Стандартизация смарт-контрактов

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 August 2022, accepted 19 August 2022, date of publication 29 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202550

RESEARCH ARTICLE

Standardizing Smart Contracts

VITTORIO CAPOCASALE[1], (Member, IEEE), AND GUIDO PERBOLI[2], (Member, IEEE)
[1]Department of Control and Computer Engineering, Polytechnic University of Turin, 10129 Turin, Italy
[2]Department of Management and Production Engineering, Polytechnic University of Turin, 10129 Turin, Italy
Corresponding author: Vittorio Capocasale (vittorio.capocasale@polito.it)

This work was supported by the Project Cyber security cOmpeteNCe fOr Research anD InnovAtion (CONCORDIA) of the European Union (EU) Commission, under Grant 830927.

V. Capocasale and G. Perboli, "Standardizing Smart Contracts," in *IEEE Access*, vol. 10, pp. 91203-91212, 2022, doi: 10.1109/ACCESS.2022.3202550. https://ieeexplore.ieee.org/abstract/document/9869650

Абстракт

Разделы документа

I.

Введение

II.

Фон

III.

Смарт-контракты: заблуждения и рекомендации

IV.

Вывод

Авторы

использованная литература

Абстракт:

В развивающемся контексте технологий распределенного реестра необходима стандартизация смарт-контрактов. Смарт-контракты — это защищенные от несанкционированного доступа компьютерные программы. Благодаря их безопасности и гибкости смарт-контракты можно использовать в самых разных случаях. В частности, можно было бы автоматизировать юридически признанные контракты, используя смарт-контракты. В связи с этим растут некоторые стандарты надлежащего управления смарт-контрактами. Однако до сих пор существует множество технологических заблуждений относительно смарт-контрактов. Это исследование

описывает смарт-контракты с разных точек зрения, а также выявляет и разъясняет некоторые из наиболее распространенных заблуждений относительно смарт-контрактов. В этом исследовании также представлены некоторые рекомендации и сведения о надлежащем управлении смарт-контрактами. Это исследование может стать ценным ресурсом для будущих стандартов смарт-контрактов.

РАЗДЕЛ I. Введение

Технология блокчейн меняет мир. Децентрализованные версии хорошо зарекомендовавших себя централизованных услуг стремительно развиваются во многих областях, таких как финансы [1], страхование [2], логистика [3], [4], энергетика [5], [6] и т. физический мир к новым границам. Цифровые токены получают все большее распространение: NFT и обмениваемые монеты увеличивают свою долю на рынке [7], [8], поскольку они позволяют токенизировать и торговать даже исторически неликвидными активами (например, мощностью майнинга [9]).

Все эти преобразования в основном возможны благодаря гибкости смарт-контрактов [10]. Смарт-контракты представляют собой защищенные от несанкционированного доступа компьютерные программы, поскольку безопасность технологии блокчейн гарантирует правильность их исполнения. Таким образом, смарт-контракты могут автоматизировать и повысить справедливость критических процессов, гарантировать качество источников данных и защитить ценные ресурсы, которые представляют интерес [11], [12], [13]. В частности, очень привлекательна идея автоматизации юридически признанных договоров [14].

Однако создать защищенные от несанкционированного доступа, безопасные, децентрализованные, экономически выгодные и юридически признанные смарт-контракты непросто: необходимо учитывать различные аспекты, как технические, так и юридические [15]. В частности, смарт-контракты редко используются как самостоятельная технология. В логистике, например, дроны автоматизируют полевые операции [16], [17], а устройства IoT собирают полевые данные для обработки с помощью подходов искусственного интеллекта, стохастического программирования и гранулярных вычислений [4], [18]. [19], [20]. Таким образом, чтобы гарантировать, что смарт-контракты используют достаточно децентрализованные потоки данных, один продукт должен отслеживаться несколькими

устройствами IoT. Тем не менее, этот подход часто неудобен из-за экономических или физических ограничений. Кроме того, обработка данных из избыточных источников увеличивает сложность алгоритмов обработки данных без повышения их точности. В целом, создание всеобъемлющей структуры смарт-контрактов является сложной задачей, поскольку необходимо одновременно решать несколько задач. Таким образом, стандарты появились в литературе совсем недавно [21].

Неудивительно, что в отношении смарт-контрактов до сих пор существует много неправильных представлений. Смарт-контракты сложно сформулировать даже с чисто технической точки зрения: многие блокчейн-платформы предоставляют функциональные возможности, связанные со смарт-контрактами, но с разными стратегиями развертывания и выполнения, которые трудно включить в единое определение. Сочетание этого разнообразия с растущим интересом к теме, особенно со стороны людей с разным опытом и приоритетами, создало идеальную среду для роста некоторых недоразумений и частичной правды. Само название смарт-контракта вводит в заблуждение, поскольку оно намекает на юридически признанные цифровые соглашения, а не на компьютерные программы общего назначения. Более того, многие свойства смарт-контрактов зависят от конфигурации и степени децентрализации базовой системы блокчейна, что часто делает попытки обобщения неточными. В таком контексте создание n правильных определений, стандартов, руководств и лучших практик для смарт-контрактов становится насущной необходимостью. Кроме того, учитывая широкую аудиторию, заинтересованную в теме, смарт-контракты должны обсуждаться точно, корректно и широко понятно, что делает задачу еще более сложной.

Исследование направлено на создание общей основы для будущих стандартов смарт-контрактов. Это исследование может быть полезно исследователям, лицам, принимающим решения, юристам и специалистам по информатике в понимании потенциала и предостережений, связанных со смарт-контрактами. В частности, это исследование предоставляет следующие материалы:

он описывает смарт-контракты с разных точек зрения, чтобы предоставить четкий обзор даже нетехническим читателям;

он выявляет и исправляет некоторые распространенные заблуждения, связанные со смарт-контрактами;

в нем представлены некоторые рекомендации по надлежащему внедрению и исполнению смарт-контрактов, чтобы подчеркнуть потенциал, компромиссы и ограничения технологии. Цель предлагаемых руководящих принципов состоит в том, чтобы установить цели, к которым должны стремиться будущие стандарты, а не в описании каких-то строгих правил. Таким образом, мы ожидаем, что будущие стандарты будут максимально соответствовать предложенным рекомендациям, если это разумно.

Оставшаяся часть этого исследования структурирована следующим образом: Раздел II кратко описывает концепцию смарт-контракта, технологию блокчейна и представляет обзор литературы; Раздел III анализирует смарт-контракты; Раздел IV завершает исследование.

РАЗДЕЛ II. Предыстория

В этом разделе обобщены основные концепции, связанные с блокчейном и смарт-контрактами. Кроме того, в этот раздел включен краткий обзор литературы.

A. Блокчейн

Блокчейн — это технология, позволяющая обмениваться данными между недоверчивыми сторонами [22], поскольку она позволяет решать вопросы доверия между недоверчивыми сторонами без использования каких-либо доверенных третьих сторон [23]. Блокчейн состоит из сети узлов, которые имеют общую базу данных [24]. Общая база данных имеет структуру леджера: в нее можно только добавлять данные. Каждый узел имеет свою копию реестра, и каждый узел управляет своей копией независимо от других узлов. Следовательно, хотя каждый узел может произвольно изменять свою копию, глобальное состояние леджера устанавливается на основе того, что хранится в большинстве копий [25]. Таким образом, состояние системы блокчейна обновляется на основе голосования большинства. В этой работе мы предполагаем равномерное распределение права голоса между узлами, чтобы упростить обсуждение. Однако, когда мы говорим о большинстве одноранговых узлов, мы на самом деле имеем в виду большинство голосов.

