Journal of
Theoretical and Applied Electronic Commerce Research

MDPI

Article

Strata Fee Management in Condominiums via Smart Contracts

Liam Scholte †, Rui Wang †, Kwok Keung Chung † and Michal Aibin *,†

Khoury College of Computer Sciences, Northeastern University, Vancouver, BC V6B 1Z3, Canada
* Correspondence: m.aibin@northeastern.edu
† Current address: 410 W Georgia St 1400, Vancouver, BC V6B 1Z3, Canada.

Abstract: Condominiums and similar properties use a stratum to manage daily operations, and owners fund it through strata fees. While existing strata fee management systems may be able to handle such funds, such systems could be more inherently transparent. It is possible to leverage the digital ledger from blockchain networks and smart contracts to build a fully transparent strata fee management system. This paper proposes designing a strata fee management system based on a smart contract in the Ethereum network. Both strata corporations and homeowners can interact with the smart contract to execute common procedures such as paying strata fees and handling expenses. Using smart contracts for strata fee management, it is believed that the chance of fraud by strata corporations is lowered compared to other systems.

Keywords: blockchain; ethereum; strata management; smart contract; decentralization

check for updates

Citation: Scholte, L.; Wang, R.; Chung, K.K.; Aibin, M. Strata Fee Management in Condominiums via Smart Contracts. *J. Theor. Appl. Electron. Commer. Res.* **2023**, *18*, 1157–1176. https://doi.org/10.3390/jtaer18030059

Academic Editor: Jemal Abawajy

Received: 10 May 2023
Revised: 13 June 2023
Accepted: 19 June 2023
Published: 4 July 2023

1. Introduction

A condominium is a building with individually owned units in the housing market. Although units in a condo are individually owned, other spaces—such as hallways, lobbies, gardens, and gyms—are collectively owned. Strata exist to ensure that certain services are provided to the residents and to handle the maintenance of the shared spaces. It is responsible for budgeting and paying these everyday expenses. Strata is considered a legal entity capable of entering contracts or being sued [1].

A condo owner has the responsibility of paying a strata fee, typically on a monthly basis. The expectation is that such funds are used responsibly by the strata. It is essential for owners to verify how and when the funds from strata fees are being used. Equally important is providing owners with a mechanism for recuperating unused funds.

Strata fees can be decentralized through blockchain technology, which produces a distributed digital ledger not controlled by a centralized authority such as the strata corporation. The distributed nature makes the digital ledger resistant to tampering [2], and the information stored within is immutable and available for anyone to see [3], creating transparency and eliminating trust issues that may arise between condo owners and the strata.

A smart contract is a self-executing program stored and replicated on a blockchain, making it immutable. The terms of a smart contract are automatically executed when predetermined conditions are met [4]. Using smart contracts, it is possible to build a decentralized application (dApp) in which homeowners can pay strata fees and verify the balance of operating funds and how they are being used in real-time. In contrast, centralized and trusted third parties need to complete traditional contracts, thus incurring longer execution times and additional costs [5].

In the context of condominium strata, smart contracts can be utilized in purchasing and servicing contracts, such as gutter cleaning, elevator services and maintenance, and roofing renewals with third parties. Contract terms such as the scope of work, completion schedule, use of material, required permits, and worker qualification [6] can be coded as variables in a smart contract. Monitoring can also be achieved in the dApp by querying the contract

variables' values. A comparison between the budget and the actual expense is available on demand. Hence the level of transparency is enhanced. A mocking environment of the proposed smart contract can be set up for owners and strata committees to understand the contract terms better, especially to familiarize themselves with the conditions that lead to the execution of the payment to contractors.

Smart contracts are not only useful in the context of strata fees but also in the context of communal insurance. An important aspect of home ownership is having insurance that protects the owner from damage that may occur. Traditionally, this would take the form of paying fees to an insurance company directly. This has a significant drawback in that the insurance company is a business that relies on not having to refund money to its customers if no incident has occurred. Alternative insurance can take the form of individual owners within a condominium pooling their money together in the event that one owner eventually needs it. Since this is not a business, funds can be returned to owners if needed.

The insurance industry relies heavily on complex processes between transacting parties to initiate, maintain and close various policies. The transaction processing time, payment settlement time, and process execution security are significant concerns for the insurance industry [7]. Therefore, it seeks solutions to minimize operating costs and the time it takes to process loss claims. Blockchain and smart contracts can make transactions more secure, processes more automated, and payments faster. As such, it is an efficient solution to the above problems [8].

This paper will apply blockchain and smart contract technology based on the Ethereum platform to improve the transparency of using strata fees. A voting system can be built into a smart contract to give everybody equal rights to decide on strata management affairs.

The remainder of this paper is organized as follows:

- Section 2 provides current uses of blockchain technology and smart contracts in areas such as supply chains and insurance.
- Section 3 presents the problem faced in current strata fee management systems and how blockchain technology can be used to improve transparency
- Section 4 details the design of a strata fee management system using smart contracts.
- Section 5 describes the uniqueness of the proposed strata fee management system compared to existing systems.
- Section 6 describes the simulation plan to verify how the proposed system solves the problem statement
- Section 7 demonstrates the results of the aforementioned system.
- Section 8 discusses the security features of our system.
- Section 9 highlights any conclusions and future work that has arisen during the implementation of the system.

2. Literature Review

The primary concept of a blockchain is its distributed nature in which it is not controlled by a centralized authority, such as a government or company, and is therefore capable of creating digital ledgers that are resistant to tampering [2]. It has a data structure similar to a linked list that is shared among all nodes on the network, where each node keeps a local copy of all its blocks starting from its genesis block. The information stored within a blockchain is immutable and available for everyone to see [3], creating transparency and eliminating trust issues that may arise when data passes through a centralized system. The combination of distributed storage and blockchain is also very important because users do not need to store much specific information on the chain but only index information [9].

A smart contract is a self-executing program stored and replicated on a distributed blockchain, at which point it becomes immutable. The terms of the smart contract are automatically executed when predetermined conditions are met. The smart contract is able to transition through predefined states as stakeholders interact with the contract [4]. In contrast, centralized and trusted third parties need to complete traditional contracts, thus

incurring long execution times and additional costs [5]. The life cycle of a smart contract comprises four phases: creation, deployment, execution and completion. Smart contracts can be deployed in public, private, or consortium blockchains platforms, such as Ethereum, Hyperledger Fabric, Corda, Stellar, Rootstock, and EOS [10].

The application of blockchain technology is becoming more and more widespread [11]. According to [12], the number of relevant articles written in 2017 was only 3. This increased by a factor of 4 in 2019 with 16 newly-written relevant articles. Some use cases include the Internet of Things (IoT), sharing economy, finance, and the public sector [13]. Supply chain management involves numerous stakeholders, including manufacturers, material providers, retailers, distributors, non-governmental organizations (NGOs), and regulatory authorities. Smart contracts can be used in determining provenance, tracking goods, executing payments, and managing reputation to improve transparency and traceability, as well as automation of contract executions [14].

Due to the potential of blockchains to enhance traceability and transparency in complex supply chains, researchers are increasingly interested in deploying blockchain-based solutions for sharing information within supply chains. Over 25% of the research in the collected literature is in the health and medical field. About 87% of the literature uses blockchain-based solutions to reduce information asymmetry or facilitate information sharing [12].

There are already explorations in different areas in the medical and healthcare field. For vaccine supply, ref. [15] proposes a blockchain-based system for tracking registration, storage, and delivery of COVID-19 vaccines and self-reporting of side effects. Ref. [16] proposed a method based on the Ethereum network to realize the distribution and delivery process of the COVID-19 vaccines. Smart contracts were used to automate the recording of events related to the distribution and delivery of vaccines. They also connected the Ethereum blockchain to off-chain storage to account for the limited available storage space on the blockchain. For personal protective equipment (PPE), ref. [9] used Ethereum smart contracts to manage orders and transactions within the PPE supply chain. It enhanced product traceability while maintaining data integrity and provenance. It also increased transparency among stakeholders. Medical waste is another concern during the pandemic, ref. [17] combined the Ethereum blockchain with the decentralized storage of Interplanetary File System (IPFS) to provide a secure, transparent, auditable, reliable, traceable and trustworthy solution for the supply chain and waste management of COVID-19 medical devices.

In the agricultural industry, the traceability of products through producers, distributors, retailers, and consumers is an area in which smart contracts are being used to document the transition of goods through the supply chain [18,19]. More specifically, in supply chain management, the Ethereum network seems to be a common choice for prototyping applications [18,20]. However, there are other choices as other systems use different networks, such as a consortium blockchain [19].

