МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Исследование технологий Web3 и их применение в разработке децентрализованных приложений

МАГИСТЕРСКАЯ РАБОТА

студентки 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Мраморнова Андрея Константиновича

Научный руководитель
к.ф.-м.н., доцент С.П. Шевырев

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент Ю.А. Блинков

Саратов 2023

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ

Биткойн был выпущен в 2009 году, и с тех пор технологии блокчейн продолжают демонстрировать множество преимуществ и применимость в различных областях промышленности (банковское дело, кибербезопасность, цепочки поставок, страхование, здравоохранение, развлечения и т.д.). Самым большим достижением в технологиях блокчейн являются смарт-контракты, которые представляют собой самоисполняемые контракты, содержащие условия соглашения между двумя частями, без необходимости в посреднике. Эти контракты позволяют разработчикам создавать автономные, самоэффективные и децентрализованные приложения (dapp).

В последнее время технологии блокчейн стали очень популярны благодаря историям успеха торговли криптовалютой и NFT, когда отдельные пользователи получали значительную прибыль. Это привлекло внимание крупных компаний и инвесторов, которые стали вкладывать большие деньги в блокчейн-проекты, в результате чего началось интенсивное развитие технологий, инструментов, фреймворков и платформ для разработки Web3-решений. Несмотря на то, что технологии кажутся достаточно развитыми, технология все еще относительно новая, и уникальный подход к разработке Web3-решений еще не сформирован.

Целью данной работы является изучение принципов разработки решений Web3, с акцентом на технологии, платформы, фреймворки и инструменты для разработки приложений. Также сравниваем подходы к разработке между различными исследованиями и отвечаем на вопросы, относящиеся к Web3 и разработке решений Web3. Демонстрируем собранных знаний происходит на примере пробного приложения.

В рамках теоретической базы рассматриваются блокчейн и технологии, связанные с блокчейном, которые лежат в основе данной работы. Затем идет развитие Web, где описывается каждая версия Web и указывается на различия между ними (Web1, Web2, Web3). В последней теоретической части рассматриваются самые распространенные решения Web3, подкрепленные примерами, а также технологии для разработки решений Web3 и выяснение того, какие технологии представляют собой полный стек Web3.

В последней части представлена реализация пробного Web3-приложения, основанного на идентифицированном стеке технологий из предыдущих глав. Описывается, какие технологии были использованы для реализации и почему были выбраны именно они. Затем представляется концепция реализации Web2-решения для того же экспериментального приложения и обсуждаются различия между ними.

1 Блокчейн

Популяризация блокчейна началась после 2008 года, когда неизвестный человек или группа людей под именем Сатоши Накамото изобрели криптовалюту Bitcoin. В 2009 году ее реализация была опубликована в открытом доступе как программное обеспечение с открытым исходным кодом. Биткойн - это децентрализованная криптовалюта, которая работает в одноранговой сети, где между узлами сети существует консенсус [4]. Консенсус означает правила, по которым работает сеть блокчейн, и подтверждает достоверность информации, записанной в блоках. Сеть состоит из узлов, и их основными задачами являются:

- Определить, является ли блок транзакций легитимным, и принять или отклонить его;
- Сохранение и хранения блоков транзакций;
- Передача истории транзакций другим узлам, которым может потребоваться синхронизация с блокчейном;

Транзакции в сети проверяются узлами с помощью криптографии и записываются в публичный реестр, называемый блокчейн. Это делает транзакции между двумя участниками возможными без посредника или центрального органа [5]. Блокчейн представляет собой последовательность блоков, в которых хранится полный список записей о транзакциях, подобно обычной финансовой книге. Каждый блок указывает на непосредственно предыдущий блок через ссылку, которая по сути является хэш-значением предыдущего блока, называемого родительским блоком. Первый блок блокчейна называется «генезис-блок» (Genesis block). Этот блок был создан самим создателем биткоина Сатоши Накамото 3 января 2009 года. Генезис-блок содержит информацию о первой транзакции биткоина, которая была совершена Сатоши Накамото на адрес, который принадлежал ему самому. В отличие от других блоков в блокчейне, генезис-блок не имеет предшествующих блоков и является началом цепочки блоков, которая продолжается до настоящего времени. Пример блокчейна показан в соответсвии с рисунком 1.1.

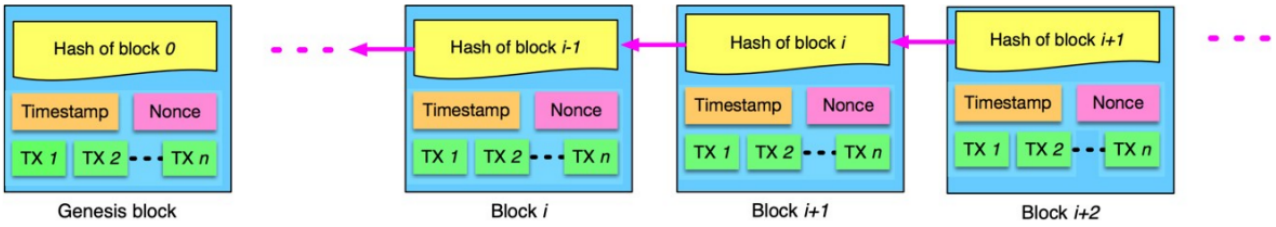

Рисунок 1.1 — Структура блоков в блокчейне

Блок в блокчейне состоит из заголовка блока и тела блока. Заголовок блока содержит:

- Версия блока: указывает, какой набор правил проверки блока следует соблюдать;
- Хэш родительского блока: 256-битное хэш-значение, указывающее на предыдущий блок;
- Корневой хэш дерева Меркла: хэш-значение всех транзакций в блоке;
- Timestamp: текущая метка времени в секундах с 1 января 1970 года;
- nBits: сжатое представление целевого значения, ниже которого должен быть хэш блока, чтобы быть действительным;
- Nonce: случайное целое число, которое представляет собой 4-байтовое поле и обычно начинается с 0 и увеличивается при каждом вычислении хэша. Nonce - это число, которое может быть использовано только один раз;

Тело блока состоит из счетчика транзакций и транзакций. Максимальное количество транзакций, которое может содержать блок, зависит от размера блока и размера каждой транзакции. Для подтверждения подлинности транзакции в ненадежной среде блокчейн использует цифровую подпись, основанную на механизме асимметричной криптографии. В цифровой подписи каждый пользователь имеет открытый и закрытый ключ. Цифровая подпись состоит из двух фаз, фазы подписания и фазы проверки, в соответствии с рисунком 1.2. На фазе подписания пользователь генерирует хэш-значение, полученное в результате транзакции, шифрует его с помощью закрытого ключа и отправляет зашифрованный хэш с исходными данными другому пользователю. Этот пользователь проверяет полученную транзакцию путем сравнения расшифрованного хэша (с использованием открытого ключа отправляющего пользо-

вателя) с хэш-значением, полученным из полученных данных с помощью той же хэш-функции, которую использовал отправляющий пользователь.

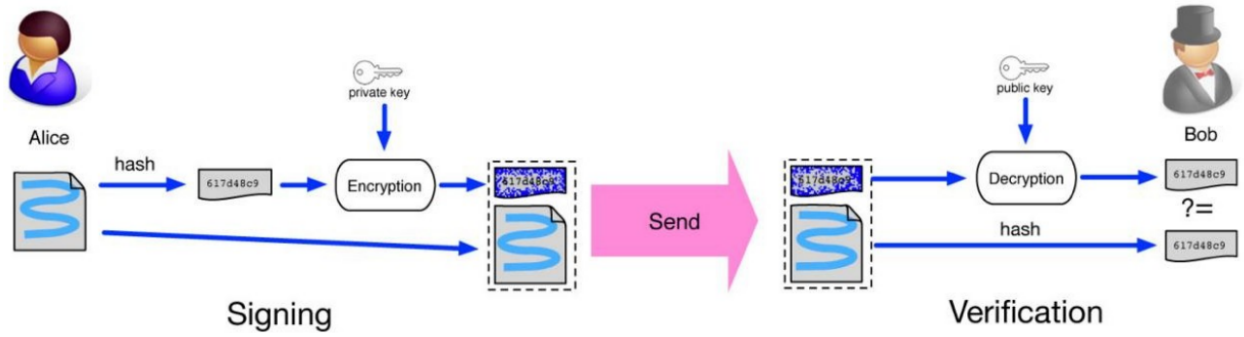

Рисунок 1.2 — Визуальный пример цифровой подписи

Чтобы понять блокчейн, необходимо разобраться в его ключевых характеристиках:

- Децентрализация: Транзакция в сети блокчейн проводится между двумя участниками без вмешательства центрального органа. Более того, любой может присоединиться к сети, а определение того, что записывается в блокчейн-бухгалтерию, также не является детерминированным. Это позволяет значительно снизить затраты на сервер и уменьшить узкие места в производительности, которые могут возникнуть на центральном сервере;
- Постоянство: практически невозможно подделать транзакцию или блок, поскольку каждая транзакция в сети должна быть подтверждена и сохранена в блоках, распределенных по всей сети. Кроме того, каждый блок должен быть подтвержден другими узлами, поэтому любая подделка может быть легко обнаружена;
- Псевдонимность: В типичном приложении Web2 пользователю предлагается создать учетную запись и предоставить адрес электронной почты и некоторую личную информацию. В блокчейне пользователи взаимодействуют с сетью блокчейна с помощью сгенерированного адреса, и каждый пользователь может даже сгенерировать несколько адресов, чтобы избежать раскрытия личности. Это позволяет обеспечить опреде-

ленную степень конфиденциальности, хотя идеальное сохранение конфиденциальности не может быть гарантировано;
- Контролируемость: каждая транзакция на блокчейне подтверждается и записывается с меткой времени, поэтому пользователи могут проверить и отследить предыдущие записи, обратившись к любому узлу сети и просмотрев так называемую реестр транзакций;

1.1 Децентрализованное приложение

Децентрализованное приложение или dapp - это любое приложение, которое либо само размещено в децентрализованном пространстве (например, IPFS), либо его логика зависит от децентрализованного пространства (например, блокчейн или смарт-контракты). Dapps полагаются на blockchain для обработки данных через распределенных сетей и выполнения транзакций с помощью смарт-контрактов. Подавляющее большинство dapps построено на блокчейне Ethereum, поскольку именно Ethereum популяризировал использование смарт-контрактов для приложений на блокчейне. Идеальный dapp должен быть полностью размещен в сети P2P и не нуждается в обслуживании и управлении со стороны первоначальных разработчиков. Децентрализованные приложения характеризуются следующими свойствами:
- Децентрализованные: они работают на децентрализованной платформе, где ни один человек или группа не имеют контроля;
- Детерминированные: они выполняют одну и ту же функцию независимо от среды, в которой они выполняются;
- Полные по Тьюрингу: Они могут выполнить любое действие при наличии необходимых ресурсов;
- Изолированные: они выполняются в виртуальной среде, так что если в смарт-контракте есть ошибка, она не помешает нормальному функционированию сети блокчейн;

1.2 Смарт контракт

Термин «Смарт контракт» или «Умный контракт» был введен Ником Сабо в середине 1990-х годов, который предложил перевести положения контракта в код и встроить их в программное или аппаратное обеспечение, сделав самоисполнимыми, чтобы минимизировать затраты на заключение контракта между сторонами и избежать случайных исключений или злонамеренных действий во время выполнения контракта. Умный контракт в разных дисциплинах имеет разное значение, в нашем случае он обозначает низкоуровневый код-скрипт, работающий на блокчейне. Умные контракты хранятся на блокчейне и могут автоматически выполняться при выполнении определенных предварительных условий. В случае с Ethereum они реализованы на языке программирования Solidity. Основными характеристиками смарт-контрактов являются автономность, прозрачность, самодостаточность и децентрализация. Автономность означает, что после их развертывания они не требуют дополнительного мониторинга или специального исполнения. Они самодостаточны благодаря способности привлекать средства, предоставляя услуги и расходуя их по мере необходимости. Они децентрализованы, так как распределены и самостоятельно выполняются между сетевыми узлами.

1.3 Ethereum

Ethereum, запущенная в 2015 году, стала первой платформой, поддерживающей смарт-контракты (смарт-контракты рассмотрим позже). Она была построена на фундаменте Bitcoin, но с существенными отличиями. Если Bitcoin - это только платежная сеть, то Ethereum программируема, поэтому в ее сети можно создавать и внедрять смарт-контракты. В ней используется полный язык Тьюринга, позволяющий поддерживать все типы вычислений, включая циклы. Он обеспечивает абстрактный уровень, позволяющий любому создавать собственные правила владения, форматы транзакций и функции перехода состояний. Это достигается за счет включения смарт-контрактов - набора криптографических правил, которые выполняют-

ся только при выполнении определенных условий. Каждое действие в сети Ethereum требует определенного количества вычислительной мощности. Эта плата вносится в виде эфира (ETH), который является родной криптовалютой Ethereum. Первоначально блокчейн Ethereum использовал механизм консенсуса «proof-of-work», однако 15 сентября 2022 года в процессе обновления, названного «The Merge», он перешел на консенсус «proof-of-stake» и снизил потребление энергии примерно на 99.95% В ходе этого процесса первоначальный уровень исполнения Ethereum был объединен с новым уровнем консенсуса, основанным на доказательстве доли, - цепочкой маяков. Это устранило необходимость в энергоемком майнинге и позволило обеспечить безопасность сети с помощью ETH. Ethereum не полностью децентрализован, поскольку подавляющее количество узлов Ethereum размещено на виртуальных машинах, таких как Amazon Web Services.

