Cyber Valley Tickets

aishift

April 5, 2025

Contents

1 Problem

Cyber Valley wants to host events and needs a convenient way to accept events offers from creators, sell tickets in crypto, verify bought tickets from customer's devices and distribution of acquired means across creator, master and dev team

2 Solution

Create Web3 mobile first web app based on Ethereum network which covers main needs

2.1 Roles

2.1.1 Customer

Public role which has the following authorities:

- List events
- Buy ticket to event

2.1.2 Staff

Assigned to each event by the master role. Has all customer's authorities and:

- Verify tickets for the assigned event

2.1.3 Creator

A.k.a shaman has all customer's authorities and:

- Send event request
- Edit own event requests

2.1.4 Master

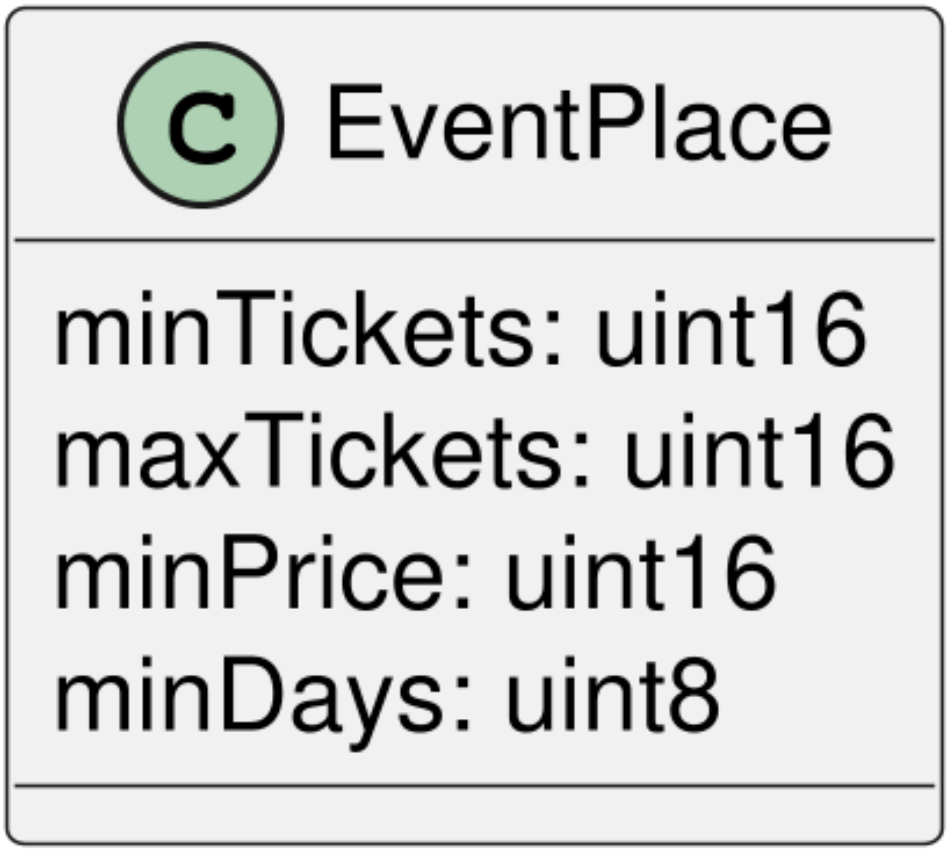
Has all customer's, staff's and creator's authorities and:

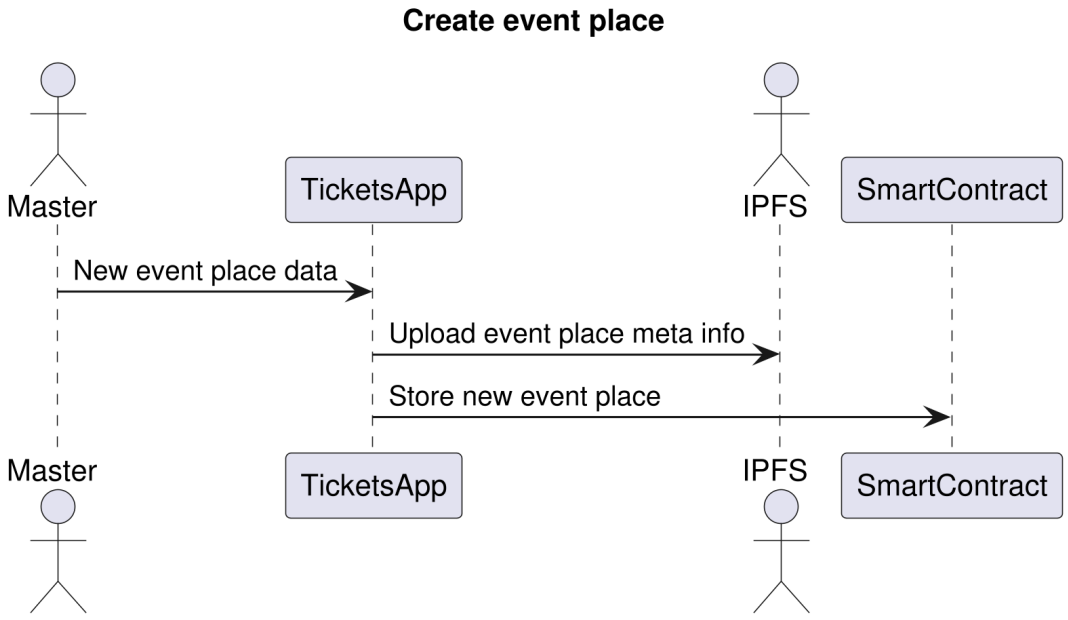
- Create event space
- Approve event requests
- Approve event changes

