White Paper

chóros

Sarocha Leelamanthep
Izabela Litwin
Austin Saunders

5th December, 2018

Abstract

Chóros is a decentralized application, built on Permanent Delegated Proof of Stake (PDPoS), designed on blockchain for the Carnegie Mellon community to reserve rooms. The goal of this application is to reduce turnaround time for booking rooms, lower third-party cost, and offer non-discriminatory access to room booking. The two currencies used in Chóros are CMU Coin and Gas. CMU Coins are used to pay for room prices while gas are used for transaction fees. The room prices are calibrated based on two variables: room capacity and surge rate. The surge rate is calculated every minute based on moving 10-minute average of demand for specific building that the room is in. Implementing this system solves rising demand for specific room and distributes usage of rooms across campus.

Introduction

Blockchain in essence is a “digital ledger in which transactions made in cryptocurrency are recorded chronologically and publicly.” There are major design decisions that determine what makes a particular blockchain unique, including but not limited to its Proof System, level of anonymity, user capacity, and transaction speed. In designing Chóros, we realize we’re in a unique position, to create a new type of blockchain that preserves the integrity of decentralization blockchain was founded on, while still maintaining Carnegie Mellon University’s overarching authority.

Chóros is a decentralized application designed on blockchain for Carnegie Mellon community to reserve rooms. The goal of this application is to reduce turnaround time for booking rooms while lowering third-party cost. Chóros includes 5 major features: instant confirmation equal opportunities to book any available location on Carnegie Mellon Campus, search functionality, and calendar exports. The main benefits of this application include instantaneous booking, equal access, and autonomy to book rooms.

Key Features

Instantaneous Confirmation
After submitting a request to book room, users will receive a confirmation receipt after the transaction is confirmed by a miner in the community. This reduces the amount of wait time for the user booking room for a person to manually confirm the booking.

Nondiscriminatory Access to Rooms
Because room requests are not necessarily approved by one entity, users are able to book any room they want, as long as they are listed as available and users have adequate CMU coins and gas for the transaction.

Location Categorizations
The application includes all rooms and spaces available at Carnegie Mellon University Pittsburgh main campus. All rooms are categorized to buildings that they are in. This categorization not only is intuitive for users to find rooms but also easy for the program to detect the demand in one main location and building, which impacts the cost of booking that room.

Search Bar
Users are able to search for rooms based on room number, building, time, capacity, type of the room and availability.

Calendar Exports
Users are able to export their reservations to their Google calendar. This feature helps users synchronize room reservation with their personal schedule.

The Platform and System Design

After analyzing current room reservation system and existing blockchains, we decided to design an entire new blockchain and platform. The purpose of this platform is to provide a simple and quick way to book any available rooms on Carnegie Mellon University Pittsburgh Campus. This system suits the extremely fast and busy lifestyle of students as well as administration. In order to make this system a reality, we decided to create an entirely new Proof System for the CMU network.

PDPoS (Permanent Delegated Proof of Stake) - In this new and modified type of DPoS, there are seven permanent delegates chosen by the university, for the purpose of this paper, these delegates are planned to be the seven colleges across Carnegie Mellon. Each college will have a dedicated server that is used for mining room reservations as soon as they come in. This may seem like a bold decision but we have two main justifications for our rationale.

First, since this platform is utilized expressly to deal with university resources, using tender distributed by the university, they are going to want a hand in this process. We are storing bookings on the blockchain because it is efficient, fast and safe, which essentially makes our platform a decentralized system. However, by granting CMU the exclusive power to booking, we are reaffirming their overall power authority in booking capabilities across campus, centralizing the system to a degree. Some may think this defeats the purpose of blockchain, but it puts this blockchain in a special position because we know that every single miner is dedicated to the blockchain, and require no further funds for rewards or repairs.

Second, DPoS works. Since DPoS's first successful implementation, it's been proven to be a highly reliable and efficient proof system for blockchain. We aim to keep this high level of efficiency by cutting down on time and energy wasted on solving puzzles. Instead of wasting time on these puzzles, all block producers will be working together to create blocks for chain, ultimately resulting in no forks in the blockchain. In the case that there is a fork, the block producers are all on the same page and will just choose the longest chain. Each BP will sign one block per timestamp and as long as there are 5/7 signatures this will be the block that is passed. However, since all of these BPs originate from the same network, there is no reason for them not to approve a transaction. Furthermore, there should never be a scenario in which a full consensus isn't reached in our blockchain. The only scenario where this would happen is if two people want the exact same transaction approved at the same time. In this case, whoever pays the most gas will end up getting their transaction approved.

Front End Design

When you log onto 25live, our current competition, you are instantly faced with a slow, clunky and outdated-looking platform that is near impossible to navigate. This may have been a tolerable design system for the early 2000s when internet was slow and visual design had to be kept as simple and compact as possible. However, we are at the forefront of CS and HCI, the platform should be prettier, simpler, and easier to navigate. We decided to create a platform that is not only aesthetically pleasing, but also allows you to find what you are looking for in the shortest amount of time. For this reason, we designed a system that is visually stunning and allows you to instantly find the location you are looking for. When you visit our site, there are different ways you can go about finding your ideal location. From the homepage, you can scroll down and find the particular location and time you are looking for and book, or you can visit the search bar and instantly find the exact location you were looking for.