B. Смарт-контракты

Смарт-контракты не привязаны к технологии блокчейн, и их первоначальное определение касалось автоматизации юридических контрактов [26]. Однако в контексте блокчейна (и других технологий распределенного реестра) смарт-контракты приобрели другое значение: они представляют собой защищенные от несанкционированного доступа компьютерные программы, которые обновляют состояние реестра [27]. На самом деле, запуская код смарт-контракта на нескольких узлах, вероятность успешного изменения его исполнения ничтожно мала. Более того, смарт-контракты могут выполнять произвольную логику, что делает их пригодными для автоматизации задач различного характера, выражая условия выполнения и реагируя на события (генерируемые пользователями или другими смарт-контрактами). Это исследование ограничит свой анализ смарт-контрактами, которые используются в системах блокчейна. Раздел III подробно анализирует смарт-контракты.

C. Постановка задачи

В настоящее время исследования смарт-контрактов быстро развиваются во многих областях. Исследователи изо всех сил пытаются понять друг друга из-за их разного опыта и точек зрения, включая информатику, экономику и право. Часто важными деталями пренебрегают, чтобы обеспечить понятное описание темы смарт-контракта, что создало идеальную среду для роста нескольких недоразумений и частичной правды. Такие ограничения затрагивают даже новые стандарты смарт-контрактов.

В этом исследовании выявляются некоторые из наиболее распространенных заблуждений и дается общее описание смарт-контрактов. Мы обсуждаем эту тему, не упуская из виду необходимые технические детали, что позволяет нам предоставить некоторые рекомендации, которые могут быть полезны как техническим, так и нетехническим читателям, чтобы демистифицировать технологию и понять ее ограничения и то, где она может найти применение.

D. Обзор литературы

Многие авторы имели дело со смарт-контрактами в литературе, особенно в последнее десятилетие. Тем не менее, это исследование анализирует тему с уникальной точки зрения, поскольку оно дает как практические рекомендации, так и философские интерпретации.

Смарт-контракты были введены для оцифровки и автоматизации юридических контрактов [26]. Затем этот термин был принят в контексте блокчейна для обозначения сценариев кода, выполняемых узлами сети блокчейна [27]. Таким образом, смарт-контракты определяют две разные концепции [15], [28]. В частности, в исследовании проводится различие между кодом смарт-контракта (который исполняется в блокчейн-системах) и смарт-юридические контракты, которые представляют собой юридические контракты в цифровой форме [15]. В исследовании смарт-контракты анализировались в основном с юридической точки зрения и подчеркивалось, что код смарт-контракта является лишь частью смарт-юридических контрактов [15]. В другом исследовании утверждается, что использование смарт-контрактов на основе блокчейна имеет лишь небольшие преимущества с практической точки зрения [29], поскольку многие термины, обозначающие свойства смарт-контрактов на основе блокчейна, часто вводят в заблуждение и неприменимы в юридическом контексте [29].

Такие двусмысленности и заблуждения значительно замедлили создание стандартов для смарт-контрактов [30]. Некоторые авторы определили основные проблемы и ограничения в автоматизации контрактов с юридической точки зрения (например, определение объема смарт-контрактов, их интернационализация, их применимость и их действительность) [31]. Другие представили основные требования, которым должны соответствовать смарт-юридические соглашения, сосредоточив внимание на преодолении разрыва между смарт-юридическими контрактами и кодом смарт-контракта и оптимизации такого процесса путем пересмотра существующих стандартов [32]. В других исследованиях изложены ключевые параметры, которые следует учитывать при обеспечении юридического признания смарт-контрактов [33]. В частности, необходимо определение универсальных API, стандартов кодирования и механизмов разрешения конфликтов [34]. Вместо этого эта работа направлена на то, чтобы демистифицировать и стандартизировать смарт-контракты на основе блокчейна с точки зрения информатики, уделяя особое внимание последствиям представленных рекомендаций на уровне приложений.

Некоторые авторы определили некоторые из основных характеристик смарт-контрактов (например, прозрачность, доступность и неизменность) [35], другие предложили новые конструкции для устранения некоторых существующих недостатков (например, зависимость от порядка транзакций, детерминизм и управление исключениями) [35]. 36], а полные стандарты можно найти в «серой» литературе [21]. Однако такие работы смещены в сторону видения смарт-контрактов, представленных протоколом Ethereum [27]. Более того, некоторые рекомендации,

предложенные в [21], следует пересмотреть. Например, использование таймеров для расторжения контрактов может привести к несоответствиям во время выполнения или проверки, как обсуждалось в разд. III-в.

В некоторых исследованиях предлагались стандарты для некоторых конкретных случаев использования. Например, в литературе доступны рекомендации по финансовым смарт-контрактам [37], а также по изменению и расторжению смарт-контрактов [38].

Другие работы были сосредоточены на парадигмах и инструментах, связанных со смарт-контрактами [28], [39], включая стратегии снижения платы за газ [40]. Одни авторы предоставили формальное описание смарт-контрактов и их характеристики [41], а другие описали вопросы программирования смарт-контрактов и предложили некоторые возможные решения [42], [43]. Однако в таких исследованиях основное внимание уделяется чисто техническим аспектам.

РАЗДЕЛ III. Смарт-контракты: заблуждения и рекомендации

В этом разделе выявляются некоторые распространенные заблуждения, связанные со смарт-контрактами на основе блокчейна, и приводятся рекомендации по их стандартизации. Таблица 1 суммирует содержание этого раздела.

ТАБЛИЦА 1 Обзор смарт-контрактов. В таблице представлены некоторые категории. В таблице обобщаются основные утверждения, представленные в этой работе для каждой категории.

Category	Claims: a smart contract...
Perspectives	is a computer program is a state-transition function is an equivalence class belongs to the system
Properties	is digital is likely tamper-proof is likely bug-free
Requirements	must be verifiable must be deterministic must use blockchain to store its output
Misconceptions	is immutable has to be stored on-chain has to be certified might not alter the state of the system has legal value has an intrinsic meaning and interpretation
Guidelines	should be independently coded (preferably) should be independently tested (preferably) should be independently executed (preferably) should be defined by what it does (not by how) should be avoided if the code must be certified should rely on oracles as little as possible should not rely on external data sources (e.g., the node's clock) should not rely on undefined behaviors (e.g., iteration order) should provide access to its source code should be integrated into a framework that defines its meaning may be integrated into a framework that grants it some legal value may leverage execution proofs

A. Смарт-контракты — это функции перехода состояния

Блокчейн-систему можно описать как конечный автомат [27]. С этой точки зрения смарт-контракты — это функции перехода состояния, которые переводят систему из одного состояния в другое: смарт-контракт можно описать как функцию $\delta:S\times I\rightarrow S$, где S — конечное и не- пустой набор состояний реестра блокчейна, а I — набор возможных входных транзакций.

В более широком плане в этом руководстве говорится, что смарт-контракты работают с двумя типами данных: (внутренними) данными реестра, которые являются надежными, и (внешними) данными транзакций, правильность которых должна быть проверена. В зависимости от варианта использования данных реестра может быть недостаточно для проверки данных транзакции. Таким образом, существует внутренний предел возможности использования смарт-контрактов. Преодолеть такое ограничение, приняв частично проверенные данные транзакций, можно, но такой подход приводит к проблеме «мусор на входе, мусор на выходе» (т. е. данные реестра больше не являются надежными).

