sustainability

MDPI

Article

Blockchain Meets Sharing Economy: A Case of Smart Contract Enabled Ridesharing Service

Shuchih Ernest Chang *, Erik Chiaway Chang and Yijou Chen

Graduate Institute of Technology Management, National Chung Hsing University, Taichung 402, Taiwan
* Correspondence: eschang@dragon.nchu.edu.tw; Tel.: +886-4-22859465

Abstract: The ideas of the sharing economy have facilitated innovative business applications, such as Uber and Airbnb. As an example of a sharing economy application, ridesharing services take advantage of underutilized resources to create economic value. However, the unruly design of ridesharing systems may make urban traffic more congested and cause other technology-organization-environment issues. This study explores the application of blockchain and smart contract technologies to enhance ridesharing services by harvesting the blockchain benefits of transaction traceability, process transparency, system automation and disintermediation. After presenting system design and implementation details for building and deploying a blockchain-based system to support the reengineered ridesharing service with required business functions, we conduct functionality/performance tests and theory-based comparative analysis to confirm its feasibility and applicability. The results reveal that our system with blockchain-enabled benefits is superior to incumbent ridesharing systems. Moreover, while prior research rarely reports the design and implementation details of blockchain-based systems to support sharing economy services, this paper primarily contributes to extant literature by not only proposing a layered system architecture adapting blockchain and smart contracts into the desired ridesharing service but also demonstrating the design and implementation details, covering the development tools, the deployment environment and the deployed smart contracts.

Keywords: blockchain; smart contract; sharing economy; ride sharing; shared mobility; distributed ledger technology (DLT)

Citation: Chang, S.E.; Chang, E.C.; Chen, Y. Blockchain Meets Sharing Economy: A Case of Smart Contract Enabled Ridesharing Service. *Sustainability* **2022**, *14*, 13732. https://doi.org/10.3390/su142113732

Academic Editors: Xueming (Jimmy) Chen and Suwei Feng

Received: 17 September 2022
Accepted: 21 October 2022
Published: 23 October 2022

1. Introduction

The advancement of Information and Communication Technology (ICT) and the prospering development of the sharing economy lead to the discussion of sharing idle/underutilized resources in society [1–3]. Transportation network companies (TNCs) such as Uber and Lyft adopt the sharing economy concept to take advantage of underutilized vehicles for the production of value and the creation of digital platform-enabled work [2]. Unfortunately, TNCs were the biggest contributor to growing traffic congestion in San Francisco between 2010 and 2016, primarily due to low occupancy rate, so we may want to consider sharing TNC rides (through a service called ride-splitting) to reduce traffic congestion [4]. Prior research argued that ridesharing/car-pooling (hereafter, referred to as ridesharing) may not only mitigate road congestion, but also reduce travel costs and greenhouse gas emissions to facilitate sustainable circulation [5–8]. However, there exist few dedicated online platforms for conducting/supporting ridesharing services, and instead, fragmented ridesharing supply/demand messages may appear on online social networking services (SNS) (e.g., Facebook, Line, etc.), informally announcing ridesharing opportunities [7,9]. Only recently have centralized online ride-sharing platforms emerged, offering drivers and passengers the opportunity to share rides [10].

Traditional e-commerce relies on the centralized governance model to conduct relevant business processes; however, the centralized model suffers the disadvantages of poor performance, single point of failure, lack of flexibility and transparency, vulnerability to malicious attacks and data theft [8,10–12]. In addition, centralized ridesharing platforms functioning as the service intermediaries for drivers and riders usually charge each order a service fee of up to 20% to maximize profits [10]. With high potential to mitigate the aforementioned issues facing the centralized model, emerging blockchain technology with distributed and decentralized characteristics may help rethink the design of business processes and operations in order to benefit from decentralized governance [10,12–17]. While ridesharing services should take advantage of information and cyber technologies to provide participants with convenience and equal opportunity for participation [7], blockchain (as a distributed ledger/database for recording transaction-related information) can help improve business process execution by utilizing its desired features of transparency, traceability, immutability, process automation and disintermediation [10,14,16–19]. However, there exists limited research carried out by adopting blockchain technology to achieve sustainability [14]. Accordingly, this study aims to explore the application of blockchain and smart contract technologies to innovate ridesharing services for harvesting the aforementioned blockchain benefits, especially the valuable features of process automation and disintermediation. Such features can make the proposed blockchain-based system more valuable to its principal participants—ridesharing organizers and passengers/riders. Moreover, the proposed blockchain-based system together with the design and implementation details can help develop a distributed governance model for achieving decentralized value creation and distribution in a new theory of value system with better support for social sharing and the sharing economy [20].

The primary goal of this study is to explore the potentials of applying blockchain technology to build sharing economy applications/services, specifically by using ridesharing services as an example, to achieve sustainability by reducing traffic congestion, energy consumption and greenhouse gas emissions. Given the aforementioned issues and deficiencies faced by TNCs and other centralized ridesharing systems, this study aims to propose and construct blockchain-based ridesharing services to fill relevant research gaps and achieve the goals of sustainability in environmental, technological, economic and social terms, by exploring research questions as follows.

(1) How can we use blockchain technology to build a dedicated, distributed ridesharing service system that enables ridesharing organizers and riders to efficiently find ridesharing partners?
(2) How can a blockchain-based decentralized system be designed to improve the ridesharing service process, especially to reduce the single point of failure, lack of transparency and security vulnerabilities faced by the centralized model?
(3) How can the proposed system be designed to eliminate the ridesharing service intermediaries that charge high service fees?
(4) In the proposed system, how can we utilize blockchain characteristics to design and implement service functions and features to improve ridesharing service quality?
(5) Given that previous research has rarely reported the implementation details of blockchain-based systems that support the sharing economy, how does this study fill in the gaps by providing these details?

Accordingly, the contributions of this research on blockchain ridesharing services are five-fold. First, this study uses blockchain technology and smart contracts to develop a dedicated distributed ridesharing service system that enables ridesharing organizers and riders to efficiently find ridesharing counterparties/partners. Second, by mitigating the shortcomings of single point of failure, lack of transparency and security risks faced by the centralized model, the proposed decentralized system leverages its blockchain characteristics of transparency, immutability, process automation and disintermediation to improve service processes execution. Third, the proposed system facilitates the development of a distributed governance model, enables decentralized value creation and distribution, and better supports social sharing and the sharing economy, especially by eliminating intermediaries that charge high service fees. Fourth, the proposed system implements three special features (identity authentication and trust evaluation, reasonable estimate of ride fare and time, and reassurance of departure/destination locations) to improve its service quality and benefit rider safety. Lastly, while prior research has rarely reported the design and implementation details of blockchain-based systems supporting the sharing economy, this paper provides such details, covering the development tools, the deployment environment and the deployed smart contracts for the proposed ridesharing system.

Following this introduction, Section 2 presents the research background regarding the sharing economy, ridesharing, blockchain technology and smart contracts. Section 3 explains how we designed and developed the proposed blockchain-based ridesharing service together with three special features to mitigate potential conflicts between service participants. Section 4 describes implementation details for development tools and environment, implemented smart contracts and service flows, and smart contract-enabled service functions. Section 5 provides discussion and evaluation of the implemented service prototype, while Section 6 concludes the paper with notes on limitations and future work suggestions.

2. Background

2.1. Sharing Economy and Ridesharing

The development of the sharing economy has brought countless innovative business ideas for the creation of value by optimally sharing underutilized resources (including goods, services, time, etc.) through various collaborative platforms [1,3,20]. According to the European Commission, the global sharing economy is booming [21]; the global market value of the sharing economy was US$14bn in 2014 and it might reach US$335bn by 2025 [3,5]. While value is an important indicator of economic activity [22], the sharing economy provides a new modality of resource allocation and exchange in societies through a new value system consisting of production of value, record of value and actualization of value [20]. This new value system may assess and distribute fair shares of value created by the sharing economy to the contributors, and at the same time, it should ensure an open and meritocratic governance of sharing economy practice, particularly in terms of participation on equal footing and a fair distribution of governing power [7,20]. In 2015, the revenue of sharing platforms from the transportation industry accounted for 47% of the entire revenue of European sharing platforms [23]. Prior research has recommended that transportation and car sharing is a suitable sector for sharing economy [1–3,7], and increasingly, established ridesharing and car-sharing services are examples of a shared approach to transport [2,5,7]. Our ridesharing research particularly fits into the important transportation sector of the sharing economy.

Car use has long been the main cause of traffic congestion, energy consumption and exhaust emissions [2,10,24]. Ridesharing mainly involves sharing a car or van (i.e., carpooling/vanpooling) with other riders traveling to similar destinations [7,10], and its riders usually conduct ridesharing activities through ICT-enabled platforms including online SNS [7,9]. Ridesharing can not only help individual participants save money by reducing the expenses of car ownership and sharing traveling budgets with other passengers, but also achieve wider benefits in terms of reduced travel times, congestion and pollution, thereby mitigating environmental and sustainability issues [6,7,9,10]. For example, it is reported that the average occupancy rate for car journeys is 1.6 people, but it is 2.8 for ridesharing rides, and ridesharing is economically attractive and popular for travel, because drivers can share the cost of petrol with travel companions and passengers can enjoy cheaper and more convenient transport than other options (e.g., train) [5].

There exist some critical success factors in terms of the design, deployment, operation and governance of desired ridesharing service systems. Such factors relate to trust and identity verification, insurance of risks, digital inclusion (reaching audiences), safeguarding checks on participants, time and cost trade-offs, availability and quality trade-offs, policies and regulations and technical solutions [4,5,7]. Among these key factors, this research aims to design and develop a blockchain-based ridesharing system emphasizing five success factors including: (1) trust and identity verification, (2) digital inclusion, (3) time and cost trade-offs, (4) availability and quality trade-offs, and (5) technical solutions.