According to previous literature reviews on blockchain usage in supply chain management, it has been identified that insurance and logistics is a promising directions for future research in blockchain applications [10]. One of the existing applications of smart contracts is its ability to protect different parties from contract breaches by imposing financial penalties depending on the nature of the violations [3]. This concept extends naturally to insurance policies for producers in a supply chain. Integrating the IoT with smart contracts makes it possible to create decentralized autonomous organizations (DAO) [21]. This concept [22–25] could be applied to provide automatic insurance policies due to, for example, adverse weather patterns. A similar system, in which a farmer enters into a smart contract with an insurance provider, exists; the system will lookup current weather data such as rainfall, and depending on whether a threshold is exceeded, the farmer may receive compensation to cover their losses [26,27]. Agriculture insurance has developed across economies in South East Asia. Ref. [28] proposed a blockchain-based smart contract framework for weather-based index insurance. Ref. [26] presented an end-to-end decentralized insurance prototype covering rainfall damages.

Protecting privacy while using blockchain technology is a critical consideration, as blockchains are inherently transparent and immutable. However, various mechanisms have been developed to enhance privacy in blockchain systems. Various encryption techniques [29,30], private/permissioned blockchains [31] or zero-knowledge proofs [32] are some of them. Taking traditional financial insurance as an example, the business process can be reconstructed by public chains and private chains. Ref. [33] compared using private and public blockchains to provide insurance services through smart contracts based on Hyperledger Fabric and Ethereum. Other use cases of smart contracts in insurance include improving customer experience and reducing operating costs, risk assessment and fraud prevention, and P2P insurance [34]. It is also a trend to further incorporate IoT technology to solve problems in the insurance industry. Ref. [35] examined how IoT sensors can be integrated into smart contracts to automate smart insurance scenarios fully. For the automotive industry, on-demand coverage can be activated when needed, and the sensors' data can be used for triggering the execution of smart contracts [36]. Ref. [37] proposed a blockchain-based framework for vehicular Usage Based Insurance (UBI) and incentives in Intelligent Transportation Systems (ITS). However, UBI requires a lot of critical driving data to determine premiums, which can lead to serious privacy breaches. At the same time, existing UBI solutions rely on a centralized entity (i.e., insurance company) to manage insurance. Due to that, ref. [38] designed a decentralized and privacy-preserving UBI scheme, called DUBI, based on blockchain technology and zero-knowledge proofs. Ref. [39] proposed PRIDE, another privacy-preserving and decentralized UBI that uses a blockchain to record encrypted driving data and a smart contract running on the blockchain to calculate premiums. Unlike existing UBI schemes, PRIDE achieves security and privacy without relying on any centralized party or trusted/tamper-proof hardware. Ref. [36] created a prototype comprising a mobile app and an electronic device to be installed on customers' vehicles. The app can modify policy coverage by interacting with smart contracts while the electronic device monitors vehicle status and triggers modifications. Ref. [40] presented a smart-contract-based ecosystem for simple and transparent car insurance called CAIPY. On CAIPY, smart contracts are not a replacement but an aid to the insurance process to save costs. Stakeholders can switch to existing approaches anytime, making a trade-off between cost efficiency and process reliability. Agriculture insurance has developed across economies in South East Asia. Ref. [28] proposed a blockchain-based smart contract framework for weather-based index insurance. Ref. [26] presented an end-to-end decentralized insurance prototype covering rainfall damages.

To the best of our knowledge, there is no system for strata management based on smart contracts and blockchain, thus, it makes our research and proof of concept novel.

3. Problem Statement

When condo owners pay strata fees, an aspect of the trust is involved regarding the use of funds. Although best practices for strata exist, such as the disclosure of budgets and financial statements [41], it is up to the strata to promote this transparency. A lot owner would be at the mercy of the strata to provide these timely and truthful financial statements.

To better guarantee transparency around expenditures, this paper proposes a strata fee management system revolving around smart contracts. As all transactions within a smart contract are stored publicly on a blockchain, this makes it possible for all stakeholders to audit the strata corporation.

Figure 1 outlines a set of use cases between two sets of stakeholders. The first group of stakeholders is the owners of individual units in the condominium and are known as the strata lot owners. Their primary responsibility is to pay monthly fees to the strata corporation to fund the maintenance and shared services of the building. Additionally, they may wish to approve or reject requests made by the strata corporation to use funds, which can optionally be done automatically by setting thresholds on expenses. The second group of stakeholders is the strata corporation. Their primary responsibility is to responsibly use the money acquired from lot owners to pay for the necessary upkeep of the building. Both

stakeholder groups can be generalized to a strata observer which can check the operating funds of the strata and view withdrawals of those funds.

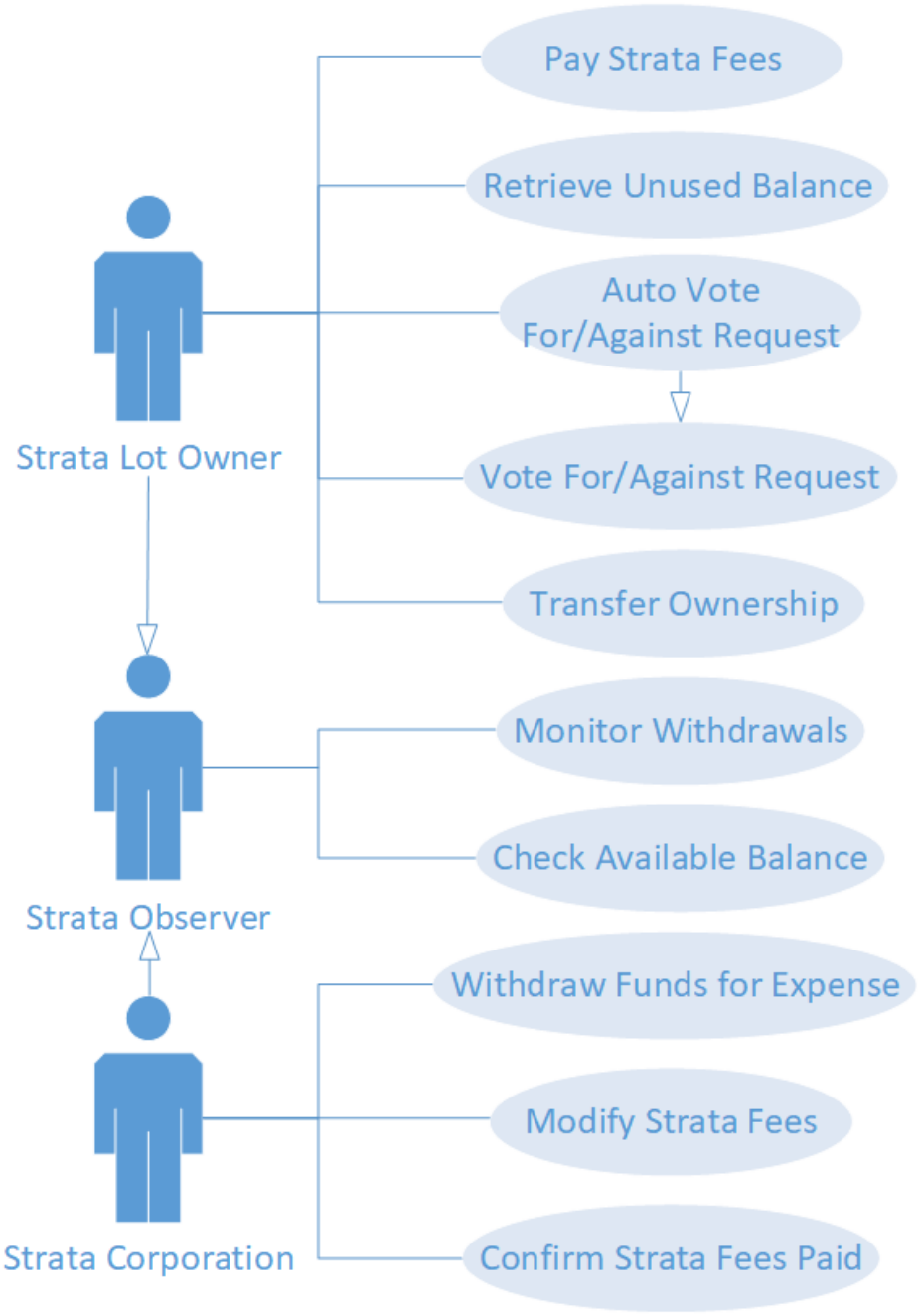


Figure 1. Use case diagram for a strata fee management system.

Stakeholders can transfer currency directly to a smart contract, held by the smart contract, instead of the strata corporation. This can realize decentralized property fee management compared to the traditional way of paying strata fees. In this system, when condo units are purchased, each owner will be required to create an account, and the amount to be paid will be determined according to the entitlement of the unit. For example, Table 1 documents a set of strata fees for each unit in a condo. Strata unit 1, for example, has an entitlement of $\frac{93}{7077}$ or approximately 1.31% of the total monthly strata fee. Therefore, unit 1 pays 1.31% of the total $\$35,339.10$, which equates to $\$464.39$. This information will establish a mapping of the unit ID and payment amount. After joining the blockchain, each

unit owner needs to pay an agreed amount of fees to the smart contract, which collects the fees monthly.

