

ОСНОВЫ БЛОКЧЕЙНА



Даниэль Дрешер

Даниэль Дрешер

**Основы блокчейна: вводный
курс для начинающих
в 25 небольших главах**

«ДМК Пресс»

2017

УДК 004.62:004.738.5
ББК 32.972.134

Дрешер Д.

Основы блокчейна: вводный курс для начинающих в 25 небольших главах / Д. Дрешер — «ДМК Пресс», 2017

ISBN 978-5-97060-591-2

Книга подробно рассматривает технические концепции технологии блокчейн, такие как пиринговые и распределенные системы, структуры данных, транзакции, криптография и хэш-значения, целостность систем и достижение консенсуса в распределенной среде. Книга написана в диалоговом стиле, без использования компьютерного и математического жаргона. Материал излагается в пошаговой, логически связанной манере, что позволяет последовательно, уровень за уровнем, наращивать знания о технологии блокчейна. Многочисленные примеры, аналогии и метафоры помогают лучше понять, как работают блокчейн-системы даже тем, кто до этого ничего не знал об этом. Издание предназначено для широкого круга читателей с различным уровнем технических знаний, желающих разобраться, что же такое блокчейн.

УДК 004.62:004.738.5

ББК 32.972.134

ISBN 978-5-97060-591-2

© Дрешер Д., 2017
© ДМК Пресс, 2017

Содержание

Об авторе	6
О Техническом рецензенте	7
Предисловие	8
Зачем нужна еще одна книга о технологии блокчейна?	9
Чего не следует ждать от этой книги	10
Чего следует ожидать от этой книги	11
Как организована эта книга	13
Дополнительные материалы	15
Часть I	16
Глава 1	17
Глава 2	21
Глава 3	29
Конец ознакомительного фрагмента.	31

Даниэль Дрешер
Основы блокчейна: вводный курс для
начинающих в 25 небольших главах

Daniel Drescher
Blockchain Basics
A Non-Technical Introduction in 25 Steps

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Copyright © 2017 by Daniel Drescher
© Оформление, издание, перевод, ДМК Пресс, 2018

* * *

Об авторе

Даниэль Дрешер (Daniel Drescher) – опытный профессионал в банковской сфере, работавший в области электронной торговли ценными бумагами в нескольких банках. В последнее время его деятельность сосредоточена на задачах автоматизации, машинного обучения и обработки больших данных в сфере торговли ценными бумагами. Кроме того, Даниэль имеет докторскую степень по эконометрике (математической экономике) в Берлинском Техническом университете и степень магистра инженерии программного обеспечения, присвоенную Оксфордским университетом.

О Техническом рецензенте



Лоренс Керк (Laurence Kirk) после успешной карьеры автора оперативного финансового прикладного программного обеспечения для делового центра Сити в Лондоне заинтересовался потенциальными возможностями технологии создания распределенного программного обеспечения для финансового учета. Он поступил в Оксфордский университет для получения степени магистра и основал компанию Extropu.io, консультирующую стартапы по разработке прикладных программ для платформы Ethereum. Увлеченный возможностями технологии распределенного программного обеспечения, сейчас он является разработчиком, экспертом-консультантом и инструктором по вопросам использования платформы Ethereum.

Предисловие

Данное предисловие отвечает на самый важный вопрос, на который обязан ответить любой автор: зачем читать эту книгу? Или на более конкретный вопрос: зачем читать еще одну книгу по технологии Blockchain? Продолжайте читать, и вы поймете, зачем написана эта книга, чего ждать от нее и чего в ней не следует искать. Вы также узнаете, для какой аудитории написана эта книга и как она организована.

Зачем нужна еще одна книга о технологии блокчейна?

Технология блокчейна (Blockchain, или цепочка блоков транзакций) сразу после своего появления привлекла большое внимание при крупномасштабных обсуждениях и в специализированных средствах массовой информации. Некоторые энтузиасты даже объявили блокчейн самым великим изобретением с момента появления Интернета. Поэтому за несколько последних лет о блокчейне было написано большое количество книг и статей. Но если вы хотите узнать больше о том, как устроен и как работает блокчейн, то вскоре можете просто потеряться в бездне книг, в которых технические подробности описываются весьма поверхностно, или базовые технические концепции излагаются на чрезмерно формализованном уровне. Первый вариант не удовлетворяет любознательного читателя, поскольку не дает описания технических деталей, необходимых для понимания и оценки по достоинству технологии блокчейна, во втором случае при изучении требуется владение именно теми знаниями, которые вы хотите получить.

Эта книга предназначена для заполнения разрыва между абсолютно технической литературой по блокчейну, с одной стороны, и книгами, в которых почти все внимание сосредоточено на специализированных приложениях, или на описаниях предполагаемого экономического эффекта от применения этих приложений, или даже на рассуждениях о будущем блокчейна, с другой стороны.

Эта книга написана, потому что концептуальное понимание технических основ блокчейна необходимо, чтобы понять функциональность специализированных блокчейн-приложений, исследовать бизнес-варианты деятельности блокчейн-стартапов или полноправно участвовать в обсуждении ожидаемых экономических эффектов. Без хорошего понимания базовых теоретических концепций невозможно дать числовую оценку реального эффекта или потенциального воздействия блокчейна вообще или числовую оценку полезности, добавляемой специализированными блокчейн-приложениями. Главное внимание в этой книге уделено основополагающим теоретическим концепциям блокчейна, так как недостаточное понимание новой технологии может привести к чрезмерному увлечению внешними ее сторонами и последующему разочарованию, когда не оправдываются иллюзорные, ничем не обоснованные ожидания.

