

AQRE WHITEPAPER

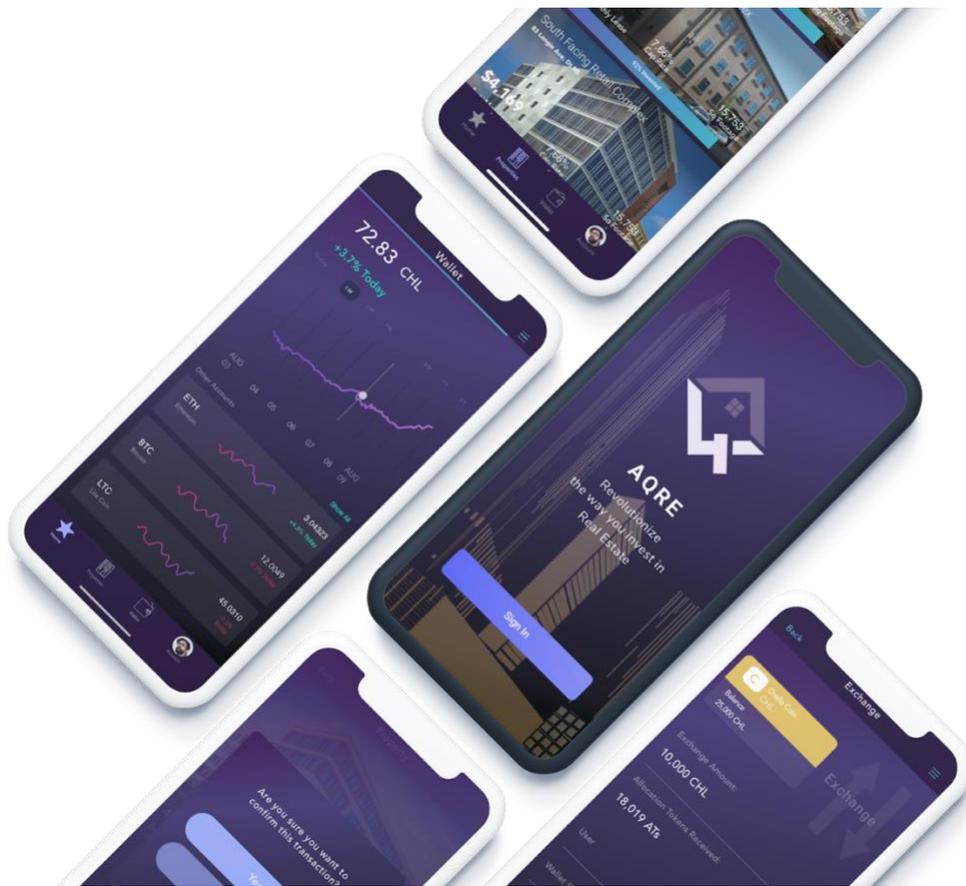


Table of Contents

<i>Introduction: Real Estate Investment Problem</i>	3
<i>Our Solution: Efficiency Through Blockchain Technology</i>	3
<i>Conceptual Description of Tokens</i>	4
Chelle Coin (CHL)	4
Premium Allocation Tokens (PATs)	4
Basic Allocation Tokens (BATs)	4
Property Token	5
Unclaimed Income - Economic Stabilizer	5
<i>Technical Description of Tokens</i>	5
Allocation Token Technical Description	5
Real Estate Properties Contract	6
<i>Conceptual Overflow</i>	7
App Intro	7
Account Creation, KYC, Ethereum Wallet	7
CHL Exchange Request and Details	8
Allocation, Process, Restrictions and Investment Returns	8
Claiming Income and Backpay	8
Unallocated Properties and Backpay	9
In-App Viewing Experiences	9
<i>Token Economics</i>	10
<i>Principal Smart Contracts Diagram</i>	11
<i>Non-Token Smart Contracts</i>	12
User KYC Contract Technical Description	12
Exchange Smart Contract	12
Investment Contract	13
ETH Dividends Contract	14
<i>Interaction Scenarios</i>	15
KYC Procedure	15
CHL Tokens Exchange	15
Investing PAT or BAT to Property	16
Depositing ETH to Distribute Property Dividends	16
Receiving Property ETH Dividends	17
User Income Formula	17

Introduction:

Real Estate Investment Problem

Investing in real estate is very challenging. Investors either need large amounts of startup capital and expertise, or they passively invest in REITs. These often lack transparency for underlying holdings, have weak portfolio growth, and have no customizability options. Investors don't have a cost-effective way to control their real estate investment conveniently. Furthermore, investors sacrifice control for cost-effectiveness when passively investing. Common passive investments made in real estate - REITs - are subject to low growth as real estate firms can only reinvest a maximum of 10% of residual income into building up their real estate book. This means that 90% of the profit is distributed to investors. While initially appealing, this restricts the real estate firm's ability to grow the underlying real estate holdings, leaving REIT holders with a potentially stagnant investment. Therefore, most real estate investment options are either extremely costly or lack features that give investors customizability features and limit investment growth.

