как правило, традиционные инструменты аналитики данных не могут обеспечить доступ к необходимым данным.

Функция Usage Analytics позволяет мгновенно осуществлять мониторинг работоспособности всех приложений Web 3.0, находящихся в одном месте, а именно, определять частоту ошибок, количество запросов в секунду и время отклика. Мониторинг использования приложения выполняется в течение заданного периода времени. Разработчики могут получать информацию об использовании приложений, сравнивая их между собой.

Функция User Insights предоставляет сводные данные о местоположении, активности и трафике без каких-либо проблем с безопасностью и конфиденциальностью.

Notify предназначен для улучшения взаимодействия пользователя с приложением Web 3.0, а именно, для организации рассылки оперативных уведомлений, например, уведомления о добытых транзакциях, событиях

смарт-контрактов и удаленных транзакциях.

Уведомления о добытых транзакциях помогают пользователям идентифицировать случаи, когда происходят депозиты, внутри-игровые действия, покупки и другие действия в сети. Уведомления об отброшенных транзакциях могут помочь пользователям принять незамедлительные меры после сбоя транзакций.

В результате пользователи приложения не будут жаловаться на исчезновение токенов, пропущенные сделки и проигранные аукционы. Кроме того, веб-перехватчики адресной активности могут включать уведомления для внешних и внутренних переводов, переводов ERC20 и переводов NFT.

Transact включает в себя набор сервисов, которые позволяют отправлять транзакции с большей надежностью, скоростью и экономичностью. В большинстве случаев транзакции Web 3.0 могут быть отменены без какой-либо причины, что приводит к потере времени и денег для пользователей. Кроме того, транзакции могут стать значительно медленнее, поскольку пользователям приходится постоянно обновлять свои приложения для проверки статуса транзакции.

Другой риск в транзакциях связан с ручным обновлением приложения для определения статуса транзакций. Transact оптимизирует транзакцию на каждом этапе, чтобы обеспечить ее более высокую скорость, безопасность, экономичность и надежность. Transact может работать с автоматическим отслеживанием и повторной отправкой транзакций, тем самым предотвращая удаление транзакций.

Кроме того, разработчики также могут получить доступ к конечным точкам частных транзакций для отправки и отмены транзакций без видимости для общественности. В результате, это может предотвратить проблемы нежелательного опережения и арбитража.

Для интеграции различных функций в приложения разработчики могут использовать API:

- NFT API мгновенно идентифицирует, проверяет и отображает любой невзаимозаменяемый токен в популярных сетях блокчейн;
- Token API упрощает процесс запроса информации о конкретных токенах, такой как баланс кошелька или метаданные;
- Transfers API извлекает истории транзакций, относящихся к конкретным адресам;
- Transact API представляет собой набор API для обеспечения скорости, безопасности и прозрачности транзакций;
- Методы Trace API позволяют получить подробную информацию об активности в сети;
- Debug API позволяет получить информацию и данные, связанные с поведением транзакций;
- Transactions Receipts API позволяет получать все квитанции о транзакциях для определенного блока в соответствии с хэшем блока или по номеру.
- Alchemy Web3 API является альтернативой Web3.js, адаптирован для работы с Alchemy без каких-либо конфликтов, обеспечивает поддержку WebSocket и автоматические повторные попытки;
- API подписки разрешает подпискам получать полные квитанции транзакций для новых неразрешенных транзакций, а не только хэш транзакции.

Пример использования API приведен на рисунке 4 (вывод баланса, вывод текущего курса GAS, вывод метаданных транзакций). Программный код использования Alchemy SDK для получения информации представлен на рисунке 5.

```

PS C:\Users\Meyer\Olya\Downloads\fe-bc\alchemy> node index.js
{
  "hash": "0xb16fa669e201d0d6c55b0ebb9ca342b07173018758830e96aac623d429ddff3",
  "type": 0,
  "accessList": null,
  "blockHash": "0xf6d0f6643f55f4f98dfdc014bce9049735efc3379ce3e38364c18c311127ea7",
  "blockNumber": 9118216,
  "transactionIndex": 0,
  "confirmations": 48482,
  "from": "0x187aA44088a0eA95bdc65fef6E5071E9468f7d8f",
  "gasPrice": "0x174876e800",
  "gasLimit": "0x5208",
  "to": "0x84068928e11168c4c3144028538c72a0507f35f8",
  "value": "0xb1a2bc2ec50800",
  "nonce": 333981,
  "data": "0x",
  "r": "0x44893108fef12c5f73a2e0cab3fa0d825463e75b12aa6725ef57e18992ec774",
  "s": "0x612ffbf18adcf7c977d58c261d2c75ed47979c357cfe041ef5d3a5dfb8465fe1",
  "v": 46,
  "creates": null,
  "chainId": 5,
  "wait": [Function (anonymous)]
}