Table 1. Example of strata fees per unit [42].

Strata Lot No.	Unit Entitlement	Monthly Operating Strata Fee	Monthly Contingency Strata Fee	Total Monthly Strata Fee
1	93	$422.28	$22.11	$464.39
2	119	$565.93	$28.30	$594.23
3	115	$546.91	$27.35	$574.26
4	108	$513.62	$25.68	$539.30
5	93	$422.28	$22.11	$464.39
…	…	…	…	…
58	238	$1131.87	$56.59	$1188.46
Number of Residential Strata Lots: 58	Total Unit Entitlement of Residential Strata Lots: 7077	Total Monthly Operating Strata Fees: $33,658.29	Total Monthly Contingency Strata Fees: $1682.81	Total Monthly Strata Fees: $35,339.10

Unit owners who do not pay strata fees on time will leave traceable records on the blockchain, which will help other stakeholders know their credit history. Penalties can be designed for untrustworthy behaviour. However, this paper will not discuss this situation in detail. The system only needs to guarantee the ability to provide relevant information about the defaulter when someone breaches the contract to ensure the fee can be recovered.

The current balance of strata funds can also be obtained by querying the smart contract. This ensures transparency in the use of strata fees. The smart contract will provide a function for the stakeholders to call. For example, all stakeholders can get the value of the current balance by invoking a `getBalance()` function of the smart contract.

As the primary purpose of the strata is to pay for the maintenance and services of the building, an essential feature of the smart contract is to allow for the withdrawal of funds received via strata fees. Since the proposed system's defining property is transparency, all stakeholders' ability to monitor such withdrawals is essential. Any fees that were paid to the smart contract remain in the custody of the smart contract itself. Therefore, the strata must submit a request when funds need to be retrieved. Because the withdrawal is a transaction on the smart contract, this transaction is automatically recorded on the underlying blockchain. This makes it possible to do a lookup on the blockchain for every transaction ID ever recorded. As such, stakeholders such as the strata lot owners could identify when and how much the strata have used funds. If many owners had concerns about how those funds were being used, they only needed to request proof of their use.

Different owners often have different preferences and opinions on expense items, particularly when the expense is large or multiple options are available. An example could be whether a security system should be upgraded or replaced or which new security system should be selected. A consensus is required such that every owner is content with the decision. For less contentious matters, the strata council, comprised of elected representatives of owners, acts as the managing body of the strata corporation and decides on daily operation matters to keep it running smoothly [43]. However, sometimes the strata council member might put their interest above the best interest of the strata corporation. A voting mechanism can be built upon the smart contract to improve the transparency of the decision-making process. By allowing owners to vote directly for a particular expense item, the owners' involvement in managing their property is elevated, increasing the sense of belongings in the strata community. Individual owners can also set up their threshold so

that if a proposed expense item is below the threshold, the owner approves the expense item automatically. This feature can keep an owner's focus on the expenses they are interested in.

In relation to expenses and strata fees, it may also be necessary for the strata corporation to adjust the strata fees from time to time. This can be approached very similarly to expenses in which the strata corporation requests the fee changes, and the owners may vote for or against the change.

Finally, owners may sell their condos for various reasons. A common practice is that the owner only pays the fee until the date they sell. After the date that they sell their property, they are no longer responsible for the strata fee. This practice can be implemented in the smart contract.

4. System Design

The design of the smart contract is based on the use cases in Figure 1, from the perspective of both strata lot owners and strata corporation, which is responsible for managing and maintaining the common property and assets of the strata development for the benefit of all of its owners [1]. Therefore, the smart contract stores strata lot information, including the owners and the entitlements, and the expense items at different stages, including expenses proposed by the strata corporation, pending approval, approved or rejected, and spent. With this data, the smart contract provides these operations:

1. For all stakeholders:
   (a) Viewing the balance of operating funds held in the contract
2. For the strata corporation:
   (a) Collecting strata fees
   (b) Proposing an expense item
   (c) Proposing a strata fee change
   (d) Withdrawing money from strata contract for an approved expense
3. For strata lot owners:
   (a) Paying strata fees
   (b) Voting on a request items
   (c) Setting up auto approve and reject thresholds for expenses
   (d) Transferring strata lot ownership
   (e) Viewing the requested items

To keep the design simple, we are making several reasonable assumptions without loss of generality:

1. The ledger of the strata corporation is public information. Everyone can see the information from the blockchain.
2. All expenses will be considered instantaneous. All lot owners will be responsible for a certain percentage of the expense at the time it occurs according to the entitlement of the lot.
3. Overdue expenses will not be charged interest or other fees.

Table 2 defines a set of variables used when describing the system in the following sections.

Table 2. The set of variables used to define the strata fee management system.

Variable	Description
A	An Ethereum wallet address, which uniquely identifies the sender of transactions to a smart contract
B_L	The balance of strata fees owed by strata lot L
E	The total entitlement of all strata lots
E_L	The entitlement of strata lot L
F	The total strata fee per month across all lots
F_1	The daily strata fee per 1 entitlement in wei
F_d	The strata fee per 1 entitlement in wei for d days
F_{d_L}	Strata Fee per month of strata Lot L for d days
L	A strata lot ID which uniquely identifies each individual unit within the condo
R	The calculated refund amount when unit ownership is transferred
X	A set of expenses that have occurred
d_m	number of days of month m

4.1. Data Relationships

Several structures are introduced to describe the data mappings, shown in Figure 2. The `Unit` structure denotes the strata lot unit's basic information and is managed by the smart contract. Each unit is owned by an `Owner`, which holds an `OwnershipRecord` over the `Unit`. The smart contract tracks all owners in the condominium.

The strata corporation proposes expenses or strata fee changes, each generating a `RequestItem`, also tracked by the smart contract. These expenses are proposed when the strata request withdrawals. Each expense can be either approved, rejected, pending, or completed.

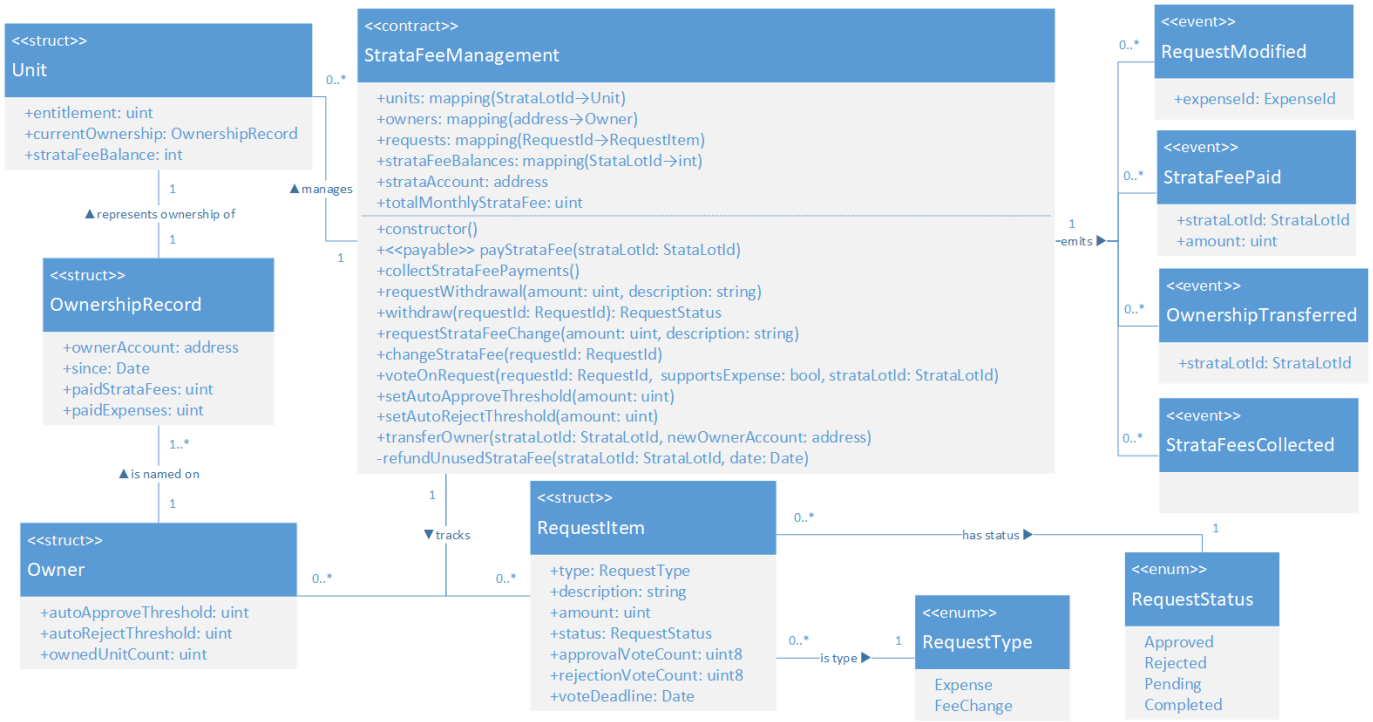


Figure 2. UML class diagram of the smart contract design.