B. Смарт-контракты должны изменять состояние системы

В отличие от общих функций перехода состояния, смарт-контракт должен изменять состояние системы: если смарт-контракт только считывает некоторые данные или выполняет только некоторую обработку данных без изменения состояния системы, становится невозможным проверить правильность выполнение смарт-контракта. Например, протокол Ethereum позволяет определять чистые функции или функции просмотра [44], как показано в следующем фрагменте кода:

```
contract Logger {

uint64 rowID;

function persist(uint64 id) external {

rowID=id;

}

function lastID() view external

returns (uint64) {

return rowID;

}

}
```

Когда вызываются чистые функции или функции представления, выполнение происходит на одном узле. Следовательно, узел может дать неверный ответ. Тем не менее, это ожидаемое поведение: чистые функции или функции просмотра — это не смарт-контракты, а всего лишь стандартные функции для извлечения данных из блокчейна Ethereum или выполнения вычислений только с данными транзакций. Рассмотрение чистых функций или функций представления, таких как смарт-контракты, является распространенным заблуждением [45], [46], вероятно, вызванным тем фактом, что их объявление происходит внутри структуры данных, определяемой ключевым словом контракт. Другие платформы явно не различают функции на основе их взаимодействия с реестром, но общая концепция по-прежнему применяется: единственный проверяемый результат смарт-контракта — это тот, который хранится в блокчейне. Причина в том, что можно проверить, достигают ли одноранговые узлы консенсуса при записи, но не при чтении: злонамеренный одноранговый узел может ответить на запрос, даже если он не должен, только используя свою копию реестра, но он не может принудительно выполнить запись. операции без изменения большинства копий.

Это руководство имеет соответствующие практические последствия. Например, при генерации сертификата (или выполнении любой другой операции только для чтения) смарт-контракт также должен хранить сгенерированный сертификат (или его отпечаток) в блокчейне. При чтении сертификата (или любых данных из блокчейна) невозможно использовать смарт-контракты, и

необходимо запрашивать достаточное количество узлов. Поскольку это часто нецелесообразно, пользователи, скорее всего, будут доверять нескольким узлам с хорошей репутацией. Таким образом, в блокчейн-системах все еще присутствует определенная степень доверия и централизации.

C. Смарт-контракты должны быть проверяемыми

Должна быть возможность проверить вывод смарт-контракта в любой момент в будущем: если это требование не будет выполнено, система может разветвиться, и консенсус никогда не будет достигнут. В частности, время проверки может сильно отличаться от времени исполнения смарт-контракта, что вводит ограничение на использование связанных со временем примитивов [47]: даже если все узлы в системе используют одни и те же атомные часы , это позволит им только синхронизировать выполнение данного смарт-контракта, но не его проверку. Такое ограничение не мешает смарт-контрактам использовать примитивы, связанные со временем, но их введение следует тщательно обдумать. Например, проверка того, что временные метки двух блоков отличаются друг от друга менее чем на месяц, вернет одно и то же значение в любой заданный момент в будущем. С другой стороны, выполнить ту же проверку с временной меткой последнего блока проблематично, так как ее результат будет меняться со временем. Таким образом, основанные на времени смарт-контракты, такие как предложенные в [21], могут вызывать несоответствия во время проверки. Следовательно, смарт-контракты должны полагаться только на данные, уже сохраненные в блокчейне или предоставленные во входной транзакции. Смарт-контракты не должны полагаться ни на что другое (например, на время, измеряемое узлом, выполняющим смарт-контракт).

Это руководство имеет важные последствия как с управленческой, так и с юридической точки зрения, поскольку некоторые варианты использования могут быть проблематичными для моделирования с помощью смарт-контрактов: можно определить, произошло ли данное событие в определенном временном окне, но не в точный момент. Например, в Биткойне [48] проверки достоверности блоков включают контрмеры, чтобы предотвратить манипулирование майнерами меток времени блоков. Такие контрмеры, однако, по-прежнему позволяют в течение двух часов колебаться. Таким образом, сроки можно проверить только приблизительно.

D. Смарт-контракты должны быть детерминированными

В предыдущем разделе представлен более общий ключевой момент. Смарт-контракт должен производить одинаковый результат на всех узлах, выполняющих его, а не только в любой момент в будущем. Другими словами, смарт-контракт должен быть детерминированным. Это требование часто недооценивается и не ограничивается временными примитивами. Например, карты — это неупорядоченные наборы пар ключ-значение в языке программирования Go. Таким образом, порядок итераций программы, перебирающей карту, не гарантируется. Это поведение также может повлиять на библиотеки сериализации. В более общем плане случайностью в смарт-контрактах необходимо тщательно управлять. Эта проблема активно исследуется, и некоторые из основных методов ее решения основаны на многопартийных вычислениях [49].

С более широкой точки зрения, смарт-контракты требуют стандартизации представления и обработки данных, поскольку это позволяет разным партнерам достигать одного и того же состояния.

E. Смарт-контракты — это классы эквивалентности

Распространенным заблуждением является то, что смарт-контракт уникален [50]. Однако в системах блокчейн узел не может проверить, какие вычисления выполняются другими, но только если они достигают того же состояния после вычисления: разные функции перехода, которые создают одно и то же результирующее состояние, неразличимы в контексте блокчейна. Таким образом, смарт-контракт представляет собой класс эквивалентной функции перехода состояния что при задании одного и того же входного состояния и входного символа создается одно и то же выходное состояние. Например, следующий фрагмент кода показывает две эквивалентные реализации одного и того же смарт-контракта:

```
func sum(a uint8, b~uint8) {

a = a + b

store(a)

}

func sum(a uint8, b~uint8) {

var i uint8

for i = 0; i < b; i++ {

a = a + 1

}

store(a)

}
```

Система блокчейн работает без проблем, даже если одни узлы используют первую функцию, а другие — вторую. Эмпирическое доказательство такого утверждения доступно на Github [51]. Это наблюдение особенно актуально, когда смарт-контракты должны иметь юридическую ценность, поскольку необходимо определить, что должны делать смарт-контракты, а не как они должны это делать [52]. Таким образом, смарт-контракты должны быть выражены на декларативных языках. Создание таких языков является активной областью исследований, и это упростит реализацию и понимание смарт-контрактов [52].

F. Смарт-контракты не нужно хранить в сети

Еще одно распространенное заблуждение, вероятно, введенное протоколом Ethereum, заключается в том, что смарт-контракты должны храниться в цепочке [53], [54]. Как обсуждалось в предыдущем разделе, узел в системе блокчейн не может проверить, какие вычисления выполняют другие узлы. Следовательно, хранение кода смарт-контракта в сети — это не способ заставить все узлы использовать одну и ту же реализацию. Хранение кода смарт-контракта в сети — это всего лишь простой способ распространения кода смарт-контракта на все узлы блокчейна. Публичные блокчейны (например, Ethereum) используют этот подход для автоматического обновления набора смарт-контрактов, развернутых на каждом узле. Другие платформы (например, Sawtooth [55]) не хранят код в цепочке, и каждый узел отвечает за установку необходимых смарт-контрактов на своем узле. Ключевая идея заключается в том, что каждому узлу не нужно знать, какой код выполняют другие, а нужно знать только, в каком состоянии они в конце концов достигнут. Если предположить, что большинство честны, будет достигнуто общее состояние. Эмпирическое доказательство этого правила доступно на Github [51].

Хранение кода смарт-контракта в сети может быть полезным в других контекстах. Например, может быть важно отслеживать, какая версия смарт-контракта работала в данный момент в прошлом для целей проверки. Более того, ончейн-версия может быть той, которую трибунал должен использовать в случае судебного разбирательства. Однако, если реализация в цепочке является единственной, имеющей юридическую ценность, партнер, который ее кодирует, может манипулировать поведением смарт-контракта в корыстных целях (например, путем сокрытия бэкдора). Даже если честное большинство разветвит систему, чтобы восстановить ее правильное состояние, трибунал может быть вынужден отменить решение сети блокчейна, основанное на большинстве, и отдать предпочтение юридически признанной ветви, созданной злоумышленником. Таким образом, если реализация в сети является единственной, имеющей юридическую ценность, должны быть реализованы дополнительные стратегии, чтобы гарантировать ее правильность (например, формальное одобрение большинства участников).