2.2. Blockchain and Ridesharing

Satoshi Nakamoto conceptualized the blockchain as a distributed ledger technology, to solve the double spending problem inherent in electronic transactions [25]. Underpinning the Bitcoin cryptocurrency, blockchain contains a consecutively growing chain of blocks, and each block securely records transaction data during a specific time span by applying cryptography that uses each block to generate the next block and to verify the authenticity of carried information, thus ensuring the immutability attribute of blockchain [15,26]. In addition to cryptography, blockchain also relies on its consensus algorithm, such as Proof-of-Work (PoW), executed by participating nodes/computers to not just broadcast transaction data to all nodes but also maintain data authenticity and availability in its distributed peer-to-peer (P2P) network without centralized authorities [26]. This kind of decentralized P2P collaboration in recording and sharing transaction information provides the opportunities for a trustless operating environment without traditional trusted authorities [18,20]. Aforementioned blockchain attributes not only enhance information transparency/traceability and responsibility attribution in the context of business operations, but also enable the tracking for product and service flows among enterprises and across borders [14,15,18,20,27].

Nick Szabo coined the concept of smart contract as "*a computerized transaction protocol that executes the terms of a contract*" [28], and later on refined it as "*a set of promises, specified in digital form, including protocols within which the parties perform on these promises*" [29]. Practically speaking, smart contract refers to a programmable protocol capable of automatically executing, verifying and updating business processes via computer programs running on top of the blockchain to facilitate, execute and enforce the terms contractual parties have agreed on, thereby mitigating issues in traditional processes about intermediaries, information latency and trust [8,15,18]. Smart contracts can execute in blockchain environment as an affiliated technology to facilitate business process automation through coding and codified conditions, which represent preset agreement on contract terms, to trigger process flows by a specific entity, event or time for achieving distributed business workflow automation [27,30,31].

While prior research findings reveal favorable arguments that blockchain together with smart contracts can facilitate virtual and physical networking symbiosis [15] and boost ridesharing efficiency [32], our research aims to achieve the reduction of intermediation along with the mitigation of transaction time and cost. In addition, our blockchain-based and smart contract-enabled ridesharing service can facilitate the support of sharing economy applications by a new system of value that helps assess and distribute value based on the dynamics of social sharing [20], particularly by utilizing their desired features of transparency, traceability, immutability (i.e., tamper-proof), process automation and disintermediation [15,18,33].

3. Materials and Methods

3.1. Function-Based Conceptual System Design

Our system design stresses the essential concept that smart contracts allow the creation of pools of resources and their allocation according to agreed criteria [34]. By incorporating smart contracts into the ridesharing service, our system will not only effectively facilitate the needed interactions and transactions between ridesharing organizers and passengers, but also achieve the production and actualization/distribution of value for the primary participants (i.e., organizers and passengers) in a more efficient and fair way, especially for eliminating unfair intermediate costs. As shown in Figure 1, our blockchain-based ridesharing system platform consists of four subsystems for supporting various functions, including: the user management, the transaction processing, the user-trust evaluation and the order creating & following subsystems. We designed and implemented those four subsystems by using blockchain technology together with seven smart contracts designed and deployed in the Ethereum private blockchain environment to support needed service functions and user interactions. The remainder of this subsection presents more details about those four subsystems, while more information about their associated smart contracts will be covered in next subsection.

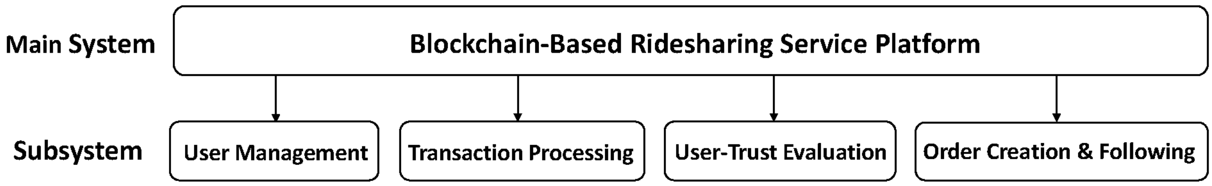

Figure 1. The proposed blockchain-based ridesharing system platform.

Every user who wants to use our ridesharing service platform can use the **user management** subsystem to register and apply for an account by filling in basic user information (such as name, nickname, gender, age, email address, etc.) and successfully providing/binding a mobile phone number with the account. After successful registration, the user may log on to the platform for further confirmation or modification of user information. This study designed and implemented an authentication smart contract (namely, Auth.sol) to deal with the registration and login/logout processes. Only after successful registration and login will users be able to post, search, follow and join ridesharing activities. To enhance the digital inclusion of our service, users can login to the system through external SNS (Facebook, LINE, etc.) as well, but they still have to fill in their basic user information, mobile phone numbers, and driver-specific information (only required for drivers) to access various ridesharing activities.

Using the **transaction processing** subsystem, users can act as a ridesharing group leader/organizer for posting related "group-riding/crowd-riding" content and announcing the group/crowd ridesharing order (event) with needed ridesharing information, including date/time, locations (departure/destination), roles (organizer/passenger), fares, vehicle types (car/van/Uber/taxi/others) and participation criteria. Users can also search, follow and join ridesharing orders from either supply or demand side. A ridesharing group can rent/charter a car or van, take an Uber or taxi, or drive a private car owned by a group member/participant. When potential passengers/followers are interested in joining a ridesharing order/event, they may pay in advance the amount of the fare to be borne by the itinerary. After successfully checking/validating followers' qualification criteria, the subsystem will accept the prepayment and then let the follower join the ridesharing group itinerary. When the agreed date/time arrives, the associated group riders may physically join this ridesharing event by following the roles and responsibility (terms/rules) detailed in the contract they agreed on. When the journey of ridesharing order executes successfully with confirmation, the system will automatically pay the fare to the leader/organizer to finalize the contract. When a no-show or last minute cancellation happens to a passenger due to any incident or personal issue, the subsystem will automatically allocate the compensation fee to the organizer according to a pre-set/agreed ratio. Figure 2 shows the above described service scenario (use case).

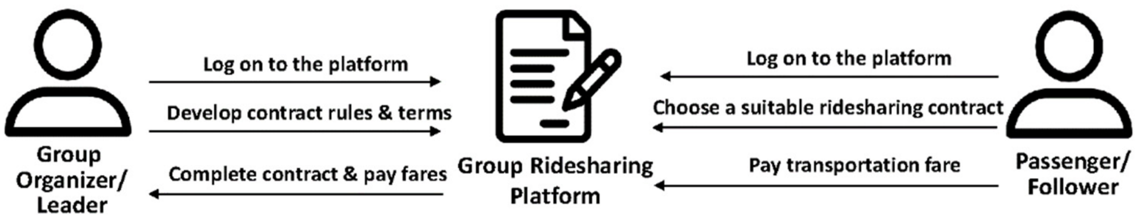

Figure 2. Blockchain-based group/crowd ridesharing scenario (use-case).

The **user-trust evaluation** subsystem aims to enhance the trust between users through P2P trust evaluation, thereby increasing their willingness to use the ridesharing platform. When the ridesharing trip ends, the participants can rate/score their counterparties (e.g., the organizer rates riders and vice versa), and after confirming with participants the subsystem will tally and display the scores. Potential ridesharing participants may reference such scores in making decisions to search, follow and join ridesharing events. By using the **order creation and following** subsystem, ridesharing organizers can create, view, manage and modify order details, and passengers can find ridesharing orders that they have joined. In addition, when the contracted/scheduled riding date/time approaches, the subsystem will send notifications to remind ridesharing event participants.

Based on the above described system functions, we can derive eight use case diagrams (four for the leader/organizer, and four for the follower/passenger) to represent their usage scenarios (see Figure 3), and according to the derived use case diagrams, we can identify business functions desired by the ridesharing service (see Figure 4). Use case diagrams describe the intended business functions and required business processes of the proposed ridesharing system, and help provide requirements to identify roles and deliverables in the design and implementation of the system [35]. Based on the use case diagrams and usage scenarios, this research conducted object-oriented analysis (OOA) and object-oriented design (OOD) processes to build class diagrams and develop class codes. Specifically, after analyzing each usage scenario and identifying the relevant ridesharing events and interactions for each use case, we developed the classes and implemented the events and interactions between classes by using the usage scenario-based modeling approach detailed by Larman [36].

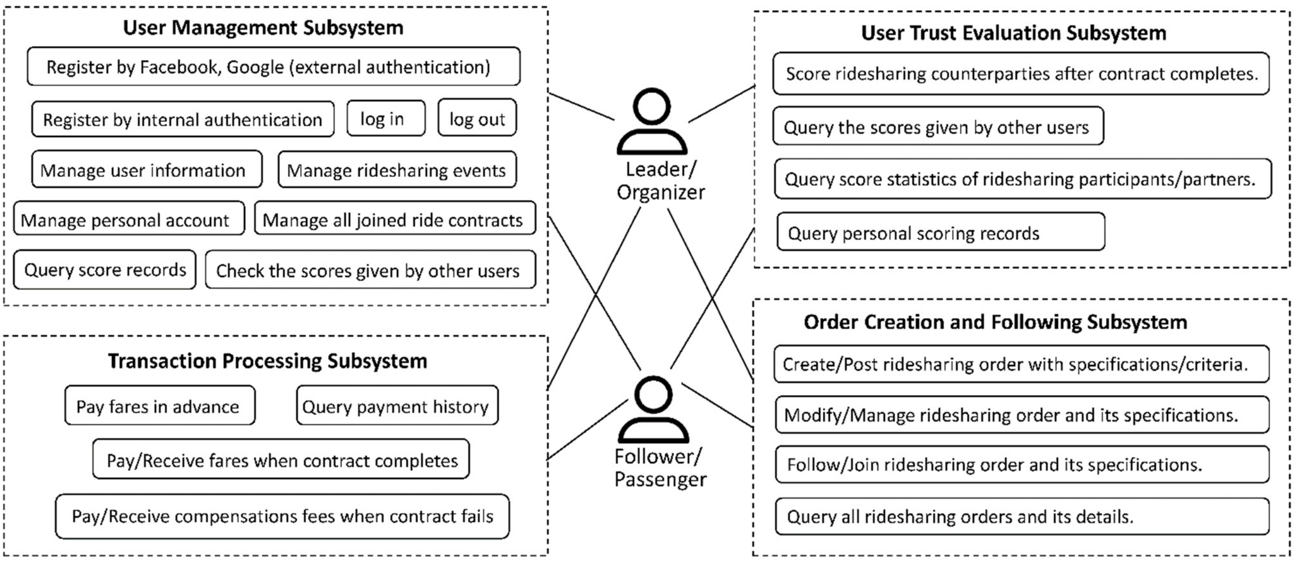

Figure 3. Use case diagrams derived by this study to represent usage scenarios.