Additionally we realized how busy students are, and wanted to take this into consideration when we were designing the front-end of this system. We decided that by adding a calendar import option to the website we would be able to help users save time by having the option to instantly add the reservation to your calendar, instead of taking the time to manually input it and risk inputting the wrong information. Overall, we tried to design this system so that it optimized not only the amount of time you spend on the platform, but also the level of ease. Booking rooms should be as painless as possible, requiring no more than three steps to book a room: search, find, and book.

Roles of Participants

We have identified two unique user types: students and professors. The first participating group, students, will comprise the majority of the platform. The second participating group will be professors. Professors often have unexpected classes, review sessions or office hours outside of the classroom and need the ability to book rooms across campus. Comparing the specific needs of the two groups, we realized that there are large variances in the specific needs of these groups and the underlying reason behind booking the rooms. We've decided that in line with university interests, professors should have some form of advantage in booking rooms over the atypical student user. In order to address this necessary parity in booking "power", we are giving professors a permanent gas price of [student full gas tank + 1]. Students will start the semester with an allowance of gas, or a "full gas tank". But, once that allowance runs out, they will have to purchase more for real money. However, to ensure a soft cap on student gas usage, the amount of gas held in their account at any given time will be capped, effectively allowing students to not overuse gas than held in their account. Professors will always have the ability to set a gas limit above the students' cap, ensuring that their room requests take precedence over students making a conflicting booking on the same block.

Addressing the issue of money, professors will be fully subsidized by the university. School, i.e. miners, will have the power to enforce their own rules concerning setting limits on how many rooms a professor can book and whether they can book cross-collegiately. On the other hand, students, as previously mentioned, will be given an allowance of gas and coins per semester. Once they pass this limit, they will have to use personal funds to buy more, adding "real" value to the coins. Gas should stay at a consistent negligible price, similar to the majority of other blockchains. Coins will shape the economy since rooms have different monetary values.

Token Economics

CMU Coin and Gas

The two major currencies used in this system are CMU Coin and Gas. To reserve rooms, users will need to use both CMU Coin and Gas. CMU coin is the currency used to pay for the price of room. Gas is the currency used to pay for transaction costs. For every room reservation, users need to pay both price of room through CMU Coin and transaction fee, through gas. This transaction works similar to any other transactions with transaction fees, with the only exception that transactions fees for Chóros are charged in a different currency.

Room Pricing

The initial price of a room is calculated based on number of people that can fit in that room in addition to initial cost of room booking. The function that captures this room pricing is:

$$Room\ Price = \alpha + \beta * y * ln(x+1)$$

α is the constant that captures initial cost to book a room, representing the base fare of room booking. Hence, α should not be impacted by surge pricing or room size.

β is another constant sets as a multiplier to the demand and room size function. β is added as a scalar for the natural log function to augment increasing price from room size.

x is a variable for room capacity. For example, if a room can fit a maximum of 25 people, variable x would be 25. This variable captures the size of the room, meaning the price of the room will increase as the room can fit more people. Adding variable x solves the problem of one single individual booking one big room when not necessary.

$ln(x+1)$ is a function that adjusts the variable for room capacity. The natural log captures how the price of room should increase with larger room size. Additionally, because the second derivative of this function is negative, the price of the of the room increases in a decreasing order. This characteristic of natural log captures how the price difference should be greater between room size of 3 and 4 than with room size of 25 and 26. Therefore, using natural log function to adjust the room size, categorizes rooms based on functionality since the purpose of room sizes may differ significantly between smaller rooms than between relatively larger rooms. For example, a room with capacity of two people may be used by one user to practice interview while a room with capacity of three people may be used for a group project. In contrast, rooms with capacity of 25 and 26 may serve the purpose of classrooms. Lastly, 1 is added to the natural function to yield a number higher than 0 when room capacity is 1.

y is a variable for surge rate, depending on demand and volume of requests. Because the surge rate is multiplied to the natural log function, surge rate is applied to room capacity and the multiplier constant. Surge therefore means higher price for users. Surge rate is visible to users as a multiplier displayed as $X.X$. Users will be able to see the surge rate from $1.0X$. The surge rate, y, is calculated based on supply and demand, driven by volume and traffic in the specific building. For example, if there are multiple users requesting to book rooms in Tepper at 6PM of the same date, surpassing our threshold demand, the surge rate increases.

$$y = Surge\ Rate = Max(1, \gamma/z)$$

γ is defined as moving average of requests for specific time in one building in the time period of rolling 10 minutes. Hence, the surge rate will differ depending on building and time.

z is the threshold for demand for specific room and time. This is a fluctuating variable, which will be adjusted based on past data and past demand for specific building and time. With this definition, z will only be calculated after we have collected data to form a distribution of volume at specific time for different days and time. z will be set at the 30th percentile value based on the distribution at that time.