Our Solution:

Efficiency Through Blockchain Technology

The recent cryptocurrency boom has attracted many investors and technologists. The ICO boom saw a rise in ERC20 tokens, many of which have zero utility. Chelle Corporation has found a use case for blockchain technology, namely customizable tokens. By utilizing self-executing smart contracts, Chelle Corp. is expanding access to accessible, customizable real estate investments to investors around the globe. Instead of being a marketing frill, our use of blockchain technology allows us to create a previously impossible real estate investment platform.

Chelle Coin is a way for people to access the AQRE real estate investment app. Users allocate tokens within the app and collect monthly residuals pro rata, paid out in ETH. We have a relatively higher fee of 20%, but this allows us to give token holders better continuous growth and opportunities. There are many tokens, processes, and smart contracts at play to make all of this happen in a sound economic environment. The platform operates on the Ethereum mainnet.

Ethereum is an open-source and public platform for creating decentralized online services that work based on smart contracts. The platform is decentralized, based on the blockchain, and runs as a virtual machine. Ethereum, as a platform, provides a solid (and extremely flexible) basis for this task, while also producing the highest level of functionality. By using smart contracts, the platform supports the transactional and even legal formalities in the digital world. The main reason we chose to build our platform on Ethereum is that it is incredibly flexible, customizable, and it has a strong network effect with a widely supported ecosystem.

Conceptual Description of Tokens

There are four tokens on the platform. Chelle Coin (ERC20 token used for fundraising and entrance vehicle), Premium Allocation Token (ERC20 token used for allocating to premium properties), Basic Sample Allocation Token (ERC20 token used for allocating to sample properties) and Property Tokens (ERC721 tokens used to store unique, malleable data regarding each property.)

Chelle Coin (CHL)

Chelle Coin is an ERC20 token with a fixed supply. It can be purchased on an exchange and it will have a variable market value once it trades on an exchange.

- This is the token that was used during the CHL token sale
- The number of CHL is capped at 59,500,000
 - <https://etherscan.io/token/0x8b8e088c7ad40d70d0a8183a399c8f9c24b5c8d8>

Premium Allocation Tokens (PATs)

PATs are ERC20 tokens. PATs have an unlimited supply and are minted during the completion of a successful CHL submission process on the AQRE app. Each PAT has an in-app allocation value of **\$1.00 USD** (all properties are valued in USD). A user can only allocate PATs to a **premium property**, with the number of PATs representing the user's 'stake' in that property. This 'stake' determines the user's pro rata residual income. When a premium property is sold, the liquid proceeds are distributed pro rata as well. PATs cannot be transferred, unallocated, sold, or used outside of the AQRE app. PATs are the primary allocation vehicle used on the AQRE Platform.

- For example: a user has 12,500 CHL and CHL has a market value of \$4
 - Net market value of \$50,000
 - Can be redeemed for 50,000 PATs

Basic Allocation Tokens (BATs)

BATs are ERC20 tokens that are almost identical to PATs. They have an unlimited supply and are only minted during the completion of a successful CHL exchange in the AQRE app. Each BAT has an in-app allocation value of **\$1.00 USD** (all properties are valued in USD). A user can only allocate BATs to a **basic property**, with the number of BATs representing the user's 'stake' in that property. This 'stake' determines the user's pro rata residual income. BATs cannot be transferred, unallocated, sold, or used outside of the AQRE app. BATs can only be allocated to basic properties.

Property Token

An ERC721 token represents each property on the AQRE platform. This means that each property has characteristics that are tokenized and communicable on the blockchain; this is a vital feature for communicating all info for the platform functionality. Each property token will contain necessary info such as market price, down payment, remaining allocation capacity, potential ETH accruals, location, monthly payout, and tenants.

Unclaimed Income - Economic Stabilizer

If properties are listed on the platform but are NOT fully allocated, unclaimed income will be held in our payment distribution smart contract. This ether will be claimed when allocations are made to the unallocated property. Users will receive some/all of the ETH as a back pay on their contribution, this backpays pro rata for each accrued payout period. This is a function that prevents people from endlessly sitting on their CHL, hoping they increase in value.

Technical Description of Tokens

Allocation Token Technical Description

ERC20 compatible Smart Contract. This token is for property allocation within the AQRE platform.

Interface

Additional properties

- exchangeContract - address, of exchange contract that is allowed to mint new tokens.
- Investment contract burns received tokens
- TODO: admin should be able to mint tokens manually

Additional methods:

- mint (address userAddress, uint256 tokens) - new AT tokens minting to userAddress. It should be called only by Exchange smart contract, otherwise it should fail.
- transfer, transferFrom - method should be overridden to disable any transfers of AT between users.

Additional events:

- Minted (address receiver, uint256 tokens).
- App description
- Submission process

Real Estate Properties Contract

ERC721 compatible Smart Contract. This Smart Contract is intended to store property info, investment restrictions, AT balances per investor and property, etc.

Property Characteristics:

Immutable characteristics: these property characteristics will not be changed throughout the lifetime of the property or our project.

- location - string - full address of property
- AllocationCapacity - uint256 - absolute amount of shares to be issued, (Let this be AV_v)
- maxInvestedATperInvestor - uint256 - absolute amount of shares that could be allocated per person (Let this be EQUAL to 20% of AV_v)
- Sq Footage

Mutable characteristics: these property characteristics are likely to change within several months or years of a property's lifecycle. Some characteristics can change through actions taken by Chelle Service Capital Inc. (CSCI), while other characteristics may change due to user interactions within the AQRE Platform or other external factors.