3.2. Adapting Blockchain and Smart Contracts into the Proposed Ridesharing Service

This research designed a four-layer service architecture to incorporate blockchain and smart contracts into the proposed "group-riding/crowd-riding" service system. As shown in Figure 5, the layered architecture consists of the transaction layer (TL), the service layer (SL), the interface layer (IL) and the application layer (AL). In TL, there are three types of smart contracts written in Solidity to support the functions, including identity verification, order management and financial management. Adopting blockchain and smart contract technologies, this layered architecture can achieve security, distributed data storage, information immutability, service process traceability/transparency and smart contract-enabled process automation.

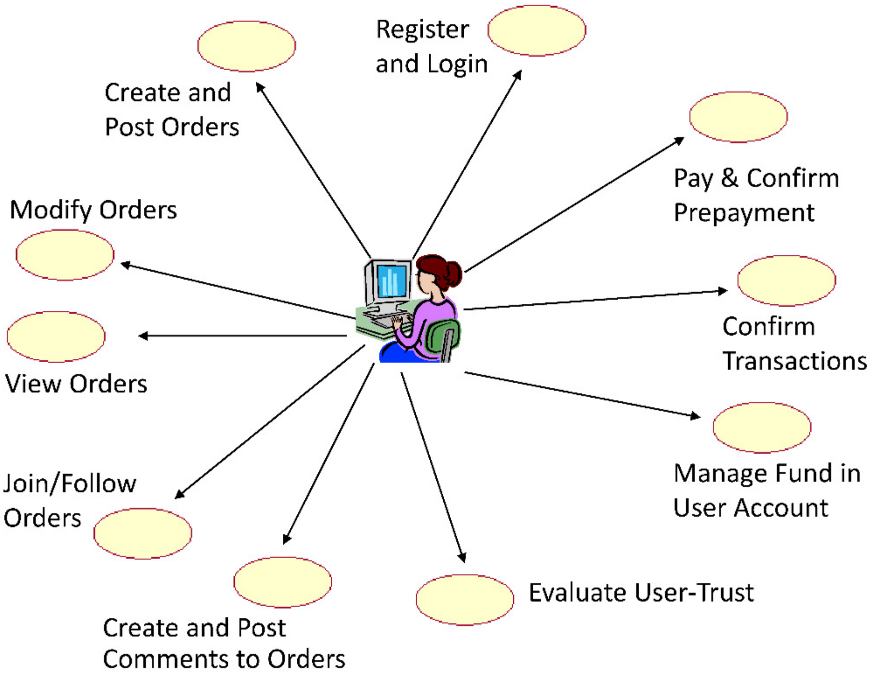

Figure 4. The desired system functions for supporting the group/crowd ridesharing service.

Application layer	**Distributed applications (Dapps), including Trust evaluation Dapp**
Interface layer	**Oracle services (Get, Post, Sent, Store)**
Service layer	**Third-party API, On-chain and off-chain data management**
Transaction layer	**Smart contracts: Identity verification (IVSC), Order management (OMSC), and Financial management (FMSC)**

Figure 5. The layered service architecture incorporating blockchain and smart contracts.

IL and TL need to use oracle as a bridge for blockchain (i.e., the on-chain environment) to communicate with the outside world (i.e., the off-chain environment) for accessing external data. AL is where users interact for acquiring information, making transaction orders to share rides and accessing distributed applications (Dapps) that support various ridesharing functions such as user-trust evaluation. After receiving data from AL for storage purposes, IL passes it to SL and then SL makes a decision to place data on-chain or off-chain. In cases where SL chooses on-chain storage, it just passes data to TL's smart contract for storage; for the alternative off-chain decision, SL simply invokes the third-party APIs to store it externally. SL also uses TL's smart contract and the third-party APIs to retrieve data residing on-chain and off-chain, respectively. IL needs to verify and confirm whether the data originally obtained from the outside world is accurate, so that only verified and confirmed data can enter TL for storage and subsequent retrieval. This layered architecture possesses four advantages, including: (1) enhanced security in accessing information; (2) better traceability in each process flow to ensure immutability; (3) facilitation and optimization of on-chain/off-chain data exchange; and (4) improved integration and interoperability of blockchain and existing systems. More descriptions of those four layers are as follows.

The transaction layer (TL) contains three types of smart contracts written in Solidity, namely, the identity verification smart contract (IVSC), the order management smart contract (OMSC), and the financial management smart contract (FMSC). These smart contracts run in the Ethereum environment to help manage users' information and their transactions; record related itinerary, riders and status of each ridesharing order; and automate the allocation process of ridesharing fees.

IVSC provides needed identity verification functions to facilitate the registration and subsequent login/logout processes. For each successful user registration, IVSC will create and record the user's ID and Ethereum blockchain address that contains 42 characters in hexadecimal format. Actually, we can derive the Ethereum address by hashing the public key generated by Ethereum. OMSC will record each order's details (such as date/time, locations, roles, fares, vehicle types and participation criteria) with needed information about its organizer and riders (such as each user's account ID and Ethereum address). An organizer can use OMSC to create ridesharing orders/events, post orders to announce riding events/opportunities, and search/view what orders the organizer announced; a rider can use OMSC to join orders posted by organizers, search and find out what orders the rider has joined. After the rider joins a ridesharing order, FMSC will check the ride's account to make sure its fund balance can cover the riding fee, and then take out the agreed amount and withhold it on the contract. When the ridesharing itinerary completes with participants' confirmation, FMSC will automatically allocate the withheld fund to the organizer's account. If a no-show or last minute cancellation happened to a rider due to any incident or personal issue, FMSC will automatically distribute the compensation fee to the organizer according to a pre-set/agreed ratio. In so doing, no third-party intermediary is involved, only P2P interaction and financial transaction automation enabled by FMSC, thus reducing the intermediary expenses/charges, improving the transparency of cash flows, and achieving better efficiency through process automation.

After receiving data from IL or TL, SL will make a decision to place data on-chain or off-chain, and then either pass the on-chain-bound data to TL's smart contract for storage, or invoke the third-party APIs to store the off-chain-bound data externally. SL also uses TL's smart contract and the third-party APIs to retrieve and process data residing on-chain and off-chain, respectively. For example, in order to allow users to plan and view their travel itineraries more conveniently, we incorporated the Google Maps API into our system so that users can see exact locations on the map. In addition, we also used the Google Maps API to provide the coordinates (longitude/latitude) data to TL for storage and AL for computation.

IL not only provides four methods (including Get, Post, Sent and Store) but uses oracles (which are third-party agents outside a blockchain) to help the smart contracts at TL obtain off-chain data from the external environment. However, blockchain systems that use oracles to facilitate information exchange with the outside world may suffer from oracle problems that can lead to single point of failure, unreliability of data and compromise of trust foundation [37]. Specifically, IL provides oracle services that may cause oracle problems, and such problems are worth further investigation by future research, which will be mentioned further in Section 6. After receiving data from Dapps running at AL, or from the data management interfaces running at SL, IL will then verify the validity of data followed by passing the validated data either to SL for storage or to AL for further computation.

We develop the Dapps and deploy them at AL. If necessary, Dapps may pass the data derived/created at AL to IL for verification/authentication purposes to improve the trustworthiness and security of data. AL will not change the blockchain settings required/specified by IL and TL, that is, blockchain configurations are transparent to Dapps' development processes, thus making it easier and more convenient for users to announce ridesharing orders. This approach can encourage participants' ridesharing behavior by making users' P2P transactions and interaction more understandable and easier to use. For example, the trust evaluation Dapp helps quantify the trust among users/participants through P2P trust evaluation, thereby increasing their willingness to use the ridesharing service. When the ridesharing itinerary completes, each participant can score against other participants and then use the Dapp to passes the scores to IL wherein the oracle services will verify and authenticate the score information and then send it to SL for storage. With the stored score information readily available from SL, the user-trust evaluation subsystem can tally and display the scores on demand from the ridesharing service participants.

3.3. Special Features of Blockchain-Based Ridesharing Services

To take advantage of the benefits provided by blockchain and smart contract, this research designed and implemented three special service features in our proposed blockchain-based ridesharing service to distinguish it from existing ridesharing services. The special service features include: (1) security enhancement through identity authentication and trust evaluation; (2) reasonable estimate of ridesharing fare and time; and (3) reassurance of departure/destination locations. More details about these special features are as follows.

We implemented a two-step authentication process in the user management subsystem, which would only accept registration of each applicant with a valid email address and mobile phone number. For example, the subsystem will send verification codes to applicants via email and mobile phone, and each applicant needs to use the verification code to complete registration within 5–10 min, thereby rejecting applicants who try to register with false contact information, and consequently making the ridesharing service more trustworthy and safe. By asking applicants to register with valid mobile phone numbers, this special two-step authentication approach is an effective and practical method of identity verification in Taiwan, because each mobile phone user must provide two valid IDs (e.g., the government issued official ID card and driver's license) when applying for mobile phone services. In fact, this special measure that requires a valid mobile phone number during registration can alleviate the security issues on Sybil attacks, in which an attacker may use multiple fake identities to achieve an exaggerated influence in SNS, thus compromising the security of the social network and the trust relationship between users [38].

In addition, the user-trust evaluation subsystem developed in our ridesharing service can help establish the trust between users. After completing ridesharing itinerary, each participant may score/rate against other participants by using the trust evaluation Dapp. The Dapp will work with relevant parts of our ridesharing system to verify, authenticate and store rating scores so that the user-trust evaluation subsystem can tally and display the scores on demand from the ridesharing service participants. As suggested by Zulfiqar et al. [38], our blockchain-based P2P rating/scoring system may not only mitigate rating frauds from fake identities but also eliminate the manipulation and tampering of rating results from the centralized authority due to any selfish motivation.