4.2. Functions

The strata contract requires calculating the strata fee of all individual lots. Intuitively, to compute the fee of the lot ID L, the total strata fee per month F, which is determined by

the budget, the entitlement ratio of each lot E_L, and the total entitlement E. Equation (1) shows their relationship for calculating the monthly fee assuming d_m days in a month:

$$F_{d_{mL}} = F \times \frac{E_L}{E} \tag{1}$$

Another way to calculate the monthly strata fee of lot L is shown in Equation (2):

$$F_{d_{mL}} = F_1 \times E_L \times d_m \tag{2}$$

where:

F_1 is the strata fee per unit entitlement per day
E_L is the entitlement of strata lot L

Equation (2) avoids storing an entitlement ratio fraction, that is $\frac{E_L}{E}$, in the smart contract. F_1 is stored in wei, the smallest possible denomination of the cryptocurrency in the Ethereum network. This will prevent the occurrence of decimal places in any intermediate steps or final results of calculations. This makes Equation (2) advantageous over Equation (1).

When a strata property is constructed, the strata lots and their corresponding entitlements are fixed. A budget, including the strata fee and budgeted expenses, is also set up for the first year of the strata. Therefore in creating a strata fee management smart contract, the strata lots and their corresponding entitlements, as well as the budget, are needed. Hence, the constructor of the smart contract receives the strata lot IDs, their entitlements and the initial budget. Upon deployment of the smart contract, the deployer's address is assumed to be the address of the strata corporation's account. This is the only account that can request and withdraw expenses from the contract.

Also, as part of smart contract initialization, the strata corporation itself will own all lots in the building and can later transfer ownership to other people. When buyers purchase strata lots from a property developer, the `transferOwner` should be invoked. The new owner will be responsible for the strata fee from that point forward, and they can view and vote on the strata expenses. When owners want to sell their units, they may also use the `transferOwner` function in the contract to do this.

The core responsibility of the smart contract is the management of strata fees. A key aspect of this is receiving and storing strata fees paid by lot owners. The contract provides a `payStrataFee` function for precisely this purpose. This function will verify the address A of the sender via a mapping of $L \mapsto A$. Once confirmed, the contract will take ownership of the funds. Strata fees that are paid will be debited in a map from $L \mapsto B_L$, where B_L represents the credit balance of strata fees for unit L.

The strata corporation is expected to invoke `collectStrataFeePayments` monthly. When called, each unit's monthly fee F_L is incremented to B_L. If the owner does not hold the strata lot for the whole month, the strata fee F_d is prorated according to the number of days of their ownership in Equation (3):

$$F_{d_L} = F_1 \times E_L \times d \tag{3}$$

where:

d is the number of days the owner owns the strata lot

The target balance of B_L should be 0, but it is possible that it may be positive or negative. A positive amount will indicate that lot L has overdue payments, while a negative amount indicates a surplus of payments.

If the strata want to know the list of lots with overdue strata fees, it can call the function `getStrataFeeBalances`, which will return $L \mapsto B_L$. This will allow the strata to iterate through the mapping and find strata lots with large positive balances, representing overdue payments to be collected.

The other key aspect of strata fee management is the usage of funds for expenses, which includes paying for expenses or raising strata fees when annual budgets change. One mechanism is allowing the strata to withdraw funds for an expense or raise fees directly. However, the lot owners may only agree with some usages of funds. Therefore, a phased approach, which includes a voting mechanism, is taken. Withdrawals and fee changes are broken into the following steps:

1. The strata makes a request via a `requestWithdrawal` or `requestStrataFeeChange` function for a particular amount and provides a reason. This triggers a vote and creates a request item. No withdrawals or strata fee changes are performed at this time.
2. Lot owners have a set time to vote on the request. In the case of expenses, if a requested amount is above or below a certain amount, an automatic approval or rejection vote may be given depending on each lot owner's chosen thresholds. Otherwise, the lot owner may vote on a request using the `voteOnRequest` function.
3. The request may transition to an approved or rejected state if a sufficient number of votes has been received. If the request is approved, the contract will automatically attempt to complete the request by finalizing the strata fee change or sending the requested amount to the strata corporation account, at which point the request transitions to a completed state. Automatic completion is not guaranteed to occur if there are conditions that would prevent it, such as insufficient strata fee balance to perform a requested withdrawal.
4. The strata may attempt to perform the withdrawal or strata fee change via `withdraw` or `changeStrataFee`, respectively. At this point, the contract checks the result of the voting process. If only a request is approved, the smart contract will finalize the withdrawal or fee change and change the request status to completed. The strata are only required to explicitly perform this action if the request was not automatically completed.

The smart contract's exact process for handling withdrawal requests and strata fee changes is outlined in Figure 3.

Commonly, the ownership will be transferred, so it is necessary to have a function `transferOwner` to handle it. When transferring ownership, a new owner's account address must be provided, creating a new mapping of $L \mapsto A$ from that point forward. The ownership start date and paid strata fees will be reset when a new owner is assigned to the unit.

Upon transfer of ownership, the smart contract will facilitate a refund of strata fee payments to the previous owner. The refund will be calculated in Equation (4)

$$R = \min\{0, -(B_L + F_{d_{pL}})\} \tag{4}$$

where d_p is the number of days between the last date of the strata fee collected and the date of ownership transfer, meaning $F_{d_{pL}}$ is the strata fee for d_p days for unit L as per Equation (3). Note that Equation (4) is taking the balance owed since the last strata fee collection and adding the balance accrued over the d_p days that have elapsed since the last collection. Since the balance is tracked where a negative amount means a surplus of strata fees have been paid, the refund amount is multiplied by -1 to account for this and is floored at 0.

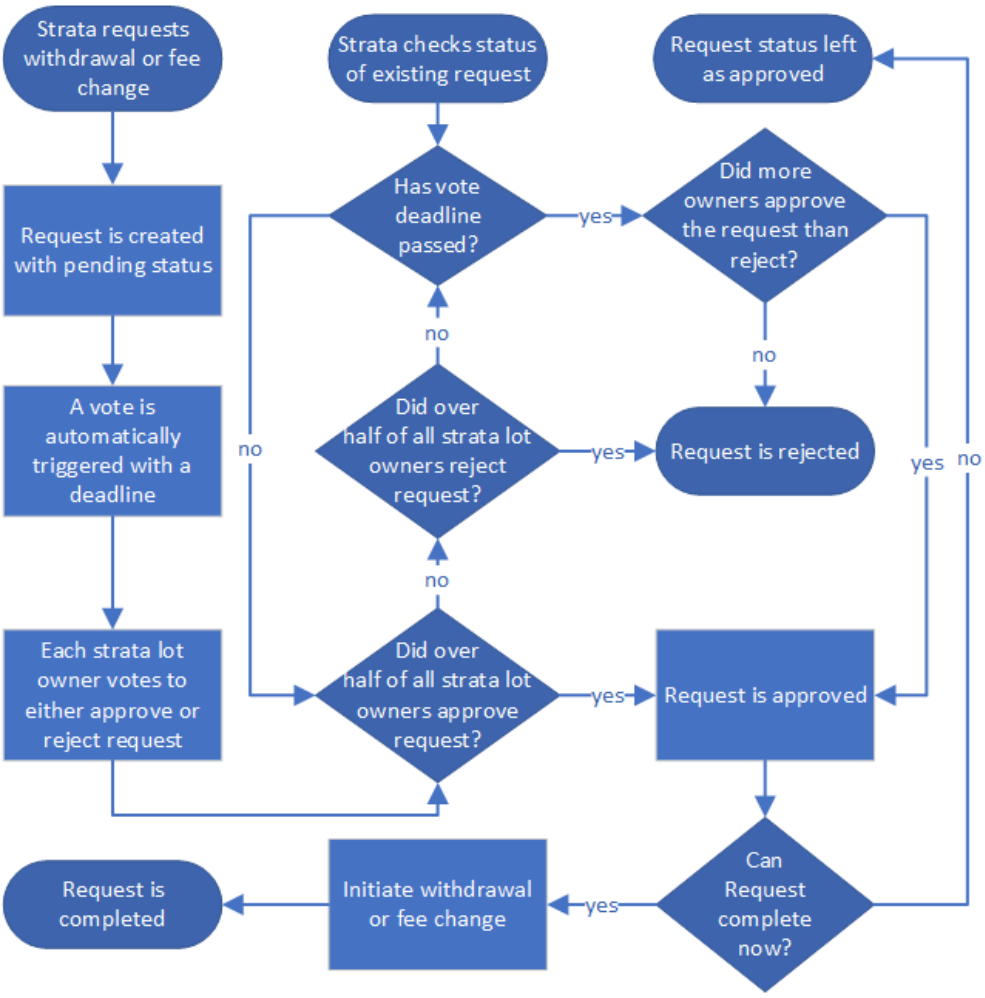


Figure 3. A flow chart outlining how to withdraw or strata fee change requests from the strata corporation are handled by the smart contract.