- marketValue - uint256 - absolute value in USD
- AT Status- address - compatible address of AT contract (could be PAT or BAT).
 - The function of this code is uncertain. We are assuming it distinguishes whether a property is premium or basic
- tenantsCount - uint256 - tenants count ((could be updated by manager)
- monthlyPayout - uint256 - monthly payout (could be updated, based on current market) (let this be P_v)
- Cap Rate
- Monthly Rent
- Residual Backpay (value linked to payout smart contract)
- Remaining allocation capacity

Interface

Properties:

- properties - mapping (uint256 => Property), it is intended to store real estate property related information, like amount of AT and Bounty AT that could be invested in total, limitations per investor re investment, investment balances, state of property (INVESTABLE, UNINVESTABLE), custom data field (that could be used to store real estate property version, investment contract compatibility, etc).
- investmentContracts - mapping (address => bool), it is intended to store addresses of Investment Contracts.
- managers - mapping (address => bool), it is intended to store managers addresses that are allowed to create/update real estate properties, enable/disable real estate property investment status, etc.

Methods:

- createProperty(bytes32 externalRef, string title, string location, uint256 maxInvestedATperInvestor, address AT, uint256 totalAllowedATinvestments) - method to register new property on platform.
- updateProperty(uint256 propertyId, uint256 tenantsNumber, uint256 monthlyPayout) - method to update updatable attributes of property.
- setManager(address manager, bool status) - set manager address status.
- getProperty(uint256 propertyId) - method to return real estate property information (combined from Real Estate Property structure).
- getPropertyInvestors(uint256 propertyId, uint256 from, uint256 to) - it should return all addresses of investors in paginated manner, that invested to real estate property.
- invest(address investor, uint256 propertyId, uint256 shares) - method could be called only by Investment contracts to define investor's shares amount of specific property.

Events:

- PropertyCreated(uint256 propertyId, bytes32 externalRef, ...)
- PropertyUpdated(uint256 propertyId, ...)
- PropertyInvested(uint256 propertyId, address allocationTokenAddress, uint256 tokens)

Conceptual Overflow

App Intro

The AQRE app is the user interface for the AQRE real estate investment platform. All actions and all user information will be accessible in the app. The purpose of building the AQRE app is to create a seamless, intuitive, and simple interface for navigating our platform's complex technical landscape. The app is where users create a profile, view properties, allocate tokens, manage holdings, and collect ETH. The app is also the primary place for user settings management.

Account Creation, KYC, Ethereum Wallet

To get started with the AQRE platform, a user must sign up and create a profile in the AQRE app. This begins with a standard KYC submission. After this is complete users will need to fill out their risk profile. Once they are KYC verified, the user profile is complete, and the user can begin integrating their Ethereum wallet to their AQRE Profile. This must be done with the user's ETH address submitted during the KYC submission process. Users connect their wallets using 'bridges,' such as Metamask or Jaxx. This allows users to exchange CHL and collect ETH residuals while maintaining their own wallet security.

CHL Exchange Request and Details

After successful profile verification and wallet integration, a user can begin preparing to invest in properties. Whether a user bought CHL during our ICO or on an exchange, to start collecting residual returns from properties, they must exchange their CHL within the AQRE app for Allocation Tokens (ATs). Recall that ATs can only be obtained within the app, they cannot be traded, transferred, or sold.

To initiate an exchange, a user must have a verified profile. The user must have readily available Chelle Coins in their Ethereum wallet. Users select the number of CHL they will include in the exchange request and complete an accompanying transaction with an equal amount. This exchange submission process undergoes an internal review, and once approved, the CHL is burned and ATs are awarded in-app. If the internal review fails, the CHL will automatically return to the user. The ATs awarded are available for allocation within the AQRE platform and app only.

The CHL<->AT exchange process is executed upon the market rate obtained by an “oraclizing” service. The AQRE app will launch before CHL is available on exchanges. During this initial period, the in-app value of CHL will be \$3.00 USD. ATs have a fixed value within the app of \$1.00 USD, and all properties are listed in USD. If a user has exchanged their CHL for ATs, market fluctuations do not affect an individual’s allocation power. ATs can be only be moved within the app when their respective user initiates an allocation.

Allocation, Process, Restrictions and Investment Returns

Once a user has Allocation Tokens, they can begin investing in properties listed on the AQRE app. When AQRE Properties LLC (AQRE’s property purchasing division) acquires properties, they list in the property browser. A user can select a property by scrolling through the list, or users can filter properties by characteristics. When a property is selected, a user decides how many ATs they will allocate to the property.

To both prevent large players from entirely allocating towards specific properties, as well as to encourage diversification, an individual’s property allocation cannot exceed 25% of the property’s allocation capacity. After confirming the transaction, the allocation is complete. Allocations cannot be undone. A user commits to an allocation for the lifetime of that property. The user will continuously receive residual income paid in ETH from the property they allocated to, pro rata, until AQRE Properties LLC decides to liquidate the property. When a premium property is sold, liquid proceeds, including potential appreciation values, will be distributed pro rata in ETH. Investment return expectations can be found in the Prospectus.