While each ridesharing group organizer should be able to collect fares to share related and reasonable expenditures, the for-profit mindset to make money by organizing riding events/orders not only fails to reduce riding cost but also hinders riders' willingness to join ridesharing services, thus negatively affecting the diffusion process of our proposed ridesharing service. Nevertheless, regulations often disallow making a profit on ridesharing, especially for private vehicle drivers who may only charge passengers an amount to share the petrol and running costs for the ridesharing journey [5]. The reason is that when ride-sharers make profits, they become taxi operators and then are subject to various government regulations, such as licenses and taxes [5]. As such, our ridesharing service implements a feature to derive an estimate of cost based on itinerary details (including date/time, departure/destination, vehicle types, etc.) so that ride-sharers can refer to the estimate for sharing reasonable expenses in a more confident and trustworthy way. For example, in case the ridesharing group opts to take an Uber and shares the expense among group riders, our ridesharing system will get a quote from Uber's website based on itinerary details and then use the quoted information to compose an estimate which contains the quoted fare, travel distance, travel time and other surcharges (if applicable). Alternatively, the ridesharing group may take a taxicab, and in this case, our system will use the itinerary details to derive travel distance, travel time, and taxi fare, by using the Google Maps API and the taxi fare schedule (or calculator) in the applicable area.

A ridesharing group organizer has to specify the departure location and destination location when creating and posting an order; such location information is essential for passengers to search, follow and join interested ridesharing orders. We incorporated the Google Maps API into our system to let users view exact locations on the map, and generate the coordinates (longitude/latitude) to pinpoint the exact departure/destination places. In so doing, both the organizer/leader and the passengers/followers can clearly double check the location information shown on the map, and then confirm/reassure regarding the exact locations they agree on, thus reducing the possibility of dispute about locations.

4. Implementation and Results

This study developed a ridesharing service system by using blockchain technology and smart contracts, together with the cloud services concept, Web application paradigm, OOA/OOD, and distributed P2P networking environment. Specifically, we adopted the Ethereum blockchain because of its worldwide popularity and solid support to build, test and deploy smart contracts. In the following subsections, we provide detailed information about the implementation of the proposed ridesharing service system.

4.1. The Development Tools and Environment

Figure 6 shows the interaction diagram of our ridesharing service in the blockchain environment. Initially, this study established the Ethereum environment and nodes by using Geth, which provided many Ethereum APIs for us to set up and manage blockchain nodes during our development efforts. Then, we used the Solidity language to develop the smart contracts (see Table 1), and used web3.js to deploy and communicate with smart contracts. We also used the Node.js server as the server side to play the role of the blockchain platform manager for building front-end applications, calling smart contracts, and accessing the off-chain data. Actually, from blockchain's point of view, there are two categories of data, on-chain data and off-chain data, needed for supporting the proposed ridesharing services. On-chain data is stored on the blockchain network and can be readily and easily available for direct access by smart contracts. However, off-chain data belongs to a different category. As for accessing off-chain data, we not only used the MongoDB to store basic personal information and application data, but also adopted the distributed InterPlanetary File System (IPFS) to store large files (e.g., documents and pictures). In this approach, we can use Provable (Oraclize) in Ethereum to access off-chain data through the RESTful API, or call the API via the Node.js server to manage the external data and subsequently propagate it to the blockchain. To perform the technical implementation of the service system, this study used the hardware and software (including tools and standards) summarized in Table 2.

4.2. Smart Contracts Design and Implementation

As mentioned in Section 3, based on the use case diagrams and usage scenarios, we conducted OOA and OOD processes to build class diagrams and develop classes/codes for smart contracts. Specifically, after identifying needed service functions for supporting the ridesharing service (see Figure 4), we conducted the object-oriented programming (OOP) approach to identify, define and implement attributes and methods for each smart contract. After iterating the development/implementation efforts through the needed smart contracts in this study (see Table 1), we created eight essential smart contract-enabled service functions for the blockchain-based ridesharing services prototype. Such functions (see Table 3) facilitate the ridesharing service in regard to: (1) registration and login; (2) ridesharing order creation/posting; (3) viewing posted orders; (4) modification of posted orders; (5) ridesharing order joining/following and prepayment confirmation;

(6) transaction confirmation; (7) evaluation of user-trust and commentary on orders; and (8) user account fund management.

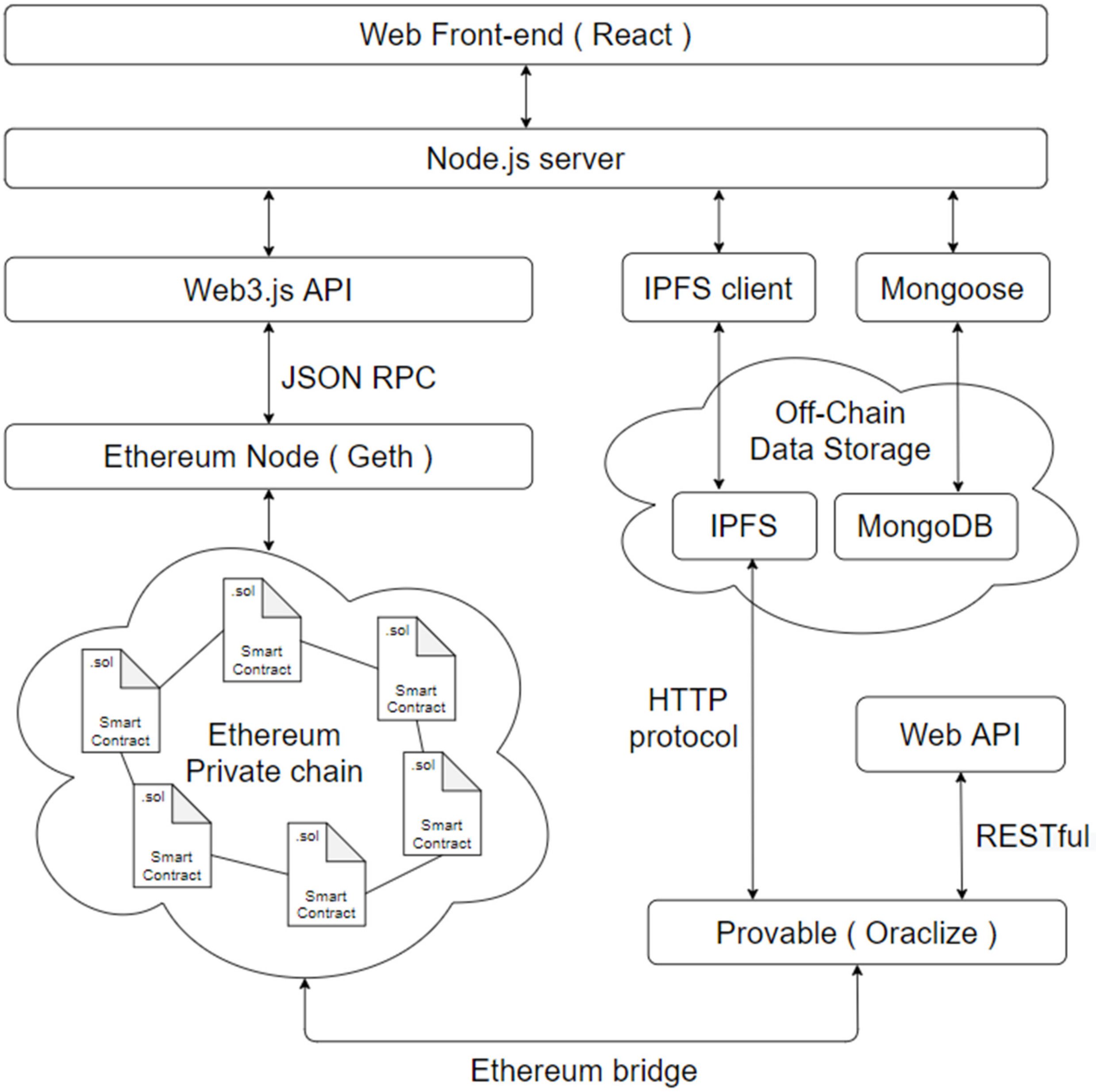

Figure 6. The interaction diagram of blockchain-based application environment.

4.3. Process Flows of the Proposed Ridesharing Service

Figure 7 shows the process flows of our implemented blockchain-based group/crowd ridesharing service (BGRS) system prototype that provides eight essential service functions shown in Table 3. Technically, we designed, implemented and deployed eleven smart contracts (see Table 1) in our blockchain application platform environment (see Figure 6) to support service functions and facilitate service process automation.

Table 1. Summary of smart contracts developed to support proposed ridesharing service functions.

Smart Contract	Descriptions of Major Features and Functionalities
Announce.sol	Search, view and join ridesharing order.
AnnounceManager.sol	Create, post, modify and view order together with its itinerary and riders information.
Auth.sol	Register and login/logout the system; Authenticate user identity; Create user object; Generate and manage blockchain keys.
User.sol	Create and manage user IDs, passwords, blockchain addresses; Create and manage blockchain addresses linking to user accounts and transactions;
UserAccount.sol	Create and use user accounts to manage users' activities including orders, transactions, and financial (cash/fund) flows.
UserManager.sol	This is a private contract with private methods and data to support functionalities of User.sol and UserAccount.sol; Manage user list, IDs and passwords by blockchain address.
Comment.sol	Create, post, and view comments.

Note: We also implement four auxiliary smart contracts with methods and data to support seven smart contracts described above, and those four auxiliary smart contracts and their descriptions are summarized as follows.

(1) AccountManager.sol	Help UserAccount.sol to manage user's fund/cash flows.
(2) TrafficTransaction.sol	Help Announce.sol, AnnounceManager.sol and Comment.sol to manage order itineraries.
(3) TransactionManager.sol	Help AnnounceManager.sol and UserAccount.sol to manage orders and transactions.
(4) Global.sol	This is an auxiliary smart contract with methods and data to support all other smart contracts.