В этой книге излагаются теоретические концепции, на основе которых сформирована технология блокчейна, в лаконичном и понятном стиле, рассчитанном на неподготовленных (с технической точки зрения) читателей. Книга отвечает на три главных вопроса, возникающих при знакомстве с любой новой технологией: что это такое? зачем это нужно мне? как это работает?

Чего не следует ждать от этой книги

В этой книге преднамеренно не рассматриваются приложения, использующие блокчейн. Несмотря на то что криптовалюты в целом и Bitcoin в частности являются основными приложениями на основе блокчейна, в книге блокчейн описывается как «технология вообще». Такой подход выбран для того, чтобы ярче выделить общие ключевые концепции и технические шаблоны блокчейна, а не ограничиваться более узкими специализированными частными случаями конкретных приложений. Таким образом:

- эта книга не о Bitcoin или какой-либо другой криптовалюте;
- эта книга не рассматривает какого-то одного специализированного блокчейн-приложения;
- в этой книге нет математических доказательств основных концепций блокчейна;
- эта книга не о программировании с использованием технологии блокчейна;
- в этой книге не обсуждаются последствия применения технологии блокчейна с точки зрения законодательства;
- в этой книге не рассматриваются социальное, экономическое и этическое воздействия технологии блокчейна на наше общество или на человечество в целом.

Тем не менее некоторые из этих тем в некоторой степени обсуждаются в соответствующих подразделах данной книги.

Чего следует ожидать от этой книги

Книга подробно описывает технические концепции технологии блокчейна, такие как транзакции, хэш-значения, криптография, структуры данных, пиринговые системы, распределенные системы, целостность систем и консенсус в распределенной среде в стиле, понятном для читателей с недостаточно высоким уровнем технической подготовки. Дидактический подход к изложению материала основан на четырех элементах:

- диалоговый («разговорный») стиль;
- отсутствие математических выкладок и формул;
- постепенное продвижение по проблемной области;
- использование метафор и аналогий.

Диалоговый («разговорный») стиль

Эта книга преднамеренно написана в диалоговом, или «разговорном», стиле. Здесь абсолютно не применяется математический и компьютерный жаргон, чтобы устранить все препятствия для читателей, не вполне подготовленных с технической точки зрения. Но здесь представлена и объяснена вся терминология, необходимая для участия в обсуждениях и для понимания других публикаций по теме блокчейна.

Отсутствие математических выкладок и формул

Главные элементы технологии блокчейна, такие как криптография и алгоритмы, основаны на сложных математических концепциях, которые, в свою очередь, предъявляют свои особые требования к их пониманию, а также приводят к необходимости изучения математических выкладок и формул, устрашающих на вид. И все же в книге намеренно не используются ни математические выкладки, ни формулы, чтобы избежать ненужной сложности и не создавать дополнительных затруднений для читателей с недостаточной технической подготовкой.

Постепенное продвижение по проблемной области

Главы в этой книге соответствуют своего рода шагам, или этапам, по вполне обоснованной причине. Такие шаги, или этапы, формируют процесс обучения, в котором последовательно, уровень за уровнем наращиваются знания о технологии блокчейна. Порядок этапов обучения был выбран с особой тщательностью. Они охватывают основы программной инженерии, подробно описывают терминологию, дают обоснование необходимости использования блокчейна и подробно рассматривают отдельные концепции, заложенные в основу технологии блокчейна, и взаимодействие ее составляющих. Строгая последовательность глав-этапов подчеркивает их взаимозависимость и дидактические цели. Тем самым обеспечивается

логически связанное изложение материала, а не набор отдельных глав, которые можно читать в любом порядке.

Использование метафор и аналогий

Каждая глава-этап, представляющая новую концепцию, начинается с образного описания ситуации из реальной жизни. Такие метафоры служат четырем основным целям. Во-первых, они готовят читателя к правильному восприятию новой технической концепции. Во-вторых, объединяя техническую концепцию с простой жизненной ситуацией, метафора устраняет психологический барьер при «исследовании новой территории». В-третьих, метафоры позволяют изучать новые концепции с помощью подобия и аналогий. Наконец, метафоры формируют простые практические правила для запоминания новых концепций без затруднений.

Как организована эта книга

Книга состоит из 25 глав-этапов, сгруппированных по пяти основным темам (частям), которые в совокупности формируют процесс обучения с постепенным наращиванием знаний о технологии блокчейна. В главах рассматриваются основы программной инженерии, объясняется необходимая терминология, обосновывается необходимость применения технологии блокчейна, описываются отдельные концепции, заложенные в основу этой технологии, а также взаимодействие между ее компонентами, рассматриваются приложения блокчейна и направления разработок и активных исследований в этой области.

Часть I: Терминология и основы технологии

В главах 1-3 рассматриваются основные концепции программной инженерии и группа терминов, необходимых для понимания последующих глав. К концу главы 3 вы получите общее представление об основных концепциях и общую картину области использования технологии блокчейна.

Часть II: Зачем нужна технология блокчейна

В главах 4-7 объясняется, зачем нужна технология блокчейна, какие задачи она решает, почему решение этих задач важно, а также описываются потенциальные возможности блокчейна. К концу главы 7 вы будете хорошо понимать проблемную область технологии блокчейна, среды, в которой применение блокчейна наиболее эффективно, и почему в этих областях применение блокчейна рассматривается в первую очередь.

Часть III: Как работает блокчейн

Третья часть является главной частью книги, поскольку подробно описывает внутреннее устройство и функционирование блокчейна. В главах 8-21 последовательно представлены 15 различных технических концепций, в совокупности составляющих основу технологии блокчейна. К концу главы 21 вы будете полностью понимать все основные концепции блокчейна, их функционирование по отдельности, а также их взаимодействие для создания крупного комплексного механизма, называемого блокчейн.