1.3.1 Виртуальная машина Ethereum

Виртуальная машина Ethereum (EVM) - это среда выполнения транзакций в Ethereum (развертывание и выполнение смарт-контрактов). Она используется для прогнозирования общего состояния Ethereum для каждого блока на блокчейне по мере его добавления в цепь. Каждый узел Ethereum работает на EVM для поддержания консенсуса в блокчейне. EVM выполняет логику смарт-контракта в изолированной среде, что означает, что код внутри EVM не имеет доступа к сети, файловой системе или другим процессам. EVM использует собственный ассемблерный, стековый и полный язык Тьюринга и состоит примерно из 150 уникальных опкодов, которые представляют собой инструкции машинного уровня, которые может выполнять компьютер. Поскольку опкоды не очень удобны для разработчиков, для написания кода используется язык программирования Solidity, который затем компилируется в байткод EVM перед развертыванием на блокчейне. Помимо Solidity, существуют и другие языки программирования для реализации смарт-контрактов, включая Rust, JavaScript, Vyper, Yul и DAML.

1.4 Газ или стоимость сделки

Узлы, на которых работает EVM, не могут предвидеть количество ресурсов, необходимых для подтверждения транзакции, что позволяет проводить атаки типа «отказ в обслуживании». Для борьбы с этим используется механизм ценообразования. Каждый вычислительный шаг в EVM оценивается в единицах газа. Для каждой транзакции отправитель должен указать максимальное количество газа, которое, как ожидается, будет потреблено при вычислениях, и цену, которую пользователь хочет заплатить за единицу газа. Цена единицы газа в эфире определяется рынком. Плата за транзакцию равна лимиту газа, умноженному на цену газа.

1.5 Криптокошелек

Криптокошельки - это программное приложение, используемое для просмотра баланса криптовалюты и совершения транзакций на блокчейне. Они хранят публичные и приватные ключи пользователей, предоставляя простой интерфейс для управления балансом криптовалюты. Публичный ключ похож на номер банковского счета и может быть предоставлен публично, а приватный ключ похож на пароль банковского счета и должен храниться в секрете. Криптовалюта физически не хранится на кошельке, вместо этого кошельки считывают публичный реестр и показывают пользователям остатки на их адресах, а также хранят приватные ключи, позволяющие совершать транзакции. Кошельки хранят один или несколько уникальных криптовалютных публичных адресов (т.е. пару ключей). Публичный адрес - это шестнадцатеричная строка с комбинацией цифр и букв в нижнем и верхнем регистре. Для получения криптовалюты его необходимо публично сообщить. Приватный ключ - это также шестнадцатеричная строка. Поскольку его трудно запомнить, он хранится в программном обеспечении кошелька. Вместо этого, чтобы получить доступ к кошельку, пользователь должен знать пинкод, связанный с каждым закрытым ключом. Для дополнительной безопасности пользователи также могут использовать кошельки с несколькими подписями, которые

требуют двух или более подписей закрытых ключей для авторизации транзакций. Существуют различные критерии для различения криптокошельков, сосредоточимся на том, который основан на устройстве, используемом для хранения ключей:

- Десктоп кошелек: программа, которую можно загрузить и установить на компьютер. Они обеспечивают один из максимальных уровней безопасности;
- Онлайн-кошелек: программное обеспечение, которое работает в облаке и может использоваться из любого места. К ним легко получить доступ, но закрытые и открытые ключи хранятся у третьей стороны. Однако ключи могут быть зашифрованы на сервере с использованием какого-либо пароля пользователя. Тем не менее, они хранятся у другой организации;
- Мобильный кошелек: программное обеспечение, которое запускается как приложение на мобильном телефоне. Ключи хранятся в мобильном кошельке;
- Аппаратный кошелек: физическое хранилище ключей на устройстве, которое работает аналогично USB-устройствам. Они совершают транзакции онлайн, но публичные и приватные ключи хранятся офлайн. Это наиболее безопасная версиякошелька;

Все упомянутые кошельки предлагают некоторые или все функции, такие как авторизация пользователя, генерация ключей, управление ключами, анонимность, поддержка мультивалютности, курсы конвертации монет или токенов, криптообмен, сканирование QR-кода для совершения криптовалютных операций, push-уведомления, а также возможность резервного копирования и восстановления ключей.

1.6 Взаимозаменяемый токен

Взаимозаменяемость — это способность актива быть взаимозаменяемым с другим идентичным активом. А токен является взаимозаменяемым, если его можно заменить другим идентичным токеном. Два разных взаимозаменяемых токена служат одной цели, даже если они разделены или обменены

на другие взаимозаменяемые токены того же типа. Взаимозаменяемый токен может быть дроблен, разделен, расщеплен или обменен, и все это без изменения его стоимости. Криптовалюты – это наиболее очевидный пример взаимозаменяемых токенов, поскольку каждая единица конкретной криптовалюты неотличима от другой единицы. Правила выпуска и реализации новых токенов определяются стандартами токенов. Стандарты обычно включают требования, определяющие общий лимит предложения токена, процесс майнинга токена, процесс сжигания токена и процесс проведения транзакций с токеном. Стандарты призваны помочь избежать мошенничества, технической несовместимости между различными токенами и выпуска токенов, не соответствующих принципам блокчейна. Наиболее популярными стандартами токенов являются ERC20, BEP2 и BEP20. ERC20 или Ethereum Request for Comment 20 - это внедренный стандарт для взаимозаменяемых токенов на блокчейне Ethereum. Он направляет создание новых токенов на блокчейне Ethereum, чтобы они были взаимозаменяемы с другими токенами, используемыми в смарт-контрактах. ERC20 содержит несколько функций и событий, которые должен реализовать токен. Минимальное количество функций для того, чтобы токен соответствовал ERC20 - это:

- TotalSupply: общее количество токенов, которые когда-либо будут выпущены;
- BalanceOf: Баланс счета владельца токена;
- Transfer: Автоматически выполняет перевод указанного количества токенов на указанный адрес для транзакций с использованием токена;
- TransferFrom: Автоматически выполняет передачу указанного количества токенов с указанного адреса с использованием токена;
- Approve: Позволяет плательщику снять установленное количество токенов с указанного счета, вплоть до определенной суммы;
- Allowance: возвращает владельцу установленное количество токенов от плательщика;
- Transfer: событие, возникающее при успешной передаче (событие);
- Approval: журнал утвержденного события (событие);

1.7 NFT

Невзаимозаменяемый токен (NFT) определяется как криптографически уникальный, неделимый, незаменимый и проверяемый токен, который представляет данный актив, будь то цифровой или физический, на блокчейне. Уникальность вводится стандартом ERC721, где добавляется переменная uint256 под названием tokenId, и каждая пара адрес контракта и uint256 tokendId должна быть глобально уникальной. Владелец NFT может легко доказать существование и владение активом, кроме того, можно отследить всех предыдущих владельцев. NFT создаются и управляются смарт-контрактами. Люди часто путаются в том, что NFT — это актив, например, что цифровое произведение искусства — это NFT, что неверно. NFT - это просто структура данных на блокчейне, которая хранит информацию о праве собственности на это произведение искусства, с некоторой дополнительной информацией, включая адрес местонахождения, где хранится актив. Поскольку цифровые произведения искусства обычно представляют собой большие файлы, слишком большие, чтобы хранить их на блокчейне, для хранения используется альтернативный способ. Одним из способов хранения активов может быть использование межпланетной файловой системы (IPFS), которая представляет собой распределенную одноранговую файлообменную сеть. Огромный всплеск популярности токенов произошел в 2021 году, когда определенные лица получили огромную прибыль, торгуя NFT. NFT в основном ассоциируются с цифровым искусством, однако они полезны в любом случае, когда требуется владение каким-либо уникальным активом. Большой потенциал имеется в индустрии видеоигр, где каждый предмет, полученный игроком в игре, может быть представлен в виде NFT. NFT создается путем майнинга актива и последующей загрузки его в блокчейн. Майнинг - это процесс создания и производства NFT, который часто стоит плату, называемую газом, цена которого зависит от платформы и блокчейна, где он развернут. С другой стороны, NFT имеют возможность добавлять плату за роялти, которая выплачивает создателю процент от транзакции каждый раз, когда NFT продается, и это является дополнительным стимулом для художников внести свой вклад в сообщество NFT.

2 Разработка Web

Так что такое Web3 и чем он отличается от Web2 и Web1? Интернет, вероятно, является одной из самых важных технологических революций в истории человечества, где Web, как одно из представлений Интернета, все еще находится в стадии развития. Люди часто ошибочно используют термины Интернет и Web как синонимы, хотя они имеют разные значения. Интернет - это глобальная сеть взаимосвязанных серверов, компьютеров и других устройств, где каждое устройство может подключаться к другому, при условии, что оба устройства подключены к Интернету с действительным IP-адресом. С другой стороны, Web - это только один из методов распространения информации через Интернет, другие включают электронную почту, протокол передачи файлов (FTP) и службу мгновенных сообщений. Веб состоит из огромного количества цифровых документов, доступ к которым идет с помощью веб-браузеров. За свою относительно короткую историю Web уже пережил несколько крупных этапов, крупнейшими из которых являются Web1 и Web2. В настоящее время много говорят о Web3, который должен стать следующей революцией в Web, и который также является центральной темой данной работы. Чтобы лучше понять, что представляет собой Web3, сначала необходимо понять, как Web развивался на протяжении всей истории.

2.1 Статический web или web 1.0

Web1 (World Wide Web), или просто Web, представляет собой первую эволюцию Web, известную как «Read-only Web», где было меньше создателей веб-сайтов и больше потребителей, получающих доступ к этим страницам. Тим Бернерс-Ли создал Web в 1989 году, работая в CERN. Он разработал первый веб-сервер, первый веб-браузер и протокол форматирования документов Hypertext Markup Language, известный как HTML. Началом создания Web считается 1991 год, когда Бернерс-Ли выпустил HTML в открытый доступ. Web1 в основном использовался для представления статического контента без возможности взаимодействия или с минимальными возможностями. Веб-сайты использовались для отображения информации, и пользователь мог лег-

ко получить доступ к ней, посетив сайт издателя. Во время Web1 необходимо было учитывать производительность веб-сервера и пропускную способность, поскольку множество страниц и огромный контент замедляли работу всего сайта. Тем не менее Web1 включал в себя некоторые возможности Web2, но реализованы они были иначе. Например, раздел комментариев в Web1 присутствовала в виде страницы гостевой книги, где посетители могли оставлять свои комментарии.

2.2 Семантический web или web 2.0

Переход от Web 1.0 к 2.0 происходил с течением времени по мере модернизации серверов, увеличения средней скорости соединения и освоения разработчиками новых навыков и методов. Термин Web2 был впервые введен Дарси Динуччи в статье «Фрагментированное будущее» в 1999 году. В своей статье она описала, как в будущем базовая структура информации и механизм гиперссылок будут использоваться на различных устройствах и платформах. Однако ее представление Web2 не имеет прямого отношения к современному использованию термина. Термин Web2 стал популярным в 2004 году, когда O'Reilly Media и MediaLive провели первую конференцию Web2. На конференции Джон Бателл и Тим О'Рейли дали определение термину «Web как платформа», согласно которому программные приложения создаются на базе Web, а не на базе компьютера. Они предложили использовать активность пользователей на сайте для создания ценности. Если Web1 назывался Web только для чтения, то Web2 можно определить как Web для чтения и записи или партисипативный социальный Web, где пользователям предлагается взаимодействовать с динамическим контентом и вносить в него свой вклад. Важными особенностями Web2 являются:

- Пользователи как первоклассная сущность в системе, с соответствующими страницами профиля;
- Возможность формирования связей между пользователями, посредством ссылок на других пользователей, отмеченных как друзья, или членства в группах различного рода, или подписки на RSS-каналы обновлений других пользователей;

- Возможность размещать контент в различных формах: фото, видео, блоги, комментарии и рейтинги;
- другие более технические возможности, такие как публичный API, позволяющий сторонним разработчикам улучшать систему или общаться с другими пользователями с помощью внутренней электронной почты или систем мгновенного обмена сообщениями;

В соответствии с рисунком 2.1, в июне 2022 года 60% всего веб-трафика будет приходиться на мобильный интернет-трафик.