Claiming Income and Backpay

Transaction fees can potentially swell and eat into investor profits. This poses a problem, as efficiency and user control are intended core features for the AQRE platform. Our solution is to create a claiming function on ETH payouts instead of automatically having funds sent to the user’s wallet. The intention behind this is to reduce fees so they are only incurred when initiated by the user. This will significantly reduce the number of transactions and network incurred by the users.

For instance, a dozen properties are paying out residuals and all these properties are fully allocated to the current group of users. Residuals will be paid in USD and collected by CSCI. CSCI pays necessary fees for maintaining the properties, takes its management fee, and then the remaining residuals are converted to ETH. Once the funds are in ETH, they are sent to into our **payment distribution smart contract** (PDSC). Users withdraw their entitled funds from the PDSC whenever they like. Their entitled funds per property are the percentage of allocation capacity they own per property multiplied by the property's monthly residual income. All management fees are already collected once the ETH is in the PDSC.

In order to claim income, users initiate a withdrawal within the AQRE app to withdraw ETH from the PDSC. Since ATs are non-transferable, all property residuals can only be claimed by the owner of the AT. This means that claimable ETH is exclusive to its intended recipient. Once the withdrawal has been initiated, the user will confirm and then the withdrawal will begin.

Unallocated Properties and Backpay

A very important aspect of the AQRE platform and our mission is to continuously grow our real estate holdings and create a profitable real estate investment platform with global availability. Part of our commitment to upholding this ecosystem is to claim no more than our 20% management fee. This fee is used to cover costs and reinvest into creating a larger real estate book. However, if premium properties are not fully allocated and have begun paying residuals, CSCI will NOT take this 'unclaimed ETH' for themselves. Instead, the unclaimed ETH will accumulate until a user has allocated PATs to the property's remaining allocation space. This will initiate backpay that pays users pro rata for all periods since the property was listed on the AQRE platform.

When a new allocation is made, any unclaimed ETH will be back paid pro rata to the user. This is an important economic incentive that encourages new entrants to purchase CHL to claim unclaimed ETH and join the AQRE platform. CHL's burning token model and high utility may cause the coin to appreciate in price. This may cause some holders to not use their CHL, to either continue to hold, or to sell it on an exchange once it has appreciated. Since the intention of CHL is for it to be used for property investment, the back-pay model for unclaimed ETH incentivizes both new entrants to join the AQRE platform and CHL usage from existing CHL bag-holders (burning and exchange for ATs).

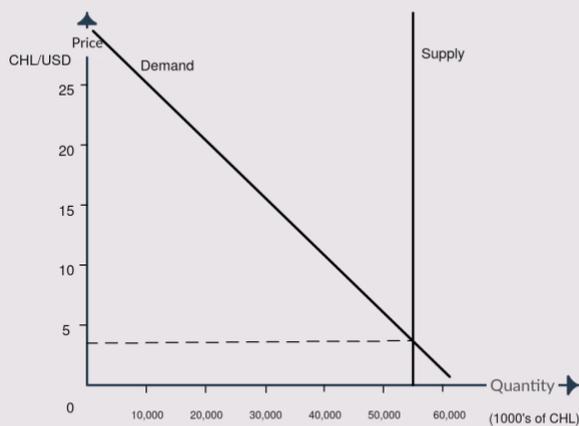
In-App Viewing Experiences

The AQRE app allows users to find all the necessary information on available properties, their individual holdings, and their individual allocation power. Users can view all available properties, all property characteristics, AT balance, CHL <-> AT rate quote (exchange rate link), current holdings, previous payouts (with etherscan links), expected income for the next month (listed in USD), and unclaimed ETH.

Token Economics

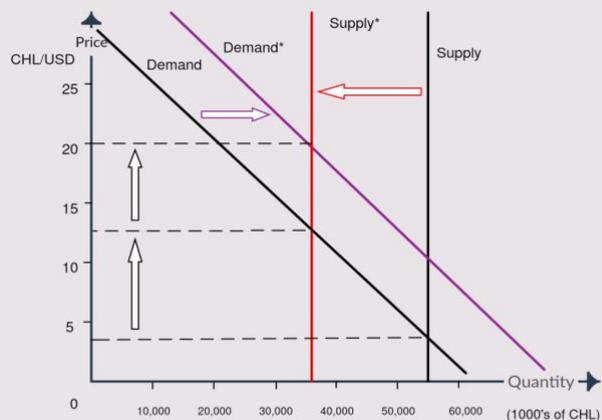
Chelle Coin has several characteristics that support a sound economic environment. This includes a fixed supply, utility (ability to invest in real estate and collect residuals), and price variability. Price variability is necessary for the price of CHL to represent the utility it offers and current needs of the economic environment. The intention for CHL is for its value to be correlated to the growing scope of AQRE Properties LLC's real estate book. The price of CHL relates to the number of properties that are paying out residuals, as CHL must be acquired and burned in order for users to collect income. This process is encouraged, because if no users allocate, accumulating backpay ETH incentives user action. This leads to the economic assumption that, over time, all properties will be at full allocation, resulting in burning of the CHL token supply. Since utility increases whenever additional properties are purchased, demand is expected to stabilize or increase.