Часть VI: Ограничения и способы их преодоления

В главах 22 и 23 главное внимание уделено основным ограничениям технологии блокчейна, описываются их причины и кратко намечаются способы их преодоления. К концу главы 23 вы будете понимать, почему основополагающая идея технологии блокчейна, подробно описанная в предыдущих главах, может оказаться не подходящей для крупных коммерческих приложений с потенциальной возможностью масштабирования, какие изменения были внесены для преодоления этих ограничений и как эти изменения повлияли на свойства блокчейна.

Часть V: Использование технологии блокчейна, общие выводы и перспективы

В главах 24 и 25 рассматриваются возможные варианты практического применения технологии блокчейна в реальном мире, а также вопросы, на которые необходимо найти ответы при выборе блокчейн-приложения. В этой части также определяются области разработок и активных исследований технологии блокчейна. К концу главы 25 вы будете полностью понимать технологию блокчейна и обладать вполне достаточной подготовкой для чтения более сложных технических материалов и участия в постоянно продолжающихся обсуждениях технологии блокчейна.

Дополнительные материалы

Веб-сайт www.blockchain-basics.com предоставляет дополнительные материалы по темам некоторых глав данной книги.

Часть I

Терминология и основы технологии

В этой части описываются основные концепции программной инженерии, а также устанавливаются правила организации и стандартизации при обсуждении основ технологии. Этот этап обучения также представляет концепции программной архитектуры и целостности программного обеспечения, а еще их связь с технологией блокчейна. К концу этого этапа вы будете хорошо понимать цели и задачи технологии блокчейна и ее потенциальные возможности.

Глава 1

Понимание уровней и аспектов

Анализ систем посредством

разделения их на уровни и аспекты

Эта глава закладывает основу для дальнейшего процесса изучения технологии блокчейна, четко определяя правила и способы организации и стандартизации при обсуждении основ технологии. В главе рассматриваются возможные методики анализа программных систем, объясняется, почему важно рассматривать программную систему как совокупность уровней. Далее наглядно демонстрируется, какие преимущества можно извлечь при анализе различных уровней системы и как такой подход помогает понять технологию блокчейна. В конце главы приводится краткое вводное описание концепции целостности программного обеспечения и подчеркивается ее важность.

Метафора

У вас есть мобильный телефон? Я почти уверен, что есть, так как у подавляющего большинства людей имеется, по крайней мере, один мобильный телефон. Что вы знаете о разнообразных протоколах беспроводного обмена информацией, используемых для отправки и приема данных? Что вы знаете об электромагнитных волнах, являющихся основой мобильной связи? Большинству из нас не слишком много известно об этих технических подробностях, потому что такие знания не являются необходимыми для практического использования мобильного телефона. К тому же почти все мы настолько заняты, что у нас вряд ли найдется время на изучение этих тонкостей. Мысленно разделяя мобильный телефон на общеизвестные составные части, мы учитываем, что обязательное присутствие этих частей невозможно игнорировать или считать само собой разумеющимся.

Такой подход к технологии не ограничивается только мобильными телефонами. Мы используем его во всех случаях, когда приходится осваивать новый телевизор, компьютер, стиральную машину и т. п. Но такие «мысленные» составные части в высшей степени индивидуальны, так как каждый по-своему решает, что считать важным, а что не зависит от наших индивидуальных предпочтений, от конкретной технологии, от наших целей и практических знаний. В результате ваше мысленное разделение мобильного телефона на составные части может отличаться от моего разделения того же самого мобильного телефона. Обычно это приводит к проблемам при обмене информацией, особенно если я пытаюсь объяснить вам, что необходимо знать об устройстве конкретной модели мобильного телефона. Таким образом, единый универсальный подход к разделению системы на составляющие компоненты является ключевым моментом при изучении и обсуждении любой технологии. В этой главе описывается, как следует разделять систему на составные части или уровни и соответствующим образом формулировать основные положения при обсуждении технологии блокчейна.

Уровни программной системы

На протяжении всей книги при разделении любой системы на составные части используются следующие две методики:

- сопоставление приложения и его реализации;
- разделение на функциональные и нефункциональные аспекты.

Сопоставление приложения и его реализации

Мысленное отделение потребностей пользователя от технических подробностей внутреннего устройства системы приводит к разделению уровня приложения и уровня реализации. Все, принадлежащее к уровню приложения, рассматривается как потребности пользователя (например, прослушивание музыки, фотографирование, заказ номера в отеле и т. д.). Все, принадлежащее к уровню реализации, рассматривается с точки зрения обеспечения выполнения вышеперечисленных действий (например, преобразование цифровой информации в акустические сигналы, определение цвета пиксела в цифровой видеокамере или передача сообщения по сети Интернет в систему бронирования номеров отеля). Элементы уровня реализации являются техническими по своей сущности и рассматриваются как средства достижения той или иной цели.

Разделение на функциональные и нефункциональные аспекты

Различие между тем, что система делает и как она это делает, приводит к разделению функциональных и нефункциональных аспектов. Примерами функциональных аспектов являются: передача данных по сети, воспроизведение музыки, фотографирование и редактирование отдельных пикселов в изображении. Примеры нефункциональных аспектов: удобный графический пользовательский интерфейс, быстрое программное обеспечение, возможность безопасного хранения пользовательских данных и защита их приватности. Другими важными нефункциональными аспектами системы являются безопасность и целостность. Целостность (integrity) означает, что система ведет себя именно так, как от нее ожидают, в то же время понятие целостности включает в себя и многие другие аспекты, такие как, например, безопасность (защищенность) и корректность [8]. Эффективным способом запоминания различий между функциональными и нефункциональными аспектами системы является аналогия с грамматикой русского или английского языка: глаголы описывают действия (что делается), а наречия – как выполняются эти действия. Например, человек может идти быстро или медленно. В обоих случаях действие «идти» одинаково, но способы выполнения этого действия различны. Поэтому в качестве практического правила можно предложить аналогию: функциональные аспекты соответствуют глаголам, нефункциональные аспекты соответствуют наречиям.