Table 2. Summary of hardware and software (tools & standards) used in system development.

Hardware Specifications (3 computers *)	• CPU: Intel® Core™ i7-7700 CPU @ 3.60GHz (Node 1), Intel® Core™ i5-4590 CPU @ 3.30GHz (Node 2), Intel® Core™ i7-8700 CPU @ 3.20GHz (Test Machine). • RAM: 8.0 GB (Node 1), 8.0GB (Node 2), 32.0 GB (Test Machine).	
Software Specifications (including tools, standards, and development environments)	• Geth v1.8.26-stable • Solidity Compiler v0.5.10 • Go (go language) v1.11.5 • Web3.js v1.2.11 • Node.js v12.14.1 • IPFS v0.4.22 • MongoDB v4.2.2	• Docker v20.10.6 • React v17.0.1 • Remix v0.12.1 • Truffle v5.1.41 • Ganache v2.5.4 • Provable v0.4.25 • Ethereum-Bridge v0.6.2

* Note: All three computers are running under the operating system Microsoft Windows 10 Enterprise.

Table 3. Summary of the smart-contract enabled ridesharing service functions.

Ridesharing Service Function	Descriptions
To register and log in/out	User registration and login/logout, enabled by Auth.sol and UserManager.sol
To create/post order	Order creation/posting, enabled by Announce.sol and AnnounceManager.sol
To view posted orders	Order viewing/searching, enabled by AnnounceManager.sol
To modify posted orders	Order modification enabled by Announce.sol
To join/follow orders and confirm prepayment	Order joining and prepayment confirmation, enabled by Announce.sol and AnnounceManager.sol
To confirm transaction	Transaction confirmation, enabled by Announce.sol
To evaluate and make comments on user trust and orders	Evaluation of user trust and commentary on order against ridesharing counterparties, enabled by Comment.sol
To manage fund in user account	Management of user account fund/cash flows, enabled by UserAccount.sol

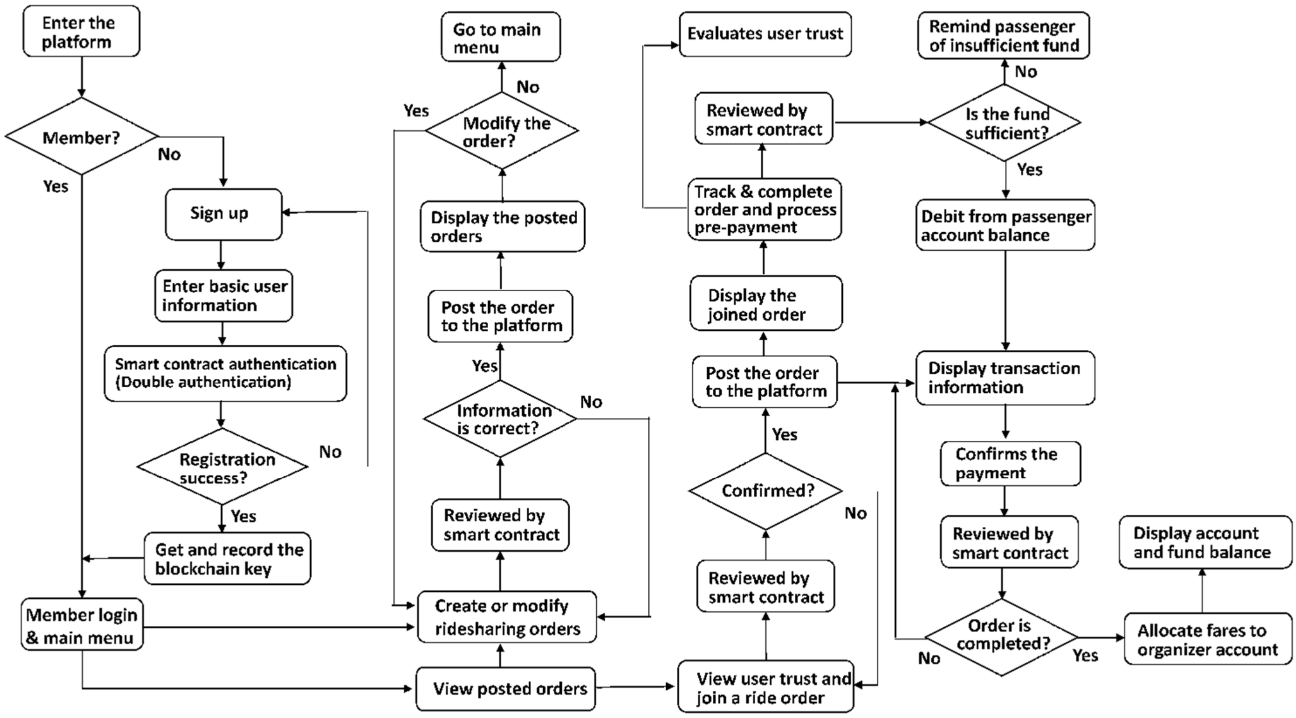

Figure 7. The process flow of the blockchain-based group/crowd ridesharing service (BGRS).

To access the BGRS, a user must register to become a member first, by entering basic user information, providing/binding a mobile phone number with the new user account, and successfully passing the two-step authentication (described in Section 3.3). After a successful member login, BGRS users can act as organizers/leaders to create, post and modify ridesharing orders/messages, or act as passengers/followers to search, view and join desired ridesharing orders. For each created or modified order, the smart contract will review and check its correctness. Afterwards, the BGRS will either post/publish the correct order on the platform and display the just posted order to the organizer for record keeping purpose, or send the incorrect order to the organizer for modification and further actions.

After successful login, a BGRS passenger/follower can search and view available ridesharing orders that organizers have posted and smart contracts have reviewed. Before joining a ridesharing order, the passenger can view the organizer's trust. For each joining order attempt, BGRS will also check the order's itinerary with the passenger, especially for checking and confirming departure/destination locations (by using the Google Maps API) and withholding ridesharing fare (by deducting and escrowing the prepayment from the passenger's account balance). After confirming the order-joining attempt, the BGRS will add the joining passenger to the rider-list in the order, withhold the prepayment of fare, post the order-joining message on the platform, and finally notify the participants (the organizer and the passengers) of the ridesharing order.

After finishing the ridesharing itinerary and using the Dapp to notify the BGRS of the completion of journey, the organizer can use the Dapp to evaluate the trust against passengers listed in the order. The passengers can also use the Dapp to evaluate the trust against the organizer of the order after confirming the completion of ridesharing journey. During the execution of the ridesharing order, the BGRS will monitor/track the order's progress for its participants by using smart contracts until the order completes. After confirming the completion of an order from its participant, the BGRS will allocate the withheld prepayments of fare to the organizer's account, check whether the passenger's account balance is sufficient (for future ride-joining attempts), and remind the passenger if the balance is insufficient. Finally, the BGRS will record and display the transaction information of each completed order for its organizer and passengers to view their ridesharing activities including the fund/cash flow details.

4.4. Implementation Results

We conducted tests on the implemented ridesharing service system against its desired service functions enabled by smart contracts (see Tables 1 and 3) in a private blockchain with three nodes running on the hardware listed in Table 2. After establishing the Ethereum nodes and the private blockchain environment by using Geth, we used web3.js to deploy and communicate with the smart contracts. Afterwards, using Node.js to play the role of the blockchain platform manager for calling smart contracts and managing off-chain data, we tested and validated the ridesharing service prototype against the desired and needed functions. Our functionality test results showed that the implemented service prototype worked smoothly as we originally designed and expected, particularly concerning the business rules realized and executed by the smart contracts. Specifically, we performed rigorous tests to validate the data flows, service scenarios and smart contract functionalities during business interactions and process workflows between ridesharing service participants and smart contracts. We extracted the test result-related screenshots for showing the correctness of service functions, and those screenshots are available from the link specified in Appendix A.

5. Discussion and Evaluation

The sharing economy has been in the center of discussion for several years, particularly for creation of new business models such as the examples of Airbnb and Uber. However, fundamental technologies for implementing sharing economy applications have not changed as much as expected. The proposed blockchain-based "group/crowd" ridesharing service (BGRS) system bring users a valuable and flexible platform for shared urban mobility services, thus achieving the benefits of reducing travel expense, time, congestion and pollution [5]. As an effective approach, we proposed and built a BGRS system prototype to take advantage of unique features of blockchain including decentralization, P2P communication, cryptography and consensus mechanism. In addition, blockchain and smart contract can provide business process automation with the desired characteristics of anonymity, immutability, transparency and trustless consensus [13,27]. To take advantage of the benefits of blockchain and smart contract technologies, this research implemented three special features in BGRS, including: (1) the security enhancement through identity authentication and trust evaluation; (2) the reasonable estimate of riding fare and time; and (3) the reassurance of departure and destination locations. These features provide efficient ways of managing user requirements by mitigating some conflicts that make the existing sharing business troublesome. With the confirmation on each ridesharing order, the system ensures the fulfillment of smart contracts by systematically tracking and automatically trigging service progress. Moreover, ridesharing organizers and passengers can conduct real-time inspections on the status and records of relevant activities by viewing data uploaded on blockchain. As a result, a BGRS offers services that are more flexible, more reliable and more trustworthy to the ridesharing organizers and passengers.

5.1. Performance Evaluation

After validating the functionalities of our BGRS system prototype, we conducted performance evaluation by using the automatic testing tool, Selenium IDE, which helps us record test procedures on Chrome and then execute the recorded test procedures (i.e., repeat the recorded interactions between Chrome and the BGRS system under test). More information about Selenium IDE is available at https://www.selenium.dev/selenium-ide/ (accessed on 15 March 2022).

