Blocktix: Decentralized Event Hosting and Ticket Distribution Network

Florian Mathieu and Ryno Mathee

September 20, 2017

Abstract

Blocktix is a trustless, decentralized platform for ticket sales and event promotion. The system operates through smart contracts on the Ethereum blockchain, enabling users to verify the validity tickets for a given event, ending counterfeit tickets.

1 Introduction

The core purpose of Blocktix is to provide a decentralized event hosting and ticket distribution network. Companies and individuals can buy and sell tickets through unforgeable cryptographic signatures stored on the blockchain. With this approach, we have simplified the process of buying and selling tickets and removed the infrastructure of centralized servers. Both parties can therefore enjoy the lowest fees and no downtime due to the decentralized nature .

Anyone who possesses enough tokens can create events and sell tickets online without worrying about any of the additional complexities that are typically encountered. Blocktix provides a user-friendly interface for the execution of smart contracts, bringing its advanced capabilities to the masses with an intuitive application. This approach reduces the complexity of the interaction with smart contracts and provides an additional layer of security minimizing the possibility of fraud or malicious third party interference with the application due to the standardization of contracts.

2 Background

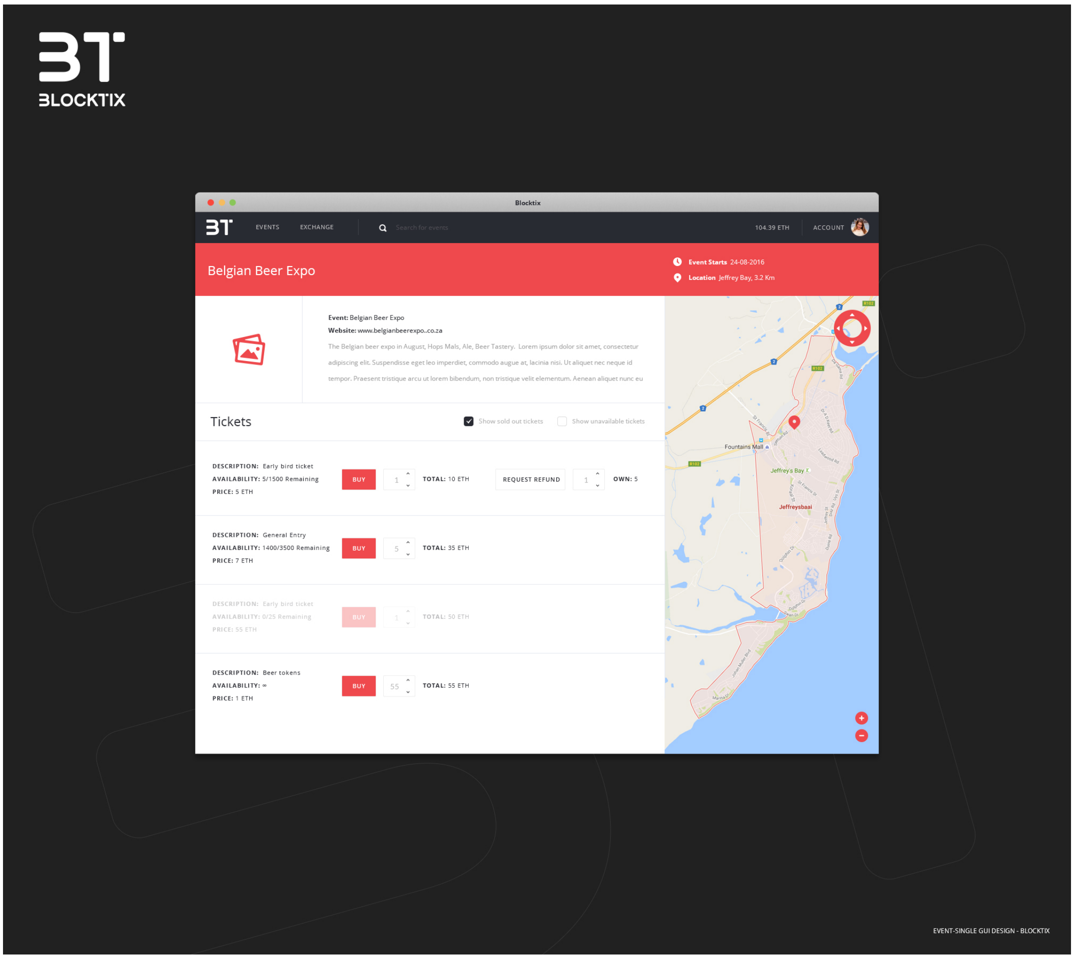
According to Meeting Professionals International, "The American meetings and events industry directly and indirectly supports 6.3 million jobs and generates almost $1 trillion a year in direct, indirect and induced spending..."[1][2], thus there is a market for events systems.