Одновременное изучение двух уровней

Определение функциональных и нефункциональных аспектов и разделение на уровень приложения и уровень реализации можно выполнять одновременно, получая в результате двумерную таблицу. В табл. 1.1 показан результат мысленного разделения на уровни системы «мобильный телефон» с одновременным определением функциональных и нефункциональных аспектов.

Таблица 1.1 Пример мысленного разделения на уровни мобильного телефона

Уровень	Функциональные аспекты	Нефункциональные аспекты
Приложения	Фотографирование Телефонные вызовы Отправка сообщений электронной почты Навигация по Интернету Отправка сообщений в чаты	Графический пользовательский интерфейс выглядит привлекательно Удобство пользования Сообщения отправляются очень быстро
Реализации	Внутренний механизм сохранения пользовательских данных Установка соединения с ближайшим узлом мобильной связи Возможность доступа к отдельным пикселям в цифровой фото(видео) камере	Эффективное хранение данных Экономия энергии Обеспечение целостности Защита приватности пользователя

Таблица 1.1 может описывать видимость (или невидимость) конкретных элементов системы для ее пользователей. Функциональные аспекты уровня приложения в большинстве своем являются видимыми элементами системы, поскольку предназначены для удовлетворения очевидных потребностей пользователей. Эти элементы обычно хорошо знакомы пользователям. С другой стороны, нефункциональные аспекты уровня реализации редко проявляют себя как основные элементы системы. Их наличие считается само собой разумеющимся.

Целостность

Целостность (integrity) – это важный нефункциональный аспект любой программной системы. Понятие целостности включает три главных компонента [5]:

- целостность данных (data integrity): данные, используемые и сопровождаемые системой, должны быть полными, корректными и непротиворечивыми;
- целостность поведения (behavioral integrity): система ведет себя, как предполагается, и не допускает логических ошибок;
- безопасность (защита) (security): система способна ограничить доступ к своим данным и функциональным возможностям, разрешая его только авторизованным пользователям.

Возможно, большинство людей считает целостность программных систем фактом, не требующим подтверждения, потому что большую часть времени имеет дело с системами, сохраняющими свою целостность. Это становится возможным благодаря тому, что программисты и инженеры затратили огромное количество времени и усилий на разработку систем, обеспечивающих собственную целостность. Иногда возможна не совсем верная оценка труда инженеров по созданию систем, обеспечивающих высокий уровень целостности. Но наше мнение может измениться, как только мы встретимся с системой, не обладающей этим свойством. Это могут быть случаи потери данных, необъяснимого поведения программного обеспечения или обнаружения факта доступа посторонних лиц к вашим личным закрытым данным. Это ситуации, когда ваш мобильный телефон, компьютер, программа электронной почты, текстовый процессор или электронная таблица заставляет вас разозлиться и забыть о хороших манерах. Во всех подобных случаях мы действительно начинаем понимать, насколько важным аспектом является целостность программного обеспечения. Поэтому не должно вызывать удивления то обстоятельство, что профессиональные разработчики программного обеспечения затрачивают огромное количество времени на кажущийся незначительным аспект уровня реализации.

Перспектива

В этой главе представлена вводная информация о некоторых общих принципах программной инженерии. Здесь рассматривались концепции целостности, функциональные и нефункциональные аспекты, уровни приложения и реализации программной системы. Понимание этих концепций поможет вам более широко взглянуть на среду, в которой существует технология блокчейна. В следующей главе будет представлена более подробная картина применения концепций, описанных в данной главе.

Резюме

- Анализ систем может выполняться с помощью разделения:
 - на уровень приложения и уровень реализации;
 - на функциональные и нефункциональные аспекты.
- Уровень приложения сосредоточен на потребностях пользователя, уровень реализации – на способах удовлетворения этих потребностей.
- Функциональные аспекты определяют, что делать, нефункциональные аспекты определяют, как это делать.
- Большинству пользователей хорошо известны функциональные аспекты уровня приложения системы, в то время как нефункциональные аспекты системы, особенно относящиеся к уровню реализации, практически невидимы для пользователя.
- Целостность является важным нефункциональным аспектом любой программной системы и включает три главных элемента:
 - целостность данных;
 - целостность поведения;
 - безопасность (защита).
- Большинство критических сбоев программного обеспечения, таких как потери данных, необъяснимое поведение, доступ посторонних лиц к личным закрытым данным, является результатом нарушения целостности системы.

Глава 2

Более подробная картина

Архитектура программного обеспечения

и ее связь с технологией блокчейна

В этой главе не только представлена более подробная картина среды, в которой существует технология блокчейна, но и более точно определяется место расположения этой технологии в общей картине. Чтобы лучше увидеть все это, в главе вводится концепция архитектуры программного обеспечения и объясняется ее связь с концепцией разделения системы на уровни и аспекты. Чтобы помочь вам в определении места расположения технологии блокчейна в общей картине, здесь особо отмечаются взаимосвязи между технологией блокчейна и архитектурой программного обеспечения. В конце главы главная цель технологии блокчейна формулируется буквально в одном предложении. Правильное восприятие этой главной цели является важнейшей основой понимания всей технологии блокчейна и содержимого всех последующих глав.

Метафора

У вас есть автомобиль? Большинство людей имеет автомобили. Даже если вы никогда не покупали машину, вероятнее всего, вы все же знаете, что машины оснащены различными типами двигателей (например, дизельными, бензиновыми или электрическими). Это пример применения принципа модульности как результата реализации идеи разделения на уровни для автомобилей. Возможность выбора одного из нескольких типов двигателя при покупке машины может привести к значительным различиям функциональных возможностей транспортных средств.