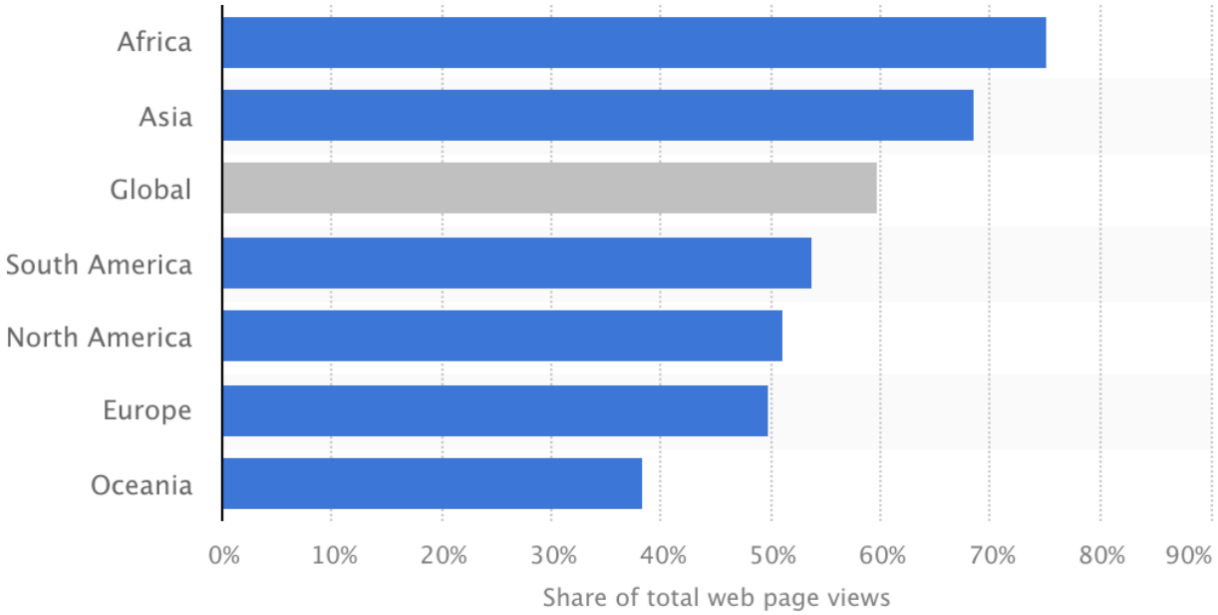

Рисунок 2.1 — Мобильный интернет-трафик в процентах от общего веб-трафика в июне 2022 года, по регионам

Интернет, каким его знают сегодня, действительно является социальным и совместным, но за это приходится платить. Большинство пользователей игнорируют или даже не знают об этом. Пользовательские данные централизуются и используются крупными корпорациями, затем они используются без согласия пользователя в маркетинговых целях, и пользователи не имеют над ними никакого контроля. Сбор пользовательских данных, их обработка и использование для получения прибыли - это то, за что выступает Web2.

2.3 Децентрализованный web или web 3.0

Основная идея Web3 заключается в том, что крупные корпорации (CA) не должны иметь контроль над данными пользователей, но каждый пользователь должен иметь возможность контролировать свои собственные данные. Web3 представляет собой идею новой итерации всемирной паутины, которая включает в себя такие концепции, как децентрализация, технологии блокчейн и экономика на основе токенов. Термин Web3 был придуман основателем Polkadot и соучредителем Ethereum Гэвином Вудом в 2014 году, обозначая децентрализованную онлайн-экосистему на основе блокчейна. Идея Web3 приобрела большую популярность в 2021 году, во многом благодаря интересу со стороны криптовалюты, энтузиастов и инвестиций от высококлассных технологов и компаний. Основными принципами Web3 являются:

- Децентрализованность. Вместо того чтобы данные контролировались и принадлежали централизованным организациям, право собственности на данные распределяется между их создателями и пользователями;
- Не требует доверия. Работает с использованием стимулов и экономических механизмов, а не полагается на доверенных третьих лиц;
- Не требует разрешения. Позволяет участвовать без необходимости получения разрешения от административного органа;
- Есть встроенные платежи. Использует криптовалюту для трат и отправки денег онлайн вместо того, чтобы полагаться на унаследованную инфраструктуру банков и платежных процессоров;

Несмотря на то, что Web3 еще не полностью разработан, многие элементы Web3 уже доступны и используются ежедневно, такие как NFT, блокчейн, распределенные реестры и облако AR.

2.4 Сравнение Web1, Web2 и Web3

Для лучшего понимания различий между Web1, Web2 и Web3 можно посмотреть таблицу различий, в соответствии с рисунком 2.2.

	Web1	Web2	Web3
Назначение	Только для чтения	Чтение-запись	Чтение, запись, владение
Тип содержимого	Статический веб-контент	Динамический веб содержание	Семантическое содержание
Содержание	Главные страницы	Блоги, вики, социальные сети	Прямые трансляции, волны, цифровые активы
Владелец данных	Централизованная организация	Централизованная организация	Пользователь
Цель	Информационный обмен	Взаимодействие	Погружение
Аутентификация	Нет	Создание новых учетных записей или SSO	Связь с криптокошельком
Инфраструктура	Централизованная инфраструктура	Инфраструктура облачных вычислений, которая в основном централизованная	Децентрализованная инфраструктура
Доступность	Удобство и доступность	Удобство и доступность	Отсутствие интеграции с современными браузеры
Реклама	Баннерная реклама	Интерактивная и поведенческая реклама	Шеринг данных за вознагражде-ние
Технологии	HTML, HTTP, URL	AJAX, JavaScript, CSS3, HTML5	Блокчейн, искусственный интеллект и децентрализован-ные протоколы, AR, VR
Доход / Прибыль	Просмотры страниц	Прибыль за клик	Создание ценности

Рисунок 2.2 — Сравнение Web1, Web2 и Web3

Первое различие между версиями Web заключалось в их назначении. Так как Web1 был ориентирован на обмен информацией, тип контента состоял из статичных веб-страниц и предназначался только для чтения без возможности взаимодействия. Web2 был основан на динамичном веб-контенте, где целью было взаимодействие пользователей с контентом, поэтому эту версию называют Participative Social Web. Веб-сайты часто просят пользователей

взаимодействовать с контентом в виде лайков и комментариев. Хотя Web2 обещал интерактивность и вовлеченность, он всегда соответствовал правилам и стратегиям монетизации существующих платформ. Web3 предлагает гораздо более захватывающий и вовлекающий опыт, где пользователи контролируют свои данные и контент. Если в Web1 и Web2 данные принадлежат централизованным организациям, то в Web3 данные принадлежат пользователям. В Web1 для размещения веб-сайтов использовалась централизованная инфраструктура, т.е. веб-серверы, которые позже перешли к облачным вычислениям в Web2. Web3 стремится быть децентрализованным и опирается на одноранговую сеть, построенную на сообществе пользователей. Веб-сайты или приложения будут размещаться на собственных устройствах этой группы, подключенных к Интернету, а не на группе мощных серверов. С точки зрения доступности, Web1 и Web2 были удобны и легко доступны с помощью веб-браузеров, при этом веб-браузеры иногда не поддерживали новейшие функции. В Web1 не было необходимости в регистрации пользователей, в то время как в Web2 почти каждый сайт пытается зарегистрировать новых пользователей или, по крайней мере, предложить Единый вход (SSO), который позволяет пользователям использовать один набор учетных данных пользователя, подтверждающих личность, для аутентификации на нескольких веб-сайтах на основе более крупной и устоявшейся платформы (например, Google, Facebook и т. д.). В Web3 отсутствует интеграция с современными веб-браузерами, поскольку аутентификация пользователя осуществляется путем подключения кошелька пользователя, необходимо использовать расширения браузера или браузер, который изначально поддерживает кошельки. В Web1 реклама осуществлялась с помощью простых баннеров на сайте, поэтому просмотры страниц имели решающее значение для получения дохода. Web2 усовершенствовал рекламу, собирая данные о пользователях и обрабатывая их с помощью рекомендательных систем для создания рекламы, ориентированной на пользователя. Это создало систему прибыли за клик, когда владелец сайта зарабатывал деньги на кликах по объявлениям, размещенным на его сайте. В Web3 новый тип рекламы, когда люди соглашаются делиться своими данными с компаниями и получают за это вознаграждение. Технологии, использованные для реализации сайтов Web1, представля-

ли собой HTML с базовым CSS без учета отзывчивости. Web2 был нацелен на создание сайтов отзывчивых, чтобы сайт подстраивался под устройства, т.е. имел разные размеры и адаптированные компоненты при отображении сайта в браузере компьютера или мобильного устройства. HTML5 и CSS3 использовались в сочетании с AJAX и JavaScript для обеспечения загрузки на странице, которая загружает динамический контент без необходимости обновлять страницу. Web3 основан на блокчейне, технологиях искусственного интеллекта и децентрализованных протоколах, и еще пока не ясно, как будет выглядеть взаимодействие с пользователями. По одним из прогнозов, пользователи будут подключаться к Web3 с помощью технологий AR или VR.

2.5 Технологии для разработки Web3

Распределенная реестр работает по заранее определенным правилам, которые согласовываются всеми участвующими узлами сети. Эти правила называются протоколом. Наиболее известным протоколом блокчейна для создания децентрализованных приложений является Ethereum, который был первым протоколом, включающим смарт-контракты. Ethereum - это публичный блокчейн и сеть без разрешений, что означает, что он может быть доступен любому для операций чтения и записи. Другие публичные сети - Polygon и Solana. С другой стороны, Hyperledger Fabric - это частная сеть с разрешением, что означает, что в ней могут участвовать только привилегированные организации и узлы. Для получения доступа необходимо получить разрешение от доверенного поставщика услуг членства. Кроме того, в Hyperledger Fabric нет необходимости в газе, поскольку каждый участник знает всех других участников сети, а вредоносные пользователи могут быть легко обнаружены и удалены из сети. В процессе разработки разработчики используют среды разработки для тестирования контрактов в локальных или публичных тестовых сетях. Популярными инструментами для создания среды разработки Ethereum являются Hardhat, Truffle, Geth и Remix IDE, причем последняя работает в браузере и одновременно является редактором кода. Среда разработки Hyperledger Fabric создается с помощью Docker. Пользовательский интерфейс или фронтенд служит мостом между пользователями и блокчей-

ном. Технологии для разработки фронтенда решений Web3 те же, что и для решений Web2. Для разработки браузерных приложений наиболее популярны React.js, Vue.js и Django. Примером для мобильных приложений является платформа Android. Взаимодействие фронтенда с блокчейн-сетью осуществляется с помощью различных библиотек. Наиболее распространенной из них является Web3.js - самая популярная библиотека на базе JavaScript для взаимодействия с сетью Ethereum. Другие библиотеки - ethers.js, web3.py, Infura API, Hyperledger Fabric Node SDK и др. Иногда нужно изучить транзакции в блокчейне. Для этого можно использовать блокчейн-проводники, такие как Etherscan или Polygonscan. Эти инструменты позволяют любому человеку просматривать блоки, адреса кошельков, хэшрейт сети, данные о транзакциях и другую ключевую информацию в блокчейне. Как уже говорилось, Web3 стремится к децентрализации, и для этого необходимо использовать децентрализованные хранилища. Межпланетная файловая система (IPFS) - это распределенная и децентрализованная сеть хранения для хранения и доступа к файлам, веб-сайтам, данным и приложениям, использующая сеть P2P для соединения множества узлов по всему миру. Доступ к содержимому, хранящемуся на IPFS, можно получить с любого шлюза IPFS. Swarm - еще одна подобная распределенная и децентрализованная сеть. Для того чтобы пользователь мог взаимодействовать с блокчейном, он должен пройти аутентификацию. В Web3 аутентификация пользователя осуществляется путем подключения кошелька блокчейна. Самый простой и распространенный способ - это использование приложения MetaMask blockchain wallet, которое может быть установлено как расширение для браузера и мобильное приложение. Оно берет на себя заботу о ключах кошелька, безопасном входе в систему, кошельке для токенов и обмене токенов. Кроме того, можно использовать специализированный браузер, например Brave. Brave - это браузер, который изначально поддерживает блокчейн-кошельки.

2.5.1 Технологии, платформы, фреймворки и инструменты для разработки решений Web3

Во время подготовки материала, были рассмотрены некоторые исследования, связанные с выбором технологий, платформ, фреймворков и технологий для реализации решений на Web3. А также, связи и зависимости Web3 от блокчейна и смарт-контрактов, необходимость несгораемых токенов и описание полного стека для Web3. По данным исследований, наиболее популярной технологией и платформой блокчейн был Ethereum, так как большинство исследований реализовали свое решение на блокчейне Ethereum. Было найдено три исследования, которые построили свое решение на блокчейне Hyperledger Fabric, и только одно исследование использовало блокчейн EOS. Языком программирования, использованным для разработки смарт-контрактов для блокчейна Ethereum, во всех исследованиях был Solidity, для Hyperledger Fabric было обнаружено использование Golang, который предпочитает сообщество, и одно использование JavaScript. Смарт-контракты блокчейна EOS были реализованы на C++. Вероятно, что причина использования блокчейна Ethereum во многих исследованиях кроется в том, что его API-реализации доступны для популярных фреймворков разработки фронтендов. В исследованиях были обнаружены библиотеки API Ethereum, написанные на различных языках программирования, например, web3.js на JavaScript, web3.py на Python и web3j на Java. Кроме того, на данный момент Ethereum занимает второе место по рыночной капитализации среди криптовалют, а на GitHub существует почти 56 000 репозиториев. Всего было найдено три вспомогательных инструмента и библиотеки, а именно Infura, Voodfy и OpenZeppelin. Infura в основном использовалась благодаря своему API для легкого соединения с IPFS. Однако она использовалась для размещения кластера узлов Ethereum, поскольку размещение узла на мобильном устройстве требует больших затрат энергии и демотивирует пользовательские агенты. Voodfy - еще одна платформа, которая использовалась для хранения потокового видео. На сегодняшний день, похоже, эта платформа больше не поддерживается, так как с лета 2021 года не было никаких обновлений. Наконец, OpenZeppelin использовалась для реализации NFT. Платформа предостав-

ляет предопределенные смарт-контракты для снижения сложности кодирования. Найденные фреймворки в основном состояли из frontend-фреймворков, и казалось, что любой frontend-фреймворк может быть использован для разработки Web3-приложений. Самые популярные фреймворки были основаны на JavaScript, лидировал Vue.js, за ним следовали Angular, Next.js и React Native. Среди других используемых фреймворков для фронтенда были ASP.net, Android и Django. Наконец, важным фреймворком, который использовался во многих исследованиях, является Truffle, который используется для компиляции и миграции смарт-контрактов. Инструменты для разработки Web3, наблюдаемые в исследованиях, использовались для создания клиента блокчейна, то есть узла блокчейна. Наиболее используемым был Ganache, который применяется для создания клиентского узла Ethereum. Другими инструментами на базе Ethereum были Geth, Parity Ethereum и Remix IDE. Вероятно, что большинство исследователей выбрали Ganache из-за его удобного и простого в использовании графического интерфейса, в то время как Geth является инструментом командной строки и может быть более сложным в использовании. В исследованиях Hyperledger Fabric для создания сети блокчейна использовался инструмент Docker. Хотелось бы отметить еще одну тему, которая не была упомянута в вопросе исследования, а именно: подход к децентрализованному хранению. Было обнаружили два различных подхода к реализации децентрализованного хранения. Более распространенным является хранение файлов в децентрализованном хранилище IPFS, а вторым - использование децентрализованного хранилища Swarm.