With current systems, event hosting and specifically the distribution of tickets is a costly and cumbersome procedure. To simplify the process, event promoters use companies such as Ticketmaster. Distribution services add a surcharge of as much as 15% of the ticket value. Promoters must advertise the event through a variety of mediums to fill the venue too. The advertising costs of an event can regularly exceed 30% of the total event budget. Ticket fraud [3] and counterfeiting losses of over $1 billion[4] per year have caused the need for trusted third party intermediaries in the peer-to-peer transfer aftermarket. These intermediaries charge an additional 20%-25% of the transferred value.

3 Concept

3.1 Blockchain

The blockchain is the heart of our application and is tasked to secure a wide range of features such as payments, tickets, and the event registry. Blocktix will be built on Ethereum[5] because it's one of the few



Belgian Beer Expo

Event Starts 24-08-2016
Location Jeffrey Bay, 3.2 Km

Event: Belgian Beer Expo
Website: www.belgianbeerexpo..co.za

The Belgian beer expo in August, Hops Mals, Ale, Beer Tastery. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse eget leo imperdiet, commodo augue at, lacinia nisi. Ut aliquet nec neque id tempor. Praesent tristique arcu ut lorem bibendum, non tristique velit elementum. Aenean aliquet nunc eu

Tickets

☑ Show sold out tickets □ Show unavailable tickets

DESCRIPTION: Early bird ticket
AVAILABILITY: 5/1500 Remaining
PRICE: 5 ETH
BUY 1 **TOTAL:** 10 ETH REQUEST REFUND 1 **OWN:** 5

DESCRIPTION: General Entry
AVAILABILITY: 1400/3500 Remaining
PRICE: 7 ETH
BUY 5 **TOTAL:** 35 ETH

DESCRIPTION: Early bird ticket
AVAILABILITY: 0/25 Remaining
PRICE: 55 ETH
BUY 1 **TOTAL:** 50 ETH

DESCRIPTION: Beer tokens
AVAILABILITY: ∞
PRICE: 1 ETH
BUY 55 **TOTAL:** 55 ETH

Figure 1: A mock up of the event page with the ability to purchase tickets and vouchers.

blockchains that provides an efficient and practical toolset for smart contracts. It is the default payment gateway of our application, meaning tickets have to be purchased with the ETH cryptocurrency. It is certainly possible to add other payment gateways such as credit cards and bank transfers through third party integrations. The application will be constructed in such a way that it is certainly possible to add other payment processors, but this is outside the scope of this whitepaper.

3.2 Token

3.2.1 Token creation

Blocktix rewards all participants of the crowdfunding with tokens (TIX) in return for the amount of Ethereum they sourced to Blocktix. TIX will be coupled to a set USD rate, with 4 tiers available: 0,12$ 0,16$, 0,20$ and 0,24$ per TIX. Each tier will have a total of 10.000.000 TIX available.

Token Details A total of 62.500.000 TIX will be created during the crowdsale, of which 64% (40.000.000 TIX) will be sold publicly. The remainder will be accredited to: Blocktix B.V.(24%), Early Investors and Founders (10%), and Promotions/Promoters (2%).

Funding Needs The minimum funding for Blocktix will be 2.500.000$. The maximum cap will be 7.500.000$.

If the desired capital is sourced then the tokens will be issued. These will become available to buyers after the end of the crowdsale. If the target is not met, then all the Ethereum and Bitcoin that was previously raised will return to their respective owners, minus transaction fees.