Две машины, внешне выглядящие абсолютно одинаково, могут совершенно отличаться по мощности их двигателей, следовательно, обеспечивать различные скорости вождения. Кроме того, выбор двигателя повлияет и на другие характеристики автомобиля, такие как цена, стоимость техобслуживания, тип потребляемого топлива, система отвода выхлопных газов, размеры тормозных колодок. Мысленно представляя схему этих взаимосвязей, можно гораздо быстрее понять роль технологии блокчейна в общей картине программной среды.

Платежная система

Применим концепцию разделения на уровни к платежной системе. В табл. 2.1 показаны некоторые потребности пользователей и некоторые нефункциональные аспекты на уровне приложения и на уровне реализации.

Таблица 2.1 Аспекты и уровни платежной системы

Уровень	Функциональные аспекты	Нефункциональные аспекты
Приложения	Денежный взнос Снятие денег Перевод денег Контроль баланса счета	Графический пользовательский интерфейс выглядит привлекательно Легкость использования Перевод денег выполняется быстро Систему используют многие люди
Реализации	?	Доступность 24 часа в сутки Защита от мошеннических действий Обеспечение целостности Гарантия приватности для пользователей

Вы обратили внимание на знак вопроса в той части таблицы, где обычно находится информация о технологии, обеспечивающей работу системы? Часть таблицы, помеченная знаком вопроса, оставлена незаполненной с определенной целью. Это то место, где вы решаете, какой «двигатель» должен использоваться, чтобы система работала. В следующем разделе более подробно рассматривается аналог двигателя в программных системах.

Два типа архитектуры программного обеспечения

Существует много способов реализации программных систем. Но одним из основных решений в процессе реализации системы становится определение ее архитектуры, то есть схемы организации ее компонентов и взаимосвязей между ними. Двумя основными типами архитектуры программных систем являются централизованная и распределенная [32].

В централизованных программных системах выделяется один центральный компонент, с которым соединяются отдельные периферийные компоненты. В противоположность такой схеме компоненты распределенных систем формируют сеть взаимосвязанных элементов без выделения какого-либо центрального элемента с функциями координации и управления.

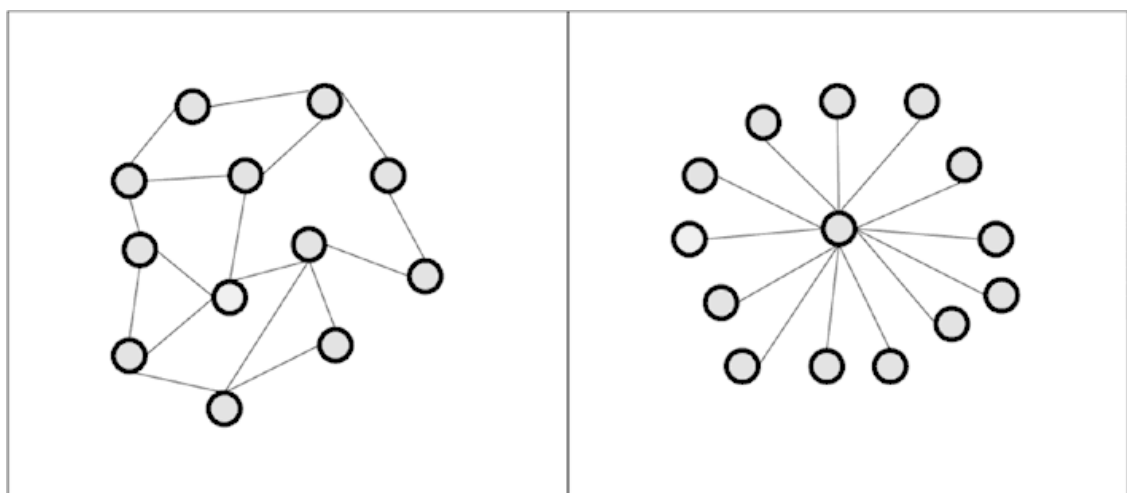


Рис. 2.1 Распределенная (слева) и централизованная (справа) архитектуры системы

На рис. 2.1 схематично изображены эти две противоположные архитектуры. Закрашенные серым цветом кружки представляют компоненты системы, также называемые узлами (nodes), а линии обозначают связи между ними. В этот момент несущественны подробности,

касающиеся того, что эти компоненты делают и какая информация передается между узлами. Здесь самым важным является сам факт существования двух различных способов организации программных систем. В левой части рис. 2.1 распределенная архитектура демонстрирует, как соединяются друг с другом компоненты без какого-либо центрального элемента. Важно понять, что в этой схеме нет компонентов, напрямую связанных со всеми прочими компонентами. Но при этом все компоненты взаимосвязаны друг с другом, по крайней мере, не напрямую. В правой части рис. 2.1 показана централизованная архитектура, в которой каждый компонент связан с одним центральным компонентом. Периферийные компоненты не имеют прямых связей друг с другом. Для каждого периферийного компонента существует единственная прямая связь с центральным компонентом.

Преимущества распределенных систем

Ниже перечислены основные преимущества распределенной системы по сравнению с отдельными компьютерами [32]:

- более высокая вычислительная мощность;
- снижение стоимости (накладных расходов, издержек);
- более высокая надежность;
- возможность естественного роста.

Более высокая вычислительная мощность

Вычислительная мощность распределенной системы определяется как сумма объединенных вычислительных мощностей всех компьютеров, входящих в состав такой системы. Таким образом, распределенные системы обычно обладают более высокой вычислительной мощностью, чем каждый компьютер в отдельности. Этот факт подтверждается даже при сравнении распределенных систем, состоящих из компьютеров с относительно низкой вычислительной мощностью, с отдельными суперкомпьютерами.