G. Смарт-контракты не являются неизменными

Это заблуждение связано с предыдущим. Если блокчейн неизменен, а смарт-контракт хранится в цепочке, то смарт-контракт неизменен [56], [57], [58]. Однако термин «неизменный» вводит в заблуждение. Точнее, блокчейн предназначен только для добавления. Следовательно, даже если то, что хранится в блокчейне, не может быть изменено, более новая версия всегда может быть добавлена в реестр. Таким образом, даже когда смарт-контракты хранятся в сети, их можно обновлять. Одна из возможных стратегий для блокчейна Ethereum описана в [59]: смарт-контракт хранит адрес другого смарт-контракта в одной из своих переменных состояния. Всякий раз, когда вызывается исходный смарт-контракт, он распространяется на смарт-контракт по сохраненному адресу. Поведение исходного смарт-контракта можно изменить, обновив сохраненный адрес. Кроме того, большинство сверстников всегда может принять решение о мягком/жестком форке системы для замены данного смарт-контракта. Таким образом, смарт-контракты Ethereum неизменны только в том случае, если они не реализуют механизм обновления, и большинство не хочет их изменять. Таким образом, смарт-контракты Ethereum защищены от несанкционированного доступа, более чем неизменны. Другие платформы (например, EOSIO [60]) позволяют обновлять смарт-контракты, перезаписывая старый код новым [61].

К сожалению, обновить смарт-контракты непросто, так как все узлы должны начать использовать более новую версию одновременно, чтобы избежать разделения системы. Поскольку каждый узел действует независимо от других, обновления смарт-контрактов не могут быть принудительно принудительными, как в централизованных системах, но должны быть предложены и приняты одноранговыми узлами. Стратегия принятия зависит от платформы и, среди прочего, может потребовать проверки правильности транзакции обновления (например, путем проверки того, что создатель исходного смарт-контракта также является лицом, осуществляющим обновление) или получения явного одобрения пиров. Таким образом, управление смарт-контрактами является более сложной задачей, чем управление централизованными программами, поскольку требует сотрудничества и координации большинства узлов.

H. Смарт-контракты не являются юридическими контрактами

Как следствие их названия, смарт-контракты на основе блокчейна часто считаются контрактами [62], [63], [64], что не соответствует действительности [10]: смарт-контракты на основе блокчейна — это компьютерная программа. Следовательно, они имеют огромный спектр возможных приложений и могут представлять собой гораздо больше, чем соглашение [10]. С этой точки зрения смарт-контракты — это больше, чем просто юридические контракты. Тем не менее, в смарт-контрактах обычно не указывается, кто должен их использовать, и пользователи подписывают в цифровой форме не смарт-контракты, а транзакции для взаимодействия с ними. Таким образом, смарт-контрактов может быть недостаточно, чтобы быть юридическими контрактами, и их юридическая ценность должна быть узаконена существующими юридическими инструментами. В некоторых юрисдикциях определение юридического контракта может уже применяться к смарт-контрактам; в других юрисдикциях может возникнуть необходимость узаконить смарт-контракты параллельными юридическими контрактами [65]. В любом случае смарт-контракты могут автоматизировать юридические контракты, по крайней мере, частично [29]. Тем не менее, это открытая область исследований и требует тщательной разработки гибридной структуры, включающей как информатику, так и аспекты, связанные с правом [15], [66].

В общем случае автономный блокчейн не может заменить существующие контракты или рассматриваться как источник неопровержимых доказательств в судебных разбирательствах. Смарт-контракты могут стандартизировать и упростить обмен данными между несколькими компаниями, но их юридическая легитимация лишь потенциальна и зависит от юрисдикции. Руководителям следует тщательно взвесить экономические последствия внедрения технологии блокчейн, не предполагая само собой разумеющееся сокращение судебных издержек.

I. Смарт-контракты не несут смысла

Смарт-контракты — это компьютерные программы: они обрабатывают последовательности битов и производят другие последовательности битов. Однако смарт-контракты не детализируют значение и правильную интерпретацию последовательностей битов, которые они производят. Например, смарт-контракт может хранить в блокчейне следующую последовательность: e, s, t, a, t, e. Хотя можно предположить, что последовательность представляет собой слово Estate, это не может быть гарантировано. Буквы могли быть результатом случайного извлечения. В таком случае их следует интерпретировать отдельно. Кроме того, слово Estate имеет разные значения

в итальянском и английском языках (это ложный друг). Следовательно, одни и те же данные могут иметь разную интерпретацию. Таким образом, смарт-контрактов (самих по себе) недостаточно: им нужны внешние стандарты, чтобы установить, как данные должны быть закодированы/декодированы и как потребители данных должны их интерпретировать. Правила интерпретации данных не могут быть наивно сохранены в блокчейне, так как проблема станет закольцованной.

В какой-то степени все протоколы блокчейна (например, Ethereum) неявно определяют стандарты кодирования и декодирования данных. Однако, поскольку они пытаются быть агностическими и универсальными, они не предоставляют достаточно подробностей, чтобы гарантировать однозначную и осмысленную интерпретацию хранимых данных, что требуется для легитимации смарт-контрактов с юридической точки зрения. С управленческой точки зрения правила интерпретации данных могут быть неявными, и никакая внешняя структура не требуется, если смарт-контракты используются только для обмена данными между несколькими компаниями. Однако, чтобы иметь юридическую ценность, такие правила должны быть четкими (или каким-то образом общепризнанными и общепризнанными), а смарт-контракты должны быть интегрированы в гибридную структуру, включающую аспекты, связанные как с информатикой, так и с законом.

J. Предпочтительно, чтобы смарт-контракты были независимо написаны и развернуты

В отличие от того, что в настоящее время делают многие сети блокчейнов (например, Ethereum), смарт-контракты не должны быть закодированы один раз и развернуты на всех узлах системы, поскольку это противоречит внутренней идее блокчейна, который представляет собой систему, в которой узлы не доверять друг другу. Следовательно, узел никогда не должен соглашаться на выполнение реализации, предоставленной другими ненадежными узлами. Вместо этого каждый узел должен автономно реализовать все смарт-контракты, чтобы быть уверенным в их правильности. Пока все честные узлы разделяют общее видение того, как смарт-контракты должны изменять состояние системы, все независимые реализации должны принадлежать к одному и тому же классу эквивалентности. Таким образом, все узлы могут достичь одного и того же состояния, даже если они используют разные реализации одного и того же смарт-контракта.

К сожалению, требование, чтобы каждый узел реализовал свой смарт-контракт, часто невыполнимо. Есть несколько вариантов использования, когда нескольким участникам нужен блокчейн консорциума с несколькими смарт-контрактами, которые все они используют (например, в логистике [67], [68]). В таких случаях каждый участник может позволить себе закодировать и развернуть свою реализацию смарт-контракта. Однако в общедоступных блокчейнах узлы обычно используют только часть всех доступных смарт-контрактов. Более того, им часто не хватает надлежащих компетенций и ресурсов для создания независимых реализаций. В таких ситуациях требование, чтобы каждый узел самостоятельно кодировал все смарт-контракты (даже те, которые он не использует), является слишком требовательным. Тем не менее, злонамеренное использование ошибки становится почти невозможным при наличии всего нескольких независимых и равномерно распределенных реализаций, как того и хотела бы ошибка. влияют только на часть сети. Таким образом, стимулирование создания нескольких реализаций и предоставление одноранговым узлам возможности выбирать, какую из них установить на своем узле, может снизить риск использования ошибок в смарт-контрактах.