2.5.2 Связь и зависимость Web3 от блокчейна и смарт-контрактов

Ранее упоминалось, что основные принципы Web3 заключаются в том, что он децентрализован, не требует наличия разрешений и имеет встроенные платежи. Блокчейн - это децентрализованный распределенный реестр, а это уже относится к децентрализованной структуре данных. В блокчейне также нет доверенной третьей стороны, что делает его ненадежным. Каждый, у кого есть криптокошелек, может подключиться к сети блокчейн. Блокчейн-

сети имеют внутреннюю валюту и позволяют осуществлять платежи. Можно сказать, что блокчейн - это фундамент, на котором будет строиться Web3, а смарт-контракты - для реализации логики. Все проанализированные исследования использовали технологию блокчейн и смарт-контракты для реализации своих решений. Поэтому можно сделать вывод, что Web3 зависит от блокчейна и смарт-контрактов.

2.5.3 Невзаимозаменяемые токены (NFT) для разработки Web3

Хотя NFT не может существовать без блокчейна, блокчейн не обязательно требует NFT. Был пример реализации платформы для потокового видео, где контент хранится в виде NFT. В другой реализации NFT использовались в качестве награды за успешную игру в головоломку. NFT в разработке Web3 играют определенную роль в специфической нише Web3. Хорошая демонстрация роли NFT в Web3 была предложена в небольшой статье All NFT Space в соответствии с рисунком 2.3. В настоящее время наиболее распространенным использованием NFT является представление прав собственности на цифровой актив, обычно цифровое искусство или предмет, заработанный в видеоигре. Однако растет число практик, использующих NFT в различных областях, например, в системах продажи билетов или персональной идентификации.

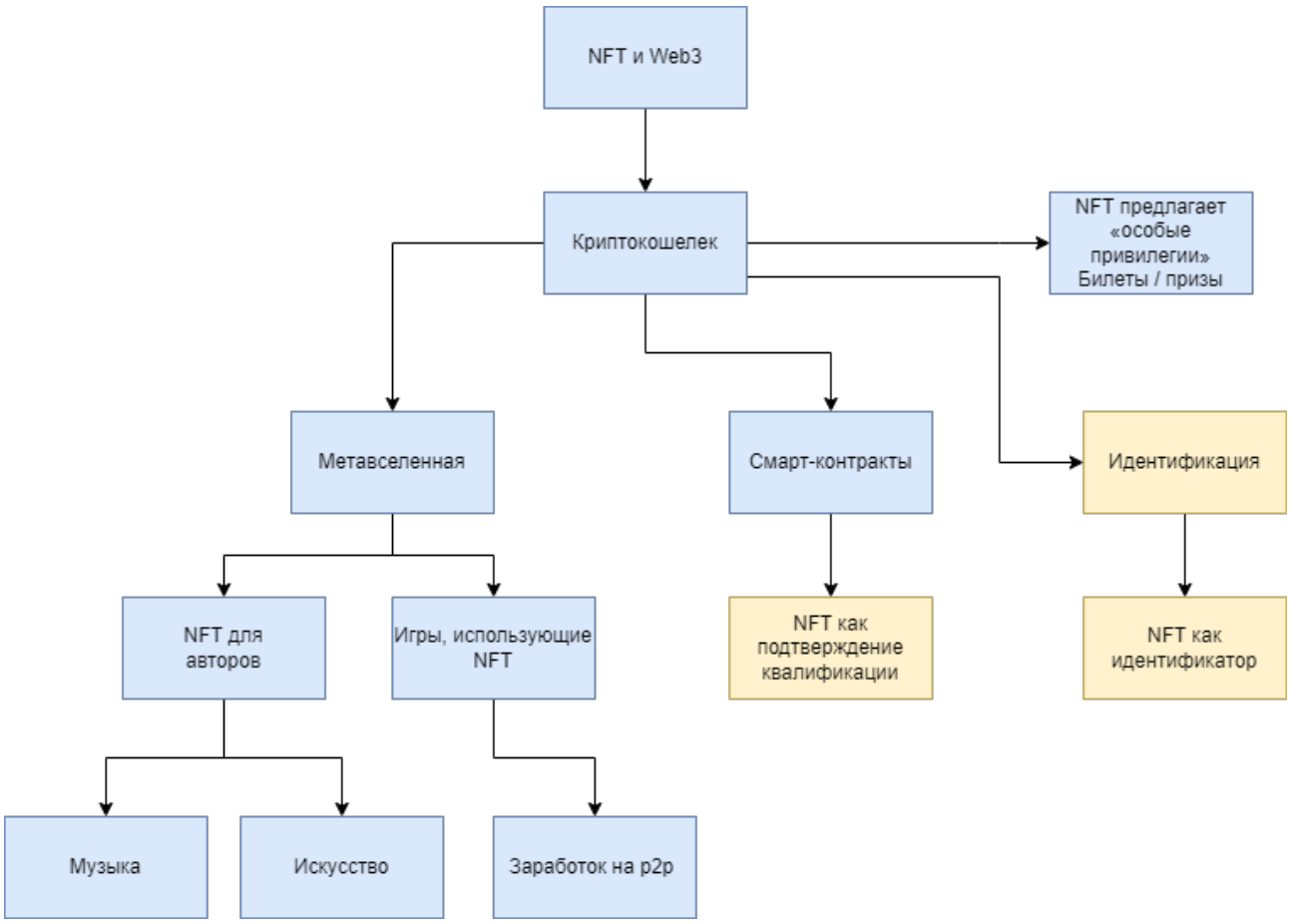

Рисунок 2.3 — Роль NFT в Web3

2.5.4 Полный стек для Web3

Сначала был собран стек технологий из изученных исследований, упорядочен в логические слои и добавлен на интеллект-карту для лучшего представления. Это можно увидеть в соответствии с рисункомw 2.4. На рисунке показана предложенная карта полного стека технологий для разработки Web3. Было определено восемь слоев, пять из которых, выделенные зеленой рамкой, являются минимальным стеком для разработки полного Web3-решения. Сетевой слой представляет блокчейн и должен быть отправной точкой для разработчиков при выборе технологий для создания решения. Среда разработчика - важный аспект, поскольку в ней будет проводиться большая часть тестирования смартконтрактов. Выбор фронтенд-фреймворка остается на усмотрение разработчика и обычно выбирается исходя из того, есть ли у разработчика уже опыт работы с ним. Коммуникационный слой - это

мост между фронтендом и смартконтрактами. Именно здесь фронтенд взаимодействует с функциями смартконтракта. Последним из минимально необходимого стека для разработки является слой идентификации, который позволяет связать кошелек с приложением. Наиболее важным дополнительным слоем является слой хранения, который необходим в любом сценарии, где необходимо хранить большие данные. Слой исследователей блокчейна добавляет возможность сканировать блокчейн и видеть детали любой транзакции в сети. Наконец, есть слой поддерживающих платформ Web3, который в последнее время привлекает наибольшее внимание, и вокруг него создаются целые сообщества. Платформы Web3 предоставляют разработчикам необходимые инструменты, чтобы сделать процесс создания приложений Web3 проще и быстрее. В то время как представленные слои, скорее всего, останутся неизменными, проекты, а вместе с ними и сообщества, имеют потенциал для кардинального развития.

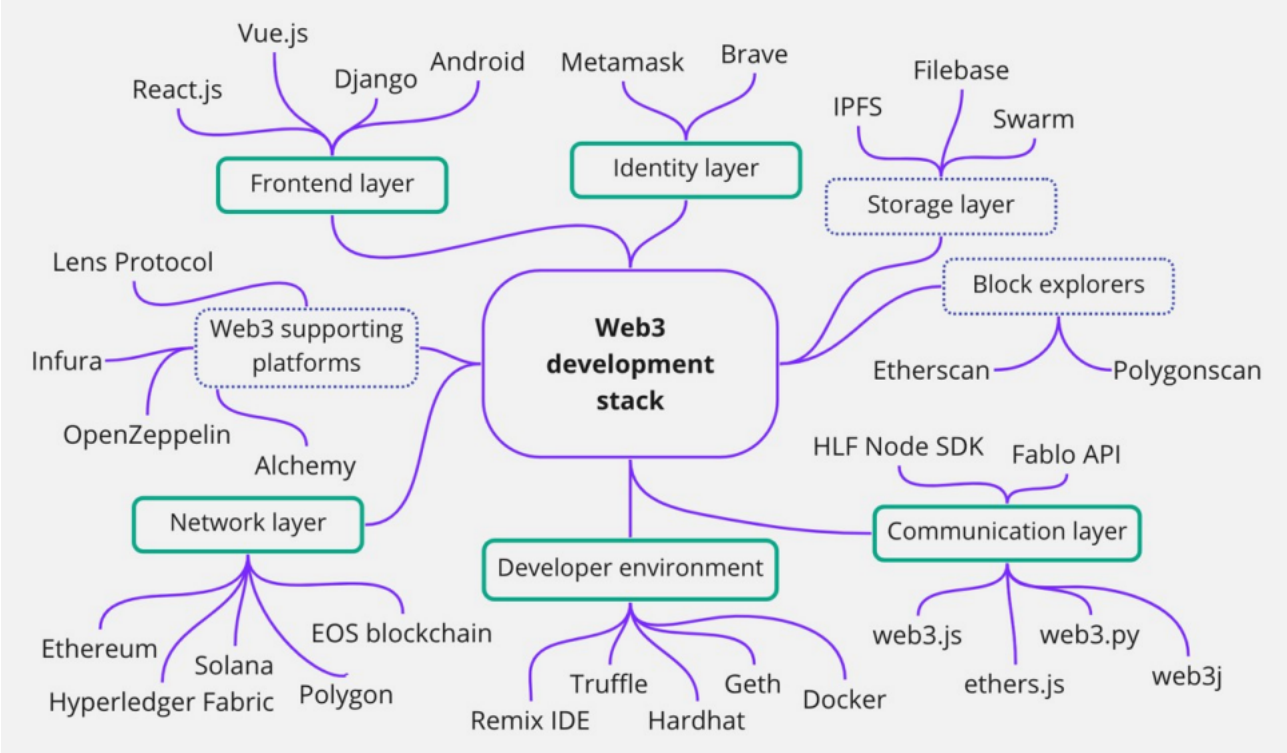

Рисунок 2.4 — Стек технологий разработки Web3

2.6 Примеры применения Web3

2.6.1 DeFi

DeFi означает децентрализованные финансы, и это был первый популярный пример решения Web3. Он представляет собой версию Web3 о более прозрачной финансовой системе с главной целью - не зависеть от регуляторов или человеческого фактора. Большинство решений DeFi позволяют пользователям управлять своими средствами в безналичной форме, используя криптокошелек. Типичным решением DeFi является децентрализованная биржа (DEX), которая представляет собой одноранговую торговую площадку, позволяющую взаимодействовать покупателям и продавцам криптовалюты. Одной из самых известных является MakerDAO, запущенная в 2017 году. Другой пример - Uniswap, протокол с открытым исходным кодом для обеспечения ликвидности и торговли токенами ERC20 на Ethereum. В отличие от других популярных бирж, таких как Coinbase или Binance, Uniswap полностью децентрализована.