Снижение стоимости (накладных расходов, издержек)

Цены на обычные типовые компьютеры, устройства памяти, дисковые накопители и сетевое оборудование постоянно и очень быстро снижаются в течение последних 20 лет. Поскольку распределенные системы состоят из множества компьютеров, первоначальная стоимость распределенных систем выше, чем первоначальная стоимость отдельного компьютера. Но затраты на создание, обслуживание и поддержку функционирования суперкомпьютера остаются гораздо более высокими, чем затраты на создание, сопровождение и обеспечение функционирования распределенной системы. Наиболее ярким подтверждением этого факта является отсутствие какого-либо заметного воздействия на систему в целом при замене отдельных компьютеров в распределенной среде.

Более высокая надежность

Повышенная надежность распределенной системы основана на том факте, что сеть компьютеров как единое целое способна продолжать работу даже при выходе из строя отдельных машин, составляющих ее. В распределенной системе нет так называемой единой точки отказа (single point of failure – SPOF). При отказе одного элемента все прочие элементы продолжают работу. Таким образом, отдельный суперкомпьютер обладает меньшей надежностью, чем распределенная система.

Возможность естественного роста

Вычислительная мощность распределенной системы определяется как сумма объединенных вычислительных мощностей всех ее компонентов. Вычислительную мощность всей системы можно увеличить, просто подключив к ней дополнительные компьютеры. Следовательно, вычислительную мощность распределенной системы можно регулировать с достаточно высокой точностью, постепенно наращивая ее. Такая методика поддерживает запросы многих организаций, которым требуется постоянное увеличение вычислительных мощностей. Постепенный рост распределенной системы абсолютно отличается от наращивания мощностей отдельных компьютеров. Вычислительная мощность отдельного компьютера остается неизменной до тех пор, пока он не будет заменен на более мощный компьютер. Такое дискретное повышение вычислительной мощности далеко не всегда является приемлемым вариантом для потребителей разнообразных современных сервисов.

Недостатки распределенных систем

Ниже перечислены недостатки распределенных систем по сравнению с отдельными компьютерами:

- издержки на координацию работы;
- издержки на организацию обмена информацией;
- зависимость от сетевой среды;
- более высокая сложность программного обеспечения;
- проблемы безопасности.

Издержки на координацию работы

В распределенных системах нет центральных объектов, которые координируют работу прочих объектов. То есть координацию осуществляют сами элементы системы. Координация совместно работающих компонентов распределенной системы представляет собой трудную задачу и требует существенных трудовых затрат и вычислительных мощностей, отвлекаемых от выполнения основной задачи. Это и называется издержками, или накладными расходами на координацию работы.

Издержки на организацию обмена информацией

Для осуществления координации необходим обмен информацией. Следовательно, компьютеры, составляющие распределенную систему, должны обмениваться информацией друг с другом. Для этого обязательно требуется наличие коммуникационного протокола, а также средства отправки, приема и обработки сообщений, что также требует трудовых затрат и вычислительных мощностей, отвлекаемых от выполнения основной задачи. Таким образом, и здесь возникают издержки (накладные расходы) на организацию обмена информацией.

Зависимость от сетевой среды

Для любого вида обмена информацией требуется носитель. Носитель отвечает за передачу информации между объектами, ведущими диалог. Компьютеры в распределенных системах обмениваются информацией в виде сообщений, передаваемых по сети. Сети обладают собственными недостатками и создают дополнительные затруднения, в свою очередь отрицательно воздействуя на координацию и обмен информацией между компьютерами распределенной системы. Но без сети распределенную систему создать невозможно, так как отсутствует обмен информацией, следовательно, нет средств координации узлов. Таким образом, распределенная система зависит от сетевой среды.

Более высокая сложность программного обеспечения

Для выполнения вычислительных (и прочих) задач требуется написание отдельных программ и комплексного программного обеспечения. Из-за проблем, описанных выше, программное обеспечение для распределенных систем обязательно должно обеспечивать решение вспомогательных задач, таких как координация, обмен информацией и поддержка работы в сетевой среде. Это увеличивает сложность программного обеспечения.

Проблемы безопасности

Обмен информацией в сетевой среде означает передачу и совместное использование данных, чрезвычайно важных для выполнения основной задачи. Но передача информации в сети создает проблемы защиты данных, так как ненадежные посторонние объекты могут воспользоваться сетью для несанкционированного доступа к приватной информации и нелегального ее использования. Следовательно, любая распределенная система непременно должна обеспечивать защиту информации. Чем меньше ограничений доступа к сети, по которой осуществляется обмен информацией между распределенными узлами, тем больше угроз для безопасности распределенной системы.

Распределенные пиринговые системы

Пиринговые (peer-to-peer), или одноранговые, сети представляют собой особый тип распределенных систем. Они состоят из отдельных компьютеров (также называемых узлами), вычислительные ресурсы которых (например, все средства обработки данных, емкость внешних накопителей, хранимые данные, пропускная способность сети и т. д.) напрямую доступны всем прочим узлам этой сети без какого-либо центрального пункта координации. Все узлы такой сети имеют равные права и одинаковые роли в системе. Более того, все они являются как поставщиками, так и потребителями ресурсов.

Для пиринговых систем существуют весьма полезные способы применения, например совместное использование файлов, распространение контента, защита частной секретной информации. Большинство этих приложений использует простую, но мощную идею: превращение обычных пользовательских компьютеров в узлы, формирующие единую распределенную систему. В результате чем больше пользователей или клиентов-покупателей использует такую программную среду, тем больше и мощнее становится система в целом. Эта идея, ее развитие, трудности, связанные с ее реализацией, будут более подробно рассматриваться в следующих главах.