- Cancel event
- Close events

2.2 Use cases

2.2.1 Create event place





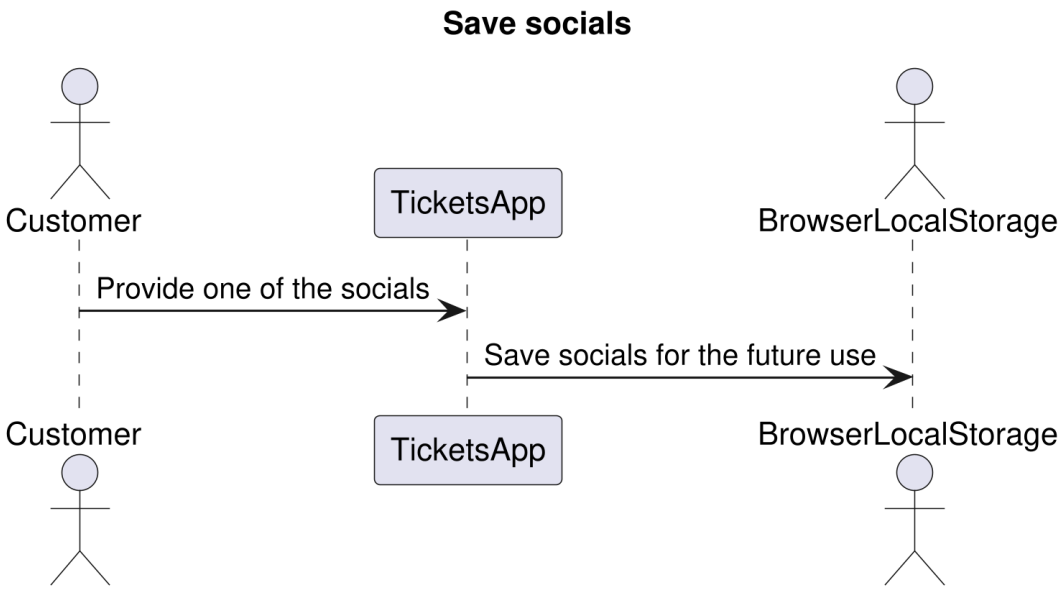
2.2.2 Save socials

Supported socials:

- Telegram
- Discord
- Whats App
- Instagram

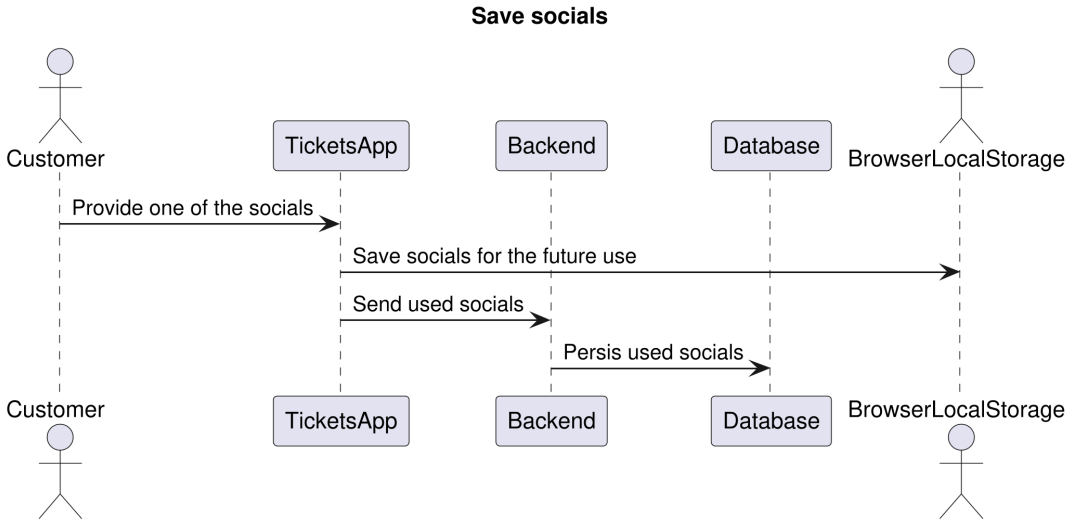
1. V1

   Used socials stored in the browser cache, so customer should input his social on each new device

