Design and architecture of Kakooyou: A web3 protocol to distribute state-bound tokens

Léon Hocquemiller

leon.hocquemiller@kakooyou.com

www.kakooyou.com

Abstract. Non-fungible tokens (NFTs) are forecasted to disrupt the ticketing industry by enabling secure, verifiable, and transparent proofs of ownership, potentially solving some of the industry's pain points. However, no implementation enabling the mass distribution of NFT tickets has yet been released. This whitepaper introduces and presents the design of Kakooyou, a Layer 1 blockchain-based protocol that addresses this technical gap, offering a framework for web3 ticketing.

Index

Economical context

Ticketing as a business application for state-bound tokens

State-bound tokens are a subclass of non-fungible tokens, which enter states in which the conditions for the exchange change. A different smart-contract must be called to shift those tokens from one state to the next logical state.

Ticketing can be seen as a logical business application for these state-bound tokens. They symbolise a proof of ownership as the possession follows a former exchange. Also, tickets have different stages:

- They are created.
- They are put on sale.
- They are bought.
- At some point they "reveal" their value (provide access).
- They are burned and loose value.

Ticketing has now been properly digitalized, and the younger generation are not using paper tickets anymore. Smartphones are mainly used to prove ownership of assets. Therefore, the consumers are ready to move forward with the adoption of NFT wallets, given that these wallets are intuitive enough to gain approbation from non-technical users.

Current challenges

The challenges that are being addressed here are scalping and fraud. Scalping happens when a ticket is being resold on the black market, and fraud happens on top on that when that ticket was cloned.

The victim of a scalped ticket will then detain a nominative ticket with false nominal data printed on it (access might be denied depending on the policies of the event-maker), and the victim of a frauded ticket will probably be denied the access to the event as each ticket can logically be burned only once.

Scalping and fraud are strictly forbidden, however making these technically impossible is difficult. These speculative actions are enforced using scalper bots which can perform at high speed.

The analysis of this economy is that proof of ownership is no longer enough in such speculation-prone systems, and a proof of integrity must be carried out as well as part of the verification process. That proof of integrity can be additionally computed if

- Each ticket is bound to only one account.
- Each account is bound to only one physical person.
- Rules over the resale of tickets are restrictive enough. Transfer from one account to another should be prohibited as these transactions might be topped with undeclared cash.

The computation of the proof of integrity together with the proof of possession would in theory allow event-organisers to protect their audience from the speculative intent of scalpers.

A web3 system is not the only way to solve that integrity problem but it is an elegant solution.

Disruptive potential

The ticketing space is a billion-dollar industry, and the impact of such a technological stack could be huge:

- It is a small improvement, but for many actors.
- The industry is growing and in demand for innovative solutions.
- While web3 is not understood the advantages of such a system can be marketed.
- This can be deployed internationally (this is an advantage of the tech space).
- Governments can be counted as positive actors on such an ethical project.

Security improvements

It is notorious that blockchain is an incredibly resilient technology in terms of cybersecurity. To corrupt some data, you need to force some advanced cryptographic algorithms, and that complexity scales with the "depth" of the data, that is the number of blocks closed after the consensus has been computed.

Blockchain is also resilient to DDOS attacks as it consists of many nodes and there is no central authority to deny.

Ideological considerations

Some ideological considerations must be listed to understand the development of such a protocol. This project is an anti-speculative technology, that aims at providing event organisers with better remunerations thanks to the control of the secondary market.

At the same time, this system could provide users with fairer prices as these would be freed from speculation. This is a "#techforgood" project as it identifies scalpers as unwanted actors in the distribution chain and specifically aims at shutting down their activities.

Technical design

Network

Red nodes

Red nodes are commonly known as "wallets". They allow the storage of a private key used to prove ownership of the tickets. These are lightweight versions of the blue nodes.

Blue nodes

Commonly known as validators, blue nodes are the full version of a node, including the drivers to validate operations which require consensus (smart contract). These blue nodes are meant to be hosted on the cloud.

Redirections

In a typical layering strategy, red nodes ask to a random blue node for consensus. The vote is then performed at the blue node layer and then the previously triggered blue node will redirect the answer to the red node, letting this red node know the status of the request.