2.6.2 Web3-игры

Web3-игра - это децентрализованная версия традиционной видеоигры, в которой игрок полностью владеет своими активами и опытом, заработанными в децентрализованной экосистеме Web3-игры. Это позволяет игрокам получить инновационное преимущество - играть, чтобы зарабатывать, поскольку игровыми активами можно торговать с помощью криптовалюты. Web3-игры также не имеют единой точки отказа благодаря своей распределенной природе, что обеспечивает высокую доступность. Кроме того, они используют консенсус голосования для внесения изменений в игровой процесс и делают игроков реальными участниками игровой экосистемы. Еще одной актуальной темой в Web3-играх является идея Metaverse - мира виртуальной реальности, к которому люди будут подключаться с помощью устройств виртуальной реальности (VR) или дополненной реальности (AR) и который будет имити-

ровать реальный мир всеми возможными способами. Примером Web3-игры является Axie Infinity, покемоноподобная игра, в которой игрок собирает существ, называемых Axies, и владеет ими как NFT. Акси можно разводить, торговать или участвовать в сражениях. Это бесплатная браузерная игра, но для игры необходимо приобрести команду из трех Axies. Взлом их сети на 615 миллионов долларов 23 марта 2022 года, их цена упала до нескольких долларов. Однако эта «случайность» не стала причиной конца эры Web3-игр, существует множество других Web3-игр и еще больше находится в разработке.

2.6.3 Социальные сети Web3

Крупные корпорации, такие как Facebook, Twitter и Instagram, в настоящее время доминируют на рынке социальных сетей и получают прибыль, собирая данные пользователей, продавая их и запуская целевую рекламу. В прошлом было много жалоб и судебных исков против таких компаний за вторжение в частную жизнь пользователей и слишком большой контроль над личной информацией пользователей. Участники Web3 стремятся создать другую социальную сеть, где платформы управляются сообществами, а пользователи имеют контроль над своей личной информацией, контентом и личностью. Протокол Lens является примером передовых решений Web3 для социальных сетей. Он позволяет создавать несколько социальных сетей и сервисов обмена сообщениями на отдельных клиентах, но использовать один и тот же протокол смарт-контрактов с открытым исходным кодом. В Lens социальные идентификаторы хранятся как NFT в кошельках пользователей и могут быть перенесены на все dapps, которые интегрируют этот протокол.

2.6.4 Маркетплейсы Web3

Web3-рынок можно описать как систему, в которой набор смарт-контрактов координирует поставщиков услуг и клиентов, а также облегчает их взаимодействие. Поставщики могут предлагать множество различных

уровней индивидуальных услуг или продуктов. И клиенты, и поставщики услуг будут зарабатывать один и тот же токен управления на основе их вклада в систему. Braintrust - хороший пример такого рынка. Он называется сетью талантов, принадлежащей пользователям, и соединяет пользователей, которые хотят работать в качестве фрилансеров, с предприятиями, которые хотят быстро найти нужный талант для своего проекта. Пока еще рано определять, будет ли такая бизнес-модель успешной в будущем и сможет ли она конкурировать с бизнесом Web2, предлагающим те же услуги, это зависит от сообщества. Важно отметить один из видов Web3-рынков, который настолько вырос за последние годы, что заслуживает отдельного упоминания. Он называется NFT marketplace, и представляет собой шлюз для торговли NFT. Существуют различные подобные торговые площадки, некоторые из самых крупных включают:

- OpenSea: Самая большая торговая площадка NFT на данный момент. Он предлагает торговлю произведениями искусства, музыкой, фотографиями, торговыми картами и виртуальными мирами;
- Rarible: пользователи могут торговать предметами искусства, коллекционными вещами и активами видеоигр. Они создали свой собственный токен RARI. Кроме того, они сотрудничают с Adobe, чтобы упростить проверку и защиту метаданных цифрового контента;
- NBA Top Shot: Это рынок самых ярких моментов в истории баскетбола, управляемый НБА. Это пример участия крупных компаний в тренде NFT;
- Binance: изначально это криптовалютная биржа, но она добавила к своим функциям торговую площадку NFT и предлагает торговлю предметами искусства, игровыми активами и коллекционными предметами;
- Nifty Gateway: Платформа известна тем, что на ней проводятся дорогие и эксклюзивные продажи NFT, включая «The Merge» цифрового художника Пака, которая была продана за $91,8 млн. Они сосредоточены на продаже только произведений искусства, особенно от знаменитостей и ведущих художников;

3 Реализация приложения на Web3

Ранее рассматривалось различия между Web2 и Web3. Однако, стоит задача дополнительно изучить, как эти различия отражаются на разработке приложений. Для этого будет реализовано децентрализованное приложение с использованием полного стека для разработки Web3, который был представлен ранее. Стоит задача использовать хотя бы по одной технологии из каждого слоя стека, чтобы эксперимент был полноценным. Затем рассмотрим, как то же самое решение можно было бы реализовать, используя технологии полного стека Web2, как выглядела бы архитектура системы и какие технологии были бы использованы для ее реализации. Наконец, сравним наблюдаемые различия между решениями Web2 и Web3. В качестве proof-of-concept приложения будет разработано WeddingFund - децентрализованное приложение, предназначенное для сбора свадебных подарков в виде криптовалы и пожеланий на свадебных открытках. Идея решения исходит из того, что молодожены могут быть молодыми парами без больших сбережений и часто предпочитают получать деньги, а не бессмысленные подарки от приглашенных, кроме того, сама свадьба стоит дорого, и они хотели бы позволить себе приятный медовый месяц. Предлагаемое решение позволяет пожертвовать любую сумму Эфира в созданный фонд для молодоженов, к которой должно прилагаться пожелание в виде цифровой свадебной открытки в формате изображения. Пожертвования перечисляются в фонд по смарт-контракту и могут быть собраны в любое время владельцем контракта. Пожелания свадебных открыток хранятся в децентрализованном хранилище IPFS.

3.1 Настройка среды

Инструменты и фреймворки для разработки были установлены и запущены с помощью Node Package Manager (npm) версии 8.5.5. и Node.js версии 16.13.1. Интегрированной средой разработчика (IDE) была выбрана Visual Studio Code. Важными расширениями, были установлены Solidity и React extensions. Для локальной среды тестирования использовался Hardhat, который предлагает развертывание интеллектуальных приложений, контрактов, выполнять тесты и производить отладку кода Solidity на локальном

блокчейне Ethereum. Это отличный инструмент для отладки Solidity, который отображает трассировку стека Solidity, console.log и явные сообщения об ошибках при неудачных транзакциях. Смарт-контракты были реализованы на Solidity и скомпилированы с помощью встроенного компилятора Hardhat. Они были развернуты в сети Hardhat и протестированы с помощью скриптов, написанных на JavaScript. Публичное тестирования Alchemy использовалась для размещения узла Ethereum в тестовой сети Goerli. Alchemy также предоставил приборную панель разработчика для получения подробной статистики. Для разработки в тестовой сети Goerli нужно владеть некоторым количеством ее криптовалюты Goerli ETH (GOR), которая была получена от Goerli Faucet. Она позволяет получать 0,2 GOR каждый день. Для подключения кошелька к сети Goerli testnet было использовано браузерное расширение MetaMask в веб-браузере Google Chrome. Для тестирования решения было создано несколько кошельков MetaMask. Приложение требовало хранения пожеланий к свадебным открыткам в формате изображений, и хранить их на блокчейне было бы неэффективно и дорого. Было принято решение использовать децентрализованное хранилище IPFS и платформу Infura для размещения узла IPFS. Infura, как и Alchemy, предоставляет подробную статистику использования размещенного узла. Фронтенд-фреймворк использовали - Next.js, JavaScript-фреймворк, построенный на React. Причина такого решения заключалась в том, что Next.js будет отвечать всем требованиям проекта. Поскольку Next.js - это JavaScript-фреймворк, нужно было выбрать API на основе JavaScript. Было решено использовать ethers.js для взаимодействия со смарт-контрактами и ipfs-http-client для взаимодействия с IPFS. Для наблюдения за транзакциями в публичной тестовой сети Goerli был использован блокчейн-проводник Goerli Etherscan. Архитектура системы децентрализованного приложения WeddingFund показана в соответствии с рисунком 3.1.

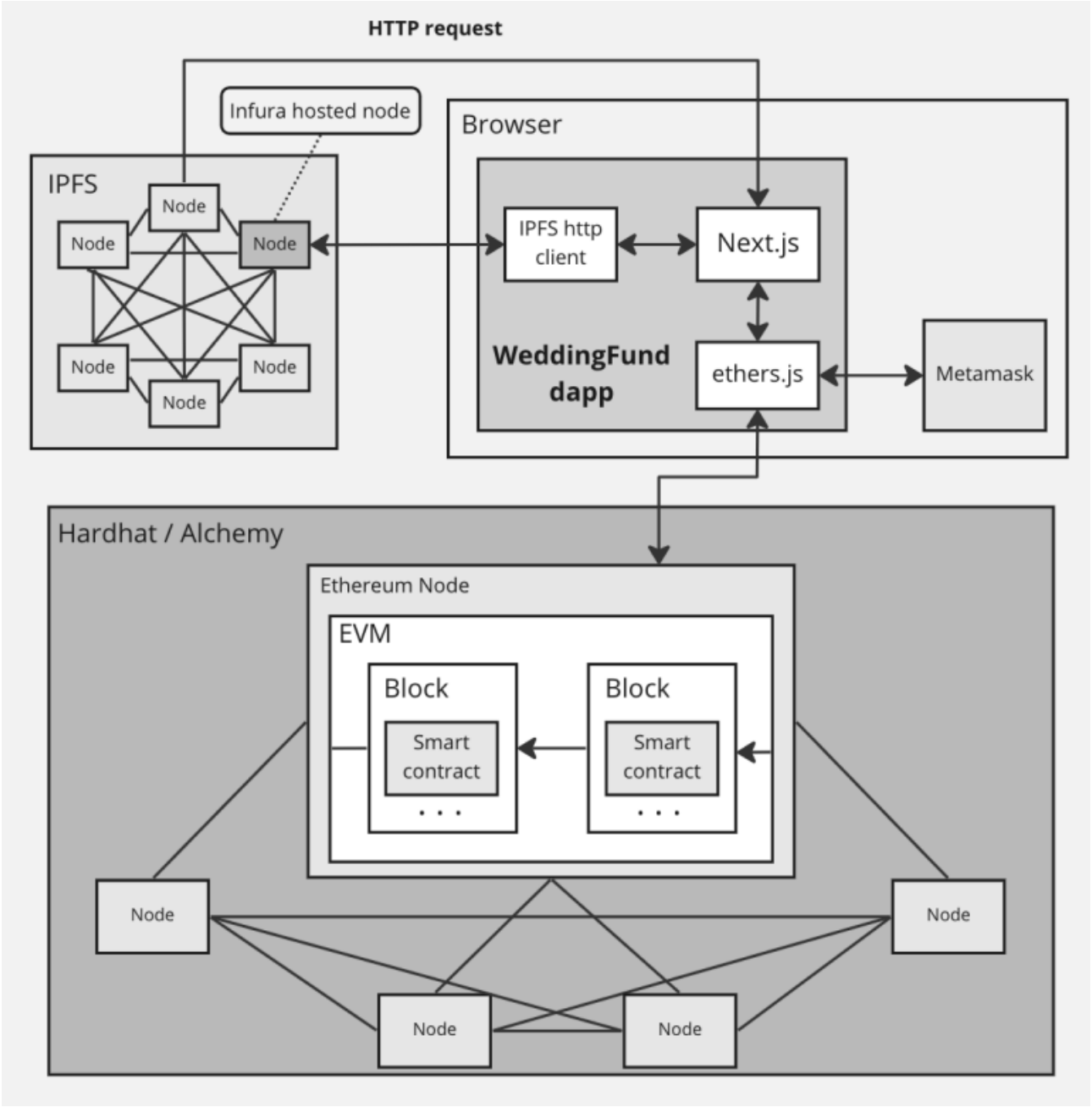

Рисунок 3.1 — Архитектура системы децентрализованного приложения WeddingFund

3.2 Внедрение решения Web3

Создания образца проекта производилось с помощью Hardhat. Из предложенных вариантов был выбран базовый образец проекта. Структура проекта, которая была создана, показана в соответствии с рисунком 3.2. Важными созданными папками являются «contracts», где хранятся файлы контрактов,

«scripts», где хранятся скрипты для взаимодействия со смарт-контрактом, и, наконец, «hardhat.config.js», где находятся конфигурации для версии Solidity и параметры развертывания. Папки «artifacts» и «cache» были сгенерированы после развертывания смарт-контракта.

