



EVENT TICKET RESELLING

BLOCKCHAIN PROJECT WITH THE AIM TO MITIGATE TICKET SCAMMERS

CONTENT



Idea



Techstack / Design decisions



Architecture



Implementation

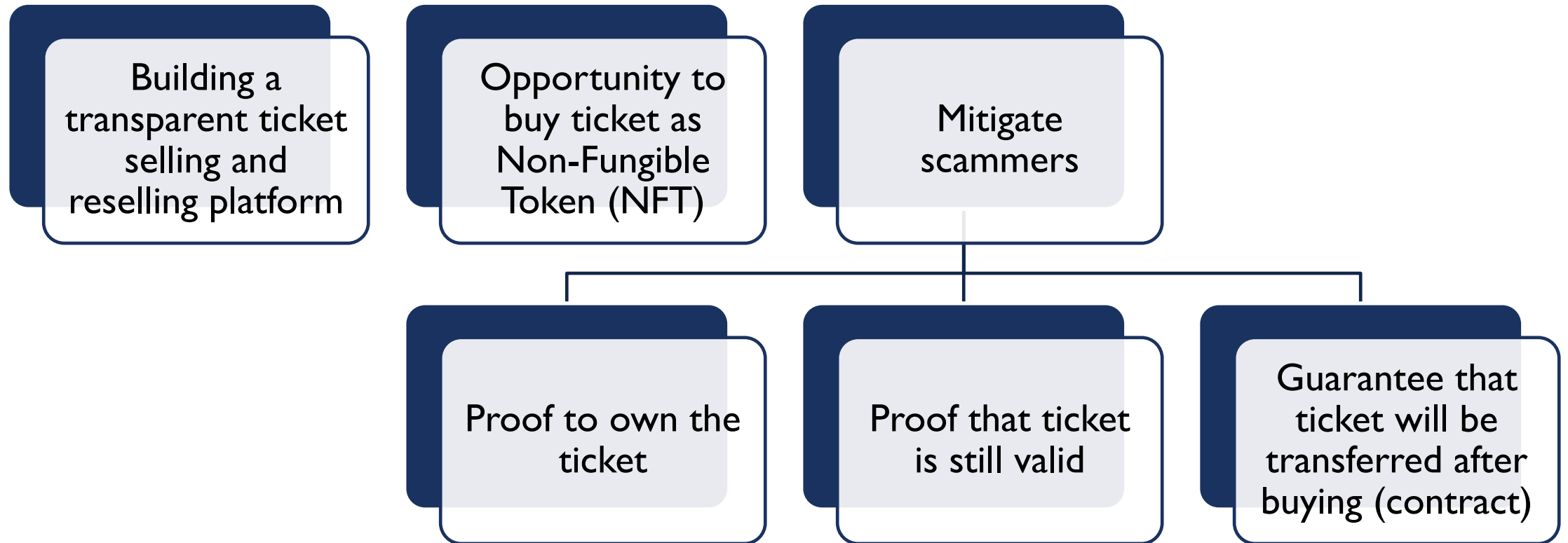


Demonstration



Conclusion

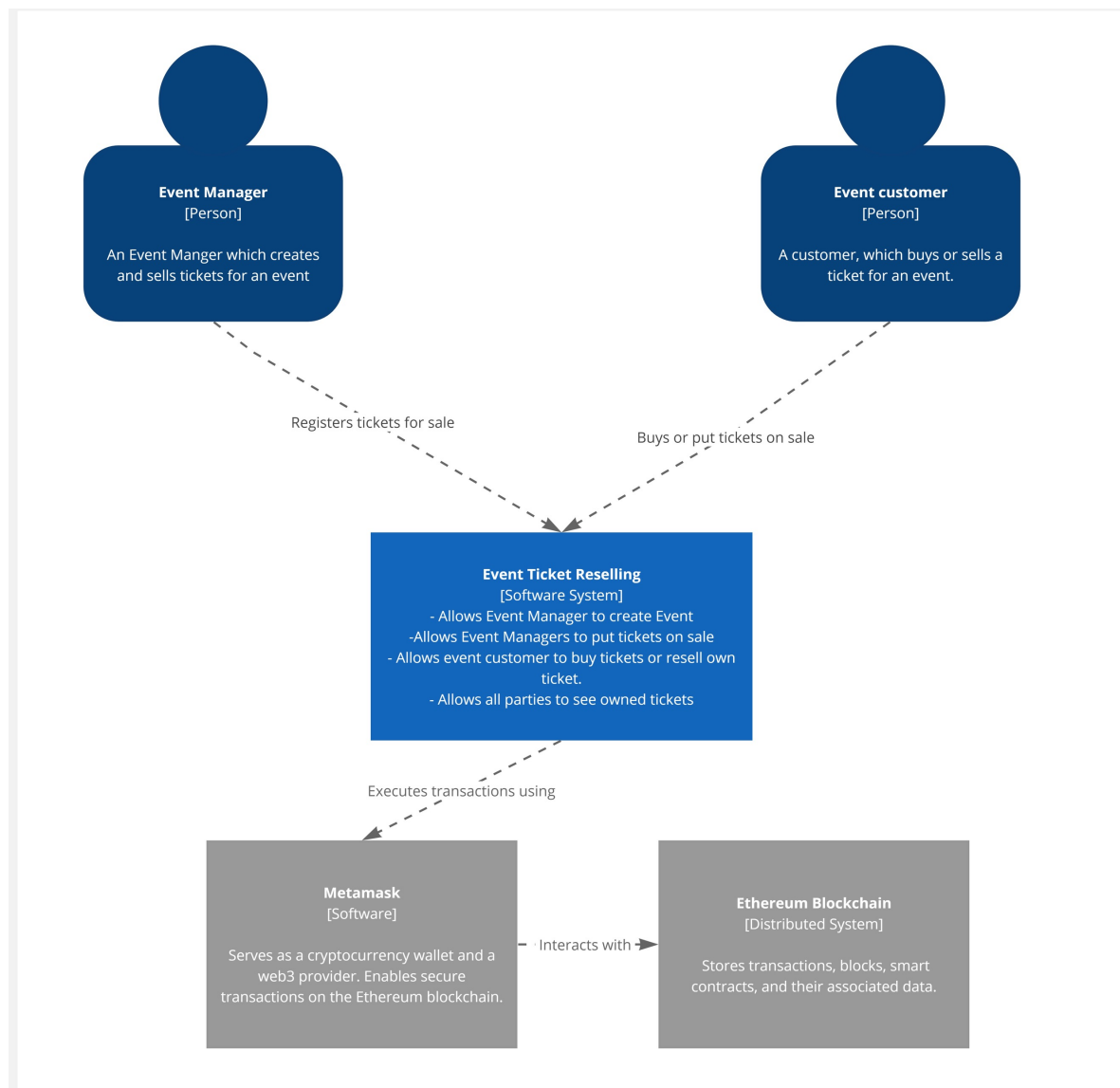
IDEA



TECHSTACK / DESIGN DECISIONS

Application	       
Smart Contract	           
Other Tooling	       

ARCHITECTURE

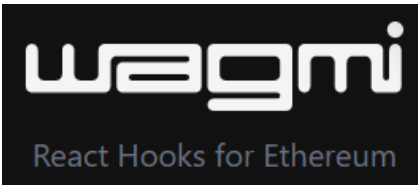


IMPLEMENTATION - APPLICATION



Reading data from the blockchain

```
const { data: transfers } = useEvmWalletNFTTransfers({  
  address: props.address,  
  chain: props.chain,  
});
```



Use read/write methods from the contracts

```
const { data: listings } = useContractRead({  
  address: exchangeContractAddress,  
  abi: loadAbi(),  
  functionName: 'getAllListings',  
});
```



Deploy contracts onto the blockchain

```
const contractCreationTransactionId = await deployContract(client,  
{  
  abi: loadAbiNftContract(),  
  bytecode: `0x${loadBytecodeNftContract()}`,  
  args: [amount, name, abbreviation, exchangeContractAddress],  
  account: address,  
});
```