Layering strategies for blue nodes

a. Simple Layering

In the simplest architectural pattern, the blue nodes network consists of a small number of blue nodes **X**. Typically **X** shall not be too big because the amount of communication is then **X!**. Each vote is performed inside that limited network and the time for consensus is **T = 3*maximum ping** (request broadcast into vote unicast into confirmation broadcast).

b. Clutch Layering

In this more advanced pattern, the blue nodes network consists of a small number of clutches of any number of nodes; typically, the number of clutches should be **Y** with **Y** not to big (same rule applies than with simple layering). Each clutch has a leader node which gathers the votes for the nodes that it represents. The time for consensus is **T = 6 * maximum ping**.

c. Tree Layering

In this strategy each clutch at index N may be composed of leaders of the clutch at index N+1. The rules on the number of nodes only apply at the highest level (it should not be possible to have a leader for the clutch 0). The time for consensus here is **T = n * maximum ping** with **n** being the maximum depth of the tree.

Data

Blocks

The first degree of data abstraction are the transactions. These are stored in blocks like how a blockchain typically works.

An administrator account is required for the mining of block. The administrator private key provides the proof of authority to close the blocks.

There are conditions for each transaction on which they can be deleted from the filesystem, freeing some storage. Typically, once an event is outdated all event-based transactions can be deleted from the filesystem, at the condition that the block has been closed properly before.

Operational data

The operational data is computed and generated automatically during the lifetime (or during the update process). It is dynamic and is used for the consensus mechanisms and it is responsible for storing the states of the tickets.

Once an event is outdated (the date + a defined period is passed) the related operational data can be deleted.

Meta data / private data

In regard with RGPD compliance, another type of data is required here: private data. It must be easily deleted from the blue nodes. In this regard it can not be stored in the filesystem or can not be attached to the transactions in the blockchain.

Specific rules apply to this data in a more traditional way, and a set of defined operations must be implemented to mass target that data on all the nodes simultaneously.

Operations

Consensus Operations

Consensus operations are a multiple stage process. The operation is first signed with the private key of the emitter. Depending on the emitting node (red node for redirection, this is the standard case scenario) it might be redirected on a blue node.

Then the consensus will take place. The operation will be broadcasted as “request”, then the votes are collected on each of the top-level nodes (see network architectures). If the vote reaches the validation rate (typically 66%) then the operation will be broadcasted anew as “validation”. It is a 3 steps mechanism (“request” into “vote” into “validation”).

In the case where an operation is marked as “validated” but the local node has a voting structure “denied” the validating node will be marked as “corrupted” locally and excluded from further voting.

Some of the actual consensus operations are:

OP_FIRST_OF_ALL, // This operation is done only once at blockchain creation

OP_CREATE_USER_TOKEN, // This operation is used to create a new user (admin nodes only)

OP_CREATE_ORGANISATOR_TOKEN, // This operation is used to create a new organiser (admin nodes only)

OP_CLAIM_USER_TOKEN, // This operation is used to claim a user token

OP_CLAIM_ORGANISATOR_TOKEN, // This operation is used to claim an organiser token

OP_DEMOTE_USER, // This operation is used to demote a user (admin nodes only)

OP_DEMOTE_ORGANISATOR, // This operation is used to demote an organiser (admin nodes only)

OP_RESTORE_USER, // This operation is used to restore a user (admin nodes only)

OP_RESTORE_ORGANISATOR, // This operation is used to restore an organiser (admin nodes only)

OP_CREATE_TICKET, // This operation is used to create a new ticket

OP_SELL_TICKET, // This operation is used to put a ticket on a sell status

OP_RESERVE_TICKET, // This operation is used to reserve a ticket

OP_TRANSFER_TICKET, // This operation is used to transfer a ticket to another user

OP_RESELL_TICKET, // This operation is used to resell a ticket

OP_ARCHIVE_BLOCK, // This operation is used to archive a block (after mining)

OP_ACCOUNT_RESET, // This operation is used to reset the private data

Non-consensus operations

Non-consensus operations are simple broadcast operations.