AQRE's Fixed CHL Supply Model



All graphs, charts, or graphics are used for illustrative purposes only and do not reflect any future event or future value or performance of any investment.

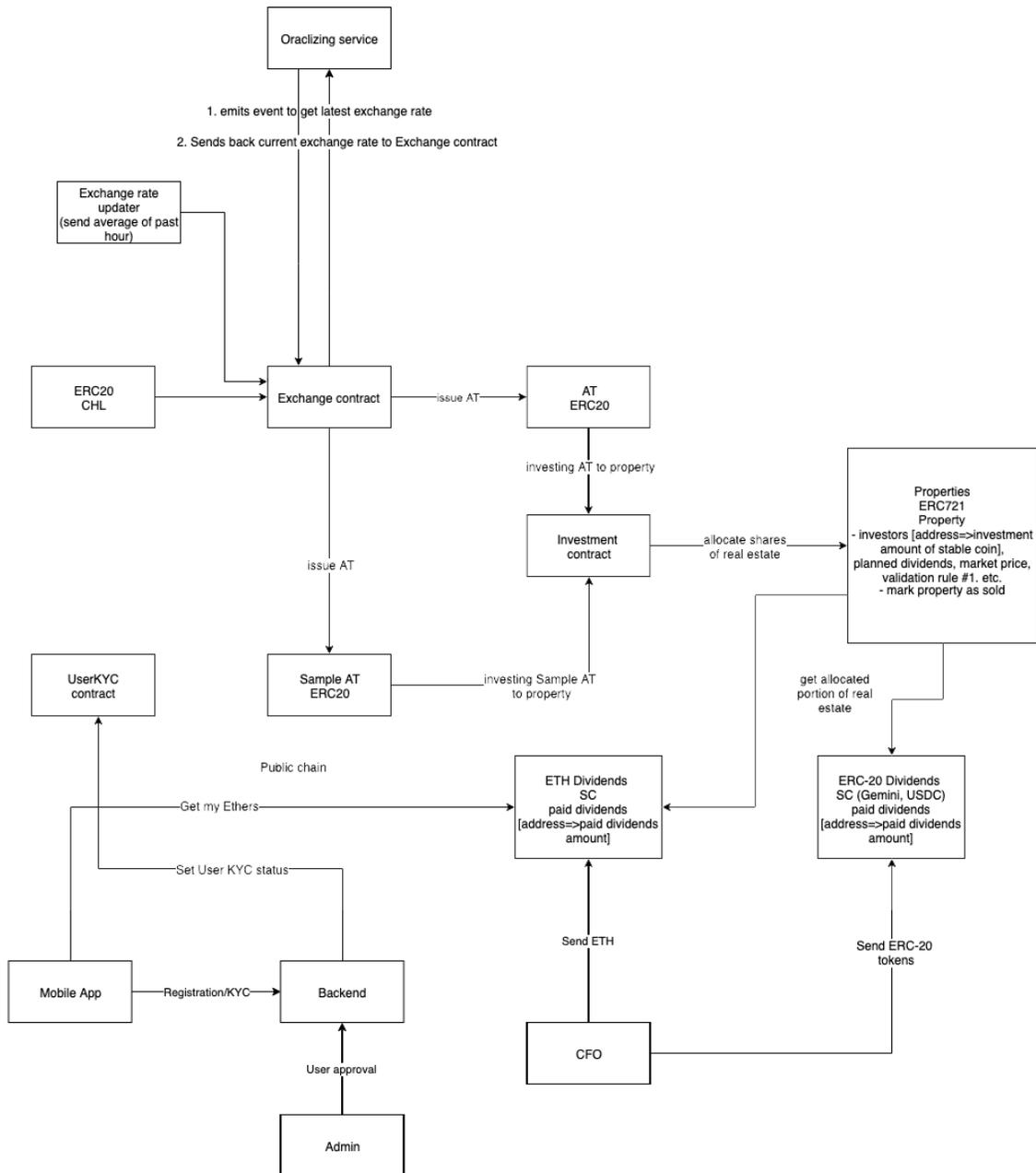
AQRE's Fixed CHL Supply Model Future Outlook



All graphs, charts, or graphics are used for illustrative purposes only and do not reflect any future event or future value or performance of any investment.

Principal Smart Contracts Diagram

The diagram below provides an overview of the AQRE Platform. All smart contract interactions will be done seamlessly within the app for an easy user experience.



Non-Token Smart Contracts

User KYC Contract Technical Description

Contract that should have a list of all users addresses that passed KYC. This contract should be used by all other Smart contracts of the Platform to validate that the sender is valid.

Interface

Properties:

- userStatuses - mapping(address=>bool) - it stores users addresses that passed successfully KYC.
- kycManagers - mapping(address=>bool) - it stores KYC managers/platform address, that can mark user address as passed KYC.