4.3. Message Signing

A key characteristic of the strata fee management system is the need to verify identity. Certain operations, such as voting or paying strata fees, are associated with lot ID L. With Ethereum as the underlying blockchain, every transaction is cryptographically signed using Elliptic Curve Digital Signature Algorithm (ECDSA) [44], which allows confidence in who sends each message to a smart contract. The implementation of this is outside the scope of this paper, as this is handled under the hood by the Ethereum blockchain and wallets, which will sign and verify the transactions automatically.

Each message sent to the smart contract provides an address A describing the Ethereum wallet that sent the message. To obtain useful information from this, the smart contract must have a mapping $L \mapsto A$, which will allow the contract to verify that nobody can act on behalf of a strata lot other than the one(s) that they own. As per Figure 2, the smart contract only needs to do a lookup on the owner of a unit to obtain the necessary address information. Similarly, the strata account is also held by the contract to verify the identity of the strata corporation itself.

5. State of the Art Features

The most common strata voting method at present is a meeting. Online meetings have become popular during the pandemic [45]. For example, BC Province has allowed electronic strata meetings during the pandemic [46]. The ability to hold meetings or hearings using telephone conference calls or an online application helped strata corporations comply with the orders and recommendations of BC's Provincial Health Officer. A more professional way

is to use a third-party company's voting system, which can eliminate time restraints and provide more information compared with strata vote meetings. For instance, PowerStrata is a strata management platform comply with BC legislation and SPA (Strata Property Act) [47]. It has a user-friendly and intuitive interface and provides centralized access to multiple property profiles for strata managers.

The strata fee is managed by smart contracts that automatically perform actions based on voting results. If the vote is passed, the smart contract automatically transfers the funds to the strata management account; if the vote fails, the transaction will not be performed. It ensures the decentralization of fund management and uses as well.

Most existing applications on strata management focus on reducing costs, improving communications, streamlining operations and providing record keeping [48–50]. However, they need to provide transparency in the ledger of the strata corporation. The financial information is only available when the expenses are disclosed or when the income statement balance sheets are available monthly, quarterly, semi-annually or annually for owners. Although the budget is pre-approved, the actual expense may exceed the budget from time to time due to various reasons, such as inflation.

Transparency can be achieved by implementing a smart contract to keep track of transactions. Every transaction, including every expense item and every strata fee collection, is recorded in the blockchain and available for every owner. Hence owners can monitor the most updated financial situation at their convenience. A voting system has been implemented to allow owners to achieve a consensus for each expense.

The smart contract on the blockchain provides transparency and retains the information indefinitely. This fulfils the record-keeping requirement of most jurisdictions. For example, in British Columbia, the voting results should be kept for at least six years [51]. The decentralized feature of blockchain enables multiple copies of the information in various locations and improves the robustness.

The smart contract provides an Application Programming Interface (API) with which dApps can interact, providing flexibility to develop features when necessary. Multiple dApps can be built for different stakeholders for different functions, for instance, an application can be created for monitoring the strata fee collection, and another application can be built for owners to vote for proposed expenses.

IoTs can also be utilized for specific scenarios in the future. One example is that in an elevator maintenance contract, IoT sensors can send out data indicating the condition of the elevator, and the fund withdrawal will be triggered only if the conditions are satisfied, ensuring the quality of the work.

6. Implementation Details

The smart contract is implemented as designed in Figure 2 and deployed to a Ganache private blockchain. It provides a suitable testing environment which includes Ethereum wallets with a large amount of ether. The Truffle development environment is used to compile and deploy the smart contract to the private blockchain.

A web application, such as the one in Figure 4 serves as a front end and proof of concept for a strata fee management system based on a smart contract could look like. The application is written using React and the web3.js library, allowing interaction with deployed smart contracts. The web app interface is created with MetaMask, a web browser extension for managing a person's Ethereum wallets. All data is obtained directly from a deployed strata fee management contract.

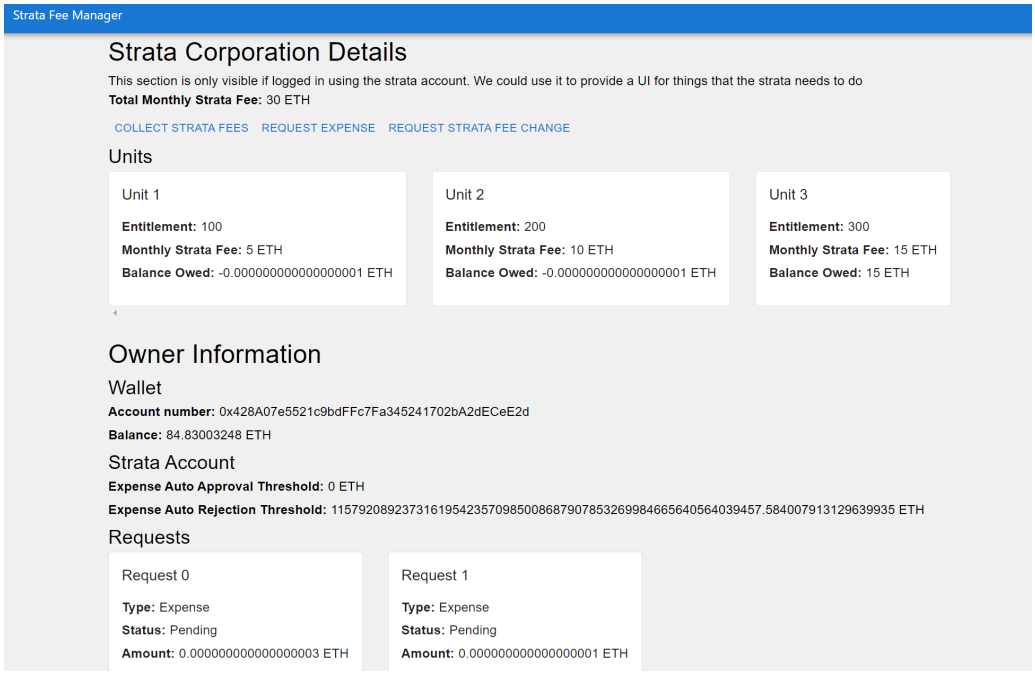


Figure 4. The React application used to demonstrate the strata fee management system.

The smart contract application manages a condominium consisting of units based on Table 1. This provides a real-world example of strata fees and entitlements with units. Strata lot fees are converted from dollars into ether or wei, which are the units of currency in Ethereum (at current prices from March 2023, 1 ether is valued around CAD 2500$. 1 ether is equivalent to 10^{18} wei, the smallest unit of ether).

The combination of both the web app and the deployed smart contract creates a dApp that provides enough capability to demonstrate every use case identified in Figure 1. Every use case was tested to verify that it is achieved and to identify the potential pitfalls of a strata fee management system based on a smart contract. By generating multiple Ethereum wallets, it is possible for a single person to simulate the behaviours of multiple actors in the strata fee management system. For example, one could act as a strata corporation with ten wallets and nine unique strata lot owners. It would be possible to transfer ownership between the accounts and to independently perform other actions, such as paying strata fees or voting on requests. Therefore, the strata corporation's and individual owners' entire life cycle can be simulated. In particular, four cycles we focused on:

1. The ownership cycle, whereby an owner sells their unit to another person
2. The strata fee collection cycle, whereby owners pay, and the strata corporation collects strata fees
3. The expense cycle, whereby the strata corporation proposes an expense, the owners vote on it, and the strata corporation withdraws the strata fee funds for the expense
4. The strata fee change cycle, whereby the strata corporation proposes a strata fee change, the owners vote on it, and the strata corporation finalizes the fee change

One of the use cases revolves around ensuring transparency with what is happening with strata fees. Ganache provides valuable tools for viewing the transaction history as well as event history on the blockchain. This gives us a proof of transparency in the strata fee management system. An example of this is in Figure 5, where the history of events emitted by the smart contract can be seen. These same events are consumed by the React application to enable dynamic updating.

