

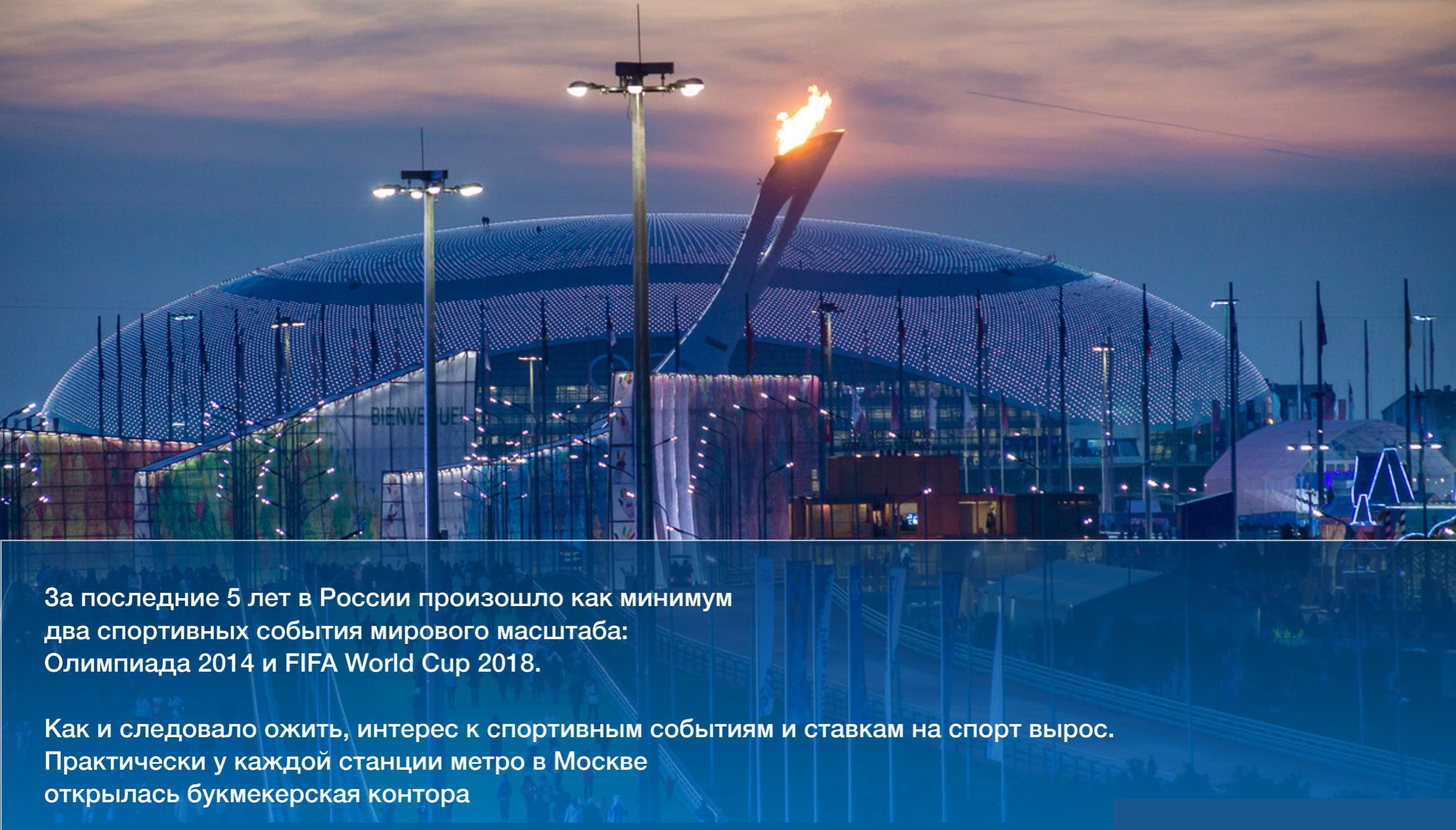
Авторы:
Адаменко Павел Алексеевич
Чуркин Никита Николаевич
Степанов Тимур Алексеевич



“Честные ставки на спорт”