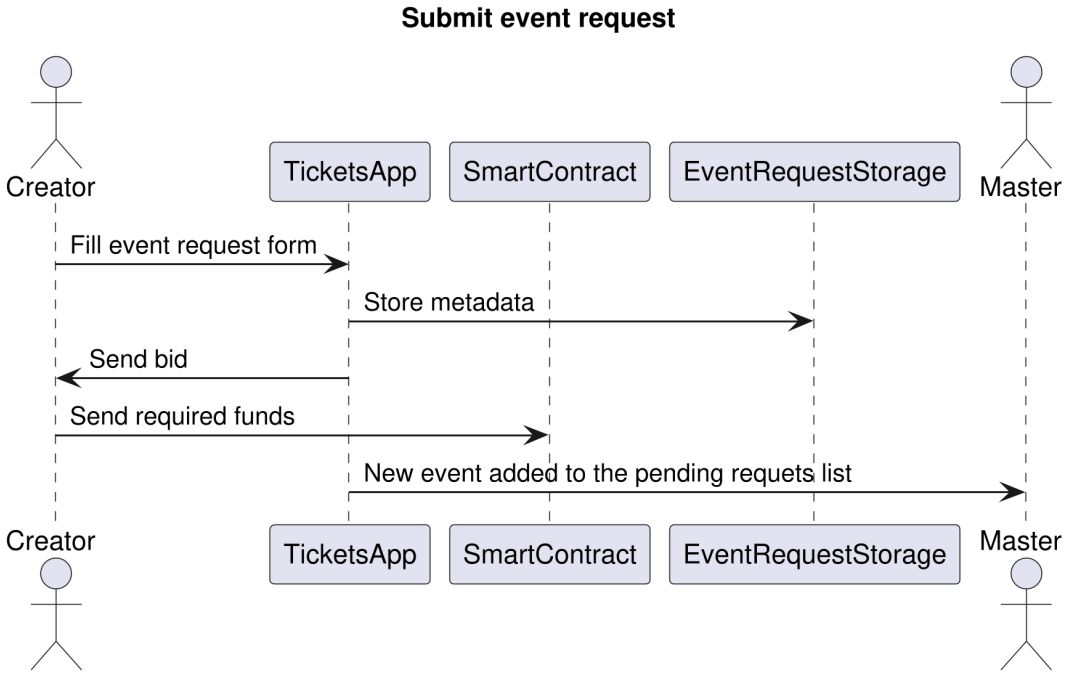


2. V2

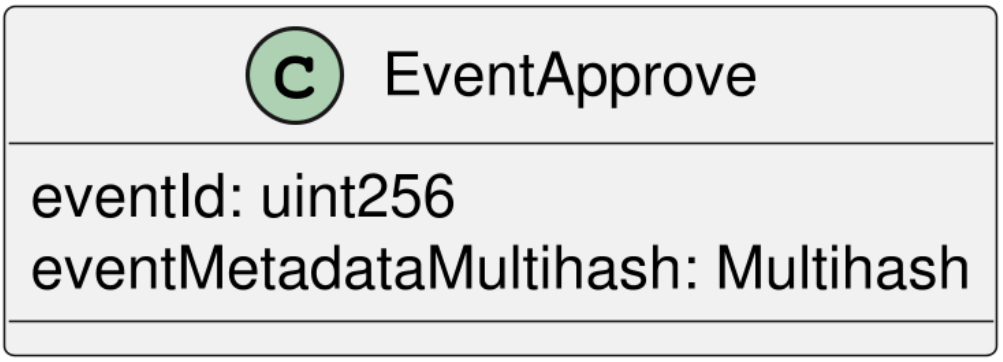
   Used socials stored in the centralized database which allows to sync state of the all devices