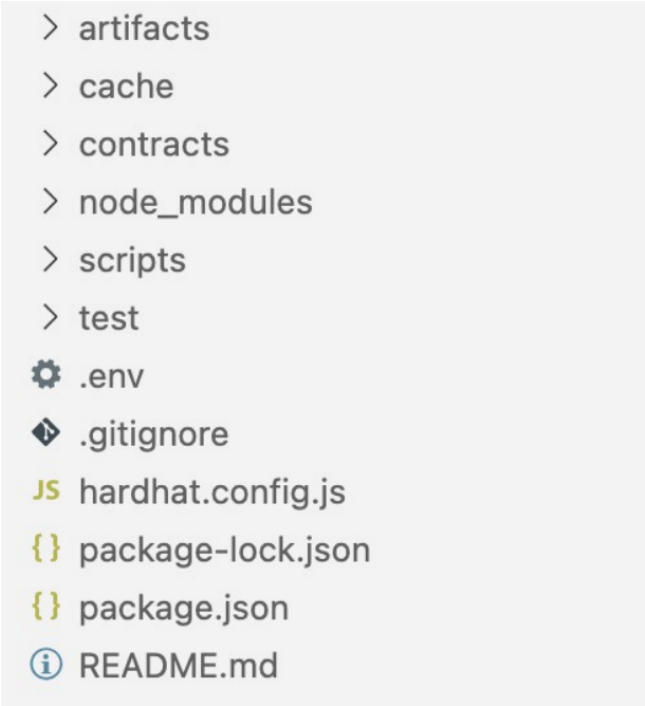

Рисунок 3.2 — Структура проекта Hardhat generated

Первое, что было реализовано, - это смарт-контракт для WeddingFund. Был определен объект Memo, в котором будет хранится адрес отправителя, временная метка, имя, сообщение и путь к файлу IPFS. Здесь же определены все операции, которые может выполнять смарт-контракт, а именно: оплата свадебного пожертвования, снятие средств и получение всех объектов Memo. Затем был реализован скрипт для развертывания контракта в сети в соответствии с рисунком 3.3.

```
const {ethers} = require("hardhat");

async function main(){
  // get the contract to deploy and deploy it
  const WeddingFund = await ethers.getContractFactory("WeddingFund");
  const weddingFund = await WeddingFund.deploy();
  await weddingFund.deployed();
  console.log("WeddingFund deployed to ", weddingFund.address);

}

main().catch((error) => {
    console.error(error);
    process.exitCode = 1;
});
```

Рисунок 3.3 — Скрипт для развертывания смарт-контракта

Здесь используется методы getContractFactory() и deploy(), которые предоставляет Hardhat. Далее был реализован скрипт для тестирования смарт-контракта в локальной сети Hardhat, в соответствии с Приложении. Здесь был использован Hardhat для создания нескольких фиктивных учетных записей, чтобы можно было протестировать контракт с несколькими участниками. По умолчанию Hardhat назначает каждому аккаунту 10000 ETH. Используется один счет, чтобы развернуть контракт и стать его владельцем, а остальные три счета - для выплаты свадебных пожертвований. После этого снимаются все пожертвования и переводятся на счет владельца. Наконец, выводим все сохраненные заметки. Отображаем остатки по всем адресам до оплаты, после оплаты и после снятия денег. Пример выполнения тестового сценария показан в соответствии с рисунком 3.4.

```
Wedding fund deployed to  0x5FbDB2315678afecb367f032d93F642f64180aa3
--- start ---
Address 0 balance:  9999.998461624375
Address 1 balance:  10000.0
Address 2 balance:  0.0
--- after payments ---
Address 0 balance:  9999.998461624375
Address 1 balance:  9998.999707521762723376
Address 2 balance:  3.0
--- after withdrawal ---
Address 0 balance:  10002.998415718991159167
Address 1 balance:  9998.999707521762723376
Address 2 balance:  0.0
--- memos ---
At 1665261646, Bob (0x70997970C51812dc3A010C7d01b50e0d17dc79C8) said: "Have fun
on honeymoon!" + hash: hash1
At 1665261647, Samantha (0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC) said: "Exci
ted for the wedding!" + hash: hash2
At 1665261648, George (0x90F79bf6EB2c4f870365E785982E1f101E93b906) said: "Enjoy
this donation. :D" + hash: hash3
```

Рисунок 3.4 — Пример выполнения тестового сценария

Адрес 0 представляет владельца контракта, адрес 1 - один из счетов дарителей, а адрес 2 - адрес развернутого смарт-контракта. Можно видеть, что небольшая сумма была выплачена со счета владельца за развертывание контракта. После платежей видно, что адрес 1 недосчитался чуть более 1 ETH, потому что 1 ETH был оплачен, а небольшая сумма была вычтена, как цена газа для транзакции. Баланс адреса 2 составляет 3,0 ETH, потому что три фиктивных счета пожертвовали по 1 ETH. После вывода средств можно увидеть, что баланс контракта вернулся к 0 ETH, а владелец получил чуть меньше 3 ETH, так как небольшая сумма была вычтена в качестве стоимости газа. После проверки работоспособности реализации, была начата работа по разработке пользовательского интерфейса, т.е. фронтенда. Сначала были настроены новые проекты на платформах Alchemy и Infura. На данный момент платформа поддерживает шесть блокчейн сетей, из которых была выбрана сеть Ethereum. Для Ethereum были доступны сети Mainnet и Goerli, другие сети были помечены как устаревшие и были отключены. Была выбрана сеть Goerli, чтобы не тратить реальную валюту для проверки приложения. После создания нового проекта можно получить доступ к API-ключу для подключения к платформе. Чтобы иметь возможность подключиться к сети Goerli,

нужно настроить конфигурационный файл Hardhat на использование URL подключения, который сгенерировал Alchemy, в соответствии с рисунком 3.5.

```
require("@nomicfoundation/hardhat-toolbox");
require("@nomiclabs/hardhat-ethers");
require("dotenv").config()

const GOERLI_URL = process.env.GOERLI_URL;
const PRIVATE_KEY = process.env.PRIVATE_KEY;

/** @type import('hardhat/config').HardhatUserConfig */
module.exports = {
 solidity: "0.8.17",
 networks:{
  goerli:{
   url: GOERLI_URL,
   accounts: [PRIVATE_KEY]
  }
 }
};
```

Рисунок 3.5 — Конфигурационный файл Hardhat.config.js

Далее нужно было настроить MetaMask для использования тестовой сети Goerli. Создание сети показано в соответствии с рисунком 3.6. Создание новой сети начинается с присвоения имени сети. Затем нужно предоставить новый RPC URL, который был сгенерирован Alchemy. Идентификатор сети Goerli - 5, а название валюты - Goerli ETH (GOR). В качестве блокчейн-проводника был выбран блокчейн-проводник Etherscan. Поскольку на тот момент криптовалюты GOR на балансе не было, пришлось посетить сайт Goerli Faucet и запросить ее. После запроса практически мгновенно было получено получили 0,2 GOR, так как Goerli Faucet предоставляет 0,02 Goerli ETH каждые 24 часа с бесплатной учетной записью Alchemy. После этого нужно было создать новый проект на платформе Infura. Для этого было использовано Infura для размещения нашего узла IPFS и для предоставления панели разработчика, чтобы легко получить представление о файлах, загруженных в IPFS. Infura

предлагает различные цены в зависимости от потребностей, но был выбран их бесплатный план. Он допускает 100 000 общих запросов в день и 5 ГБ хранилища на IPFS. Был план размещать на IPFS только изображения, и 5 ГБ было более чем достаточно для пробного приложения.

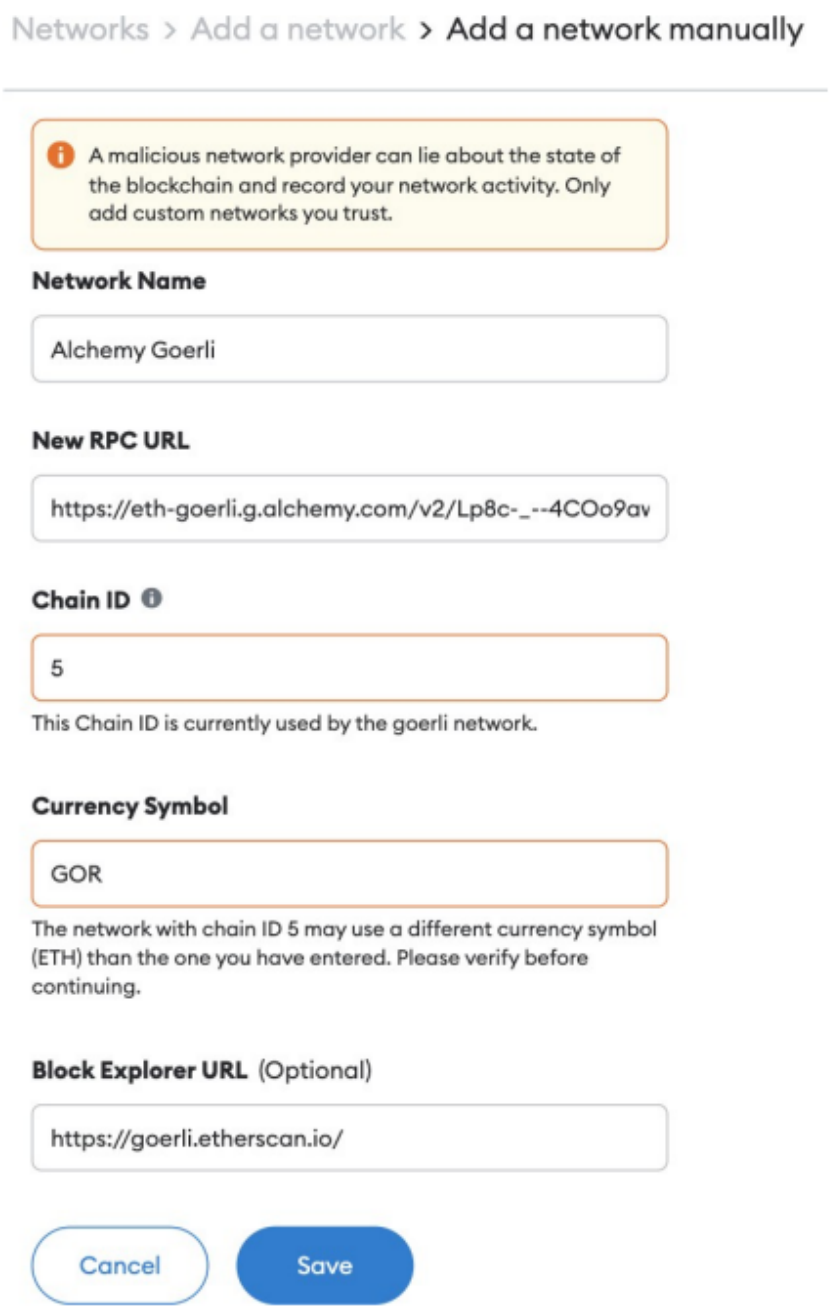

Рисунок 3.6 — Создание тестовой сети Goerli с помощью MetaMask

В качестве фронтенд-фреймворка был выбрали Next.js. Была использовали команда create-next-app CLI, чтобы создать структуру проекта. Затем

сначала были установлены необходимые зависимости, в соответствии с рисунком 3.7.

Рисунок 3.7 — Файл package.json

Был установлен bootstrap для создания компонентов оформления, ethers.js для реализации связи с блокчейном и ipfs-http-client для связи с IPFS. Для того чтобы ethers.js мог вызывать связь со смарт-контрактом, пришлось сгенерировать Application Binary Interface (ABI) смарт-контракта WeddingFund и импортировать его в файл index.js. ABI - это JSON-файл, содержащий информацию о точных именах и типах, связанных с операциями смарт-контракта. В приложении показан сгенерированный ABI контракта WeddingFund. Первое, что было реализовано во фронтенде, это проверка того, подключен ли кошелек пользователя. Если он не подключен, пользователю будет предложено подключить его. По сути, это аутентификация в Web3. Реализованные функции для проверки того, был ли подключен кошелек и для подключения кошелька в соответсвии с рисунком 3.8.

```
// Wallet connection logic
const isWalletConnected = async () => {
  try {
    const { ethereum } = window;

    const accounts = await ethereum.request({method: 'eth_accounts'})
    console.log("accounts: ", accounts);

    if (accounts.length > 0) {
      const account = accounts[0];
      console.log("wallet is connected! " + account);
    } else {
      console.log("make sure MetaMask is connected");
    }
  } catch (error) {
    console.log("error: ", error);
  }
}

const connectWallet = async () => {
  try {
    const {ethereum} = window;

    if (!ethereum) {
      console.log("please install MetaMask");
    }

    const accounts = await ethereum.request({
      method: 'eth_requestAccounts'
    });

    setCurrentAccount(accounts[0]);
  } catch (error) {
    console.log(error);
  }
}
```

Рисунок 3.8 — Логика подключения кошелька

Начальную страницу можно увидеть в соответствии с рисунком 3.9. Когда пользователь нажимает на кнопку, ему будет предложено подключить свой кошелек с помощью MetaMask. На начальной странице также отображается текущее значение фонда, поскольку для запроса сети кошелек пользователя не требуется.

Рисунок 3.9 — Начальная страница приложения

После того как пользователь подключит кошелек, он может отправить пожертвование и пожелание свадебной открыткой, используя форму, показанную в соответствии с рисунком 3.10.