Methods:

- setUserAddressStatus(address userAddress, bool passedKYC) - method that should be called only by KYC managers, to mark that user passed KYC. It emits event `UserStatusUpdated` on every change.
- getAddressStatus(address userAddress) - method that should be called by all contracts to validate that sender address is valid and known to the platform. It returns bool to indicate that passed address is valid or not.

Events:

- UserStatusUpdated(address user, bool status).
- Overview smart contract description with images
 - Detailed description with focused images
- Payout system and incentive mechanisms

Exchange Smart Contract

Smart Contract intended to convert ERC-20 token (Chelle Coin) to AT tokens or Bounty AT tokens. Exchange review process has manual overweight components so that the AQRE Platform can accurately award PATs or BATs deservingly.

Interface

Properties:

- erc20Token - address of ERC-20 token allowed to be exchanged.
- allocationToken - address of Allocation Token Smart Contract.
- state - enum (ACTIVE, INACTIVE), current state of Contract that indicates allowance of exchange through this contract.
- exchangeManager - address, that can manage exchange rates of ERC-20 (CHL, Bounty) tokens to AT tokens.

- allocationTokensPerErc20Token - uint256, exchange rate of ERC-20 tokens to AT tokens.

Methods:

- exchange(uint256 tokens) - method should exchange supplied tokens, by calling method transferFrom to send tokens to 0x0, and minting AT tokens based on current exchange rate. Method could be payable to pay oraclize commission.
- calculateAllocationTokens(uint256 tokens) - method to calculate amount of AT tokens that will be minted.
- setState(uint256 state) - method, that should set current state of contract, access should be limited only to owner of smart contract.
- setExchangeRate(uint256 allocationTokensPerErc20Token) - method, that should update exchange rate of ERC-20 tokens to AT tokens.

Events:

- TokensExchanged(uint256 _operationId, address userAddress, address erc20Token, address allocationToken, uint256 erc20Tokens, uint256 allocationTokens)

Investment Contract

Smart Contract intended to manage investment process and pro rata distribution, also referred to as the payment and distribution smart contract (PDSC). It should validate restrictions of Property. Investment process could be triggered only by AT holder. Investment Contract is separated from Real Estate Property Contract to have upgradability of investment process, like adding new restrictions, etc.

Interface

Properties:

- allocationTokens - mapping(address=>bool) - addresses of ERC-20, that could be invested through this contract.
- propertyAddress - address of Real Estate Properties ERC721 Contract.
- state - enum (ACTIVE, INACTIVE), current state of contract to indicate is it allowed to invest AT tokens through this contract.

Methods:

- invest(uint256 propertyId, address tokenAddress, uint256 tokens)

Events:

- Invested(uint256 propertyId, address investor, address tokenAddress, uint256 tokens)

Property Token Each property is its own ERC721 token. This means that each property has characteristics that are tokenized and communicable on the blockchain; this is a vital feature for communicating all info for the platform functionality. Each property token will contain necessary info such as market price, down payment, remaining allocation capacity, potential ETH accruals, location, monthly payout, and tenant number.

- **Note:** In the event that properties are listed on the platform but are NOT fully allocated, **unclaimed income** will be held in a smart contract. This ether will be claimed when allocations are made to the unallocated property. Users will receive some/all of the ETH as a back pay on their contribution. This backpays pro rata for each accrued payout

period. This is a function that prevents users from endlessly holding their CHL hoping they increase in value.

- In the event that there are two funding methods with asymmetric return profiles (ex. CHL and BTC), we will adjust the payout criteria as necessary. This will likely occur in a future version of the app and not be ready for initial launch

ETH Dividends Contract

Smart Contract to manage dividends distribution.

Interface

Properties:

- dividends - mapping (uint256 => Dividend), it is intended to store dividend related information, key of mapping should indicate auto incremented id of dividends. Dividend should include the total amount of dividends to be distributed among all property investors, the total amount of investments to property, and per investor status of dividend payment.
- managers - mapping (address => bool), it is intended to store managers addresses that are allowed to pay dividends to investors.
- state - enum (ACTIVE, INACTIVE), current state of contract to indicate is it allowed to pay/receive dividends through this contract

Methods:

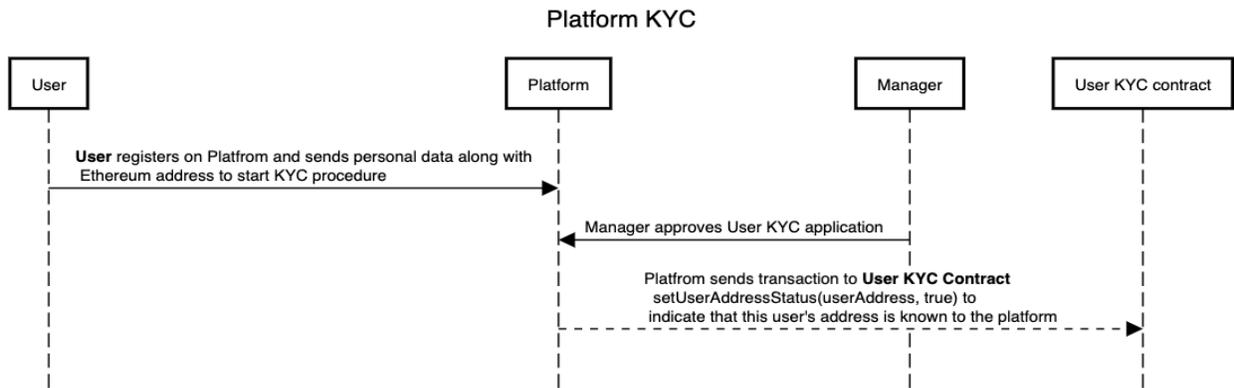
- payDividend(uint256 propertyId) - method register new dividend for specific property. It should be called only by managers.
- delegatedDividendWithdrawal(uint256 dividendId, address investor) - method that should be called to automate the process of sending dividends to investors addresses by dividend id.
- withdrawDividend(uint256 dividendId) - method to manually receive dividend by dividend id.
- delegatedDividendWithdrawal(address investor) - method that should be called to automate the process of sending dividends to investors addresses.
- withdrawDividend() - method to manually receive all dividends.
- setState(uint256 state) - method to enable/disable adding dividends and withdrawing them from Smart Contract.