We used the Selenium IDE to record the interactions/procedures while a new BGRS user went through their login process first time, and then repeatedly ran the recorded interactions/procedures for 1, 5, 10 and 20 time/times. The test results (see Table 4) revealed that the average time spent by a new user's login process was 3.0 s (testing 1 time), 3.2 s (testing 5 times), 3.2 s (testing 10 times) and 4.2 s (testing 20 times), respectively. Next, we performed the similar Selenium IDE tests measuring the average time spent by a ridesharing event organizer to post an order, and the results (see Table 4) showed that the order posting process would take 11.0 s (testing 1 time), 9.4 s (testing 5 times), 9.4 s (testing 10 times) and 10.6 s (testing 20 times), respectively. According to the test results, it takes around 3 to 4 s for BGRS users to complete their login process, and for ridesharing order/event organizers to get their orders posted on BGRS may take around 9 to 11 s.

Table 4. Average time spent in a new user's login process and in an organizer's order posting process.

Iteration(s) of Testing	Average Time Spent in Login Process	Average Time Spent in Order Posting
1 time	3.0 s	11.0 s
5 times	3.2 s	9.4 s
10 times	3.2 s	9.4 s
20 times	4.2 s	10.6 s

Afterwards, we used the Selenium IDE to conduct the same login process and the same order posting process with various multiple concurrent users (2 users, 5 users, 15 users and 30 users) using BGRS at the same time. The test results (see Table 5) showed that the average time spent by a new user to log in BGRS was 3.0 s (2 users), 3.6 s (5 users), 4.0 s (15 users) and 4.5 s (30 users), respectively. Finally, we found that the average time spent by a ridesharing event organizer to post an order in BGRS with various concurrent users was 9.0 s (2 users), 9.4 s (5 users), 10.0 s (15 users) and 10.4 s (30 users), respectively. Based on the test results from Tables 4 and 5, we concluded that a new BGRS user may typically spend 3 to 4.5 s in average to complete the login process, and a ridesharing event/order organizer may typically spend 9 to 11 s in average to post a ridesharing order.

Table 5. Average time spent in a new user's login process and an organizer's order posting process with multiple concurrent users (2 users, 5 users, 15 users, and 30 users) in BGRS.

Number of Concurrent Users	Average Time Spent in Login Process	Average Time Spent in Order Posting
2 users	3.0 s	9.0 s
5 users	3.6 s	9.4 s
15 users	4.0 s	10.0 s
30 users	4.5 s	10.4 s

According to a 2018 study conducted by Google Research, the average loading time—which is defined as the average amount of time to display all of the content on a webpage from when a user clicks on a page link to when the page is fully loaded in the user's browser—of a mobile web page is 15.3 s [39]. Generally speaking, the simple data loading in this study can be completed in less than 3 s. For the user login process involving the off-chain database reading function that may take a longer time, it can be completed from an average of 3.0 s when there are 2 concurrent users to an average of 4.5 s when there are 30 concurrent users, i.e., a new user's login process takes much less than 15.3 s. For the ridesharing event/order organizer's order posting process in the BGRS, it can be completed in from an average of 9.0 s for 2 concurrent users to an average of 10.4 s for 30 concurrent users, i.e., a ridesharing event organizer's order posting process takes less than 15.3 s.

According to Google, the page loading time used for comparisons described above comprises two components: (1) network and server time, and (2) browser time, and actually, the page loading time may be further divided into (1) redirection time, (2) domain lookup time, (3) server response time, (4) page download time and (5) browser time for parsing and executing the JavaScript and rendering the page [40]. As such, the average page loading time is a commonly-adopted composite performance metric that comprises the above-mentioned five metrics, and it is therefore deemed appropriate by this research for comparing the average loading time of the blockchain-based system with the commonly used standard values such as the value of 15.3 s reported by Google Research. However, it is never overcautious to conduct comparisons by using other commonly-available standards. Backlinko, a company dedicated to search engine optimization training and link-building strategies, analyzed 5.2 million desktop and mobile devices for webpage loading in 2019, and the results showed that the average page loading time was 10.3 s on desktop and 27.3 s on mobile device [41]. Compared with these two values reported by Backlinko, the average page loading time derived by this research (shown in Tables 4 and 5) are also considered acceptable.

The last—but never the least—factor in terms of the BGRS performance evaluated in this study: if the hardware and software resources invested in actual business operations can be improved and optimized, the loading time of the BGRS can be further reduced, which is expected to improve the usability and acceptance of the BGRS in actual practice. Although the preliminary performance evaluation results shown in Tables 3 and 4 may be considered acceptable, a properly configured centralized database may provide better transaction throughput and lower latency than blockchain-based systems performance [27]. As suggested by [27], we may explore applicable blockchain configurations such as (1) on-chain/off-chain data storage and computation, (2) block sizes and (3) degrees of centralization, to evaluate potential performance improvement, for achieving better usability of a BGRS with more attractive performance.

5.2. Comparison of the Blockchain-Based Ridesharing Service with the Traditional Approach

A BGRS has the advantages of reducing search and information transaction costs due to the provision of a dedicated and integrated service platform, while in the traditional approach the passengers have to search multiple SNSs to identify desired ridesharing events/orders, followed by contacting the organizers and following various communication methods and diversified interaction protocols to get ridesharing opportunities. As mentioned in Section 2.2, through the designed features of service transparency, information immutability, process automation and disintermediation, a BGRS can facilitate the support of sharing economy applications by a new value system in assessing and distributing innovation value via social sharing [20]. Consequently, BGRS achieves higher service transparency, lower risk of data tempering, faster and automated fund transfer and decentralized service management/control as compared with the traditional approach.

In addition, we also implemented three BGRS special features to enhance its security through identity authentication and trust evaluation, estimate riding fare and time, assure the departure and destination locations, thus alleviating potential conflicts (such as miscommunication, misunderstanding, and disputes) among BGRS participants. Therefore, compared with the existing ridesharing methods, BGRS enjoys higher confidence in identity authentication, service trust and the risk of default in carpooling agreements. Nevertheless, with the required confirmation from the participants of each ridesharing order/event, BGRS ensures the fulfillment of smart contracts by systematically tracking and automatically trigging critical service processes, including the smart contract-enabled account fund allocation and distribution. In particular, when a participating passenger fails to show up or cancels the trip in the last minute due to any personal issue or incident, BGRS will distribute the compensation fee to the ridesharing order organizer according to a pre-set and agreed ratio. This particular compensation fee, deducted from the user account associated with the no-show or last minute cancellation, is an effective way to reduce the default risk of ridesharing agreement as well. Table 6 summarizes the results of comparison between the proposed BGRS and the traditional ridesharing approach.

Table 6. Comparison between the blockchain-based ridesharing service and the traditional approach.

Feature/Function	Blockchain-Based Service (BGRS)	Traditional Approach (Existing Services)
Dedicated/Integrated platform	Yes (Users use BGRS only)	No (Users have to use several SNS sites)
Transparency of service process	Higher	Lower
Risk of data tempering	Lower	Higher
Payment of riding fare	Automated by smart contract	Cash, fund transfer (via bank), credit card
Service management/control	Decentralized/Peer-to-peer (P2P)	Centralized
Identity authentication	Higher (level of confidence)	Lower (level of confidence)
Service trust	Higher	Lower
Default risk of sharing agreement	Lower	Higher

5.3. Theory-Based Analysis with Discussion and Implication

We conduct the theory-based analysis on BGRS by starting with the economics theory, network effects, which suggests that the user's adoption value of a service or product becomes more valuable when the number of adopters increases [42]. One of the essential design goals of a BGRS is to optimize its network effects. By offering the advantageous features/functions shown in Table 6, a BGRS can increase users' willingness to participate in ridesharing practice, thereby enhancing its network effects. Indeed, BGRS participants can conveniently use one dedicated platform to post ridesharing orders or to join desired orders after searching posted orders in the integrated BGRS platform, instead of hopping around and searching through multiple SNS followed by devoting diversified and overwhelmed efforts in communication, clarification, negotiation and decision making to satisfy their ridesharing needs. This dedicated platform and other advantageous features shown in Table 6 can enhance users' intention to adopt a BGRS by offering various benefits over the traditional approach, as described in Section 5.2, particularly in terms of higher transparency, higher security and trust, lower risks and better process efficiency through automation. Moreover, there exist special features in a BGRS (see Section 3.3) to further increase its security and trust, and ameliorate the possibility of conflicts/disputes by estimating and confirming critical itinerary information about locations, date/time, travel expense and travel time.

The success of a BGRS lies in its network effects to induce/attract BGRS adopters, the more the better, and to reach a critical mass in adoption, the faster the better [42]. However, to reach a critical mass for adopting a BGRS, we may pay attention to the innovation diffusion theory (IDT) with five important factors (relative advantage, compatibility, complexity, trialability and observability) that influence users' decision to adopt or reject an innovation [43]. Since this study contributes to the innovation of ridesharing service, it is suitable for us to analyze the value of the proposed BGRS against these five factors regarding innovation adoption. Our IDT based analysis results together with relevant implications are as follows.

5.3.1. Relative Advantage

Prior research found that ridesharing can increase vehicle occupancy and lead to reductions in traffic congestion, energy consumption and emission, but ride-sharers have to do traditional ridesharing informally through various and different SNS sites [9]. In addition to the advantage of providing a dedicated and integrated service which is more convenient and easier for participants to use than the incumbent ridesharing approach, a BGRS also has other advantages over the existing approach. Through the comparative advantages listed in Table 6, a BGRS can help users create, announce, join, and complete ridesharing events/activities in a less expensive, less risky, more convenient, more secure, and more trustworthy way.

5.3.2. Compatibility

We built BGRS through designing/deriving its use cases, activity diagram, process flows, user interface, service functions and system operations in a strategic way to emulate the traditional design and development processes of e-commerce applications for improving user experience in using the service. Specifically, we used Node.js server to replace the web server and used React.js library to develop user interfaces for the clients, thereby helping ridesharing participants use BGRS in a way similar to and compatible to their e-commerce usage experience.