Объединение централизованных и распределенных систем

Централизованные и распределенные системы являются полными противоположностями с точки зрения архитектуры. Технические противоположности, казалось бы, не совместимы друг с другом, всегда вдохновляли инженеров на создание объединенных, гибридных систем, наследующих все сильные стороны своих предков. В этом плане централизованные и распределенные системы не являются исключением. Существуют два основных прототипа объединения противоположных архитектур, и их необходимо хорошо понимать, потому что сущность подобного объединения чрезвычайно важна для изучения функционирования блокчейн-приложений в реальном мире. Первый прототип – это центральное положение одного из узлов в распределенной системе, второй – распределенная система как управляющий узел централизованной системы.

Схема в левой части рис. 2.2 изображает архитектуру, в которой определен центральный компонент в распределенной системе. На первый взгляд кажется, что это обычная распределенная система с обычными компонентами. Но если приглядеться, то выясняется, что

все кружки-узлы соединены с одним кружком большего размера, расположенным в центре схемы. Таким образом, подобная система только кажется распределенной при поверхностном осмотре, но в действительности это централизованная система.

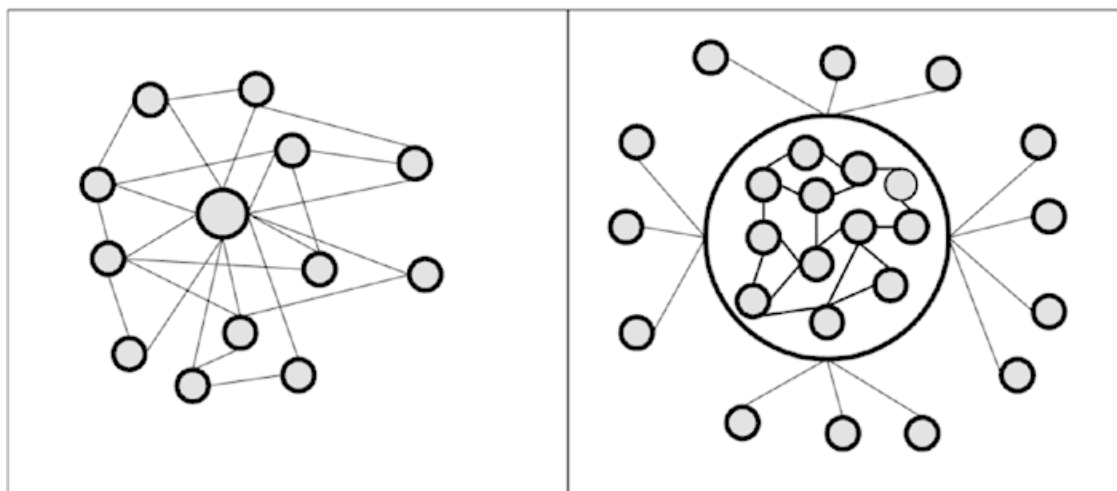


Рис. 2.2 Прототипы объединения распределенной и централизованной архитектур

Схема в правой части рис. 2.2 демонстрирует противоположный подход. Система выглядит централизованной на первый взгляд, поскольку все периферийные кружки-узлы имеют только одно соединение с большим центральным компонентом. Но внутри центрального компонента размещена распределенная система. При этом периферийные компоненты могут даже не знать о существовании распределенной системы внутри центрального узла.

Описанные выше подходы объединяет затруднение при определении их истинной сущности. Эти системы являются распределенными или централизованными? Возможно, нет необходимости в однозначном определении названий таких архитектур. Здесь гораздо важнее понять их двойственную сущность, потому что далеко не всегда можно с легкостью увидеть признаки централизованности или распределенности в сложных системах. К этой теме я вернусь позже, при обсуждении способа коммерциализации технологии блокчейна.

Идентификация распределенных систем

Появление смешанных архитектур затрудняет однозначную идентификацию распределенных систем. Впрочем, формулирование общего приемлемого определения распределенных систем выходит за рамки тематики данной книги. Но при изучении книги важно четко понимать, что такое распределенная система и чем она отличается от других программных систем. Если вы сомневаетесь, является ли рассматриваемая конкретная система распределенной, то найдите в ней отдельный компонент (например, базу данных, имя или пользовательскую запись в реестре, компонент регистрации в системе или выхода из нее, кнопку аварийного выключения, наконец), который может завершить работу всей системы в целом. Если такой компонент найден, то рассматриваемая система не является распределенной.

Примечание

Если существует отдельный компонент, например кнопка выключения, который может остановить работу всей системы в целом, то такая система не является распределенной.

Цель технологии блокчейна

При проектировании программной системы можно выбрать используемый тип архитектуры, подобно тому, как выбирается двигатель для автомобиля. Решение по выбору архитектуры может быть принято независимо от функциональных аспектов уровня приложения. Кто-то может создать распределенную систему, кто-то – централизованную систему, но с одинаковой функциональностью на уровне приложения. Архитектура является всего лишь одним из средств достижения конечной цели при реализации системы. Таким образом, платежная система, описанная в табл. 2.1, может быть реализована либо как распределенная, либо как централизованная система.

Каждая из двух описанных выше архитектурных концепций обладает собственными достоинствами и недостатками, а также своими особыми способами достижения конечной цели. Выбор конкретной архитектуры определяет в дальнейшем, как вы будете реализовывать функциональные и нефункциональные аспекты системы. В частности, эти архитектурные концепции предлагают совершенно различные подходы к обеспечению целостности. Это как раз тот момент, когда на сцене появляется блокчейн. Технология блокчейна (blockchain) (цепочка блоков транзакций) представляет собой инструментальное средство обеспечения целостности распределенных программных систем. Таким образом, блокчейн можно рассматривать как инструмент реализации нефункционального аспекта на уровне реализации.

Примечание

Главная цель технологии блокчейна – реализация и поддержка целостности в распределенных системах.