3.2.2 Token reward model

Ticket fees Anyone that owns Blocktix Tokens is a token holder and is granted several benefits over non-token holders. By participating in the event verification process, the token holder will receive a reward for the work being done. The initial reward will be the TIX that is used to back the event.

Advertising Anyone will be able to purchase advertising rights for an event. These funds will be distributed to partly to 'Blocktix Sales B.V.' and partly to the users on the system that view the advertisements. The event feed always starts with two "promoted events". When an event is being promoted, the advertisement has a random chance to appear in one of these two promoted spots.

Let $TX_1, TX_2, \ldots, TX_m$ be a sequence of m transactions sent to an advertising contract for event e.

First you would calculate the total amount the advertising contract for a given event e has received by a summation S_e of all m transactions for a given event e.

$$S_e = TX_1 + TX_2 + \cdots + TX_m$$

Percentage wise, the amount of views for a given event e should be equal to roughly C_e

$$C_e = \frac{S_e}{S_1 + S_2 + \cdots + S_n} * 100$$

Every time the event feed updates a new number is generated by a pseudorandom number generator (PRNG) ranging between 0 and Sum.

$$Sum = S_1 + S_2 + \cdots + S_n$$

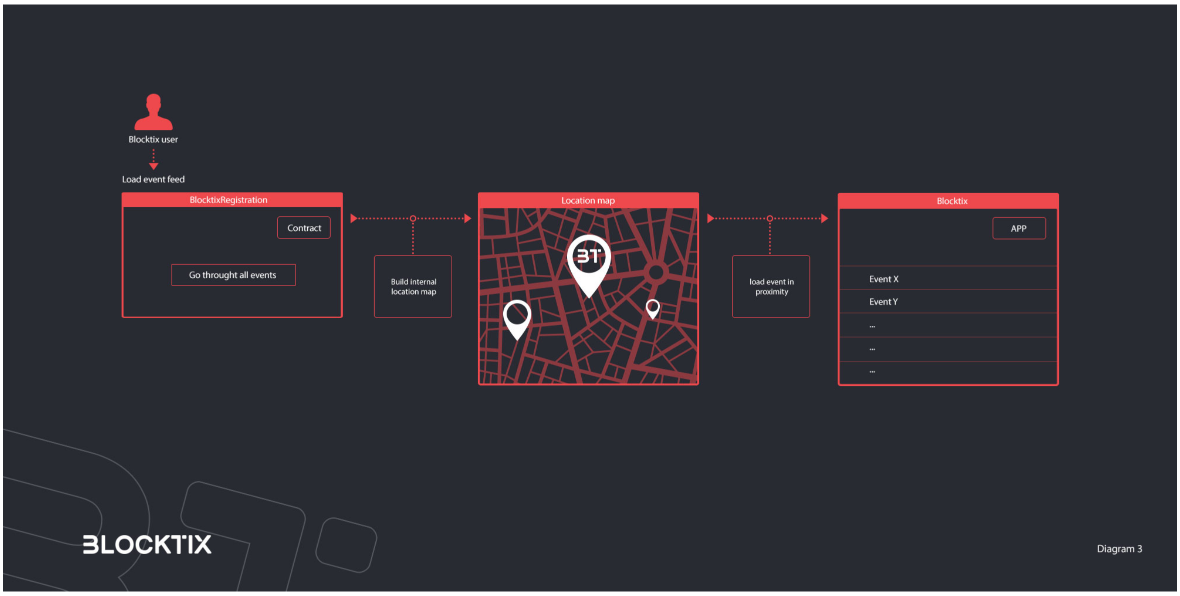


Figure 2: When a user starts the application it will load all the events in proximity

3.3 Event feed

The system keeps track of all the ongoing events and displays these in the event feed. Variables such as distance between the event and the user, starting date, category tags, and advertising fees determine the hierarchy within the list.

Location based Events happening near the user are shown first, as these are more likely to be attended. As the distance between an event and a user increases, the location factor becomes less influential, and the categories start to matter more.