OP_JOIN_NETWORK, // This operation is used to join the network

OP_ANNOUNCE_NEW_NODE, // This operation is used to announce a new node

OP_CHANGE_NODE_TYPE, // This operation is used to announce a change in the node type

OP_GET_BLOCK, // This operation is used to pull the blocks

OP_FETCH_MARKETPLACE_EVENTS, // This operation is used to pull the marketplace events

OP_FETCH_MARKETPLACE_DATES, // This operation is used to pull the dates for a specific event

OP_FETCH_MARKETPLACE_CATEGORIES, // This operation is used to pull the categories for a specific event

OP_FETCH_MARKETPLACE_SEATS, // This operation is used to pull the seats for a specific category

OP_FETCH_INVENTORY_PERMISSION, // This operation is used to ask permission to pull the tickets in an inventory

OP_FETCH_INVENTORY, // This operation is used to pull the tickets in an inventory (after permission)

OP_GET_LAST_ACTIVE_BLOCK, // This operation is used to get the last active block

OP_GET_BLOCK_ZERO, // This operation is used to fetch the block zero exclusively

OP_SET_VERIFIED_FLAG_FOR_EVENT, // This operation will set the verified flag for an event

OP_FETCH_PRIVATE_DATA_AVAILABILITY, // This operation will check on the availability of the private data

OP_FETCH_PRIVATE_DATA, // This operation will retrieve the private data once available

OP_MAKE_PAYMENT_INTENT, // This operation is used to make payment intent

Account types

Standard

Standard accounts are the default, general purpose, accounts on which it is possible to buy tickets. Additionally, it is also possible to resell those tickets. It is impossible to gift tickets with those accounts. Additionally, each account is bound to a mobile number and a mail address, to ensure the policy linking one account to one physical person.

Organisers

Organisers accounts are for organisers. These allow the creation of tickets, the posting of them and editing the events details.

Administrators

Administrators accounts are for the administration tasks. With an administrator account it is possible to transfer tickets, create other accounts and blacklist other accounts.

Additionally with an administrator private key it is possible to manage the marketplace and the private data bases for the whole network.

The administrator private keys are generated at the start of the blockchain.

Tickets

States

The possible states for the tickets (defined as state-bound tokens) are:

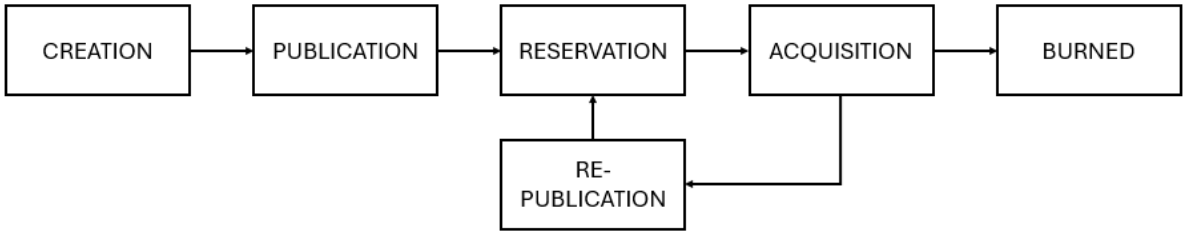

Figure 1- Possible states for tickets

Parameters

Some parameters of the tickets are set dynamically, and some are set by the organisers and users during the lifecycle of the ticket:

Ticket id	Automatic
Owner ID	Automatic
Owner reference (index of creation operation)	Automatic
Organiser ID	Automatic
Organiser reference (index of creation operation)	Automatic
Date of event	Set at creation
Seat ID	Set at creation
Event ID + name	Set at creation
Category ID + name	Set at creation
Rush date	Set at creation
Visual data (image address)	Set at creation
Price + maximum resell price + currency	Set at publication or republication

Rules

Rules on the tickets are enforced by smart contracts. It impossible to gift tickets, but it is possible to resell them. However, the resell price is capped by the organiser.

Inventories

All the ticket related data is contained on the blue nodes in public inventories. With the associated permission, it is possible to download that data in private inventories on the red nodes. There are 3 types of inventories for possessed tickets, tickets to be sold, sold tickets.

Marketplace

Description

Kakooyou has a built-in marketplace with limited features. By default, it will return the events which have been verified by administrators.

Depending on the stage of the tickets the marketplace will display or not the tickets. If the tickets are not on sale yet or sold out it will display an error message.

Event and category data

Some filters can be selected based on some additional data the event organiser can provide. This data may include the location of the event. It is recommended that organisers include that data but not mandatory.