Events:

- DividendPaid(uint256 propertyId, uint256 dividendId, uint256 ethAmount)
- DividendWithdrawn(uint256 propertyId, uint256 dividendId, address investor)

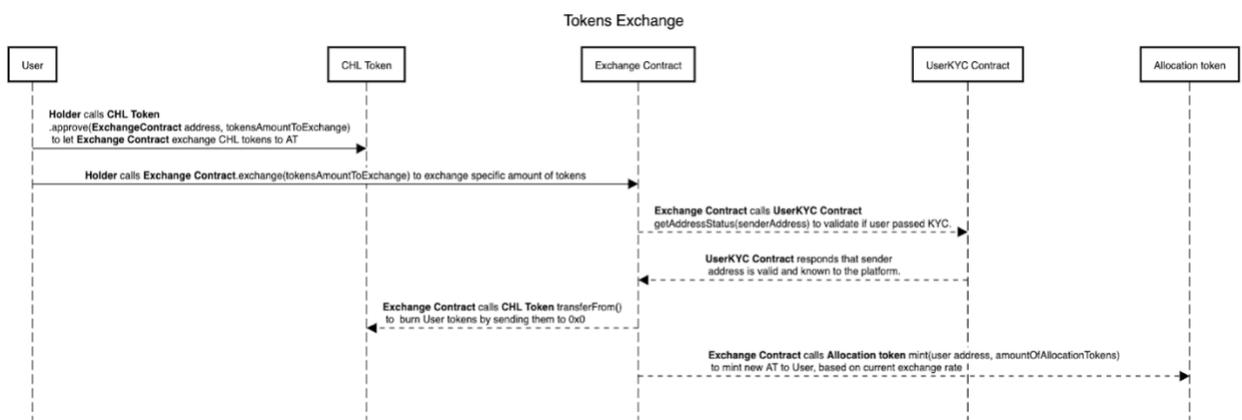
Interaction Scenarios

KYC Procedure



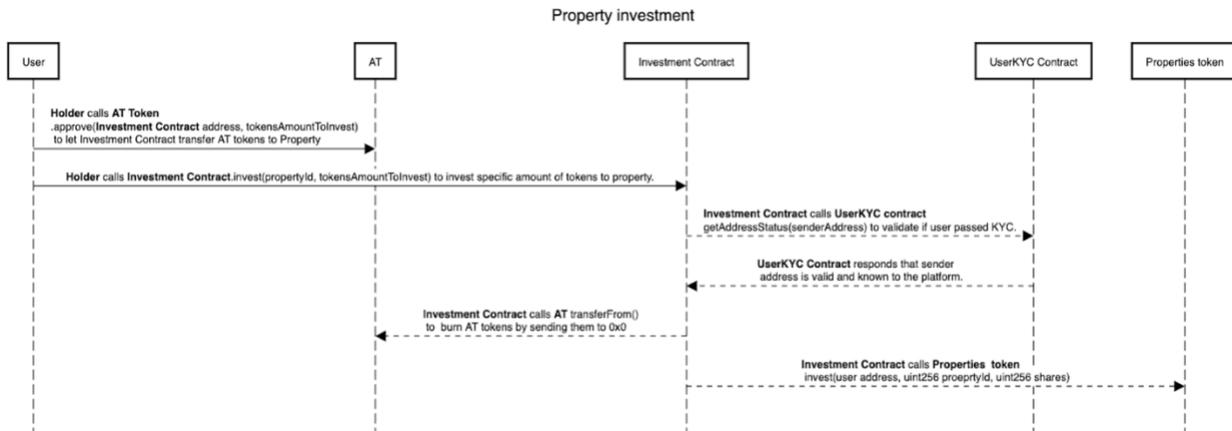
After users download the app and begin the registration process, they create a user profile and begin submitting all necessary information for KYC. KYC submission yields an outcome: successful/unsuccessful. If successful, the user is granted successful profile completion and their profile is communicable on the blockchain as a participant who can: exchange CHL for ATs, allocate ATs to properties, and collect ETH monthly residuals. If unsuccessful, the user cannot participate in-app features such as CHL<->AT exchange, allocating ATs, or claiming ETH residuals.