Category based Events can also be tagged with multiple categories by their creator. A user can 'like' categories such that these are more likely to show up in the event feed. Optionally we could track the behavior of the user offline and build the list of categories without visible user input.

Advertising There will always be two advertised events on every page in the event feed. There is no guarantee that the advertised events take place near the user or that they share any relevance to categories liked by the user.

Mandatory TIX The creator of an event must pay a mandatory 1000 TIX for it to show up on the event feed. This is a measure to prevent spammers from overtaking the event feed with bogus events. The mandatory amount is subject to change. It is also worth noting that it is technically possible for someone other than the event creator to pay the mandatory amount, allowing the event to show in the feed as a form of charity.

3.4 User interface

The user interface will be created in HTML and JavaScript. Currently, we have created the designs of the basic pages to get a clear idea of how it would look. The main key ingredients are beauty and intuitiveness. One look at the application and it should be immediately clear how it works.

Event page You can see a preview of a design for the event page, figure two on page 2.

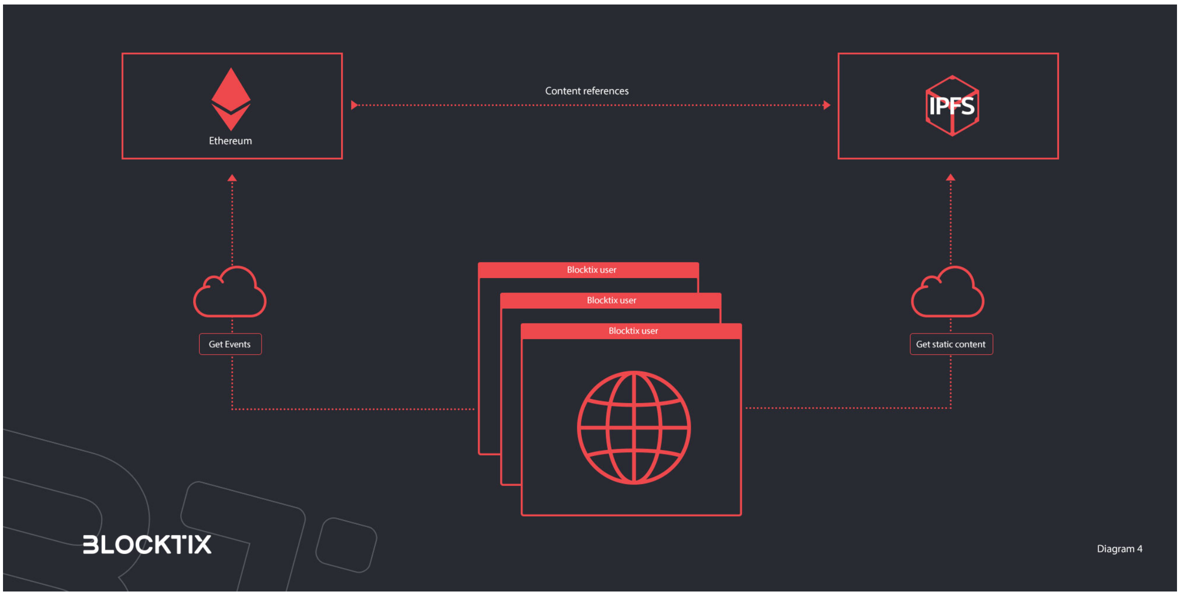


Figure 3: Ethereum contracts store the hashed references to files stored on IPFS.

3.5 IPFS

IPFS is a public peer-to-peer distributed file system. Anyone can upload files free of cost. You can even turn off your computer, and it will still be available to everyone due to its network topology (given that the file is pinned on another computer). Over the past few years, it has accumulated a broad support for many programming languages, JavaScript being one of them. This makes it an ideal candidate to build our infrastructure upon. The reason we need to adopt IPFS is because most blockchains, Ethereum included, do not support sharded databases, rendering them unsuitable for the storage of larger data. This integration will enable event administrators to customize their event pages with images and videos without losing the decentralized topology.

4 Contracts

The Blocktix smart contract programming language will be Solidity[6] as it is currently the flagship language of Ethereum and the most popular language for smart contracts.