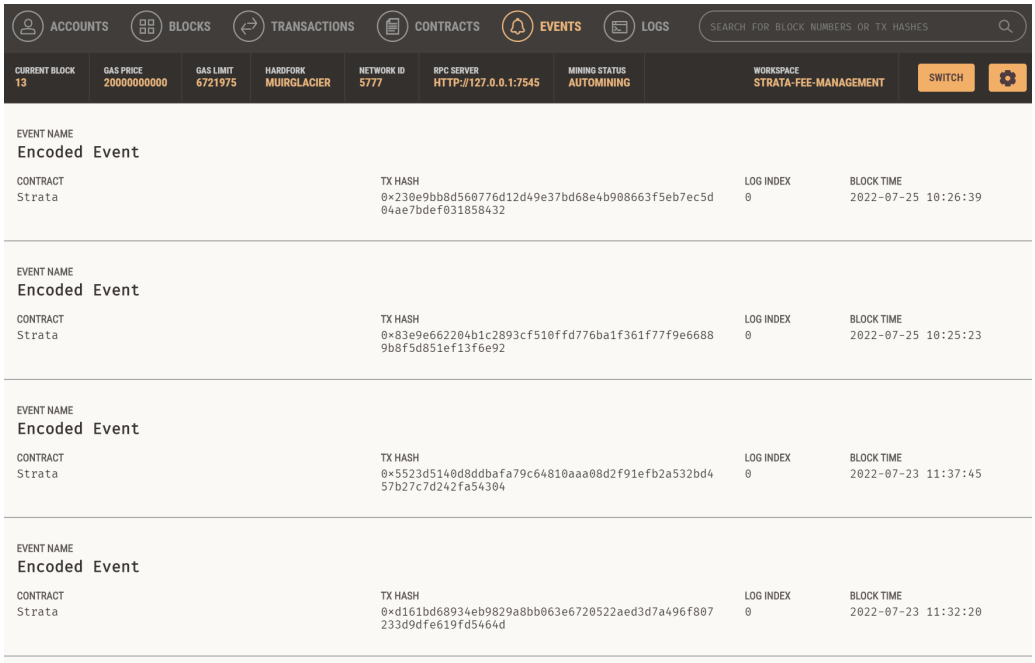


Figure 5. Example of the event history on the in the blockchain shown in Ganache.

7. Discussion

The smart contract provides an API presented in the UML of Figure 2. The contract itself is capable of validating who has the capability of invoking certain functions and rejecting transactions from unauthorized actors. For example, as per Figure 1, only strata lot owners are allowed to vote for a strata fee change request, while others, including the strata corporation, are not. The mechanism of API authorization has been built based on the digital wallets of the Ethereum network. Strata lot owners and the strata corporation are identified by the addresses of digital wallets, which are stored in the smart contract.

7.1. Ownership

At the creation of the smart contract, it is assumed that the wallet that created the smart contract is the strata corporation and that the strata corporation owns all lots. Thus, the strata corporation can later transfer the ownership to the actual owner when the property developer sells the lots to buyers.

With the use of a digital wallet address as the identifier of an owner in the smart contract, unlike other systems, owners do not need to create an account in other systems to pay strata fees or vote, which means they do not need to remember an extra username and password.

The smart contract only stores the digital wallet address of the owners. Other information about the owners is not recorded in the smart contract. As a result, it does not provide the functionality of the title search of the strata lots. The strata corporation would need to store the owners' information, such as phone numbers, separately in other systems or smart contracts for communication and other purposes.

7.2. Strata Fee Collection

A smart contract-based strata fee management system is superior to low-tech solutions that pay fees via cheque. Payment of strata fees is similar to using online banking to transfer funds. The key difference is that this system is based on the Ethereum network and uses ether as the currency. Therefore, the amounts that strata lot owners pay may fluctuate when converted into CAD. It may be possible to use a stablecoin, which is a cryptocurrency tied to another currency, such as CAD, but this possibility has yet to be explored. However,

if feasible, it would mitigate this flaw and leave the smart contract-based system on par with a more traditional strata fee management system.

An advantage of a smart contract compared to a centralized application is that the strata corporation or any other third party holds no money. It is instead held by the smart contract, which is not controlled by any single entity. This provides a layer of confidence to the strata lot owners that their funds will not be misused.

Smart contracts, unfortunately, do have limitations in what they are capable of doing. One significant limitation is that it is impossible to execute code at a predetermined time. Instead, an actor in the system is required to send a transaction to the smart contract. At this point, the contract could determine if a predetermined time has passed and either proceed with or cancel the transaction. This limitation is significant when updating how much each strata lot owes at any given time. To counteract this, a `collectStrataFeePayments` function is necessary to update the amounts owed since the last time the function was called. Non-blockchain-based software does not need to do this, as no limitation prevents code from running periodically.

While it is possible to implement floating point arithmetic within the smart contract, this increases gas consumption, a fee paid to execute transactions on the Ethereum network. The usage of floating point numbers also introduces rounding errors. Finally, floating point arithmetic must be more natively supported in Ethereum-based smart contracts. We implemented the Equation (2) instead of Equation (1) to avoid these issues. The round error is eliminated, and the dependency on external library handling floating point calculation is removed, which is crucial in financial systems. All calculations are in the whole number, saving computation costs [52].

7.3. Expense and Strata Fee Change Requests

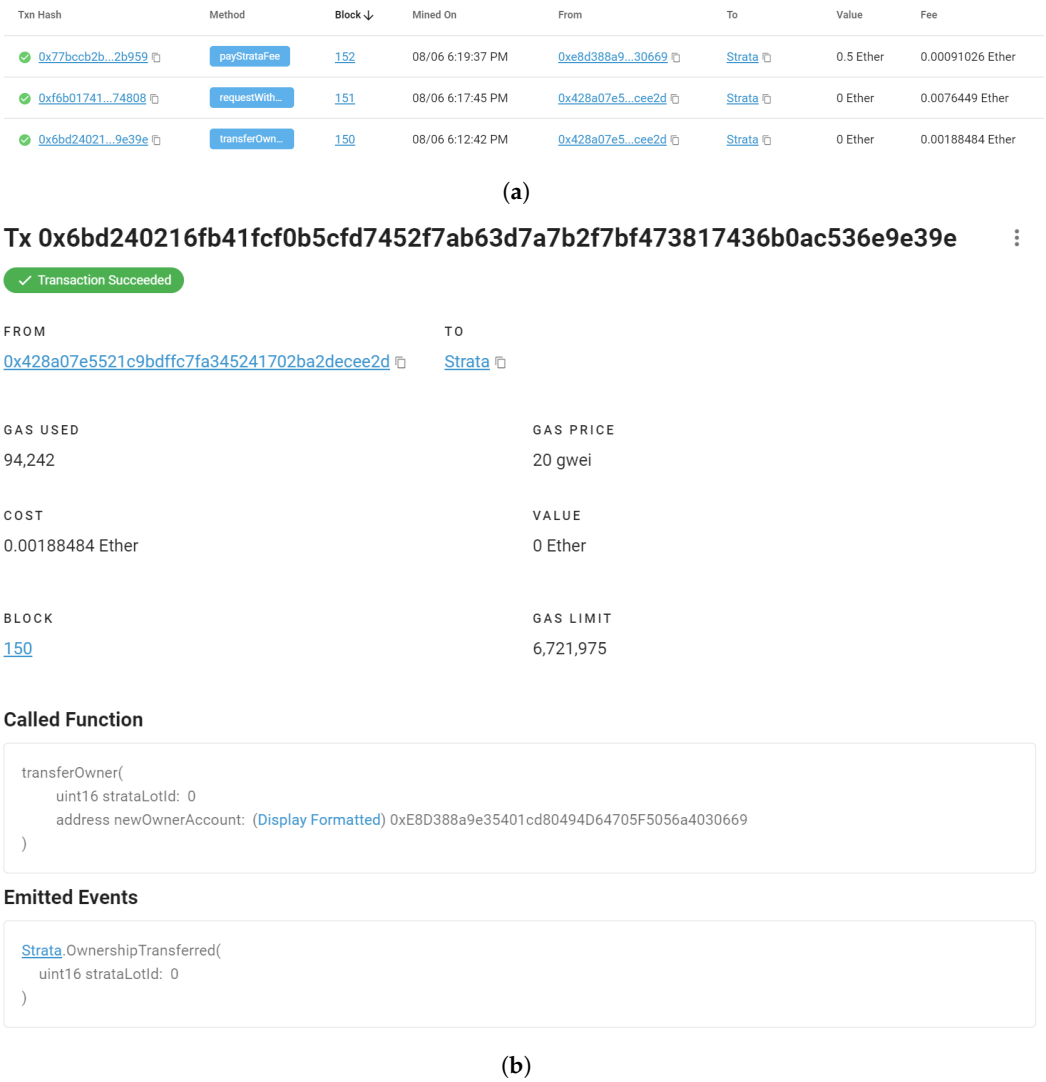
Expenses and strata fee changes need to go through an approval process by the strata lot owners. The requested amount can be entered, and the reason for the change can be provided. When the voting approval conditions are met, the smart contract will automatically transfer funds to the strata account or adjust the strata fee. This means the owners no longer need to gather and vote. Furthermore, the whole process is managed by smart contracts without any third-party intervention, which means even strata management cannot misuse the fee. Also, since the contract and transactions are on the blockchain, all voting results are transparent and tractable.