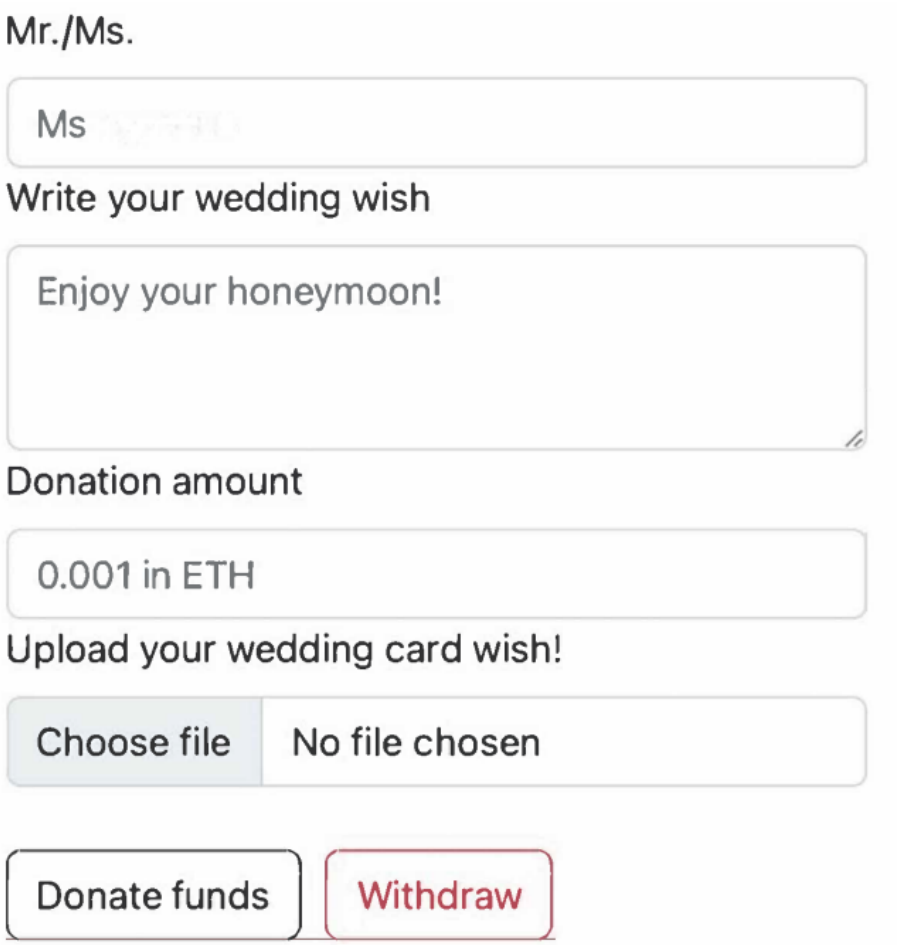

Рисунок 3.10 — Форма для пожертвований приложения WeddingFund

Сначала нужно было настроить IPFS. Сумма пожертвования указана в ethereum, а минимальная сумма, которую можно отправить, составляет 0.001 ETH. Эта функция включает в себя реализацию загрузку изображения пожелания свадебной открытки в IPFS. После успешной загрузки изображения извлекается путь IPFS, созданный для этого изображения, и сохраняет его в смарт-контракт вместе с другими данными из формы. Если подключенный кошелек является владельцем смарт-контракта, он также может вывести средства из фонда. Фрагмент кода функции withdraw показан в соответсвии с рисунком 3.11.

```
const withdrawFunds = async () => {

  try {
    const {ethereum} = window;
    if(ethereum){

      const provider = new ethers.providers.Web3Provider(ethereum, "any");
      const signer = provider.getSigner();
      const weddingFund = new ethers.Contract(
        contractAddress,
        contractAbi,
        signer
      );

      const contractBalance = await getBalance(provider, weddingFund.address);
        console.log("Current balance of contract: ", await getBalance(provider, weddingFund.address),
"ETH");

      // Withdraw funds if there are funds to withdraw.
      if (contractBalance !== "0.0") {
          console.log("withdrawing funds..")
          const withdrawTxn = await weddingFund.withdrawDonations();
          await withdrawTxn.wait();
      } else {
          console.log("No funds to withdraw!");
      }
      console.log("Funds withdrawn!");


    }
    } catch (error) {
      console.log(error);
    }
  }
```

Рисунок 3.11 — Реализация функции withdrawFunds во внешнем интерфейсе

Когда владелец выводит средства, баланс контракта возвращается к 0 ETH, но заметки с изображениями остаются на блокчейне и отображаются на странице. Подключенные пользователи также могут видеть все пожертвования от других дарителей, в соответствии с рисунком 3.12.

Рисунок 3.12 — Отображение пожертвований от всех дарителей

Реализация извлечения всех заметок из смарт-контракта приведена в соответствии с рисунком 3.13.

```
const getMemos = async () => {
  try {
    const { ethereum } = window;
    if (ethereum) {
      const provider = new ethers.providers.Web3Provider(ethereum);
      const signer = provider.getSigner();
      const weddingFund = new ethers.Contract(
        contractAddress,
        contractAbi,
        signer
      );

      console.log("fetching memos from the blockchain..");
      const memos = await weddingFund.getMemos();
      console.log("fetched!");
      setMemos(memos);
    } else {
      console.log("Metamask is not connected");
    }
  } catch (error) {
    console.log(error);
  }
};
```

Рисунок 3.13 — Реализация функции getMemos во внешнем интерфейсе

Сначала получаем список заметок из смарт-контракта, перебираем каждую из них и отображаем их на веб-странице. Для отображения изображений используется путь IPFS к изображению, чтобы найти место хранения каждого изображения.

Блокчейн-проводник Etherscan был использован для наблюдением за транзакциями решения в публичной тестовой сети Goerli. В соответствии с рисунком 3.14 можно видеть все транзакции, которые были выполнены, с соответствующими ценами на газ в транзакции. Первая транзакция, начиная с нижней, представляет собой развертывание контракта в сети, где можно понять, с какого адреса была отправлена транзакция и стоимость транзакции. Начальная стоимость контракта была равна 0. Три транзакции в середине представляют собой пожертвования в фонд, где каждое пожертвование составляло 0,001 ETH. Из этих транзакций уже можно найти адрес контракта

в колонке «To». Транзакция сверху - это вывод средств, и можно заметить, что во время этой операции стоимость газа (т.е. Txn Fee) самая низкая, что объясняется тем, что для вывода средств не требуется много вычислительной мощности. Самая высокая цена газа была во время пожертвований, потому что здесь Memos был добавлен в блок.

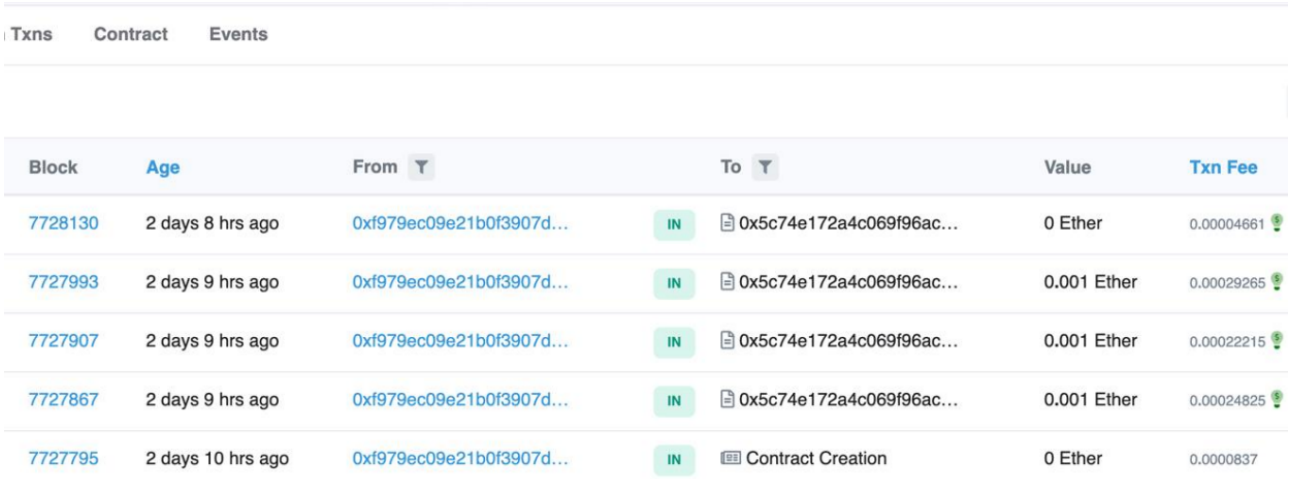

Рисунок 3.14 — Транзакции на блоке Explorer Etherscan

3.3 Концепция реализации решения Web2

Чтобы иметь возможность сравнить реализованное решение Web3 с аналогичным решением Web2, была подготовлена концепцию системной архитектуры приложения Web2. Для реализации фронтенда был выбран тот же фреймворк Next.js. При разработке Web2-решения пришлось бы дополнительно реализовать бэкенд. Для этого нужно будет использовать Node.js, а точнее Express.js, который является де-факто стандартным серверным фреймворком для Node.js. При разработке концепции решения Web2 стояла задача охватить следующие функциональные возможности:

- Аутентификация: Нужно позволить только аутентифицированным пользователям дарить свадебные подарки. Для этого было решено использовать OAuth с Firebase API. OAuth - это протокол аутентификации, который позволяет пользователям одобрить одно приложение, взаимодействующее с другим от их имени, не сообщая свой пароль. Google Firebase - это платформа, предоставляющая API для реализа-

ции OAuth и панель для разработчика, которая показывает статистику пользователей;

- Пожертвование средств: Для пожертвования средств используется API PayPal для безопасной отправки средств на указанный счет PayPal. PayPal - это компания, которая работает в качестве платежного процессора для онлайн-платежей, за что взимает комиссию. После того как пользователь подтвердит пожертвование нажатием кнопки, откроется окно PayPal с дальнейшими инструкциями по входу в аккаунт PayPal и подтверждению пожертвования;
- Хранение пожеланий на свадебной открытке: Здесь было два варианта. Можно было бы хранить изображения на внутреннем сервере, и в этом случае, как у владельца сервера, был бы полный контроль над хранимыми данными. Однако это привело бы к плохому пользовательскому опыту для пользователей, которые пытаются подключиться к приложению, но живут далеко от сервера, из-за большой задержки. Второй вариант - использование облачного хранилища Amazon Cloud, а именно Amazon S3, для размещения данных и выбора лучшего сервера, когда пользователь пытается подключиться к приложению. Для данной концепции был выбран последний вариант;
- Хранение других данных: Информация, полученная в результате аутентификации пользователей, и данные о пожертвованиях будут храниться на сервере в реляционной базе данных MySQL. Эти данные будут размещаться и контролироваться разработчиком;

Полная системная архитектура приложения аналога WeddingFund для Web2 представлена в соответствии с рисунком 3.15.

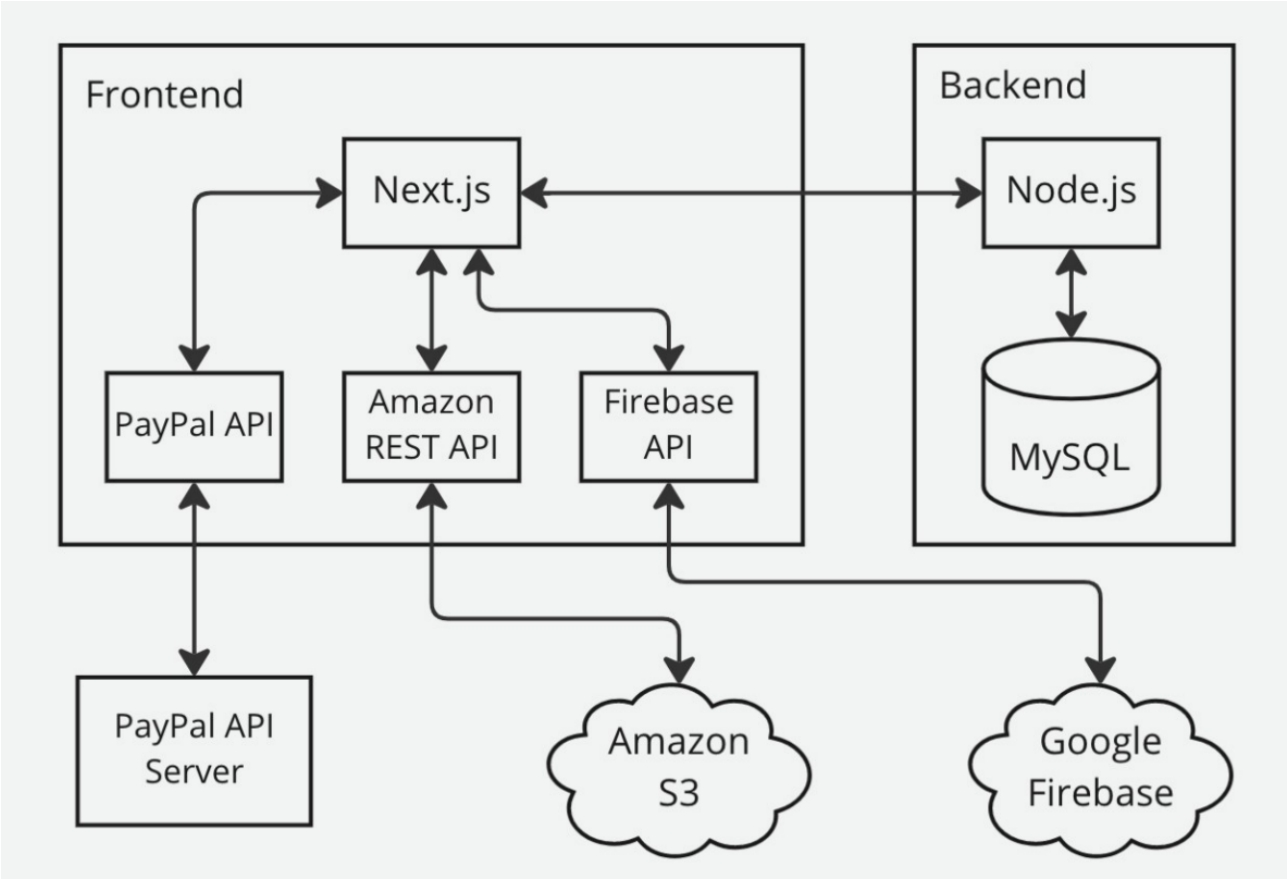

Рисунок 3.15 — Концепция системной архитектуры приложения Web2

3.4 Сравнение концепций Web3 и Web2 в рамках разработанного приложения

Реализованное решение WeddingFund представляет собой полностью децентрализованное приложение, построенное на блокчейне Ethereum, основной целью которого является пожертвование средств и пожелания молодоженам в свадебной открытке. Основными преимуществами реализованного децентрализованного приложения являются:

- Оно децентрализовано: Не существует центрального органа, который мог бы контролировать данные пользователей. Нет единой точки отказа, что означает, что данные о пожертвованиях и пожеланиях свадебных открыток должны быть всегда доступны без простоев. За обработку пожертвований не нужно платить, кроме небольшой комиссии за транзакционный газ;

- Оно псевдонимно: взаимодействие с приложением создается с помощью кошелька MetaMask, где совместно используется только публичный адрес пользователя. Если этот адрес не был скомпрометирован, анонимность гарантирована;
- Оно прозрачно: Все пожертвования публично видны;
- Оно защищено от взлома: Данные надежно хранятся в блокчейне Ethereum и не могут быть подделаны;

Реализованное децентрализованное приложение было протестирован на тестовой сети Goerli. В таблице в соответствии с рисунком 3.16, можно увидеть стоимость каждой операции и их стоимость в Российских рублях на данный момент. Самая низкая комиссия за транзакцию была во время снятия средств, а самая высокая - во время отправки пожертвования, что вполне логично, ведь данные записывались в блок.