С управленческой точки зрения это руководство оказывает существенное влияние на экономику проектов, основанных на блокчейне, поскольку затраты на разработку не распределяются, а реплицируются между различными узлами. Более того, требуются дополнительные усилия, чтобы гарантировать принадлежность всех независимых реализаций к одному и тому же классу эквивалентности. В частности, создатели широко используемых смарт-контрактов могли бы финансировать создание независимых реализаций, чтобы компенсировать риск использования ошибок. Подобные инициативы уже существуют в виде программ Bug Bounty.

K. Желательно, чтобы смарт-контракты подвергались независимому аудиту и тестированию

Смарт-контракты должны быть независимо проверены и протестированы узлами, прежде чем они начнут их использовать. Таким образом, по-прежнему применимы соображения, аналогичные тем, которые были сделаны для предыдущего руководства. Однако это руководство, вероятно, найдет более широкое применение, поскольку тестирование смарт-контракта должно быть проще, чем его кодирование. Для оптимизации процесса тестирования рекомендуется публиковать исходный код смарт-контрактов, поскольку это упрощает аудит машинного кода.

С практической точки зрения узлы вряд ли будут тестировать смарт-контракты из-за той же нехватки ресурсов и компетенций, о которой говорилось в предыдущем руководстве, в частности, в общедоступных сетях блокчейна. Таким образом, особенно при отсутствии экономических стимулов для тестирования, смарт-контракты могут быть не такими безопасными и надежными, как их часто считают.

L. Смарт-контракты могут использовать доказательства исполнения

В некоторых ситуациях требование независимого выполнения смарт-контрактов, требующих больших вычислительных ресурсов, может быть слишком сложным, и использование одного выполнения и доказательства его правильности может быть предпочтительнее. Например, найти решение кубика Рубика может быть сложно. Тем не менее, если ходы заданы, легко проверить, составляют ли они решение. В следующем фрагменте кода показаны два альтернативных подхода:

```
func rubik_heavy(problem Problem) {

solution:= solve(problem)

if solution != nil {

store(solution)

}

}
```

```
func rubik_light(problem Problem, moves []Move) {

solution:= verify(problem, moves)

if solution != nil {

store(solution)

}

}
```

Интересно, что упрощенный подход по-прежнему требует выполнения части протокола в виде смарт-контракта (т. е. проверки и хранения решения). Таким образом, смарт-контракты имеют внутреннюю базовую сложность, поскольку этапы проверки и хранения всегда должны выполняться децентрализованным образом. Тем не менее, многие блокчейн-платформы успешно применяют доказательство выполнения для улучшения масштабируемости. В этом отношении доказательства с нулевым разглашением особенно полезны [69] и в настоящее время используются в таких протоколах, как zkSync [70].

M. Смарт-контракты не нуждаются в сертификации

Независимо от варианта использования смарт-контракты не нуждаются в сертификации: органы по сертификации являются доверенными третьими сторонами, а технология блокчейна пытается устранить все такие стороны. Конечно, в интересах узлов должным образом тестировать и проверять свои реализации смарт-контрактов [71], что также может включать использование внешних служб тестирования программного обеспечения. Однако система блокчейна не должна признавать только несколько органов для проверки смарт-контрактов. Корректность смарт-контрактов должна гарантироваться только тем, что узлы могут достигать одного и того же состояния. Если возможно, каждый узел должен кодировать и тестировать свою реализацию в соответствии со стратегиями, которые узел считает подходящими. В случае, если система блокчейна хочет полагаться на некоторые внешние органы для сертификации смарт-контрактов, следующие стратегии являются лучшими альтернативами:

перейти на централизованную систему, контролируемую органом по сертификации. Фактически, оставив бэкдор в смарт-контракте, орган по сертификации все равно будет иметь полный контроль над смарт-контрактом;

использовать оракулы. Оракулы — это доверенные третьи стороны, которые предоставляют данные, поступающие из реального мира в систему блокчейна. Таким образом, оракулы часто необходимы, поскольку данные реального мира не могут быть (разумно) получены иначе. Если смарт-контракт должен быть сертифицирован, было бы проще и эффективнее позволить органам по сертификации запустить код и сохранить результат в блокчейне. Используя децентрализованную сеть оракулов [72], можно ограничить влияние каждого оракула. Таким образом можно проверить согласованность результатов, полученных различными органами по

сертификации. Проблема этого решения в том, что степень децентрализации пропорциональна количеству органов по сертификации, а не количеству узлов.

С более широкой точки зрения, любая форма централизации подрывает ценность смарт-контрактов на основе блокчейна. Таким образом, прежде чем соглашаться на решения на основе блокчейна, лица, принимающие решения, должны рассмотреть, до какой степени можно децентрализовать систему. Во многих ситуациях непрактичность децентрализованных решений может побудить лиц, принимающих решения, пойти на централизованные компромиссы в своих блокчейн-системах. Тем не менее, этот подход может противоречить цели создания блокчейна.

N. Желательно, чтобы смарт-контракты не зависели от оракулов

Проблема полагаться на оракулы не является исключительной для процесса сертификации кода. Никогда нельзя полностью доверять данным, предоставляемым оракулами: оракул всегда является доверенной третьей стороной, что снижает степень децентрализации системы [29]. К сожалению, устранение оракулов из системы блокчейна редко осуществимо. Таким образом, смарт-контракты должны как можно меньше полагаться на данные, предоставляемые оракулами. Кроме того, должны применяться стратегии, препятствующие неправомерным действиям оракулов. В этом смысле службы оракулов (например, Chainlink [72]) используют экономические стимулы и множественные потоки данных для уменьшения попыток манипулирования.

В более общем плане смарт-контракты не решают проблему «мусор на входе и выходе», когда задействованы оракулы. Таким образом, блокчейн можно использовать для предотвращения ретроактивных (но не упреждающих) манипуляций с данными. В зависимости от конкретного варианта использования лица, принимающие решения, должны оценить, оправдывают ли такие гарантии принятие технологии или нет.

O. Смарт-контракты (вероятно) не содержат ошибок

Если предположить, что смарт-контракт независимо написан несколькими узлами, маловероятно, что все реализации имеют одну и ту же ошибку. Конечно, все реализации могут полагаться на одну и ту же библиотеку. В таких случаях ошибки, влияющие на библиотеку, также влияют на все реализации. Тем не менее, общая идея, лежащая в основе трилеммы масштабируемости [73], по-прежнему применима: увеличение числа независимых реализаций увеличивает усилия, необходимые для их кодирования, и вероятность получения смарт-контракта без ошибок, как уже обсуждалось в разд. III-J.

Мы подчеркиваем, что, вероятно, можно получить аналогичные программы без ошибок в централизованных настройках, поддерживая аналогичные усилия по внедрению. Поскольку это не является стандартным промышленным поведением, рентабельность такой стратегии сомнительна. Тем не менее, мы считаем, что смарт-контракты, которые управляют ценными активами (например, протоколами для децентрализованных финансов), могут рассматривать эту стратегию как способ компенсировать риск кибератак.

P. Смарт Контра cts (вероятно) защищены от несанкционированного доступа

Поскольку смарт-контракт независимо выполняется/проверяется несколькими узлами, нарушение выполнения смарт-контракта практически невозможно. Следовательно, смарт-контракты защищены от несанкционированного доступа, но только до тех пор, пока у узлов нет мотивации для сговора. Более того, свойство защиты от несанкционированного доступа не означает, что смарт-контракт ведет себя так, как ожидается: свойство защиты от несанкционированного доступа является следствием многократного независимого исполнения смарт-контракта, а его корректность является следствием его многократного и независимого выполнения. Следовательно, даже если бы все узлы использовали идентичные реализации, смарт-контракт все равно был бы защищен от несанкционированного доступа (но вряд ли без ошибок).