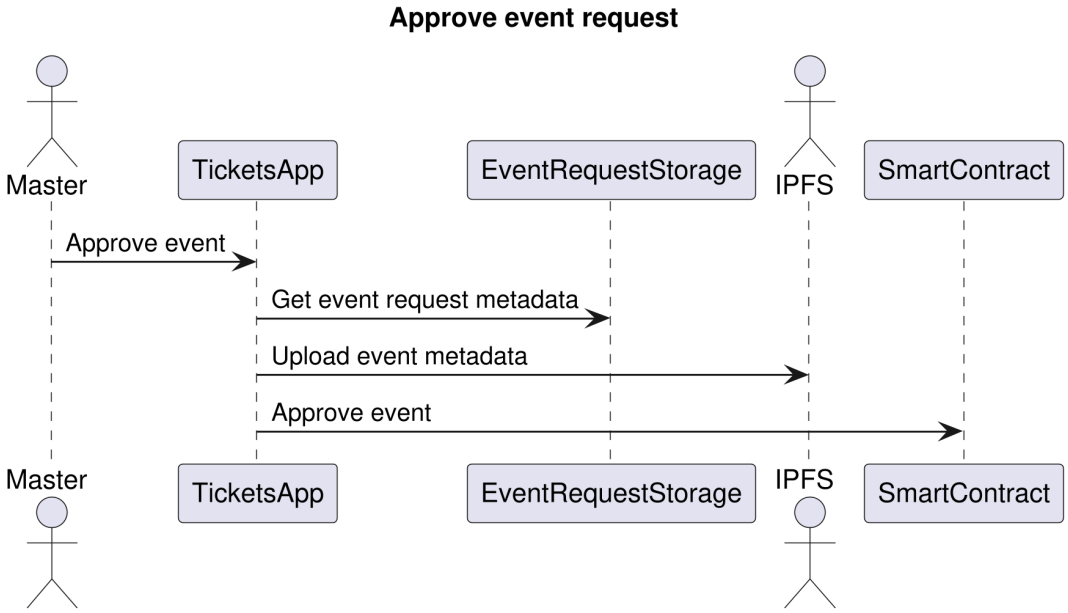


2.2.3 Submit event request

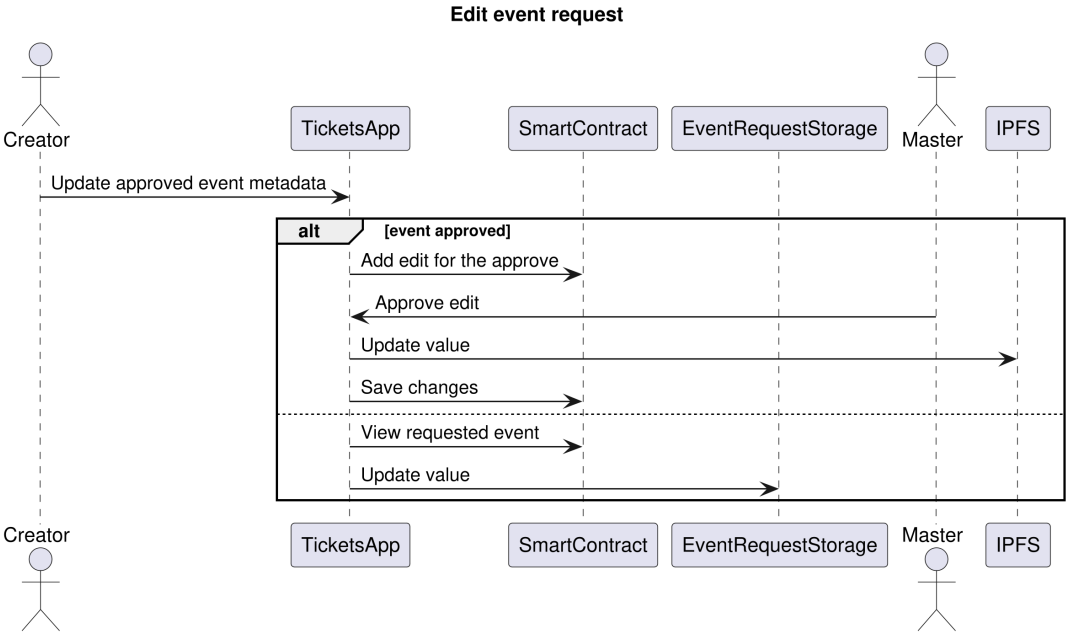


2.2.4 Approve event request

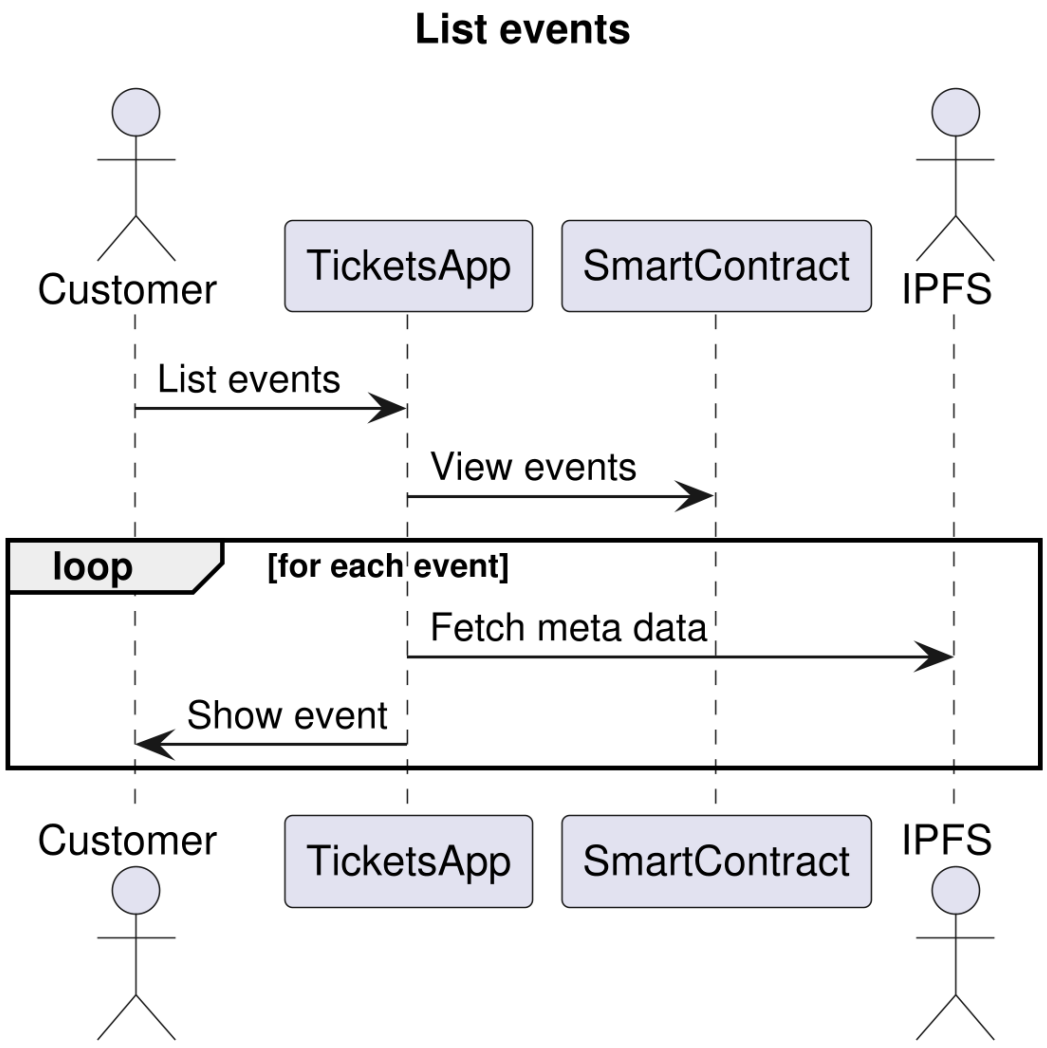




2.2.5 Edit event request

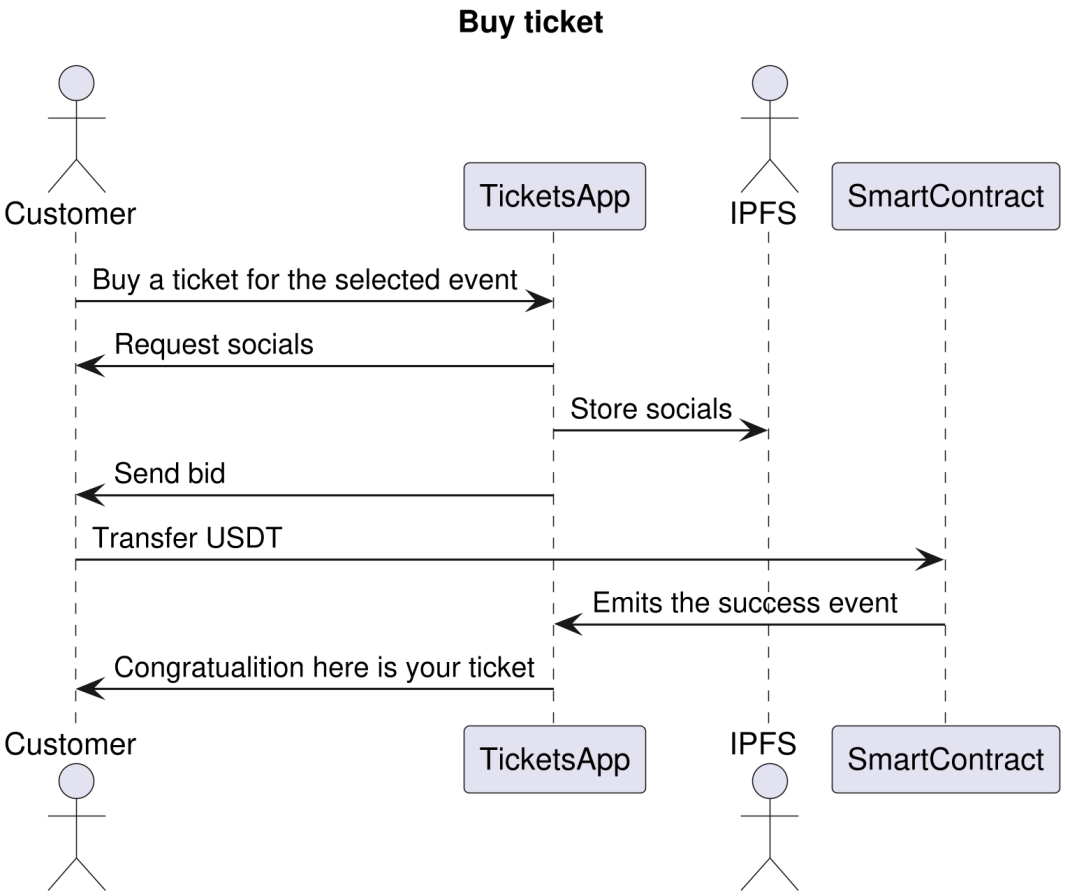


2.2.6 List events