```

Рисунок 4. Вывод результата работы Alchemy API

```

1 import { Network, Alchemy } from 'alchemy-sdk';
2
3 const settings = {
4   apiKey: "0bPynfNZ496v6LyTvVNDV1udeR6GoiX",
5   network: Network.ETH_GOERLI,
6 };
7
8 const alchemy = new Alchemy(settings);
9
10 // get the latest block
11 const latestBlock = alchemy.core.getBlockByNumber(123456789).then(console.log);
12 const price = alchemy.core.getGasPrice().then(console.log);
13 const transaction = alchemy.core.getTransaction("0xb16fa669e201d0d6c55b0ebb9ca342b07173018758830e96aac623d429ddff3").then(console.log);
14

```

Рисунок 5. Программный код вывода метаданных транзакций

Заключение

DArr работает как программное обеспечение с открытым исходным кодом, реализованное на децентрализованной цепочке блоков, и работает с использованием токенов, которые генерируются с использованием согласованного протокола или алгоритма. DArr обеспечивает масштабируемость, безопасность, высокую пропускную способность, низкую стоимость транзакций и др. Существует множество различных инструментов, каждая из которых имеет свои преимущества и недостатки. Выбор инструментов блокчейн зависит от целей и задач DArr.

В последнее время среди разработчиков блокчейн существует запрос на масштабируемые поставщики серверной

инфраструктуры, объединяющие множество различных инструментов, которые позволили бы обеспечить работу внутренней инфраструктуры блокчейн, позволяя разработчикам сосредоточиться на разработке внешнего интерфейса и веб-сервисов на стороне клиента, а также упростить работу разработчиков и сократить время написания программного кода.

Список литературы

1. Huang H., Kong W., Zhou S. A Survey of State-of-The-Art on Blockchains / H. Huang, W. Kong, S. Zhou [et al.] // ACM Computing Surveys. — 2021. — Vol. 54, No. 2. — DOI 10.1145/3441692. — EDN KKXOEU.
2. Kasireddy P. The Architecture of a Web 3.0 application. [Электронный ресурс].

URL: <https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application/> (дата обращения: 09.06.2023)

3. Hofmann, A., Gwinner, F., Winkelmann, A., Janiesch, C. Security Implications of Consortium Blockchains: The Case of Ethereum Networks / A. Hofmann, F. Gwinner, A. Winkelmann, C. Janiesch. // JIPITEC – Journal of Intellectual Property, Information Technology and E-Commerce Law. – 2021. – Vol. 12. – No. 4. – PP. 347-359

4. Shamili, P. Enhancing the Decentralized Application (Dapp) for E-commerce by Using the Ethereum Blockchain / P. Shamili, B. Muruganatham // Lecture Notes in Networks and Systems. – 2022. – Vol. 341. – P. 679-694. – DOI 10.1007/978-981-16-7118-0_58. – EDN IDOIKS.

5. Ye H., Ji Kim. Kim H., Ho. K. Oh. DAPP: automatic detection and analysis of prototype pollution vulnerability in Node.js modules / H. Ye. Kim, Ji. H. Kim, Ho. K. Oh [et al.] // International Journal of Information Security. – 2022. – Vol. 21, No. 1. – P. 1-23. – DOI 10.1007/s10207-020-00537-0. – EDN GWRIQA.

6. Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., Bani-Hani, A. Blockchain smart contracts: Applications, challenges, and future trends / S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, A. Bani-Hani. – Текст: электронный // Springer Science+Business Media, LLC, part of Springer Nature. – 2021. URL: <https://doi.org/10.1007/s12083-021-01127-0> (дата обращения: 01.03.2023)

7. Balcerzak A. P., Nica E., Rogalska E., Poliak M., Klietk T., Sabie O. M. Blockchain Technology and Smart Contracts in Decentralized Governance Systems / A. P. Balcerzak, E. Nica, E. Rogalska, M. Poliak, T. Klietk, O. M. Sabie. – Текст: электронный // Administrative Sciences. – 2022. – Vol. 12. URL: <https://doi.org/10.3390/admsci12030096> (дата обращения: 18.03.2023)

8. Gada, A. G., Mosa, D. T., Abualigah, L., Abohany, A. A. Emerging Trends in Blockchain Technology and Applications: A Review and Outlook / A. G. Gada, D. T. Mosa, L. Abualigah, A. A. Abohany // Journal of King Saud University - Computer and Information Sciences. – 2022. – Vol. 34. – No. 9. – PP. 6719-6742.

9. Lohachab A., Garg S., Kang B. Towards Interconnected Blockchains: A Comprehensive Review of the Role of Interoperability among Disparate Blockchains / A. Lohachab, S. Garg,

B. Kang [et al.] // ACM Computing Surveys. – 2022. – Vol. 54, No. 7. – DOI 10.1145/3460287. – EDN IYKWPU.

10. Giron, A. A. Steganographic analysis of blockchains / A. A. Giron, J. E. Martina, R. Custódio // Sensors. – 2021. – Vol. 21, No. 12. – DOI 10.3390/s21124078. – EDN ARYYSX.