However, compared with the traditional vote system, there will be an extra cost for the lot owners in gas fees. The gas consumption will be evident if an owner has many properties. In our test, if a house owner has more than 50 units, the transaction fee of each request vote will be surprisingly high. However, we consider this an edge case, as a person will usually only hold 1 unit under a stratum.

7.4. Other Discussions

All operations are recorded as transactions in a blockchain which are publicly visible and immutable. As Figure 6 shows, all stakeholders can check the transaction history and related details to ensure there is no fraudulent behaviour from any party. The history could be used as legal evidence if there are any disputes. Furthermore, the smart contract cannot be modified once it is deployed. Every strata lot owner effectively agrees to a contract that cannot be broken.

Due to smart contracts being immutable, a new version of the contract must be created and deployed if a bug is ever found in the strata fee management contract. This will re-incur the gas costs associated with contract deployment. In addition, the data from the previous contract will need to be correctly migrated to the new contract during deployment. This data includes the state of each strata lot unit as well as the strata fee balance stored within the previous contract. Unfortunately, this is a more complex process than upgrading a traditional software system.



Txn Hash	Method	Block	Mined On	From	To	Value	Fee
0x77bccb2b...2b959	payStrataFee	152	08/06 6:19:37 PM	0xe8d388a9...30669	Strata	0.5 Ether	0.00091026 Ether
0xf6b01741...74808	requestWith...	151	08/06 6:17:45 PM	0x428a07e5...cee2d	Strata	0 Ether	0.0076449 Ether
0x6bd24021...9e39e	transferOwn...	150	08/06 6:12:42 PM	0x428a07e5...cee2d	Strata	0 Ether	0.00188484 Ether

(**a**)

(**b**)

Figure 6. A demonstration of transaction history being transparent. (**a**) Transaction history in the blockchain including gas fees. (**b**) An example of an ownership transfer transaction.

Also, using a smart contract may save some cloud storage or server expenses as this information is stored in Ethereum. As a trade-off, gas consumption costs will be associated with storing strata fee data within the smart contract. These gas costs can also be seen in Figure 6a. In particular, operations that modify the data stored by the smart contract will tend to be more expensive in terms of gas costs than operations that do not mutate data.

The blockchain provides built-in redundancy to ensure that the strata fee management system is always available. This would be on par with other web service providers that provide sufficiently high availability per their service-level agreement.

8. Security

Security in any financial system that handles transactions over communication networks is crucial [53–55]. It is the same for the strata fee management systems. To ensure the transactions are made only by legitimate users, message signing is required to verify the identity before requesting the smart contract. With Ethereum as the blockchain storing the smart contract, all transactions are signed cryptographically using Elliptic Curve Digital Signature Algorithm [56]. The signature in the message provides proof that it comes from a certain address, which is used in the smart contract to identify the ownership of a unit. The smart contract keeps a table of each strata lot and the address of its owner; when a

transaction, such as voting on a particular expense or transferring the ownership of a strata lot, is requested, the address of that request must match the one in the table.

When an expense is withdrawn from the smart contract, the expense must be transferred to the address of the strata corporation specified in the smart contract. This ensures no expense is assigned to unwanted addresses. The front-end web application, the interface between users and the smart contract, utilizes MetaMask to manage Ethereum wallets. It stores private information and performs message signing locally on the user's machine. Therefore no transfer of any credentials through the Internet is needed. The smart contract's immutability prevents the contract code from being modified maliciously. Combining all these security measures, the smart contract can ensure the processes in the smart contract works as designed by legitimate users.

9. Future Work

Gas costs are the major drawback identified in the strata fee management system. Specifically, storing strata fee data is one of the most costly aspects leading to high gas costs. If gas costs are too high, it is unlikely that strata would choose to adopt a strata fee management system based on a smart contract. With careful optimization of the smart contract code, it is possible to reduce the gas costs to something that may not be prohibitively expensive. However, it may be possible to avoid storing information such as requests or strata lot information within the smart contract and instead store it in off-blockchain storage. This strategy for reducing gas costs would require additional exploration.

Another direction of future work is to expand to various voting mechanisms. The current voting mechanism is a majority vote such that if more than half of the votes are in favour, the request gets passed. According to the legal requirements of BC [57], there are currently acceptable voting methods such as a 3/4 vote or an 80% vote. 80% Vote is similar to the majority: if more than 80% of the owners agree, it will be passed. This ratio can be determined according to the results discussed by all strata members in advance. A 3/4 vote requires that at least 3/4 of the votes are cast in favour by eligible voters at the time the vote was taken. However, if a request requiring a 3/4 vote was passed at a general meeting by persons holding less than 50% of the votes in the strata corporation, then the strata council cannot implement the resolution.

Strata voting entitlements can be varied as well. While one strata lot usually has one vote, in some cases, one can have multiple votes. For example, a non-residential strata lot can have a vote in proportion to its unit entitlement divided by either the average residential unit entitlement in mixed-use developments [58].

Author Contributions: Conceptualization, L.S., R.W., K.K.C. and M.A.; methodology, L.S. and R.W.; software, L.S., R.W. and K.K.C.; writing—original draft preparation, L.S., R.W. and K.K.C.; writing—review and editing, M.A.; supervision, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy concerns.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Strata Corporations. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/roles-and-responsibilities/strata-corporations (accessed on 2 June 2022).
2. Yaga, D.; Mell, P.; Roby, N.; Scarfone, K. *Blockchain Technology Overview*; Technical Report; 2018. Available online: https://nvlpubs.nist.gov/nistpubs/ir/2018/nist.ir.8202.pdf (accessed on 5 May 2023).