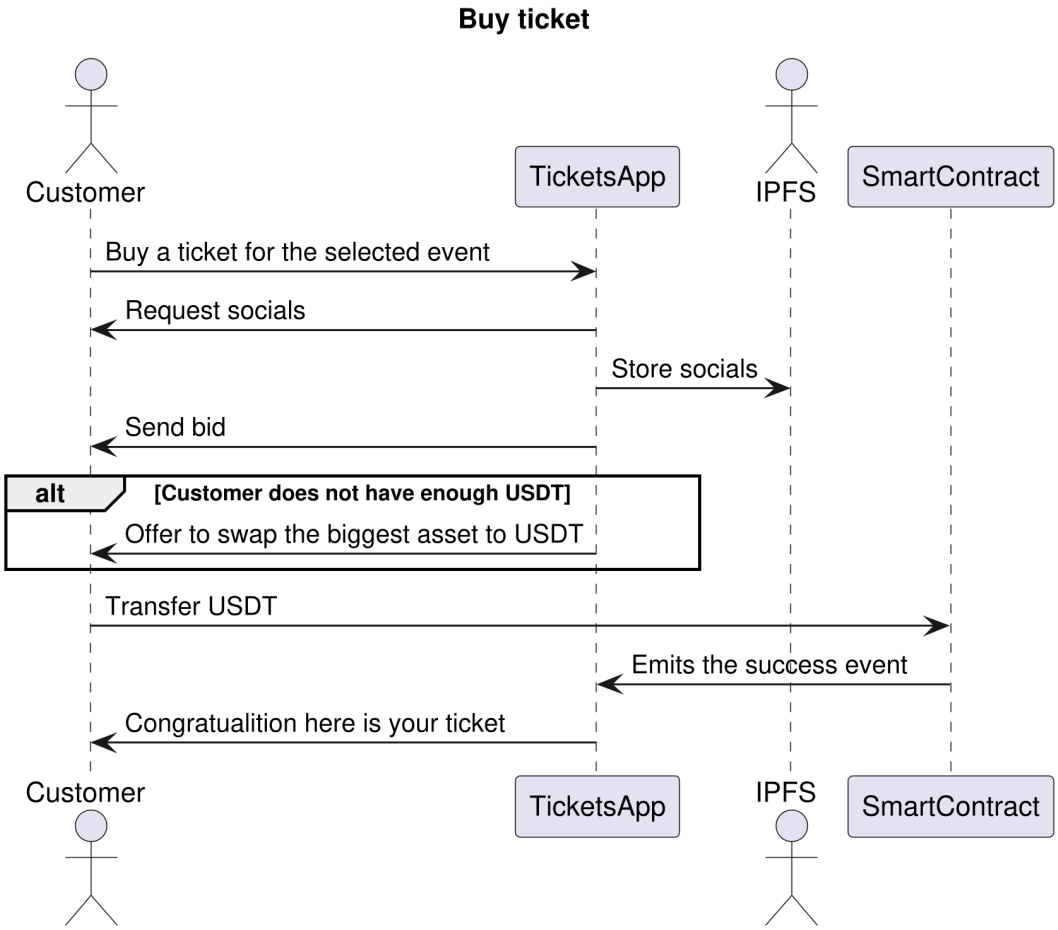


2.2.7 Buy ticket

1. V1


Buy ticket
Customer
TicketsApp
IPFS
SmartContract
Buy a ticket for the selected event
Request socials
Store socials
Send bid
Transfer USDT
Emits the success event
Congratualition here is your ticket