Мы подчеркиваем, что устойчивость смарт-контрактов к взлому пропорциональна степени децентрализации сети блокчейна. Смарт-контракты не дают никаких гарантий, если один узел может влиять на другие или если у некоторых узлов есть сильная мотивация для сговора. Таким образом, лица, принимающие решения, должны проанализировать отношения между узлами, прежде чем присоединиться к сети блокчейна. В более общем плане блокчейн не решает полностью проблемы доверия, поскольку сверстники по-прежнему должны верить в то, что большинство честны.

Q. Смарт-контракты принадлежат системе

Право собственности на смарт-контракты является спорной темой. Владелец смарт-контракта часто является его создателем. В качестве альтернативы смарт-контракт может предоставлять особые привилегии конкретному объекту. Однако в любом случае смарт-контракт управляется и исполняется узлами системы блокчейн. Если большинство из них решат изменить смарт-контракт, у владельца не будет власти (или права) остановить их. Таким образом, фактическим владельцем смарт-контракта является сама система блокчейн. Хотя это философское замечание, вероятно, не имеет большого практического значения, важно подчеркнуть, что концепция собственности приобретает другое значение в системе блокчейн.

РАЗДЕЛ IV. Заключение

Смарт-контракты могут произвести революцию в мире, поскольку они гибкие и безопасные. Тем не менее, смарт-контракты не следует использовать в качестве отдельной технологии, поскольку они должны быть интегрированы в более широкие рамки, чтобы полностью раскрыть свой потенциал. В этой связи стандартизация смарт-контрактов является необходимым шагом.

В этом исследовании смарт-контракты были проанализированы с разных точек зрения и в контексте технологии блокчейн. В частности, это исследование: предоставило несколько определений смарт-контрактов; выделены их основные свойства и требования; выявлены и исправлены некоторые распространенные заблуждения по теме; предоставил некоторые рекомендации по правильной реализации, развертыванию и контекстуализации смарт-контрактов в общем стандарте. Это исследование является первым шагом к созданию четкого и единого видения смарт-контрактов. В частности, это исследование может быть полезным для

менеджеров, юристов и нетехнических читателей, которым необходимо принять решение о возможном внедрении смарт-контрактов в свои области знаний.

Анализ, проведенный в этом исследовании, выявил множество неправильных представлений о смарт-контрактах, которые могут помешать созданию универсального стандарта. Само название смарт-контракта вводит в заблуждение и должно быть заменено более подходящим словом «чейнкод» [74]. Совместное использование общего определения смарт-контрактов и связанных с ними свойств является необходимостью. В частности, четкое понимание тем, связанных с компьютерными науками, имеет основополагающее значение для того, чтобы видеть сквозь путаницу, создаваемую абстракциями и неправильными толкованиями.

В этом исследовании смарт-контракты не анализировались как отдельная технология, поскольку они тесно связаны с базовой технологией блокчейна. Следовательно, необходимо создать стандарты для надлежащего использования смарт-контрактов и их интеграции во внешние технологии и структуры, что особенно актуально для легитимации и юридического признания смарт-контрактов. В частности, рекомендуется изучить следующие исследовательские вопросы.

Какие условия делают систему блокчейна достаточно децентрализованной и безопасной? Кто может доверять, отвергать, отвергать или игнорировать данные, которые он хранит?

Какие узлы должны кодировать/тестировать/выполнять данный смарт-контракт? Какие факторы и метрики должны определять степень надежности смарт-контракта?

Каким стандартам, связанным с кодированием/декодированием данных, должны следовать смарт-контракты? Где и как указаны эти стандарты?

Как формализовать, какие выходные данные должен генерировать смарт-контракт для заданных входных данных? Например, как формальный язык может определить юридический договор? Можно ли использовать декларативные языки?

Как смарт-контракт может быть связан с его концептуальным значением? Например, если смарт-контракт сравнивает два числа, сравнивает ли он две температуры?

В какой степени смарт-контракты могут заменить юридические контракты?

Основной вывод из руководящих принципов, предложенных в этом исследовании, заключается в том, что полной децентрализации достичь трудно, и она создает множество дополнительных проблем. Учитывая трудности о Создавая действительно децентрализованные смарт-контракты, мы задаемся вопросом, до какой степени можно согласиться на компромиссы. Какой смысл в децентрализации, если мы готовы пожертвовать ею ради практичности? В конце концов,

являются ли децентрализованные решения, ориентированные на компромисс, более надежными, чем централизованные?

References

1.
I. Pavlova, "Blockchain ETFs: Dynamic correlations and hedging capabilities", *Managerial Finance*, vol. 47, no. 5, pp. 687-702, Apr. 2021.
Show in Context CrossRef Google Scholar
2.
V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda and V. Santamaría, "Blockchain and smart contracts for insurance: Is the technology mature enough?", *Future Internet*, vol. 10, no. 2, pp. 20, Feb. 2018.
Show in Context CrossRef Google Scholar
3.
H. Hellani, L. Sliman, A. E. Samhat and E. Exposito, "Overview on the blockchain-based supply chain systematics and their scalability tools", *Emerg. Sci. J.*, vol. 4, pp. 45-69, Aug. 2021.
Show in Context CrossRef Google Scholar
4.
S. Pan, W. Zhou, S. Piramuthu, V. Giannikas and C. Chen, "Smart city for sustainable urban freight logistics", *Int. J. Prod. Res.*, vol. 59, no. 7, pp. 2079-2089, Apr. 2021.
Show in Context CrossRef Google Scholar
5.
A. Ruffini, A. Salerno and F. Simões, "Net-zero emissions: Main technological geopolitical and economic consequences of the new energy scenario", *SSRN Electron. J.*, vol. 2022, pp. 1-14, Apr. 2022.
Show in Context CrossRef Google Scholar
6.
A. A. Khan, A. A. Laghari, D.-S. Liu, A. A. Shaikh, D.-D. Ma, C.-Y. Wang, et al., "EPS-ledger: Blockchain hyperledger sawtooth-enabled distributed power systems chain of operation and control node privacy and security", *Electronics*, vol. 10, no. 19, pp. 2395, Sep. 2021.
Show in Context CrossRef Google Scholar
7.
M. Nadini, L. Alessandretti, F. Di Giacinto, M. Martino, L. M. Aiello and A. Baronchelli, "Mapping the NFT revolution: Market trends trade networks and visual features", *Sci. Rep.*, vol. 11, no. 1, pp. 1-11, Dec. 2021.
Show in Context CrossRef Google Scholar
8.
G. Attanasio, L. Cagliero, P. Garza and E. Baralis, "Quantitative cryptocurrency trading: Exploring the use of machine learning