IMPLEMENTATION - CONTRACT

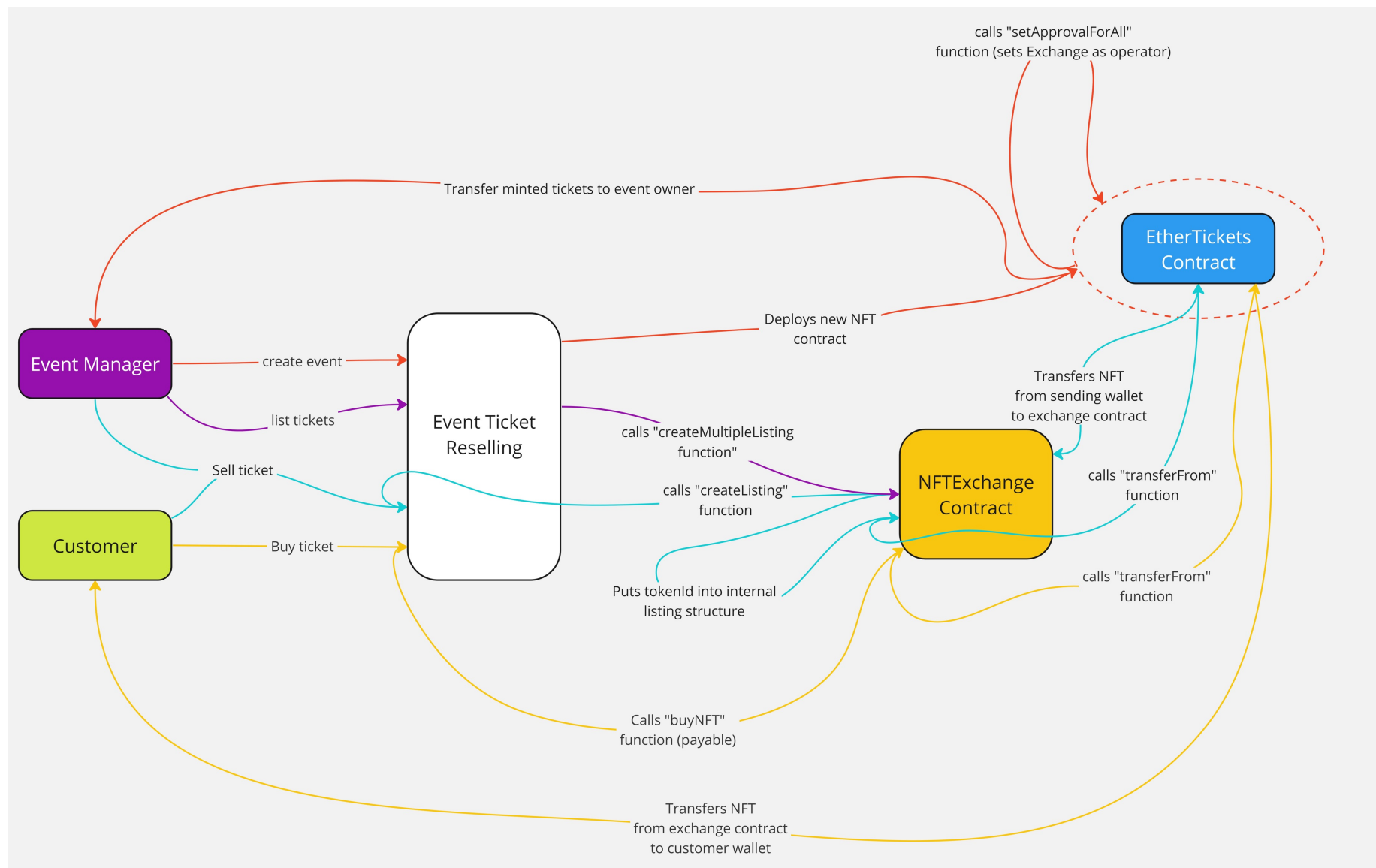
NFTContract “EtherTickets”

- Based on ERC721, ERC721Enumerable, ERC721Burnable and Ownable
- Minting of tickets
- Ownership transfers
- Source [OpenZeppelin](#)

Exchange “NFTExchange”

- Based on IERC721 and Ownable
- Used to sell Tickets (NFT's)
- Payment handling
- Source [Hackermoon.com](#)

IMPLEMENTATION - WORKFLOW





LIVE-DEMO



CONCLUSION

Further improvements

- Feature to validate tickets for customer/event entrance
- Include only our EtherTickets-NFTs in ticket view
- Include reference to further event informations, which are stored off-chain

Retrospective

- Overall fun project with good learning curve
- Idea was maybe to complex/big for this challenge task
- Using next.js frontend framework without prior knowledge was not a good idea



THANK YOU