4.1 BlocktixRegistration

The registration contract is tasked to store the latest addresses of the Blocktix and BlocktixEvent contracts. It acts as an update mechanism and can only be changed by the Blocktix administrators or through voting by the token holders. The voting is a long term goal and is expected to be added near the end of the development process such that the system can sustain itself.

Blocktix Updates When development is complete, token holders will be tasked to take responsibility for updates to Blocktix contracts. They will have to regulate the system by approving or declining proposed changes to contracts. A wide range of variables are subject to change by the token holders that apply for every new event contract.

4.2 Blocktix

The Blocktix contract houses an array with all event addresses and stores global variables that are enforced on the events and tickets. These include but are not limited to ticket fees, required TIX to enlist

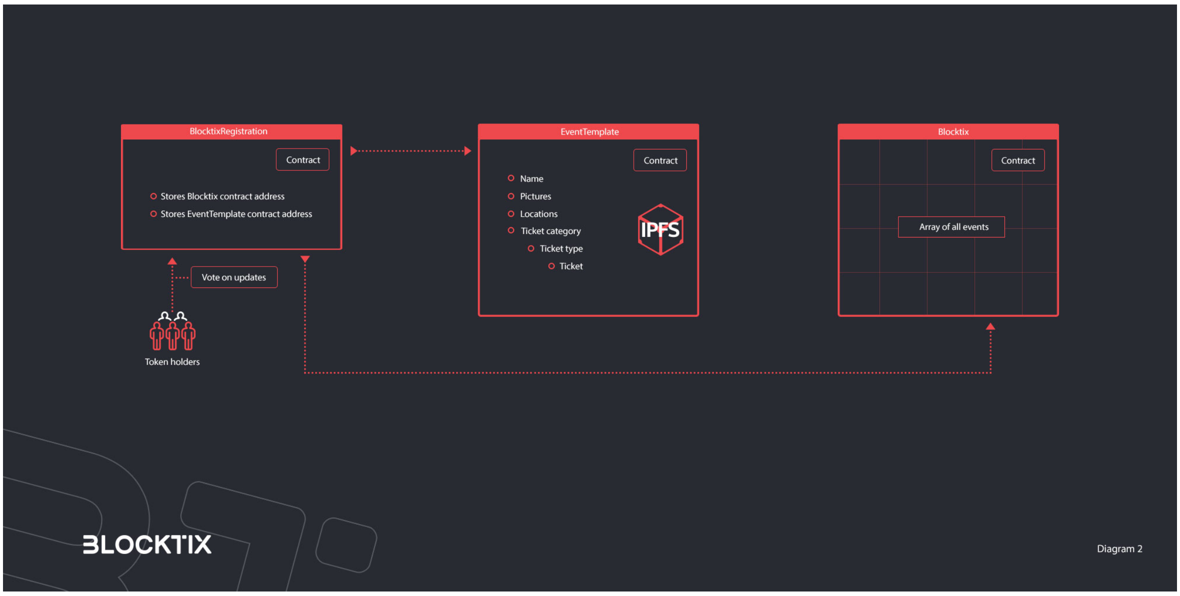


Figure 4: The Blocktix structure can be updated by the token holders who update the BlocktixRegistration contract which essentially acts as an update contract

event in the feed, and most elements related to the advertisement.

4.3 BlocktixEvent

4.3.1 Event Creation

The BlocktixEvent contract is what houses everything related to one specific event. It is brought into existence by the Blocktix contract when creating an event and the creator address is given administrator privileges. Only that address is granted the permission to change settings such as name, location, description, photos, etc. Nevertheless, it is possible to add extra administrators to the event.

Anyone will be able to create an event, but it will only show up on the event feed if the event is backed by a minimum of 1000 TIX. The required value for listing will be voted for by the token holders every three months. Non-token holders can still ask token holders to back the event, as a willing token holder could allow anyone to make use of the event feed.

Security There is a chain of contracts when creating a new event in the system. This ensures the sanity of the contracts being loaded and thus essentially disallowing attackers from launching their own malicious contracts and inserting them into the Blocktix system. Event creators call the Blocktix contracts (retrieved by BlocktixRegistration) which in turn create the BlocktixEvent contract, thereby removing the possibility of tampering.