Tokens

Communication tokens are core to Kakooyou's functioning. Some of these tokens act a "reference data" to find a set of tickets in the marketplace and some of these tokens are signed by the organisers / users / administrators to allow further manipulation of the data.

These tokens are mostly transmitted as QR codes.

Account claim tokens

Account claim tokens are signed by the administrator. Claiming an account claim token is necessary to create a user or an organiser account. Typically, these are transmitted via mail.

Export tokens

Exports tokens are common to all account type, and they allow to export the account details (including the private key). As these tokens contain the private key, they contain the proof of possession for all the tickets in the inventory.

These tokens are typically bound to a physical person as they contain the private key. Exchanging the token will not reset the account's data (mail and mobile) so it is not recommended to exchange them. Typically, these should be used only when a user / organiser / admin wants to use its account on another device. Exchanging accounts is not allowed and resetting the account data is not possible this way.

Ticket tokens

Ticket tokens must be generated before the access to the event. The event organiser has the possibility to scan and burn the ticket at that time.

These tokens contain a timestamp. It is recommended that these are generated at the very least moment before entering the event. In an anti-scalping context this policy must be diffused to prevent ticket-tokens resales.

Reference tokens

Reference tokens are automatically sent to event organisers by request. These can be used to promote the event (printed / diffused as QR codes). Scanning these tokens will provide the users with the data for the event/category promoted.

Claim tokens

Claim tokens are used during the payment process. They contain data related to the ticket that must be transferred. After a successful payment process a node with administrator privilege will execute the transfer command on the claim token, and the ticket will then be added to the inventory of the purchasing user.

Security

On network

The first layer of security is on the network itself, as a password is required to join the network. This password is different for red nodes and blue nodes. As this system aims at staying privately owned it is recommended to not diffuse the password for the blue nodes.

Joining the network can only be done through entry points; those entry points can be deactivated for blue nodes and for red nodes and this way the network can operate in a "closed" mode.

Additionally, the operations are encoded using a private key which is specific to the network.

On operations

Operations are signed with account private keys, which are stored locally. Additionally, and to avoid cloning operation, timestamps are added to those and being checked on reception. This way an operation has a validity period of only a few minutes.

Account prk policies

We do recommend having one account for each user, that is one private key for each user. This is part of the anti-scalping policy.

The option shall be given to store these private keys on some offline cloud services in case where the device gets stolen or the user forgets the encryption password. This is the backup policy for private keys.

Prk encryption

The private key is stored locally. It is encoded in a secured way depending on the hardware where the node is running. Additionally, a password is required to decrypt the private key. That way if a device is stolen the private key still cannot be read without knowledge of the password.

Limits and future considerations

Some old hardware does not have what is required to encrypt the private key. However, all modern smartphones have it which is why this limitation can be ignored.

In terms of security the only incoming issue will be quantum computing. It has to be assessed on which level this might be an issue and what should be done to control the disruption. In the best case the encryptions algorithm could simply be rewritten to be quantum-proof and in the worst-case architecture changes must be made.

Architecture and performance

Parallel processing

Kakooyou features parallel processing. That is achieved by having each operation following its flow independently of the others. However, in the case where two operations target the same data, one of these must be rejected. This is achieved locally by adding an anti-collision module.

Software modules

Some base software component can be deducted from the technical specifications.

a. Client / server modules

The client and server modules are operating the network stack. They keep in memory the network architecture and the address of all remote nodes.

b. Collector

The collector module is responsible for collecting the votes. This is key in the consensus process.

c. Writer

The writer module manages the blocks and the transactions. When a transaction is performed it adds the transaction in the block and generates the additional data.

d. Checking module and checking threads

The checking module performs the checks for the consensus mechanisms. These are done in different thread contexts so that multiple consensus can be made at the same time.

e. Node module

The node module is responsible for creating the operations. It is the highest-level module and can be interacted with through the console or an API.

f. Feedback module

The feedback module is responsible for executing code and returning the correct output once an operation has been approved or rejected by the blockchain.

Language and speed

C++ and Rust (high performance languages) are recommended for this implementation.

Technically this is a lightweight program that could be runed everywhere, however it is recommended that the blue nodes use some powerful hardware to boost the transactions per seconds. If a blue node does not have the required computing power, it will fall behind and get ghosted by the rest of the network.