Customer
TicketsApp
IPFS
SmartContract

2. V2

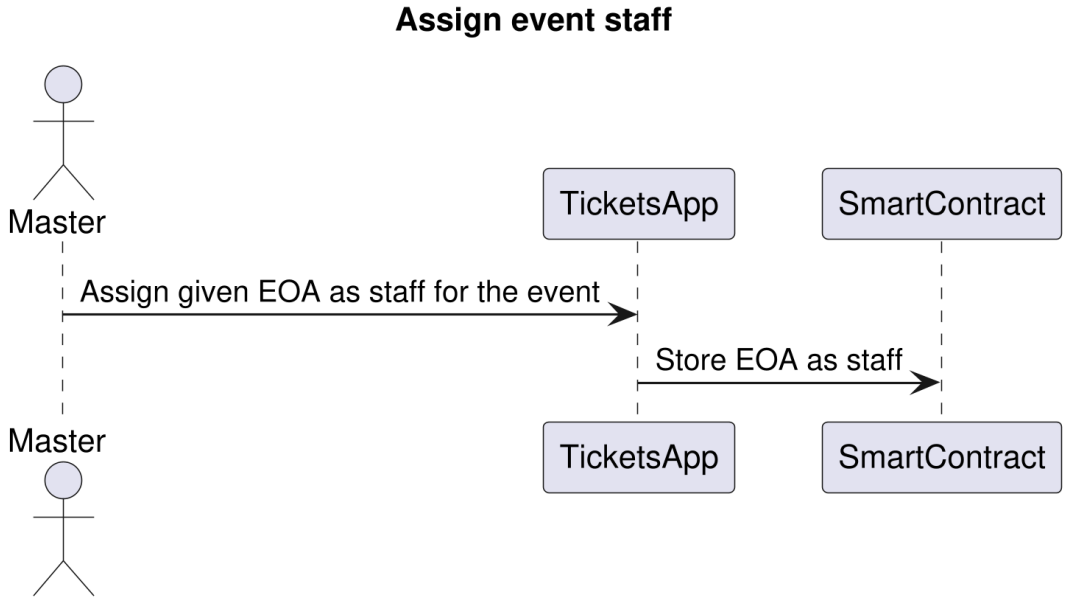


2.2.8 Assign event's staff

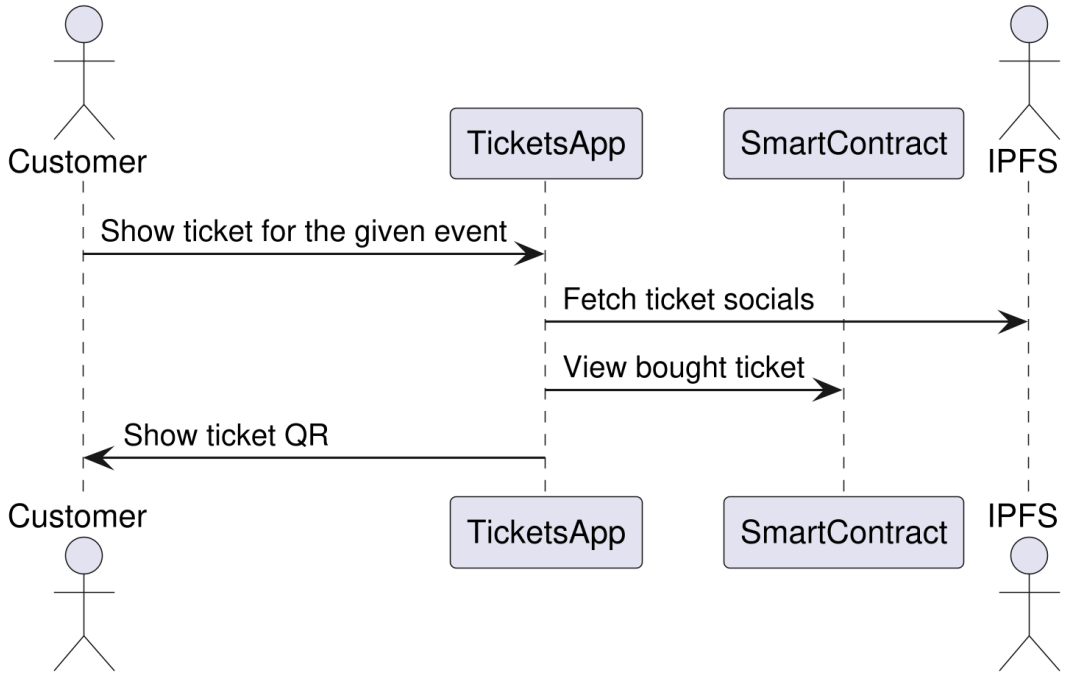
This could be changed to the array of staff independent from the event which can be edited by the master.