11. MetaMask: [Электронный ресурс]. URL: <https://metamask.io/> (Дата обращения: 09.06.2023)

12. Alchemy: [Электронный ресурс]. URL: <https://www.alchemy.com/> (Дата обращения: 10.06.2023)

References

1. Huang H., Kong W., Zhou S. A Survey of State-of-The-Art on Blockchains / H. Huang, W. Kong, S. Zhou [et al.] // ACM Computing Surveys. – 2021. – Vol. 54, No. 2. – DOI 10.1145/3441692. – EDN KKXOEU.

2. Kasireddy P. The Architecture of a Web 3.0 application. [Electronic resource]. URL: <https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application/> (date of application: 09.06.2023)

3. Hofmann, A., Gwinner, F., Winkelmann, A., Janiesch, C. Security Implications of Consortium Blockchains: The Case of Ethereum Networks / A. Hofmann, F. Gwinner, A. Winkelmann, C. Janiesch. // JIPITEC – Journal of Intellectual Property, Information Technology and E-Commerce Law. – 2021. – Vol. 12. – No. 4. – PP. 347-359

4. Shamili, P. Enhancing the Decentralized Application (Dapp) for E-commerce by Using the Ethereum Blockchain / P. Shamili, B. Muruganatham // Lecture Notes in Networks and Systems. – 2022. – Vol. 341. – P. 679-694. – DOI 10.1007/978-981-16-7118-0_58. – EDN IDOIKS.

5. Ye H., Kim Ji., Kim H., Ho. K. Oh. DAPP: automatic detection and analysis of prototype pollution vulnerability in Node.js modules / H. Ye. Kim, Ji. H. Kim, Ho. K. Oh [et al.] // International Journal of Information Security. – 2022. – Vol. 21, No. 1. – P. 1-23. – DOI 10.1007/s10207-020-00537-0. – EDN GWRIQA.

6. Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., Bani-Hani, A. Blockchain smart contracts: Applications, challenges, and future trends / S. N. Khan, F. Loukil,

C. Ghedira-Guegan, E. Benkhelifa, A. Bani-Hani. – Текст: электронный // Springer Science+Business Media, LLC, part of Springer Nature. – 2021. URL:

<https://doi.org/10.1007/s12083-021-01127-0> (date of application: 01.03.2023)

7. Balcerzak A. P., Nica E., Rogalska E., Poliak M., Klietk T., Sabie O. M. Blockchain Technology and Smart Contracts in Decentralized Governance Systems / A. P. Balcerzak, E. Nica, E. Rogalska, M. Poliak, T. Klietk, O. M. Sabie. – Текст: электронный // Administrative Sciences. – 2022. – Vol. 12. URL: <https://doi.org/10.3390/admsci12030096> (date of application: 18.03.2023)

8. Gada, A. G., Mosa, D. T., Abualigah, L., Abohany, A. A. Emerging Trends in Blockchain Technology and Applications: A Review and Outlook / A. G. Gada, D. T. Mosa, L. Abualigah, A. A. Abohany // Journal of King Saud

University - Computer and Information Sciences. – 2022. – Vol. 34. – No. 9. – PP. 6719-6742.

9. Lohachab A., Garg S., Kang B. Towards Interconnected Blockchains: A Comprehensive Review of the Role of Interoperability among Disparate Blockchains / A. Lohachab, S. Garg, B. Kang [et al.] // ACM Computing Surveys. – 2022. – Vol. 54, No. 7. – DOI 10.1145/3460287. – EDN IYKWPU.

10. Giron, A. A. Steganographic analysis of blockchains / A. A. Giron, J. E. Martina, R. Custódio // Sensors. – 2021. – Vol. 21, No. 12. – DOI 10.3390/s21124078. – EDN ARYYSX.

11. MetaMask: [Electronic resource]. URL: <https://metamask.io/> (date of application: 09.06.2023)

12. Alchemy: [Electronic resource]. URL: <https://www.alchemy.com/> (date of application: 10.06.2023).

СВЕДЕНИЯ ОБ АВТОРАХ/ABOUT THE AUTHORS

Окладникова Ольга Дмитриевна, магистрант факультета программной инженерии и компьютерной техники, Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО», 197101, Санкт-Петербург, Кронверкский пр., д. 49, лит. А, hathaway@gmail.com

Окладникова Светлана Владимировна, к.т.н., доцент, доцент кафедры систем автоматического проектирования и моделирования, ГБОУ АО ВО «Астраханский государственный архитектурно-строительный университет», 1414056, г. Астрахань, ул. Татищева, 18, okladnikova.s.v@yandex.ru

Olga D. Okladnikova, Master's student of the Faculty of Software Engineering and Computer Technology, ITMO University, 49 lit. A Kronverksky Av., St. Petersburg, 197101

Svetlana V. Okladnikova, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Automatic Design and Modeling Systems, State Budgetary Educational Institution of the Astrakhan Region of Higher Education "Astrakhan State University of Architecture and Civil Engineering", 18 Tatishcheva str., Astrakhan, 141405