CHL Token Conversion



Once verified, users can start a CHL->AT conversion request within the app. Number of CHL and the CHL <-> AT conversion rate will provide an estimate for the number of ATs that will be awarded. Once a conversion request has been made, it cannot be withdrawn. Its outcome is decided by the outcome of the submission review. Whether a user has successful KYC, whether the user's wallet and CHL transaction are valid, and whether the user receives PATs or BATs are all decided during the exchange request. If everything is valid, then the user is granted ATs and the submitted CHL are burned. If the exchange request fails, the user's CHL is returned to their wallet.

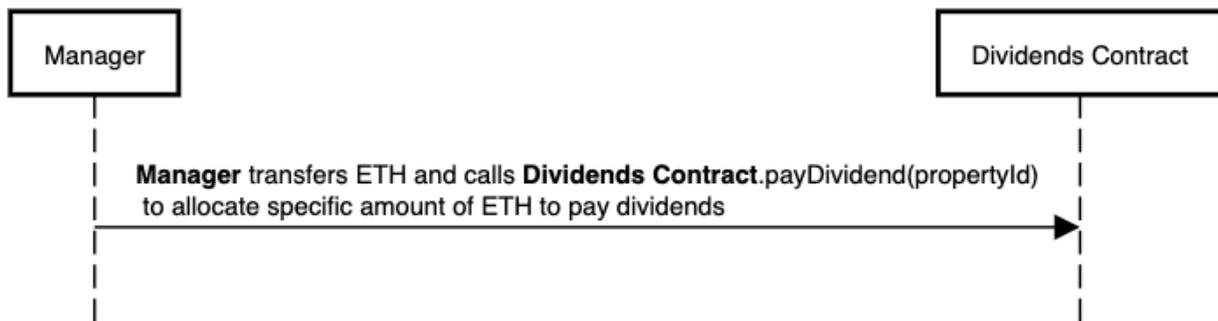
Investing PAT or BAT to Property



A user ‘possesses’ ATs in-app. A user initiates an investment transaction by assigning ATs to a specific property. This transaction interacts with several smart contracts. The Investment Contract combines user investment transactions, property characteristics, and the user income formula to determine how much residual income a user should get from each property every month. The user first interacts with the Investment Contract (PDSC) by initiating an investment transaction. Then the investment transaction checks to confirm that the client is KYC approved and has made a valid AT allocation (sufficient balance, capacity, etc.). If the investment transaction is approved and submitted, then the ATs are burned. In return, the number of burned tokens combined with the property characteristics at the time of allocation conclusively award the user’s profile a R_{xy} value. The R_{xy} value is the percentage or *rate* of distributed monthly income from property Y that is awarded to user X. Thus, $R_{xy} = U_x Q_y / AV_y$.

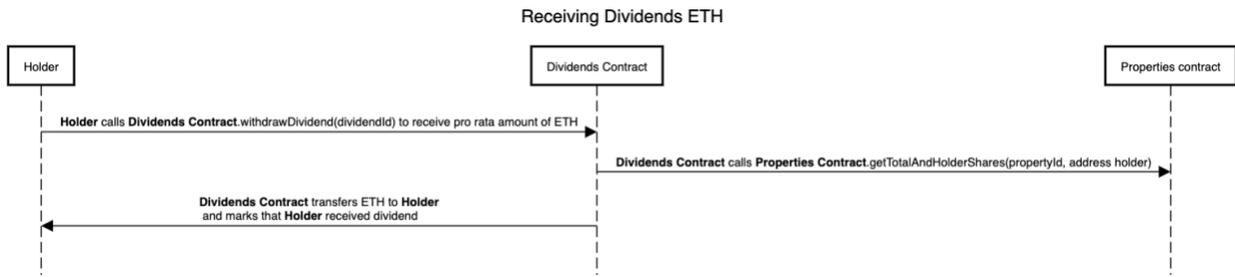
Depositing ETH to Distribute Property Dividends

Transferring ETH to distribute dividends between Property share holders



The AQRE Platform will convert all USD after fees and commission into ETH. It will then transfer this ETH into the dividends contract (PDSC). Once the ETH is in the PDSC, users with unclaimed income can withdraw their entitled amount of ETH whenever they choose.

Receiving Property ETH Dividends



The PDSC is designed in a way that reduces network fees for the user and supports our backpay ETH model. The AQRE app provides an intuitive interface that allows users to interact with our various smart contracts with minimal smart contract familiarity required. Users initiate a request to withdraw their unclaimed ETH. This request causes the dividends contract to pull data from the all property tokens and the R_{xy} value; this confirms the user's unclaimed ETH balance for all previous monthly payouts. A user's unclaimed ETH is a value that they can always see in their profile. This value is equal to the R_{xy} value for each property for each residual payment.

User Income Formula

This User Income Formula can be found here https://docs.google.com/document/d/1--f89qgld-jsQ3fk97b_fis0K-gToa_yEsJs6xgd4tk/edit?usp=sharing

Thus, a user's unclaimed income is their UI_x for all periods. Users can withdraw this amount from the PDSC minus whatever amount they have previously withdrawn (user pays the gas when withdrawing ETH from the pool).