Also given approach makes it difficult to list events for the given staff's address and requires GAS for each edit.

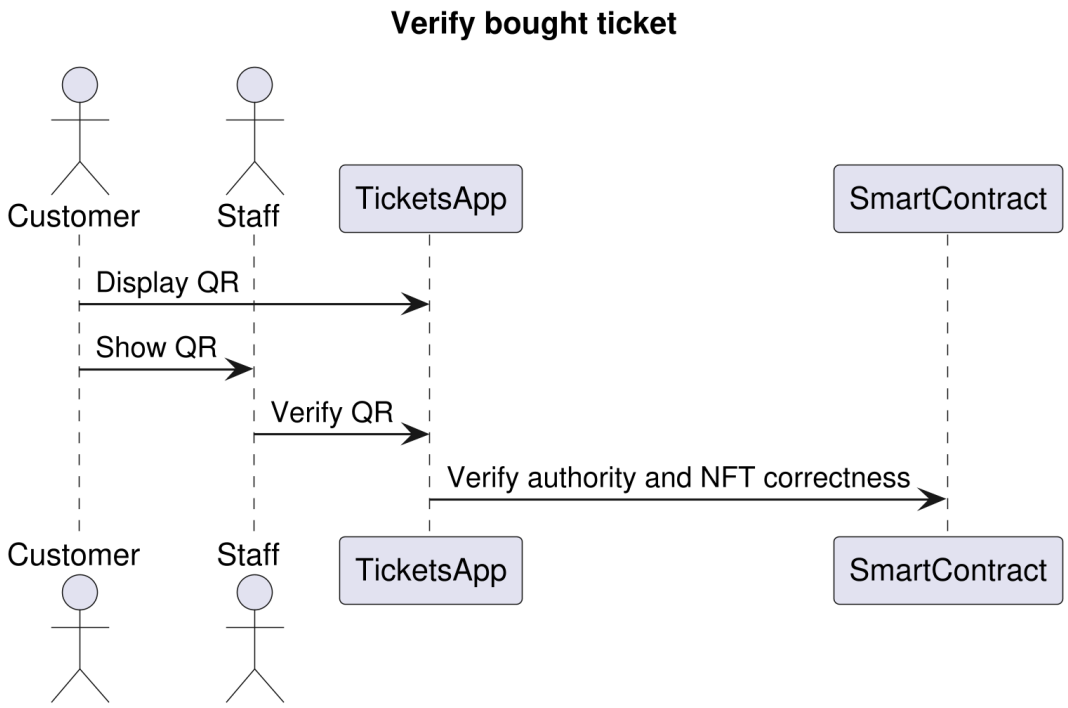
As and alternative we can store staff addresses in the IPFS, but it'll introduce some latency in exchange of less GAS cost.