3. Kononets, Y.; Treiblmaier, H.; Rajcaniova, M. Applying Blockchain-Based Smart Contracts to Eliminate Unfair Trading Practices in the Food Supply Chain. *Int. J. Logist. Syst. Manag.* **2020**, *43*, 297–316. [CrossRef]
4. Wang, S.; Yuan, Y.; Wang, X.; Li, J.; Qin, R.; Wang, F.Y. An Overview of Smart Contract: Architecture, Applications, and Future Trends. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 108–113. [CrossRef]
5. Musamih, A.; Salah, K.; Jayaraman, R.; Arshad, J.; Debe, M.; Al-Hammadi, Y.; Ellahham, S. A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain. *IEEE Access* **2021**, *9*, 9728–9743. [CrossRef]
6. Procurement: Purchasing and Negotiating Services and Products. Available online: https://choa.bc.ca/wp-content/uploads/Sept-14_SLIDES_Procurement.pdf (accessed on 3 June 2022).
7. Raikwar, M.; Mazumdar, S.; Ruj, S.; Sen Gupta, S.; Chattopadhyay, A.; Lam, K.Y. A Blockchain Framework for Insurance Processes. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–4. [CrossRef]
8. Aleksieva, V.; Valchanov, H.; Huliyan, A. Application of Smart Contracts based on Ethereum Blockchain for the Purpose of Insurance Services. In Proceedings of the 2019 International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 8–9 November 2019; pp. 1–4. [CrossRef]
9. Omar, I.A.; Debe, M.; Jayaraman, R.; Salah, K.; Omar, M.; Arshad, J. Blockchain-based Supply Chain Traceability for COVID-19 personal protective equipment. *Comput. Ind. Eng.* **2022**, *167*, 107995. [CrossRef] [PubMed]
10. Chang, S.E.; Chen, Y. When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications. *IEEE Access* **2020**, *8*, 62478–62494. [CrossRef]
11. Guo, H.; Yu, X. A Survey on Blockchain Technology and its security. *Blockchain Res. Appl.* **2022**, *3*, 100067. [CrossRef]
12. Wan, P.K.; Huang, L.; Holtskog, H. Blockchain-Enabled Information Sharing Within a Supply Chain: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 49645–49656. [CrossRef]
13. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
14. Law, A. Smart Contracts and Their Application in Supply Chain Management. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2017.
15. Antal, C.; Cioara, T.; Antal, M.; Anghel, I. Blockchain Platform For COVID-19 Vaccine Supply Management. *IEEE Open J. Comput. Soc.* **2021**, *2*, 164–178. [CrossRef]
16. Musamih, A.; Jayaraman, R.; Salah, K.; Hasan, H.R.; Yaqoob, I.; Al-Hammadi, Y. Blockchain-Based Solution for Distribution and Delivery of COVID-19 Vaccines. *IEEE Access* **2021**, *9*, 71372–71387. [CrossRef]
17. Ahmad, R.W.; Salah, K.; Jayaraman, R.; Yaqoob, I.; Omar, M.; Ellahham, S. Blockchain-Based Forward Supply Chain and Waste Management for COVID-19 Medical Equipment and Supplies. *IEEE Access* **2021**, *9*, 44905–44927. [CrossRef]
18. Shahid, A.; Almogren, A.; Javaid, N.; Al-Zahrani, F.A.; Zuair, M.; Alam, M. Blockchain-Based Agri-Food Supply Chain: A Complete Solution. *IEEE Access* **2020**, *8*, 69230–69243. [CrossRef]
19. Yang, X.; Li, M.; Yu, H.; Wang, M.; Xu, D.; Sun, C. A Trusted Blockchain-Based Traceability System for Fruit and Vegetable Agricultural Products. *IEEE Access* **2021**, *9*, 36282–36293. [CrossRef]
20. Zhang, X.; Sun, P.; Xu, J.; Wang, X.; Yu, J.; Zhao, Z.; Dong, Y. Blockchain-Based Safety Management System for the Grain Supply Chain. *IEEE Access* **2020**, *8*, 36398–36410. [CrossRef]
21. Zhang, Y.; Wen, J. The IoT electric business model: Using blockchain technology for the internet of things. *Peer Netw. Appl.* **2017**, *10*, 983–994. [CrossRef]
22. Chohan, U.W. The Decentralized Autonomous Organization and Governance Issues. 2017. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3082055 (accessed on 2 June 2022).
23. Zichichi, M.; Contu, M.; Ferretti, S.; D'Angelo, G. LikeStarter: A Smart-contract based Social DAO for Crowdfunding. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 313–318.
24. Dwivedi, V.; Pattanaik, V.; Deval, V.; Dixit, A.; Norta, A.; Draheim, D. Legally enforceable smart-contract languages: A systematic literature review. *ACM Comput. Surv. CSUR* **2021**, *54*, 1–34. [CrossRef]
25. Beniiche, A.; Ebrahimzadeh, A.; Maier, M. The way of the DAO: Toward decentralizing the tactile internet. *IEEE Netw.* **2021**, *35*, 190–197. [CrossRef]
26. Iyer, V.; Shah, K.; Rane, S.; Shankarmani, R. Decentralised Peer-to-Peer Crop Insurance. In Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure, BSCI '21, Hong Kong, China, 7–11 June 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 3–12. [CrossRef]
27. Schwarze, R.; Sushchenko, O. Climate Insurance for Agriculture in Europe: On the Merits of Smart Contracts and Distributed Ledger Technologies. *J. Risk Financ. Manag.* **2022**, *15*, 211. [CrossRef]
28. Nguyen, T.; Das, A.; Tran, L. NEO Smart Contract for Drought-Based Insurance. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–4. [CrossRef]
29. Ben Sasson, E.; Chiesa, A.; Garman, C.; Green, M.; Miers, I.; Tromer, E.; Virza, M. Zerocash: Decentralized Anonymous Payments from Bitcoin. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; pp. 459–474. [CrossRef]

30. Noether, S.; Mackenzie, A. Ring confidential transactions. *Ledger* **2016**, *1*, 1–18. [CrossRef]
31. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
32. Sun, X.; Yu, F.R.; Zhang, P.; Sun, Z.; Xie, W.; Peng, X. A survey on zero-knowledge proof in blockchain. *IEEE Netw.* **2021**, *35*, 198–205. [CrossRef]
33. Aleksieva, V.; Valchanov, H.; Huliyan, A. Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services. In Proceedings of the 2020 International Conference Automatics and Informatics (ICAI), Arna, Bulgaria, 1–3 October 2020; pp. 1–4. [CrossRef]
34. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough? *Future Internet* **2018**, *10*, 20. [CrossRef]
35. Mahmoud, O.; Kopp, H.; Abdelhamid, A.T.; Kargl, F. Applications of Smart-Contracts: Anonymous Decentralized Insurances with IoT Sensors. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Espoo, Finland, 22–25 August 2018; pp. 1507–1512. [CrossRef]
36. Lamberti, F.; Gatteschi, V.; Demartini, C.; Pelissier, M.; Gomez, A.; Santamaria, V. Blockchains Can Work for Car Insurance: Using Smart Contracts and Sensors to Provide On-Demand Coverage. *IEEE Consum. Electron. Mag.* **2018**, *7*, 72–81. [CrossRef]
37. Singh, P.K.; Singh, R.; Muchahary, G.; Lahon, M.; Nandi, S. A Blockchain-Based Approach for Usage Based Insurance and Incentive in ITS. In Proceedings of the TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kochi, India, 17–20 October 2019; pp. 1202–1207. [CrossRef]
38. Qi, H.; Wan, Z.; Guan, Z.; Cheng, X. Scalable Decentralized Privacy-Preserving Usage-Based Insurance for Vehicles. *IEEE Internet Things J.* **2021**, *8*, 4472–4484. [CrossRef]
39. Wan, Z.; Guan, Z.; Cheng, X. PRIDE: A Private and Decentralized Usage-Based Insurance Using Blockchain. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE , Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Espoo, Finland, 22–25 August 2018; pp. 1349–1354. [CrossRef]
40. Bader, L.; Bürger, J.C.; Matzutt, R.; Wehrle, K. Smart Contract-Based Car Insurance Policies. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–7. [CrossRef]
41. Financial Best Practices for Stratas. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/roles-and-responsibilities/strata-corporations (accessed on 10 June 2022).
42. The JERVIS-Disclosure Statement. Available online: https://bccondos.net/uploads/2015/11/11/disclosure_statement-wm.pdf (accessed on 11 June 2022).
43. Strata Council-Role and Responsibilities. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/roles-and-responsibilities/strata-councils#role (accessed on 11 June 2022).
44. Wood, G.; Davis, D. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
45. COVID-19 Information for Strata Housing. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/covid-19-and-strata-housing/covid-19-information-for-strata-housing#meetings (accessed on 28 June 2022).
46. Changes to Strata Legislation in BC. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/legislation-and-changes/changes-to-legislation (accessed on 28 June 2022).
47. Power Strata Products. Available online: https://www.powerstrata.com/products (accessed on 28 June 2022).
48. StrataMax-Comprehensive Strata Software. Available online: https://www.stratamax.com/What-is-StrataMax/Comprehensive-Strata-Software (accessed on 30 June 2022).
49. StrataCommons Organizer-Building Vital and Productive Strata Communities. Available online: https://stratacommons.ca/organizer (accessed on 30 June 2022).
50. All-in-One Cloud-Base Strata and Property Management Solution. Available online: https://learn.condocontrolcentral.com/strata-management-software-demo/?utm_source=google_ad&utm_medium=cpc&utm_campaign=google_ad_search_bc&gclid=Cj0KCQjw8O-VBhCpARIsACMvVLNCbtvVajtE-xtoRknrkWHjDImTWLURv_fHiASTMB-lLgygN__MDkMaAtQDEALw_wcB (accessed on 30 June 2022).
51. Information and Record Keeping in Stratas. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/information-and-record-keeping (accessed on 30 June 2022).
52. Aibin, M.; Walkowiak, K. Resource requirements in fixed-grid and flex-grid networks for dynamic provisioning of data center traffic. In Proceedings of the 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, Canada, 15 May 2016; pp. 1–4. [CrossRef]
53. Furdek, M.; Wosinska, L.; Goścień, R.; Manousakis, K.; Aibin, M.; Walkowiak, K.; Ristov, S.; Gushev, M.; Marzo, J.L. An overview of security challenges in communication networks. In Proceedings of the 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), Halmstad, Sweden, 13–15 September 2016; pp. 43–50. [CrossRef]
54. Aibin, M.; Walkowiak, K.; Haeri, S.; Trajković, L. Traffic Prediction for Inter-Data Center Cross-Stratum Optimization Problems. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 393–398. [CrossRef]

55. Aibin, M.; Walkowiak, K. Monte Carlo Tree Search with Last-Good-Reply Policy for Cognitive Optimization of Cloud-Ready Optical Networks. *J. Netw. Syst. Manag.* **2020**, *28*, 1722–1744. [CrossRef]
56. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
57. Office of Housing and Construction Standards. Types of Strata Voting. Province of British Columbia. 4 January 2021. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/meetings-and-voting/types-of-voting#typesofvotes (accessed on 18 August 2022).
58. Strata Voting Entitlements. Available online: https://www2.gov.bc.ca/gov/content/housing-tenancy/strata-housing/operating-a-strata/meetings-and-voting/voting-process (accessed on 18 August 2022).