5.3.3. Complexity

Blockchain is not just an information technology, but also many other things, including: a distributed ledger, a trust network, a cryptographic system, a consensus/coordination mechanism, a means of offering decentralized services, a tool proving the existence of any digital object at a particular time and so on [44]. The cross-disciplinary and inevitably complex nature of blockchain raises the barrier for businesses to implement blockchain applications [16,17]. Aiming to lower such a barrier, this study demonstrates a systematic and feasible way to build blockchain-based applications by integrating blockchain into the traditional design and implementation of e-commerce services. Moreover, Gartner Research also predicts that blockchain will reach mainstream adoption in ten years [45], while the impact of complexity on building blockchain applications will be alleviated in the next five to ten years.

5.3.4. Trialability

We designed and implemented BGRS in a way that participants can try to use it similar to how they access popular online services (such as e-commerce services and SNS). Furthermore, interested developers and managers may try to build desired blockchain applications by referencing the procedures described in this article. Indeed, many innovative blockchain-based applications have been designed and deployed in multiple industries including financial services, marketing, supply chain and logistics, ICT, agriculture, pharmaceutical, circular economy and other sectors such as waste management [8,13,16,17,31,33,46]. Judging from the exploding number of trials testing blockchain applications in various industries, the trialability of blockchain-based services is practically increasing, beyond inflated expectation, illusionary hype and speculation [47].

5.3.5. Observability

Although the observability of blockchain applications used to be low [46], we have continuously witnessed more pioneering trials (including this BGRS prototype) and industrial implementations of blockchain applications announced with use cases, reports and publications for us to reference and follow. We expect that the observability of blockchain applications will significantly increase in next few years, as more and more blockchain services and applications are moving to production in 2022 and beyond with huge demand and interest in integrating and interacting with blockchain to enhance enterprise productivity [47]. We may soon observe more mainstream firms begin to adapt blockchain technology into their business processes [47].

5.4. Research Contribution: A Recap

Following the discussion about and evaluation of the blockchain-based ridesharing system that we have built to achieve the goals of sustainability in environmental, technological and social terms, it is perhaps appropriate to end this section by summarizing the contribution of this research as follows:

(1) This study demonstrates a feasible and applicable approach to use blockchain technology and smart contracts to develop a dedicated, distributed ridesharing service system that enables ridesharing organizers and riders to efficiently find ridesharing counterparties/partners.
(2) By mitigating the shortcomings of single point of failure, lack of transparency and security risks faced by the centralized model, the proposed decentralized, distributed

system leverages its blockchain characteristics of transparency, immutability, process automation and disintermediation to improve service processes execution.

(3) The proposed system facilitates the development of a distributed governance model, enables decentralized value creation and distribution and better supports social sharing and the sharing economy, especially by eliminating intermediaries that charge high service fees.

(4) The proposed system implements three special features (identity authentication and trust evaluation, reasonable estimate of ride fare and time, and reassurance of departure/destination locations) to improve its service quality and particularly enhance the service trust to benefit ride safety.

(5) While prior research has rarely reported the design and implementation details of blockchain-based system supporting the sharing economy, this paper provides such details covering the development tools, the deployment environment and the deployed smart contracts for the proposed ridesharing system.

6. Concluding Remarks, Limitation and Future Work

This study explores the application of blockchain and its affiliated technology, smart contracts, to innovate ridesharing services, particularly for taking advantage of blockchain and smart contract-enabled features including transparency, traceability, immutability, process automation and disintermediation. In addition to creating value from better use of underutilized vehicles, the proposed blockchain-based group/crowd ridesharing service (BGRS) system can effectively increase vehicle occupancy rate as compared with TNCs (e.g., Uber and Lyft), reduce traffic congestion and emission/pollution and facilitate sustainable circulation. Through the blockchain-based and smart contract-enabled features, BGRS can help ridesharing participants with dedicated and integrated services to improve transportation costs, service transparency, convenience, quality, security and trust. Utilizing the features of process automation and disintermediation, the proposed blockchain service architecture (see Figure 5) and platform environment (see Figure 6) can help service providers develop a distributed governance model for decentralized value creation and distribution in a transparent and fair way [20]. Not only can such distributed model help assess and distribute fair shares of value created by the sharing economy to its contributors, but achieve transparent and fair governance in terms of equal opportunity of participation and fair distribution of governing power, thus facilitating social sharing and sharing economy [20].

Technology usability, together with its acceptance, is undoubtedly a highly important arena deserving further investigation to fill research gaps of blockchain applications [13], mainly because it is one of the key barriers facing incumbent companies when they try to use blockchain but encounter technical difficulties [15–17,48]. Correspondingly, this paper tries to demonstrate how to effectively design, implement, and evaluate blockchain-based and smart contract-enabled applications, by describing relevant technical procedures related to BGRS design, implementation and evaluation. In so doing, we mainly used traditional OOA/OOD/OOP development approaches, to derive or generate use case diagrams, usage scenarios, service functions and software codes for classes and smart contracts. In addition to describing traditional development approaches, we presented the important technical part in Section 3.2 for adapting blockchain and smart contracts into our proposed ridesharing service. We truly hope the details of technical design and implementation of a BGRS can serve as a blockchain application prototype for developers, practitioners and managers to reference while building desired blockchain applications. While the extant studies of blockchain-based applications rarely report the details of implemented smart contracts [31], this paper particularly contributes by not only proposing a layered system architecture adapting blockchain and smart contracts into the desired ridesharing service but also presenting the implementation details covering the development tools, the deployment environment, and the developed smart contracts.

It is highly important to design a blockchain application system with reasonably good performance in terms of minimizing its computational and communicational overheads [10], and this research does not thoroughly explore the issue of system performance, particularly for achieving higher transaction throughput and lower latency. In addition to the performance issue, this study inevitably faced several other limitations. Firstly, the blockchain-based ridesharing system constructed by this study suffers technical limitations related to transaction throughput, service latency, scalability, security and privacy, which urgently need future work to investigate and improve their technical solutions in a comprehensive way [27]. Secondly, the blockchain of the proposed ridesharing system will continue to grow in length and size as it evolves through its lifecycle. As such, another limitation of our blockchain-based ridesharing system is the length and size of the blockchain, which can lead to efficiency and even feasibility issues due to various communication constraints [49]. Thirdly, the results of the blockchain-based system built by this research is limited by the acceptance of its targeted users. As suggested by prior research [13], empirical research to evaluate the usability and user adoption of the system for clarifying uncertainties of the blockchain-based ridesharing services is considered as undoubtedly important and valuable future work, which can not only shed light on the effectiveness and usefulness of the proposed ridesharing system but also guide the practical design and implementation of required blockchain-based services to achieve sustainability goals. Lastly, as mentioned in Section 3.2, blockchain systems using oracles as data bridges to facilitate information exchange with outside world may suffer from oracle problems [37], and our blockchain system is no exception because it acquires off-chain information. The oracle problem arises because the blockchain collects information from the outside world through oracles, yet those oracles cannot check the authenticity of the information, thus creating the reliability risk of information on a blockchain [37,50]. To cope with oracle problems, a recent article argues for treating oracle as third-party service organization and proposes control objectives to address the problems, but unfortunately it does not present needed control activities to achieve the control objectives [37]. After all, solving oracle problems is a non-trivial research agenda that requires serious future work to address this complex problem.

From a multidisciplinary perspective, service systems with feasible implementation approaches to optimize performance may influence users' behavior in perception, adoption and satisfaction. We plan to explore various blockchain configurations against on-chain/off-chain data storage and computation, block sizes and degrees of centralizations to improve BGRS performance. Indeed, investigating various technological approaches to optimize service performance in future research is definitely worth trying. Developing and deploying blockchain-based services is an exciting business arena with valuable research and application potentials [12,13,16,17,19,31,33,34,38,48]. Research into detailed designs of commercially-applicable and valuable blockchain services in various sectors is helpful not only to broaden the scope of its practical applications but also to shed light on developing innovative theories in terms of the design, implementation and management of blockchain-based business services and applications.

This research addresses the issue of shared transportation (focusing on ridesharing) for achieving the sustainability of urban transportation in the context of new technological revolution, particularly for harvesting the benefits of blockchain technology. Ridesharing may, more or less, supplement urban public transportation and therefore public administrative authorities can consider the potential of blockchain-based ridesharing for their planning and policy of sustainable public transportation. In addition, academic researchers and industry practitioners may also look into various methodologies and case studies for applying the blockchain-based ridesharing services described in this paper to various applicable scenarios with the ultimate goal of sustainable public transportation.

Author Contributions: Conceptualization, S.E.C., E.C.C. and Y.C.; methodology, S.E.C.; software, Y.C.; validation, E.C.C. and Y.C.; resources, S.E.C.; writing—original draft preparation, S.E.C. and E.C.C.; writing—review and editing, E.C.C. and S.E.C.; supervision, S.E.C.; project administration, S.E.C.; funding acquisition, S.E.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan, under contract numbers MOST-109-2221-E-005-043 and MOST-110-2221-E-005-079.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A

The screenshots for checking the correctness of service functions are available at: https://drive.google.com/drive/folders/1A76yigP3XdEePg2WRfzAI5rf_sn-inmL?usp=sharing (accessed on 15 March 2022).