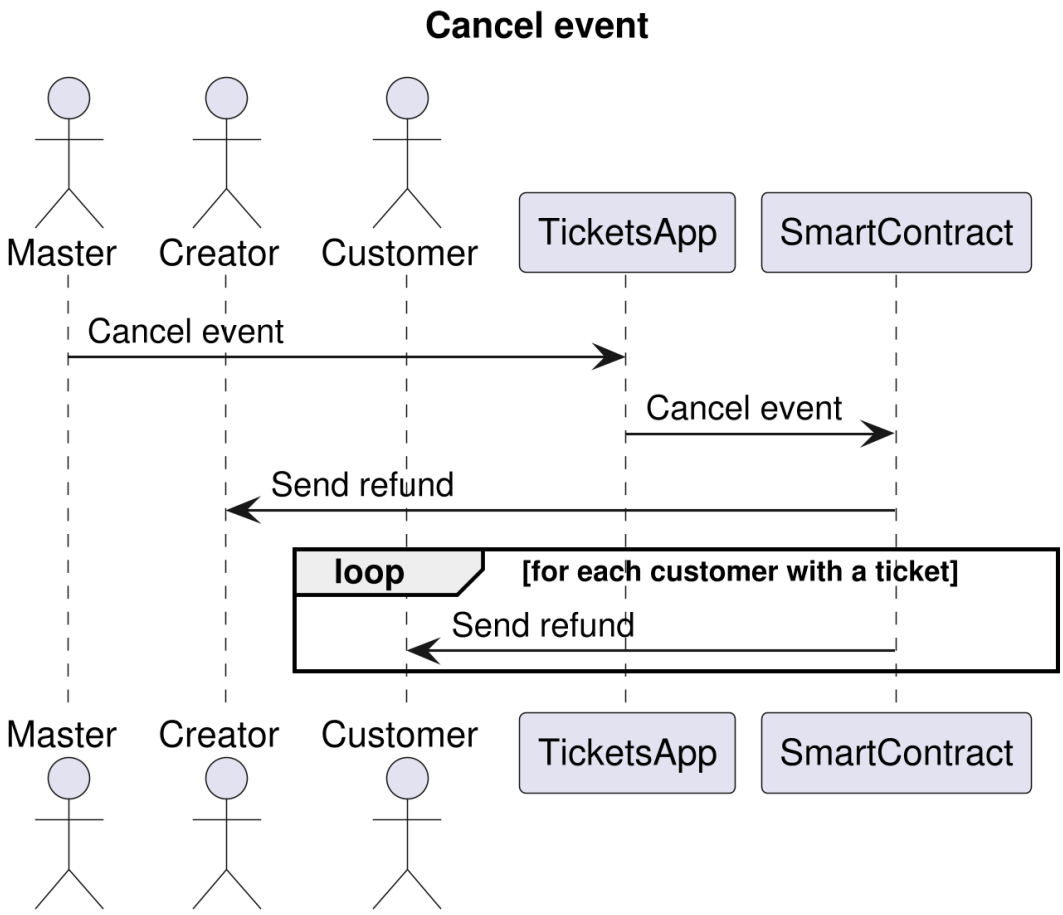
2.2.9 Show ticket



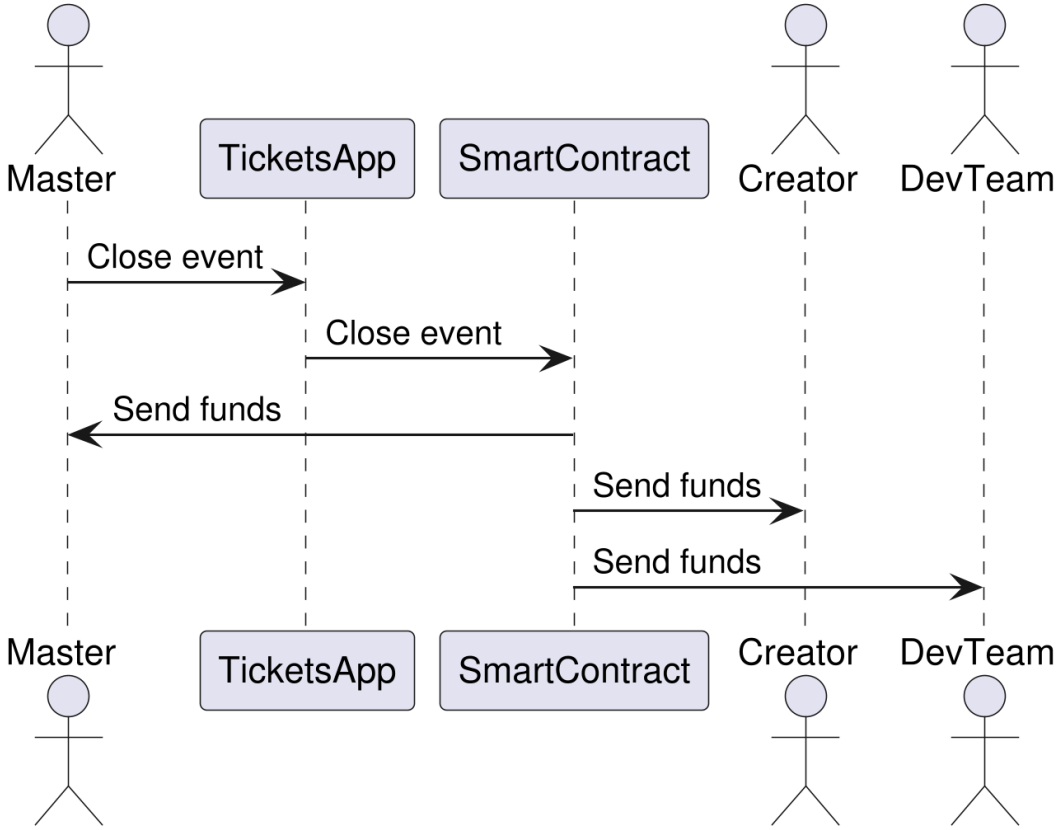
2.2.10 Verify bought ticket



2.2.11 Cancel Event



2.2.12 Close event



2.3 Tech stack

Solidity, OpenZeppelin, React, TypeScript, Tailwind, ethers.js, IPFS

Also a thin backend over database is required to provide free of charge ability to change event request data before it's approve, so it'll be implemented with Python, SQLite and Litestar

2.4 Excluded features from the first stage

Given list of features can be interpreted as obviously required or any section below can unintentionally imply them, so they explicitly mentioned

- Tickets refund
- Cancel or refund event request submission
- Any sort of push notifications about any updates or new data
- Ticket price change on sold out and increasing available seats
- Remove assigned staff person to the event

2.5 Proxy contract vs multiple versions

Because of big amount of reads from the blockchain (which lead to spending gas on call delegation in proxy) we offer to use multiple versions and support them on the client side. To prevent difficulties of funds & data migration between versions, we'll create new events in a new version, but still support the previous ones until all events there will be closed or canceled.

3 Questions

3.1 Both desktop and mobile are required?

Mobile only

3.2 Is it required to verify tickets without internet connection?

No

3.3 Will be there multiple masters or the only one in foreseeable future?

Only one

3.4 Event request price fixed in ETH, depends on ETH/USD rate or could be changed by the master?

Smart contract should work with USDT

3.5 Is a ticket transfer allowed e.g. customer A bought a ticket, but sent it to the customer B?

Yes

It requires additional UI and flows to properly update ticket's meta data, so this feature will be skipped in the V1

3.6 Will tickets have some metainfo about the owner (name, number etc)

Yes, socials i.e. one or many {Telegram, Discord, Instagram, Whats App}

3.7 Is it applicable to show available seats count for all (so the creator and master can see it as well without additional screen)?

Yes

3.8 UI design references