Операция	Цена на газ в ETH	Цена на газ в рублях (RUB)
Развертывание смарт-контракта	0.000083	12,29
Пожертвование 1	0.000248	36,72
Пожертвование 2	0.000222	32,87
Пожертвование 3	0.000292	43,24
Снятие средств	0.000046	6,81

Рисунок 3.16 — Цены на газ в ETH и RUB

Чтобы убедиться, что реализованное приложение действительно является Web3-приложением, было произведена его оценка его на основе принципов Web3-приложений. Для удобства восприятия оценку можно посмотреть в таблице в соответствии с рисунком 3.17. Приложение является децентрализованным, поскольку все данные приложения хранятся на децентрализованных объектах, где транзакции хранятся на смарт-контракте Ethereum blockchain, а файлы изображений - на IPFS. Он также является недоверенным, поскольку не полагается на какую-либо доверенную третью сторону. Доверие распределяется между заинтересованными сторонами сети Ethereum, то есть

разработчиками, майнерами и потребителями. Каждый, у кого есть криптокошелек, может участвовать в работе приложения, не требуя разрешения административных органов, поэтому оно не требует разрешений. В приложении есть встроенные платежи; пользователи делают все свои пожертвования, отправляя родную криптовалюту Ethereum - Ether.

Основные принципы Web3	WeddingFund
Децентрализованность	✓
Без доверия	✓
Без разрешения	✓
Используются платежи в виде криптовалюты	✓

Рисунок 3.17 — Основные принципы Web3 в WeddingFund

Когда сравниваются архитектуры систем Web3 и Web2, сразу становится видно, что в Web2 нет децентрализации. Если рассматривать разработанное приложение в рамках архитектуры Web2, то имеется, что каждый компонент централизован и используется централизованный внутренний сервер, и три централизованных (incloud) сервиса, а именно PayPal, Amazon S3 и Google Firebase. Это означает, что данные из приложения передаются и хранятся этими компаниями, и они полностью контролирует эти данные. Помимо данных, обычно еще разработчики платят этим компаниям за использование их услуг или облака, даже если они могут предлагать бесплатные ограниченные планы. В Web2 также нет никакой формы анонимности, потому что требуется аутентификация, и используется Google Firebase, который уже предоставляет как минимум адрес электронной почты пользователя. Еще одно отличие - отсутствие прозрачности, так как PayPal обрабатывает платежи, только даритель и владелец фонда могут знать сумму пожертвования. Хранение пожеланий свадебных открыток в обоих решениях распределенное, где IPFS использует одноранговую сеть узлов для хранения файлов, а Amazon использует сети распространения контента (CDN). Разница между подходами заключается в том, что Amazon S3 централизован и контролируется Amazon, а IPFS децентрализован и контролируется пользователями. Тяжело

точно определить сложность разработки решения Web2, чтобы сравнить ее со сложностью разработки решения Web3, но исходя из концепции системной архитектуры и собственного опыта разработки приложений Web2, можно отметить, что Web2 является более сложным, поскольку в нем дополнительно используется бэкенд-хранилище MySQL. Различие подходов представлено в таблице в соответствии с рисунком 3.18.

Подход	Web3 приложение	Web2-приложение
Фронтэнд	Next.js	Next.js
Бэкэнд	-	Node.js
Внутреннее хранилище	-	MySQL
Хранение изображений	IPFS	Amazon S3
Аутентификация	ethers.js и MetaMask	Google Firebase
Платежи	ethers.js и смарт-контракт	API PayPal

Рисунок 3.18 — Различие в подходах в приложениях Web3 и Web2

Есть некоторые идеи, как можно было бы реализовать это приложение по-другому, например если бы нам требовалось реализация приложения с использованием NFT. Ранее уже говорилось, что NFT не обязательно нужны для разработки Web3, поэтому NFT не было реализовано для данного приложения. Однако можно было бы использовать NFT в качестве приглашения на свадьбу, где после того, как пользователь отправит пожертвование на свадьбу, будет сгенерирован уникальный NFT, который послужит входным билетом на свадьбу. Это работало бы как, генерирование мультипликационного изображения молодоженов, например, с уникальной цветовой комбинацией для каждого гостя свадьбы. В приложении Web2 реализовать такую систему нельзя. Вместо этого можно было бы сгенерировать PDF-файл с изображением молодоженов вместе со сгенерированным QR-кодом и отправить его по электронной почте каждому гостю в качестве билета на свадьбу.

ЗАКЛЮЧЕНИЕ

Основной целью данной дипломной работы было изучение технологий для разработки решений Web3 и исследование того, какие технологии в совокупности образуют полный стек для разработки Web3. Для целей исследования были изучены концепции блокчейн-технологий и Web3. Были ввели такие понятия, как Ethereum, смарт-контракты, криптокошельки, невзаимозаменяемые токены, а также газ или стоимость транзакции. Описание развития Web было продолжено, где подробно представили каждую версию Web по отдельности, а затем сравнили их, чтобы проиллюстрировать различия. Также были представлены технологии, используемые для разработки решений Web3, наиболее распространенные типы решений Web3 и примеры каждого из них. Наконец, на основе выявленных технологий, было реализовано приложение. Полученное децентрализованное приложение было назвали WeddingFund, а его целью является сбор средств для молодоженов. Сначала была представлена подготовка к практической части, где рассматривалась разработка, среда и все используемые технологии. Затем был подробно описан процесс внедрения решения и результат. Кроме того, получившееся приложение было сравнено с эквивалентной концепцией приложения Web2, чтобы легче представить различия между Web2 и Web3. В конце была разобрана возможная идеия иной реализации похожего приложения.

Технология блокчейн изменила и продолжает изменять мир. Она привнесла инновации в самые разные отрасли и позволяет компаниям внедрять новые и интересные услуги и возможности. Децентрализованные приложения являются одним из побочных продуктов технологии блокчейн, которые предлагают безопасное программное обеспечение с открытым исходным кодом для повседневных пользователей и предприятий. Как и в случае со всеми достижениями, неизбежно, что многие из используемых в настоящее время практик устареют, однако основной подход, скорее всего, останется прежним.

В заключении работы, можно отметить, что технология Web3 имеет большой потенциал для развития и применения в различных областях.

На основании проведенного исследования, можно сделать следующие рекомендации по применению принципов Web3 в практике:

- Уделять большое внимание безопасности в разработке и использовании децентрализованных приложений и смарт-контрактов;
- Использовать стандарты и протоколы, разработанные сообществом, для обеспечения совместимости между различными блокчейн-системами;
- Учитывать экономические аспекты разработки и использования Web3.0 решений, такие как стоимость транзакций, оплата за использование ресурсов сети и другие факторы;
- Развивать и использовать инструменты и платформы для разработки и тестирования смарт-контрактов и децентрализованных приложений, такие как Hardhat, Truffle и другие;
- Активно участвовать в сообществе разработчиков и учиться новым технологиям и методам разработки Web3.0 решений;

Вместе с тем, Web3 является технологией, которая только начинает свой путь развития. В будущем можно ожидать еще большего развития децентрализованных приложений и смарт-контрактов, а также появления новых инструментов и платформ для их разработки и использования.

Можно предположить, что в будущем Web3.0 станет неотъемлемой частью многих отраслей, включая финансовую, медиа, здравоохранение и другие.

Таким образом, можно сделать вывод, что разработка и использование Web3.0 решений имеет большое будущее, и настоящее время является хорошей возможностью для исследования и применения этих технологий в практике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Абрамов, А. А., Черный, А. И. Web 3.0 [Текст] / А. А. Абрамов, А. И. Черный. – СПб.: БХВ-Петербург, 2018. – 336 с.

2 Богданова, И. В., Куликова, А. А. Web 3.0 как возможность развития электронного обучения [Текст] / И. В. Богданова, А. А. Куликова / Современные технологии в науке, образовании и производстве : материалы VII Международной научно-практической конференции (г. Томск, май 2021 г.). – Томск : Издательство Томского государственного университета, 2021. – С. 24-28.

3 Васильев, А. С. Основы Web 3.0 [Текст] / А. С. Васильев. – М.: ООО «Бизнес-школа ИнтелСофт», 2018. – 320 с.

4 Григорьев, Н. Н., Гуревич, Е. М., Соколов, А. Н. Разработка децентрализованных приложений на базе технологии блокчейн [Текст] / Н. Н. Григорьев, Е. М. Гуревич, А. Н. Соколов. – М.: ФИЗМАТЛИТ, 2019. – 192 с.

5 Дроздов, В. И. Протоколы Web 3.0 [Текст] / В. И. Дроздов. – М.: Издательский дом «ЛКИ», 2019. – 240 с.

6 Карякин, Д. А. Web 3.0: взгляд изнутри [Текст] / Д. А. Карякин / Журнал «Хакер», № 229, 2018. – С. 28-37.

7 Климанов, И. А., Шишкин, А. Г. Web 3.0. Концепции и технологии [Текст] / И. А. Климанов, А. Г. Шишкин. – М.: ФИЗМАТЛИТ, 2021. – 192 с.

8 Дугин, А. С. Технологии Web 3.0 и их роль в развитии интернет-приложений / А. С. Дугин / Информационное общество. – 2018. – Т. 7. – С. 24-32.

9 Жуков, Д. И. Исследование применения блокчейн технологий в Web 3.0 / Д. И. Жуков / Проблемы современной экономики. – 2020. – № 2 (74). – С. 112-117.

10 Кашин, С. А. Технологии Web 3.0: основные аспекты и применение / С. А. Кашин / Информационные технологии. – 2021. – Т. 27. – № 5. – С. 297-303.

11 Мурашко, С. В. Разработка децентрализованных приложений на базе технологии Web 3.0 / С. В. Мурашко, С. Н. Шишкин / Системы управления, связь и безопасность. – 2019. – Т. 16. – № 4. – С. 75-82.

12 Немченко, А. В. Технологии Web 3.0: понятие, особенности, применение / А. В. Немченко / Информатизация образования и науки. – 2018. – Т. 15. – № 1. – С. 100-105.

13 Попов, В. Ю. Web 3.0 – новый этап развития сети Интернет / В. Ю. Попов / Мировая наука и образование. – 2021. – Т. 4. – № 3 (101). – С. 58-60.

14 Селиванов, И. П. Развитие Web 3.0: новые технологии и перспективы применения / И. П. Селиванов / Информационные технологии и вычислительные системы. – 2019. – Т. 1. – С. 29-37.

15 Сергеев, А. В. Децентрализованные приложения на основе технологий Web 3.0 / А. В. Сергеев / Информационные технологии и системы. – 2018. – № 4 (24). – С. 60-66.

16 Глушкова, А.В. Инновационное развитие российской экономики на основе технологий Web3 [Текст] / А.В. Глушкова, М.В. Чистова / Экономика и управление. – 2019. – № 4 (155). – С. 44-50.

17 Разуваев, О. Web3 и децентрализованные приложения [Текст] / О. Разуваев / Хакер. – 2018. – № 8. – С. 54-57.

18 Карягина, Т. Web3.0 – технология децентрализации интернета [Текст] / Т. Карягина / Наука и технологии. – 2019. – № 4 (30). – С. 87-92.

19 Шереметьев, А. Web3 и его роль в развитии децентрализованных приложений [Текст] / А. Шереметьев / Инновации и инвестиции. – 2021. – № 3. – С. 15-20.

20 Гришин, Е. Web3.0 – будущее интернета [Текст] / Е. Гришин / Банковское дело. – 2019. – № 4. – С. 48-51.