$Max(1, \gamma/z)$ function sets the minimum of the surge rate to be at least 1, even in the case where there is low traffic.

This surge rate is re-calibrated every minute, meaning the 10-minute moving average is recalibrated every minute. Re-calculating surge rate frequently based on demand not only distinguishes users based on their willingness to pay but also creates an even distribution of room usage across campus. For example, in the case which Tepper rooms have high traffics, users who only need any room for a group meeting that are not willing to pay for the search, can explore other availabilities which may not have surge pricing.

Transaction Cost and Mining Reward

In addition to paying for room price, users will pay for transaction cost to facilitate transaction in Gas. This transaction cost is the price that users pay to other miners within the community to confirm their room reservation. The transaction cost is the maximum value between floor rate and users willingness to pay to confirm the room transaction.

$$Transaction\ Cost\ = Max(Floor\ Transaction\ Cost,\ Willingness\ to\ Pay)$$

$Floor\ Transaction\ Cost$ is the minimum cost users will pay for miners to verify this transaction.

The gas that users pay as transaction cost will be transferred to miners. If users are in a rush for transaction to happen, users can pay a higher price for this transaction cost. By natural economics, miners will first verify transactions with higher transaction costs. Hence, transaction cost acts as an incentive for miners to verify room request.

Cancellation

Users are able to cancel their room reservation and get a refund for room pricing. However, users will have to pay for cancelation fee in gas. This transaction fee will be 5 times the cost of floor transaction cost. By having a cancelation fee higher than the price of the floor transaction cost, users will be penalized for not using rooms that they reserve. This prevents users from booking room far in advance unless users are somewhat certain they will need the room. The refund in CMU coin for room prices that users get back still incentivize users to cancel reservations they may not need. This solves the economics' moral hazard problem of users holding supplies that they may not need.

Coin Distribution, Usage, and Resets

As discussed in Roles of Participants section, there are two main groups of users: students and professors/administration. The professor and administration will have more priority over students. Administrations have power to override the system while professor have unlimited amount of gas and more coins. They also have access to room booking for classes before students. By implementing this timeline, we have enforced and prioritized school's necessity for reserving spaces for classes before students.

Two timelines will be implemented for student's usage. The first usage is for personal use: students will only be able to reserve rooms at a maximum of three months in advance. The timespan of three months will be a rolling time. For example, the latest date that users will be able to reserve a room from November 1st will be January 30. However, on the next day, November 2nd, users will be able to book rooms available on January 31. The second usage is for organizational use: students will be able to reserve rooms at maximum of one semester in advance after the administrations and professor have reserved rooms. However, they will need approval from the administration. In the case that users need to reserve rooms over 6 months in advance, they will need to submit a request for permission to administration for approval.

Users will be given CMU coins quarterly or every mid-semester, which will be reset with no rollover every mid-semester or end of semester. This rule prevents inflation of CMU coin supply as well as enforces the ceiling surge rate. Users will also be allotted gas quarterly. In contrast to CMU Coins, gas does roll over but resets every semesters. Resetting gas relatively less frequently incentivizes miners to continue to to verify transactions, especially because miners who are also users in other scenarios, need gas for transactions.

Scalability and Application

Right now, Chóros can only be used at Carnegie Mellon Pittsburgh main campus. Therefore, Chóros can be easily adopted by other CMU campuses, other schools and organizations and/or companies that have demand for room reservation. Schools or organizations planning to use Chóros will only have to update locations and solve for the optimal number of coins, gas limits, and time reset for usage.

Additionally, Chóros application framework can be applied to operations that work with borrowing and lending objects. For example, this framework and blockchain system can be applied to borrowing books in library or gym equipments. Users will be able to reserve gym equipments online in advance prior to usage. Users will have a different equilibrium limit in terms of coins and gas. Regardless, users will still be able to reserve these systems in advance and provide proof of their reservation upon usage.

In terms of Chóros application within CMU in the future, we would encourage students with smartphones to download the Chóros mobile application. The application would also have confirmation key that users need before entering the room they reserve. This application not only eliminates issues with middle-man needing to confirm

Future System Improvements

For future implementations of our system, there are two main areas that we want to explore to maintain a system that is fair, easy to use and flexible. The first improvement we would like to make is the ability to pool transactions. We think that adding contracts that support multiple people contributing coins towards a transaction, would be an ideal improvement to the system. This addition makes transactions among groups more transparent. Under this system, no free-riders would exist because the contract would not be executed until a set number of accounts contribute a set amount of coins. The user creating the contract would be able to figure out exactly how many people need to contribute and exactly how much each person will contribute towards the contract. Additionally, this new feature helps foster CMU's culture of collaboration. As we've already made a point about the amount of unused space on campus, we believe that if we can combine the opportunity to make use of this space more efficient with a more equitable process for group bookings, even more people would use the system.