techniques", *Proc. 5th Workshop Data Sci. Macro-Modeling Financial Econ. Datasets (DSMM)*, pp. 1-6, 2019.
Show in Context CrossRef Google Scholar
9.
C. Kohler, What is the Blockstream Mining Note?, 2022, [online] Available: https://thebitcoinmanual.com/articles/what-blockstream-mining-note/.
Show in Context Google Scholar
10.
M. Dell'Erba, "Demystifying technology. Do smart contracts require a new legal framework? Regulatory fragmentation self-regulation public regulation", *SSRN Electron. J.*, 2018, [online] Available: https://ssrn.com/abstract=3228445.
Show in Context CrossRef Google Scholar
11.
R. Aringhieri, S. Bigharaz, D. Duma and A. Guastalla, "Fairness in ambulance routing for post disaster management", *Central Eur. J. Oper. Res.*, vol. 30, no. 1, pp. 189-211, Mar. 2022.
Show in Context CrossRef Google Scholar
12.
W. Serrano, "Verification and validation for data marketplaces via a blockchain and smart contracts", *Blockchain Res. Appl.*, vol. 2022, Jul. 2022.
Show in Context CrossRef Google Scholar
13.
M. Sookhak, M. R. Jabbarpour, N. S. Safa and F. R. Yu, "Blockchain and smart contract for access control in healthcare: A survey issues and challenges and open issues", *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021.
Show in Context CrossRef Google Scholar
14.
M. Giancaspro, "Is a 'smart contract' really a smart idea? Insights from a legal perspective", *Comput. Law Secur. Rev.*, vol. 33, no. 6, pp. 825-835, Dec. 2017.
Show in Context CrossRef Google Scholar
15.
A. Janssen and F. Patti, "Demystifying smart contracts", *Osservatorio Diritto Civile Commerciale*, vol. 9, no. 1, pp. 31-50, 2020.
Show in Context Google Scholar
16.
M. Boccia, A. Mancuso, A. Masone and C. Sterle, "A feature based solution approach for the flying sidekick traveling salesman problem", *Proc. Int. Conf. Math. Optim. Theory Oper. Res.*, vol. 1476, pp. 131-146, 2021.
Show in Context CrossRef Google Scholar
17.
M. Boccia, A. Mancuso, A. Masone, A. Sforza and C. Sterle, "A two-echelon truck-and-drone distribution system: Formulation and heuristic approach" in Optimization and Decision Science, Cham, Switzerland:Springer, pp. 153-163, 2021.

Show in Context CrossRef Google Scholar
18.
A. Diglio, A. Mancuso, A. Masone, C. Piccolo and C. Sterle, "A MILP formulation for the reorganization of the blood supply chain in italian regions" in Optimization and Data Science: Trends and Applications, Cham, Switzerland:Springer, pp. 51-66, 2021.
Show in Context CrossRef Google Scholar
19.
M. Gajda, A. Trivella, R. Mansini and D. Pisinger, "An optimization approach for a complex real-life container loading problem", *Omega*, vol. 107, Feb. 2022.
Show in Context CrossRef Google Scholar
20.
G. Chiaselotti, T. Gentile and F. Infusino, "Lattice representation with algebraic granular computing methods", *Electron. J. Combinatorics*, vol. 27, no. 1, pp. 1-34, 2020.
Show in Context Google Scholar
21.
ETSI GS PDL 011 V1.1.1, Sophia Antipolis, France, 2021.
Show in Context Google Scholar
22.
G. Perboli, M. Stefano and R. Mariangela, "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases", *IEEE Access*, vol. 6, pp. 62018-62028, 2018.
Show in Context View Article
Google Scholar
23.
Z. Zheng, S. Xie, H.-N. Dai, X. Chen and H. Wang, "Blockchain challenges and opportunities: A survey", *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352-375, 2018.
Show in Context CrossRef Google Scholar
24.
M. Hribernik, K. Zero, S. Kummer and D. M. Herold, "City logistics: Towards a blockchain decision framework for collaborative parcel deliveries in micro-hubs", *Transp. Res. Interdiscipl. Perspect.*, vol. 8, Nov. 2020.
Show in Context CrossRef Google Scholar
25.
S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, [online] Available: https://www.debr.io/article/21260-bitcoin-a-peer-to-peer-electronic-cash-system.
Show in Context Google Scholar
26.
N. Szabo, "Formalizing and securing relationships on public networks", *1st Monday*, vol. 2, no. 9, pp. 1-21, Sep. 1997.
Show in Context CrossRef Google Scholar

27.
V. Buterin, A next-generation smart contract and decentralized application platform, 2014, [online] Available: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf.
Show in Context Google Scholar

28.
W. Zou, D. Lo, P. S. Kochhar, X.-B.-D. Le, X. Xia, Y. Feng, et al., "Smart contract development: Challenges and opportunities", *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2084-2106, Oct. 2021.
Show in Context View Article
Google Scholar

29.
E. Mik, "Smart contracts: Terminology technical limitations and real world complexity", *Law Innov. Technol.*, vol. 9, no. 2, pp. 269-300, Jul. 2017.
Show in Context CrossRef Google Scholar

30.
P. Tolmach, Y. Li, S.-W. Lin, Y. Liu and Z. Li, "A survey of smart contract formal specification and verification", *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1-38, Sep. 2022.
Show in Context CrossRef Google Scholar

31.
P. S. Bayón, "Key legal issues surrounding smart contract applications", *KLRI J. Law Legislation*, vol. 9, no. 1, pp. 63-91, 2019.
Show in Context Google Scholar

32.
C. D. Clack, V. A. Bakshi and L. Braine, "Smart contract templates: Essential requirements and design options" in arXiv:1612.04496, 2016.
Show in Context Google Scholar

33.
A. Dixit, V. Deval, V. Dwivedi, A. Norta and D. Draheim, "Towards user-centered and legally relevant smart-contract development: A systematic literature review", *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022.
Show in Context Google Scholar

34.
S. A. McKinney, R. Landy and R. Wilka, "Smart contracts blockchain and the next frontier of transactional law", *Washington J. Law Technol. Arts*, vol. 13, pp. 313, Apr. 2017.
Show in Context Google Scholar

35.
N. Fotiou and G. C. Polyzos, "Smart contracts for the Internet of Things: Opportunities and challenges", *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, pp. 256-260, Jun. 2018.

Show in Context View Article
Google Scholar
36.
L. Luu, D.-H. Chu, H. Olickel, P. Saxena and A. Hobor, "Making smart contracts smarter", *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 254-269, 2016.
Show in Context CrossRef Google Scholar
37.
W. Brammertz and A. I. Mendelowitz, "From digital currencies to digital finance: The case for a smart financial contract standard", *J. Risk Finance*, vol. 19, no. 1, pp. 76-92, Jan. 2018.
Show in Context CrossRef Google Scholar
38.
B. Marino and A. Juels, "Setting standards for altering and undoing smart contracts", *Proc. Int. Symp. Rules Rule Markup Lang. Semantic Web*, pp. 151-166, 2016.
Show in Context CrossRef Google Scholar
39.
B. Hu, Z. Zhang, J. Liu, Y. Liu, J. Yin, R. Lu, et al., "A comprehensive survey on smart contract construction and execution: Paradigms tools and systems", *Patterns*, vol. 2, no. 2, Feb. 2021.
Show in Context CrossRef Google Scholar
40.
Y.-W. Jeng, Y.-C. Hsieh and J.-L. Wu, "Step-by-step guidelines for making smart contract smarter", *Proc. IEEE 12th Conf. Service-Oriented Comput. Appl. (SOCA)*, pp. 25-32, Nov. 2019.
Show in Context View Article
Google Scholar
41.
K. Hu, J. Zhu, Y. Ding, X. Bai and J. Huang, "Smart contract engineering", *Electronics*, vol. 9, no. 12, pp. 2042, Dec. 2020.
Show in Context CrossRef Google Scholar
42.
D. Macrinici, C. Cartofeanu and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study", *Telematics Inform.*, vol. 35, no. 8, pp. 2337-2354, 2018.
Show in Context CrossRef Google Scholar
43.
T. M. Hewa, Y. Hu, M. Liyanage, S. S. Kanhare and M. Ylianttila, "Survey on blockchain-based smart contracts: Technical aspects and future research", *IEEE Access*, vol. 9, pp. 87643-87662, 2021.
Show in Context View Article
Google Scholar