4.3.2 Ticket Creation

The contract is also the entry point for anything related to tickets of the particular event. The event creator has the possibility of adding multiple ”ticket types” to their event contract such that they can differentiate better and offer multiple types of tickets. For example, they could offer early bird entry tickets, 1 day pass tickets if the event is for multiple days. Each TicketType maintains an array with all of the valid tickets addresses, allowing the autonomous sale of tickets by the BlocktixEvent contract itself and allowing event administrators to manually insert tickets. The latter opens the gap for third parties that wish to integrate with the Blocktix network.

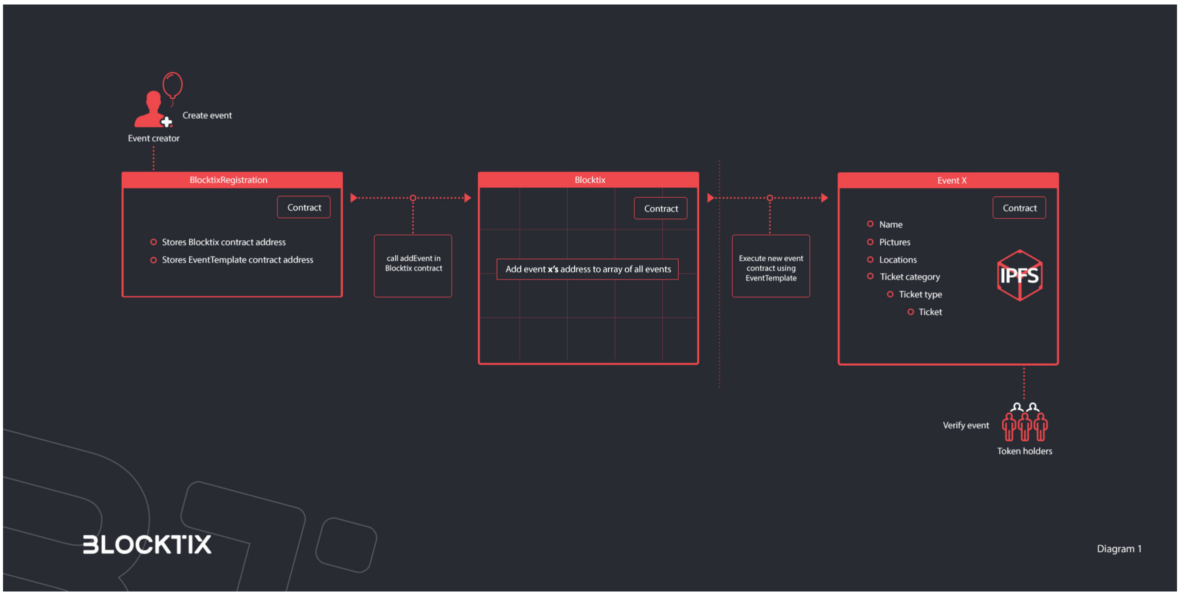

Create event
Event creator
BlocktixRegistration
Contract
Stores Blocktix contract address
Stores EventTemplate contract address
call addEvent in Blocktix contract
Blocktix
Contract
Add event x's address to array of all events
Execute new event contract using EventTemplate
Event X
Contract
Name
Pictures
Locations
Ticket category
Ticket type
Ticket
IPFS
Verify event
Token holders
BLOCKTIX
Diagram 1

Figure 5: Flowchart of a user creating an event

High level overview The event contract stores an array of TicketCategory objects, which in turn stores an array of TicketType objects. The TicketCategory object is merely used to optimize the category system. The TicketType objects store most of the information, including an array of the ticket addresses of all previously sold tickets of that specific TicketType.

4.3.3 TicketCategory

Tickets are not limited to entry tickets. They can also be utilized as vouchers for food and drinks. That's why Blocktix provides categories to allow clean separation of tickets and consumables.

- name (Name of category examples: tickets, food, drinks, ..)
- types (Array of TicketTypes. examples: entry ticket, hamburger, beer,...)