References

1. Chen, A.; Lu, Y. Protective behavior in ride-sharing through the lens of protection motivation theory and usage situation theory. *Int. J. Inf. Manag.* **2021**, *61*, 102402. [CrossRef]
2. Distelmans, M.; Scheerlinck, I. Institutional strategies in the ridesharing economy: A content analysis based on Uber's example. *Sustainability* **2021**, *13*, 8037. [CrossRef]
3. Li, C.Y.; Fang, Y.H. The more we get together, the more we can save? A transaction cost perspective. *Int. J. Inf. Manag.* **2022**, *62*, 102434. [CrossRef]
4. Erhardt, G.D.; Roy, S.; Cooper, D.; Sana, B.; Chen, M.; Castiglione, J. Do transportation network companies decrease or increase congestion? *Sci. Adv.* **2019**, *5*, eaau2670. [CrossRef]
5. Wosskow, D. *Unlocking the Sharing Economy: An Independent Review*; Department for Business, Innovation and Skills: London, UK, 2014. Available online: http://gesd.free.fr/unlocksharing.pdf (accessed on 25 July 2021).
6. SFCTA. *TNCs Congestion: Draft Report*; San Francisco County Transportation Authority: San Francisco, CA, USA, 2018. Available online: https://archive.sfcta.org/sites/default/files/content/Planning/TNCs/TNCs_Congestion_Report_181015_Final.pdf (accessed on 25 July 2021).
7. Gandia, R.; Antonialli, F.; Nicolaï, I.; Sugano, J.; Oliveira, J.; Oliveira, I. Casual carpooling: A strategy to support implementation of Mobility-as-a-Service in a developing country. *Sustainability* **2021**, *13*, 2774. [CrossRef]
8. Park, A.; Li, H. The effect of Blockchain technology on supply chain sustainability performances. *Sustainability* **2021**, *13*, 1726. [CrossRef]
9. Mouratidis, K.; Peters, S.; van Wee, B. Transportation technologies, sharing economy, and teleactivities: Implications for built environment and travel. *Transp. Res. D Trans. Environ.* **2021**, *92*, 102716. [CrossRef]
10. Namasudra, S.; Sharma, P. Achieving a decentralized and secure cab sharing system using blockchain technology. *IEEE Trans. Intell. Transp. Syst.* **2022**, in press. [CrossRef]
11. Rodrigues, B.; Bocek, T.; Lareida, A.; Hausheer, D.; Rafati, S.; Stiller, B. A blockchain-based architecture for collaborative DDoS mitigation with smart contracts. In Proceedings of the IFIP International Conference on Autonomous Infrastructure, Management and Security, Zurich, Switzerland, 10–13 July 2017; pp. 16–29. [CrossRef]
12. Lahkani, M.J.; Wang, S.; Urbański, M.; Egorova, M. Sustainable B2B E-commerce and blockchain-based supply chain finance. *Sustainability* **2020**, *12*, 3968. [CrossRef]
13. Chang, S.E.; Chen, Y.C. When blockchain meets supply chain: A systematic literature review on current development and potential applications. *IEEE Access* **2020**, *8*, 62478–62494. [CrossRef]
14. Yadav, S.; Singh, S.P. Blockchain critical success factors for sustainable supply chain. *Resour. Conserv. Recycl.* **2020**, *152*, 104505. [CrossRef]
15. Kumar, N.M.; Chopra, S.S. Leveraging blockchain and smart contract technologies to overcome circular economy implementation challenges. *Sustainability* **2022**, *14*, 9492. [CrossRef]
16. Peres, R.; Schreier, M.; Schweidel, D.A.; Sorescu, A. Blockchain meets marketing: Opportunities, threats, and avenues for future research. *Int. J. Res. Mark.* **2022**, *102*, 451–464. [CrossRef]

17. Huang, L.; Zhen, L.; Wang, J.; Zhang, X. Blockchain implementation for circular supply chain management: Evaluating critical success factors. *Ind. Mark. Manag.* **2022**, in press. [CrossRef]
18. Chang, S.E.; Chen, Y.C.; Wu, T.C. Exploring blockchain technology in international trade. *Ind. Manag. Data Syst.* **2019**, *119*, 1712–1733. [CrossRef]
19. Utz, M.; Johanning, S.; Roth, T.; Bruckner, T.; Strüker, J. From ambivalence to trust: Using blockchain in customer loyalty programs. *Int. J. Inf. Manag.* **2022**, in press. [CrossRef]
20. Pazaitis, A.; De Filippi, P.; Kostakis, V. Blockchain and value systems in the sharing economy: The illustrative case of Backfeed. *Technol. Forecast. Soc. Change* **2017**, *125*, 105–115. [CrossRef]
21. Basselier, R.; Langenus, G.; Walravens, L. The rise of the sharing economy. *NBB Econ. Rev.* **2018**, *3*, 57–78. Available online: https://www.chrisbauman.com.au/Content/Documents/Shared%20economy.pdf (accessed on 25 July 2021).
22. Perez, C. Technological revolutions, paradigm shifts and socio-institutional change. In *Globalization, Economic Development and Inequality: An Alternative Perspective*; Reinert, E.S., Ed.; Edward Elgar Publishing: Cheltenham, UK, 2004; pp. 217–242.
23. Arrowsmith, L. Smart cities: Business models, technologies and existing projects. In *Information Technology Service Research Report*; IHS Technology: London, UK, 2014.
24. Kenworthy, J.R.; Laube, F.B.; Newman, P.; Barter, P.; Raad, T.; Poboon, C.; Guia Jr, B. *An International Sourcebook of Automobile Dependence in Cities 1960–1990*; University Press of Colorado: Louisville, CO, USA, 1999.
25. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 25 July 2021).
26. Pilkington, M. Blockchain technology: Principles and applications. In *Research Handbook on Digital Transformations*; Olleros, F.X., Zhegu, M., Eds.; Edward Elgar Publishing: Cheltenham, UK, 2016; pp. 225–253. Available online: https://ssrn.com/abstract=2662660 (accessed on 25 July 2021).
27. Chang, S.E.; Chen, Y.C.; Lu, M.F. Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process. *Technol. Forecast. Soc. Change* **2019**, *144*, 1–11. [CrossRef]
28. Szabo, N. Smart Contracts. Unpublished manuscript. 1994. Available online: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (accessed on 25 July 2021).
29. Szabo, N. Smart contracts: Building blocks for digital free markets. *EXTROPY #16 J. Transhumanist Thought* **1996**, *8*, 50–53 & 61–64. Available online: https://ia801806.us.archive.org/24/items/extropy-16/Extropy-16.pdf (accessed on 25 July 2021).
30. Zamani, E.D.; Giaglis, G.M. With a little help from the miners: Distributed ledger technology and market disintermediation. *Ind. Manag. Data Syst.* **2018**, *118*, 637–652. [CrossRef]
31. Leduc, G.; Kubler, S.; Georges, J.P. Innovative blockchain-based farming marketplace and smart contract performance evaluation. *J. Clean. Prod.* **2021**, *306*, 127055. [CrossRef]
32. Khanji, S.; Assaf, S. Boosting ridesharing efficiency through blockchain: Greenride application case study. In Proceedings of the 2019 10th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 11–13 June 2019; pp. 224–229. [CrossRef]
33. Wang, X.; Ma, D.; Hu, J. Recycling model selection for electronic products considering platform power and blockchain empowerment. *Sustainability* **2022**, *14*, 6136. [CrossRef]
34. Savelyev, A. Contract law 2.0: 'Smart' contracts as the beginning of the end of classic contract law. *Inf. Commun. Technol. Law* **2017**, *26*, 116–134. [CrossRef]
35. Cox, K.; Phalp, K.T. Practical experience of eliciting classes from use case descriptions. *J. Syst. Softw.* **2007**, *80*, 1286–1304. [CrossRef]
36. Larman, C. *Applying UML and Patterns—An Introduction to Object-Oriented Analysis and Design and the Unified Process*; Prentice-Hall: Englewood Cliffs, NJ, USA, 2002.
37. Sheldon, M.D. Auditing the blockchain oracle problem. *J. Inf. Syst.* **2021**, *35*, 121–133. [CrossRef]
38. Zulfiqar, M.; Tariq, F.; Janjua, M.U.; Mian, A.N.; Qayyum, A.; Qadir, J.; Sher, F.; Hassan, M. EthReview: An Ethereum-based product review system for mitigating rating frauds. *Comput. Secur.* **2021**, *100*, 102094. [CrossRef]
39. Thinkwithgoogle.com. 15.3 s is the Average Load Time for a Mobile Page. Available online: https://www.thinkwithgoogle.com/consumer-insights/consumer-trends/mobile-shopping-ecosystem-page-load-speed/ (accessed on 8 July 2022).
40. Support.google.com. In *Deciphering "Site Speed": Where to Find the Metrics, and What They Mean*; Google LLC: Mountain View, CA, USA, 2022. Available online: https://support.google.com/analytics/answer/2383341?hl=en#zippy=%2Cin-this-article (accessed on 10 October 2022).
41. Dean, B. *We Analyzed 5.2 Million Desktop and Mobile Pages: Here Is What We Learned about Page Speed*; Backlinko: Boston, MA, USA, 2019. Available online: https://backlinko.com/page-speed-stats (accessed on 10 October 2022).
42. DiMaggio, P.; Garip, F. Network effects and social inequality. *Annu. Rev. Sociol.* **2012**, *38*, 93–118. [CrossRef]
43. Rogers, E.M. Diffusion of innovations: Modifications of a model for telecommunications. In *Die Diffusion von Innovationen in der Telekommunikation*; Stoetzer, M.W., Mahler, A., Eds.; Springer: Berlin/Heidelberg, Germany, 1995; pp. 25–38. [CrossRef]
44. Swan, M. *Blockchain: Blueprint for a New Economy*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.

45. Panetta, K. *Top Trends in the Gartner Hype Cycle for Emerging Technologies*; Gartner, Inc.: Stamford, CT, USA, 2017. Available online: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/ (accessed on 25 July 2021).
46. Friedlmaier, M.; Tumasjan, A.; Welpe, I.M. Disrupting industries with blockchain: The industry, venture capital funding, and regional distribution of blockchain ventures. In Proceedings of the 51st Annual Hawaii International Conference on System Sciences (HICSS), Hilton Waikoloa Village, HI, USA, 3–6 January 2018; pp. 3517–3626. [CrossRef]
47. Benjamin, G. *Gartner Blockchain Hype Cycle 2021: Where We Are & What's Next*; Mini Blockchain: Zurich, Switzerland, 2021. Available online: https://imiblockchain.com/gartner-blockchain-hype-cycle (accessed on 15 March 2022).
48. Kumar, A.; Liu, R.; Shan, Z. Is blockchain a silver bullet for supply chain management? Technical challenges and research opportunities. *Decis. Sci.* **2020**, *51*, 8–37. [CrossRef]
49. Chang, S.E.; Chen, T.-Y. Application of blockchain technology to podcast-based enterprise content marketing. *IEEE Access* **2022**, *10*, 106324–106333. [CrossRef]
50. Caldarelli, G. Understanding the blockchain oracle problem: A call for action. *Information* **2020**, *11*, 509. [CrossRef]