Перспектива

Обеспечение целостности в распределенной системе является исключительно технической задачей, поэтому ее изучение может показаться достаточно скучным занятием. Тем не менее это вопрос чрезвычайной важности для многих людей, вопрос, который зависит от того, что именно будет делать распределенная система и какой тип централизованной системы она заменяет. В следующей главе вы узнаете, как пиринговые системы изменили наш мир и почему блокчейн как инструмент обеспечения целостности в распределенных программных системах также способен изменить весь мир.

Резюме

- Архитектура программной системы определяет, как организованы ее компоненты и как они связаны друг с другом.
- Централизованная и распределенная программные архитектуры могут считаться полностью противоположными друг другу.
- Распределенная система состоит из некоторого количества независимых компьютеров, которые взаимодействуют друг с другом, используя среду обмена информацией для достижения определенной цели, без какого-либо центрального управляющего или координирующего элемента.
- Практическое правило: если в системе имеется отдельный элемент, способный полностью остановить ее работу, то можно с уверенностью утверждать, что такая система не является распределенной, вне зависимости от сложности ее архитектуры.

- Технология блокчейна является частью уровня реализации распределенной программной системы.
- Цель технологии блокчейна – реализация конкретного нефункционального аспекта распределенной программной системы, а именно реализация и поддержка целостности системы.

Глава 3

Определение потенциальных возможностей *Как пиринговые системы могут изменить мир*

В этой главе мы углубляем понимание главной цели технологии блокчейна, рассматривая конкретный тип распределенной системы: пиринговую систему. Это поможет лучше понять, почему блокчейн вызывает столь высокий интерес у технических специалистов и бизнес-профессионалов. Здесь также описана область приложений, в которой от блокчейна ожидается наибольшая отдача. Кроме того, в этой главе оцениваются некоторые последствия практического применения пиринговых систем.

Метафора

Вы можете вспомнить, когда в последний раз покупали компактдиск (CD) в музыкальном магазине или в универсальном торговом центре? В наши дни люди уже давно не покупают компакт-дисков, потому что музыкальная отрасль изменилась коренным образом. Сейчас люди скачивают отдельные композиции с музыкальных порталов, делятся файлами в формате mp3 с друзьями или используют потоковые музыкальные программы на мобильных устройствах вместо покупки компакт-дисков. Эти изменения начались вместе с появлением программных средств, которые позволили делиться музыкальными файлами друг с другом. Но что такое особенное предлагают подобные программы? Вот что сказал по этому поводу один из создателей нового направления в музыкальной индустрии:

«Самым интересным в этой системе является то, что вы взаимодействуете с равными партнерами, вы обмениваетесь информацией с такими же людьми, как вы сами».

Шон Фэннинг (Shawn Fanning), сооснователь пиринговой сети Napster

Фэннинг и его коллеги изобрели пиринговую (peer-to-peer), или одноранговую, систему для совместного использования музыкальных файлов. В конце 1990-х гг. это программное обеспечение открыло новый путь формирования бизнес-модели в музыкальной индустрии. Далее в этой главе подробно рассматривается, как появление сети Napster, снижение продаж компакт-дисков и коренные изменения в музыкальной индустрии повлияли на технологию блокчейна.

Как пиринговая система изменила целую отрасль промышленности

В течение долгих лет музыкальная индустрия работала по следующей схеме: музыканты заключали контракты со студиями, которые выполняли записи композиций, переносили записи на различные виды носителей (винил, магнитная лента или компактдиск) и приводили их в товарный вид, затем товарные экземпляры носителей продавались потребителям по разнообразным каналам, включая универсальные торговые центры и специализированные музыкальные магазины. В действительности студии звукозаписи выступали как посредники между музыкантами и любителями музыки. Студии звукозаписи способны были выполнять функции посредников благодаря своим особым знаниям и практическому опыту в продюсировании, маркетинге и распространении музыкальных записей. Но в первом десятилетии XXI века среда, в которой действовали студии звукозаписи, изменилась коренным образом.

Возможность оцифровки музыкальных записей, доступность записывающего оборудования по приемлемым ценам, бурный рост количества персональных компьютеров, находящихся в частном владении, появление и быстрое развитие Интернета – все это привело к тому, что студии звукозаписи перестали быть обязательным компонентом музыкальной индустрии. Три основные функции студий – продюсирование, маркетинг, распространение – теперь могли выполнять сами музыканты и потребители. Сеть Napster сыграла главную роль в устранении студий звукозаписи как посредников. Пользуясь Napster, люди перестали считать студии звукозаписи единственным источником доступа к самым свежим хитам. Появилась возможность совместного использования отдельных музыкальных файлов людьми по всему миру без необходимости покупки каких-либо компакт-дисков. Методика одноранговой сети, реализованная в Napster, в действительности стала разновидностью общедоступного цифрового огромного торгового зала для файлов в формате mp3, предоставляющего потребителям доступ к более широкому ассортименту музыки, чем когда-либо ранее. При этом студии звукозаписи оказались в определенной степени не у дел и понесли значительные убытки [15, 22].

Потенциальные возможности пиринговых систем

Успех сети Napster наглядно показал, что пиринговые системы обладают потенциальными возможностями, способными изменять целые отрасли промышленности, и основаны на простой идее: замена посредника на прямое взаимодействие между равными партнерами. В случае с музыкальной индустрией давно существующие студии звукозаписи и их каналы маркетинга и распространения, действующие как посредники между музыкантами и слушателями, были заменены пиринговыми системами совместного использования файлов. Основными характеристиками, которые сделали музыкальную индустрию столь уязвимой и способствовали распространению пиринговых систем, являлись нематериальная сущность музыки и низкие накладные расходы на копирование и передачу данных.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

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

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