4.3.4 TicketType

Here is an overview for all the properties a TicketType would offer which are all customizable by the event creator.

- name (Name of ticket type)
- description (Description of ticket type)
- quota (Maximum amount of tickets per type)
- price (Price of ticket sale)
- startVendingTime (Start date time of ticket type sales)
- endVendingTime (End date time of ticket type sales)
- startEntryTime (Start date time of ticket type entry)
- endEntryTime (End date time of ticket type entry)
- owners (Amount of tickets sold)
- paid (mapping of paid prices to addresses)
- refundable (is the ticket refundable)

4.3.5 Ticket Trade

The tickets can be freely traded directly between users without risk of counterfeit. Sending a ticket to a friend is easy since you just need to the friend's Ethereum address. Selling tickets for ETH can be arranged in such a way that both the transfer of the ticket and the payment only happen together. This means that the ticket is only transferred if the money is transferred too, and the other way around. This removes the risk of getting scammed due to the other party not upholding their end.

4.3.6 Ticket Auction

The event creator can also auction tickets to maximize revenue or in case of scarce ticket supply to allow the market mechanism to find equilibrium.

TicketBids Within the BlocktixEvent is an array of TicketBid objects. It tracks the bids by users and has the following properties:

- owner (Address of the bidder)
- buy (Boolean - is bid to buy)
- price (bid price)
- ticketID (ID of ticket type)

The buy variable is in place such that it can be extended to allow sell bids as well. This essentially extends the structure to enable both the ticket auctions by the official event creator and the free ticket market where people can trade peer to peer.

Whenever a bid is placed, the array is updated. If it's the only bid in the array, then a time interval starts (set by the event creator) which on expiration automatically sells the ticket to the highest bid in the array. The highest bid is then removed, and all others remain untouched. A new interval then starts, and this loop continues until the array is empty.

4.3.7 Ticket Verification

Ticket verification is as easy as verifying a digital signature and making sure the public key is listed as a valid ticket for the specified event. There is no need for event organizers to hire or purchase specialized equipment to verify the validity of tickets. A smartphone with a camera and an internet connection can be used to scan the ticket QR code at the entrance of the event and subsequently execute the smart contract that deactivates the ticket. The contract is then executed on the Ethereum network to prevent anyone re-using the QR code to gain entrance.

5 Possible attack vectors

Fake events Although tickets are cryptographically secure, there is a risk that they are for the wrong event. This can happen if users could get tricked into believing they are buying tickets for a real event, not realizing a malicious seller has created a fake, duplicate event. There are a few ways to minimize this attack from occurring, but most of them rely on user awareness.

Several indicators will be placed on the event page that make it more obvious to spot fake events. One of them is the amount of tickets that are already sold. Big events most likely will already have sold a decent amount of tickets. Another precaution is to include other events happening at the same location. If a fake event picks the same location, then it will be more visible to both the real event and the visitors. Additionally, if the event has a website, it could host a hidden confirmation identifier proving that the event you are browsing has control of the mentioned website.

Events will be verified by Blocktix Token Holders, requiring them to approve events. Event reporting features will allow users who notice fake events to report them, kicking off a new vote for verification. Invalid events will be removed from the system as this will be the easiest, most reliable and fastest method for getting rid of them. Third party scanners could also be created to compare and red flag events if they are too similar.

References

[1] Kristi Casey Sanders. The true value of the u.s. meetings and events industry. 2011. URL `http://planyourmeetings.com/the-true-value-of-the-u-s-meetings-and-events-industry/`.

[2] PwC. The economic significance of meetings to the u.s. economy, 2012.

[3] Internet fraud n.d. wikipedia, the free encyclopedia. 2008. URL `https://en.wikipedia.org/wiki/Internet_fraud#Internet_ticket_fraud`.

[4] Richard E. Cascarino. *Corporate Fraud and Internal Control.* 2013.

[5] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. 2013. URL `https://github.com/ethereum/wiki/wiki/White-Paper`.

[6] Christian Reitwiessner and Gavin Wood. Solidity. 2015. URL `http://solidity.readthedocs.org/`.