44.
P. Chapman, D. Xu, L. Deng and Y. Xiong, "Deviant: A mutation testing tool for solidity smart contracts", *Proc. Blockchain*, pp. 319-324, Jul. 2019.
Show in Context View Article
Google Scholar
45.
M. Saetran, J. Seo and S. Park, "Leverage sidechains to reduce the workload of smart contracts through parallelization", *J. Comput. Sci. Eng.*, vol. 15, no. 3, pp. 125-133, Sep. 2021.
Show in Context CrossRef Google Scholar
46.
N. Vashistha, M. M. Hossain, M. R. Shahriar, F. Farahmandi, F. Rahman and M. M. Tehranipoor, "EChain: A blockchain-enabled ecosystem for electronic device authenticity verification", *IEEE Trans. Consum. Electron.*, vol. 68, no. 1, pp. 23-37, Feb. 2022.
Show in Context View Article
Google Scholar
47.
M. Abdelhamid and G. Hassan, "Blockchain and smart contracts", *Proc. 8th Int. Conf. Softw. Inf. Eng.*, pp. 91-95, 2019.
Show in Context CrossRef Google Scholar
48.
J. Lánský, "Bitcoin system", *Acta Inf. Pragensia*, vol. 6, no. 1, pp. 20-31, Jun. 2017.
Show in Context CrossRef Google Scholar
49.
M. Du, Q. Chen, L. Liu and X. Ma, "A blockchain-based random number generation algorithm and the application in blockchain games", *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, pp. 3498-3503, Oct. 2019.
Show in Context View Article
Google Scholar
50.
A. Hassan, M. I. Ali, R. Ahammed, M. M. Khan, N. Alsufyani and A. Alsufyani, "Secured insurance framework using blockchain and smart contract", *Sci. Program.*, vol. 2021, pp. 1-11, Nov. 2021.
Show in Context CrossRef Google Scholar
51.
V. Capocasale, Smart Contract Equivalence, 2022, [online] Available: https://github.com/vittoriocapocasale/SmartContractEquivalence.
Show in Context Google Scholar
52.
G. Governatori, F. Idelberger, Z. Milosevic, R. Riveret, G. Sartor and X. Xu, "On legal contracts imperative and declarative smart contracts and blockchain systems", *Artif. Intell. Law*, vol. 26, no. 4, pp. 377-409, Dec. 2018.

Show in Context CrossRef Google Scholar

53.
S. Rouhani and R. Deters, "Security performance and applications of smart contracts: A systematic survey", *IEEE Access*, vol. 7, pp. 50759-50779, 2019.
Show in Context View Article
Google Scholar

54.
F. Schär, "Decentralized finance: On blockchain-and smart contract-based financial markets", *FRB St. Louis Rev.*, vol. 2021, pp. 1-22, Apr. 2021.
Show in Context CrossRef Google Scholar

55.
K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton and C. Montgomery, Sawtooth: An introduction, 2018, [online] Available: https://www.hyperledger.org/wp-content/uploads/2018/01/Hyperledger_Sawtooth_WhitePaper.pdf.
Show in Context Google Scholar

56.
I. Sergey, V. Nagaraj, J. Johannsen, A. Kumar, A. Trunov and K. C. G. Hao, "Safer smart contract programming with scilla", *Proc. ACM Program. Lang.*, vol. 3, pp. 1-30, Oct. 2019.
Show in Context CrossRef Google Scholar

57.
S. Sayeed, H. Marco-Gisbert and T. Caira, "Smart contract: Attacks and protections", *IEEE Access*, vol. 8, pp. 24416-24427, 2020.
Show in Context View Article
Google Scholar

58.
J. Kongmanee, P. Kijsanayothin and R. Hewett, "Securing smart contracts in blockchain", *Proc. 34th IEEE/ACM Int. Conf. Automated Softw. Eng. Workshop (ASEW)*, pp. 69-76, Nov. 2019.
Show in Context View Article
Google Scholar

59.
Y. Hu, T. Lee, D. Chatzopoulos and P. Hui, "Analyzing smart contract interactions and contract level state consensus", *Concurrency Comput. Pract. Exper.*, vol. 32, no. 12, pp. e5228, Jun. 2020.
Show in Context CrossRef Google Scholar

60.
D. Larimer, J. Lavin, N. Hourt, Q. Ma and W. Prioriello, EOS. IO technical white paper v2, 2018, [online] Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md.
Show in Context Google Scholar

61.
V. Y. Kemmoe, W. Stone, J. Kim, D. Kim and J. Son, "Recent advances in smart contracts: A technical overview and state of the art", *IEEE Access*, vol. 8, pp. 117782-117801, 2020.
Show in Context View Article
Google Scholar
62.
P. Sreehari, M. Nandakishore, G. Krishna, J. Jacob and V. S. Shibu, "Smart will converting the legal testament into a smart contract", *Proc. Int. Conf. Netw. Adv. Comput. Technol. (NetACT)*, pp. 203-207, Jul. 2017.
Show in Context View Article
Google Scholar
63.
Y. Liu and J. Huang, "Legal creation of smart contracts and the legal effects", *J. Phys. Conf. Ser.*, vol. 1345, no. 4, Nov. 2019.
Show in Context CrossRef Google Scholar
64.
A. Norta, "Designing a smart-contract application layer for transacting decentralized autonomous organizations", *Proc. Int. Conf. Adv. Comput. Data Sci.*, pp. 595-604, 2016.
Show in Context Google Scholar
65.
A. Janssen and M. Djurovic, "The formation of blockchain-based smart contracts in the light of contract law", *Eur. Rev. Private Law*, vol. 26, no. 6, pp. 753-771, Dec. 2018.
Show in Context CrossRef Google Scholar
66.
K. Lauslahti, J. Mattila and T. Seppala, "Smart contracts—How will blockchain technology affect contractual practices?", 2017.
Show in Context CrossRef Google Scholar
67.
G. Perboli, V. Capocasale and D. Gotta, "Blockchain-based transaction management in smart logistics: A sawtooth framework", *Proc. COMPSAC*, pp. 1713-1718, 2020.
Show in Context View Article
Google Scholar
68.
V. Capocasale, D. Gotta, S. Musso and G. Perboli, "A blockchain 5G and IoT-based transaction management system for smart logistics: An hyperledger framework", *Proc. IEEE 45th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, pp. 1285-1290, Jul. 2021.
Show in Context View Article

Google Scholar
69.
M. He, H. Wang, Y. Sun, R. Bie, T. Lan, Q. Song, et al., "T²L: A traceable and trustable consortium blockchain for logistics", *Digit. Commun. Netw.*, 2022.
Show in Context CrossRef Google Scholar
70.
A. Gluchowski, Introducing Zksync: The Missing Link to Mass Adoption of Ethereum, 2019, [online] Available: https://blog.matter-labs.io/introducing-zk-sync-the-missing-link-to-mass-adoption-of-ethereum-14c9cea83f58.
Show in Context Google Scholar
71.
M. Almakhour, L. Sliman, A. E. Samhat and A. Mellouk, "Verification of smart contracts: A survey", *Pervas. Mobile Comput.*, vol. 67, Sep. 2020.
Show in Context CrossRef Google Scholar
72.
L. Breidenbach et al., Chainlink 2.0: Next steps in the evolution of decentralized oracle networks, 2021, [online] Available: https://research.chain.link/whitepaper-v2.pdf.
Show in Context Google Scholar
73.
A. Altarawneh, T. Herschberg, S. Medury, F. Kandah and A. Skjellum, "Buterin's scalability trilemma viewed through a state-change-based classification for common consensus algorithms", *Proc. 10th Annu. Comput. Commun. Workshop Conf. (CCWC)*, pp. 727-736, Jan. 2020.
Show in Context Google Scholar
74.
S. Kim, Y. Son and Y. Lee, "A study on the security weakness analysis of chaincode on hyperledger fabric and ethereum blockchain framework", *J. Green Eng.*, vol. 10, no. 9, pp. 6349-6367, 2020.
Show in Context Google Scholar