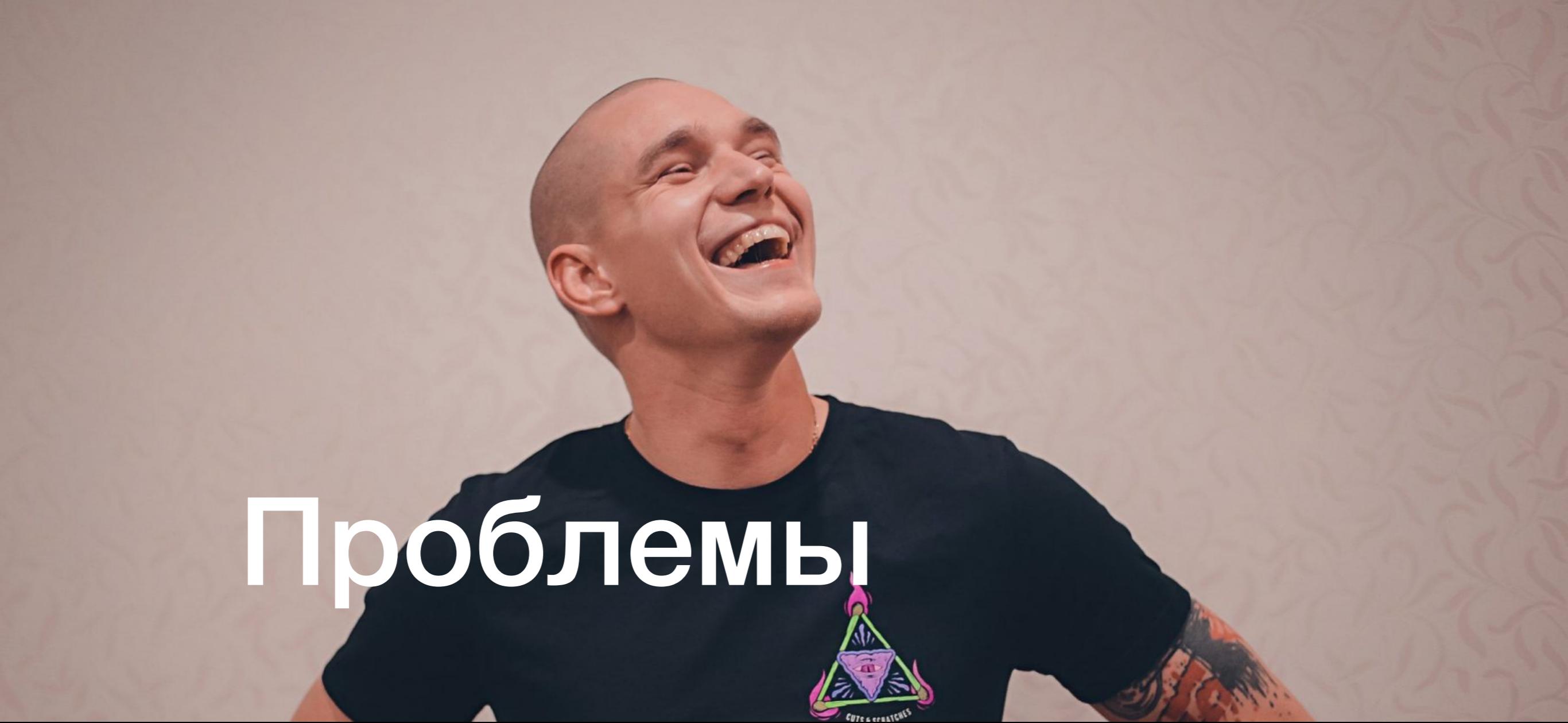
(реализованные на смарт-контракте,
для прозрачности сделок)

Введение



За последние 5 лет в России произошло как минимум
два спортивных события мирового масштаба:
Олимпиада 2014 и FIFA World Cup 2018.

Как и следовало ожидать, интерес к спортивным событиям и ставкам на спорт вырос.
Практически у каждой станции метро в Москве
открылась букмекерская контора



Проблемы

Тем не менее, сомнения в честности работы букмекеров никуда ни ушли,
особенно в интернете

- Где гарантия, что мне вернут деньги?
- Каким образом выставляются коэффициенты?



Решение

Использование blockchain технологии, в частности, смарт-контрактов, позволяет избавить пользователя от нежелательного риска, проверить коэффициенты исхода игр.

Как это работает ?

Смарт-контракт — компьютерный алгоритм,
предназначенный для заключения и поддержания
комерческих контрактов в технологии blockchain

Как это работает?

Классическая модель ставок:

$$Win = bet + \left(\frac{bet}{Total\ bets\ on\ your\ team} * Total\ bet\ on\ the\ other\ team \right)$$

Преимущества blockchain технологии для данной модели:

- Прозрачность схемы
- Отсутствие посредников

Как это работает ?

В рамках смарт-контракта реализован следующий интерфейс:

- Поставить на Зенит
- Поставить на Спартак
- Посмотреть коэффициенты на победу/проигрыш зенита
- Посмотреть на кого была сделана ставка с известного адреса
- Посмотреть все ставки на Зенит
- Посмотреть все ставки на Спартак
- Oracle может определить кто выиграл
- Oracle может уничтожить контракт
- dapp деплоится oraclom

ИСТОЧНИК

https://github.com/padap/hse_smart_contracts

The screenshot shows the Remix Solidity IDE interface. The main area displays the Solidity code for the `Betting` contract. The sidebar on the left shows two files: `browser` and `config`. The right side of the interface includes tabs for `Compile`, `Run` (which is highlighted), `Analysis`, `Testing`, `Debugger`, `Settings`, and `Support`. Below the code editor, there's a transaction history section with a single entry: a call to `TotalLose` with data `0x43c...569f4`. The `Run` tab contains a list of function signatures with their parameters and return types.

```
pragma solidity ^0.4.2;
// Победа зенита [Зенит-Спартак 25/01/2019 18:00 Кубок Матч Премьер]
// There are two command 1(Зенит) abd 2(Спартак)
// Win - win command Зенит(1)
// Lose - lose command Спартак(2)
//
contract Betting {
    string public contractInfo = "Победа зенита [Зенит-Спартак 25/01/2019 18:00 Кубок Матч Премьер]";
    uint256 constant MinBet = 1;

    address public owner;
    uint256 public minimumBet;
    uint256 public totalBetsOne;
    uint256 public totalBetsTwo;

    address[] public players;
    struct Player {
        uint256 amountBet;
        uint16 teamSelected;
    }
    // Адресс персоны <=> информация пользователя
    mapping(address => Player) public playerInfo;
    function() public payable {}
    function Betting() public {
        owner = msg.sender;
    }
}
```

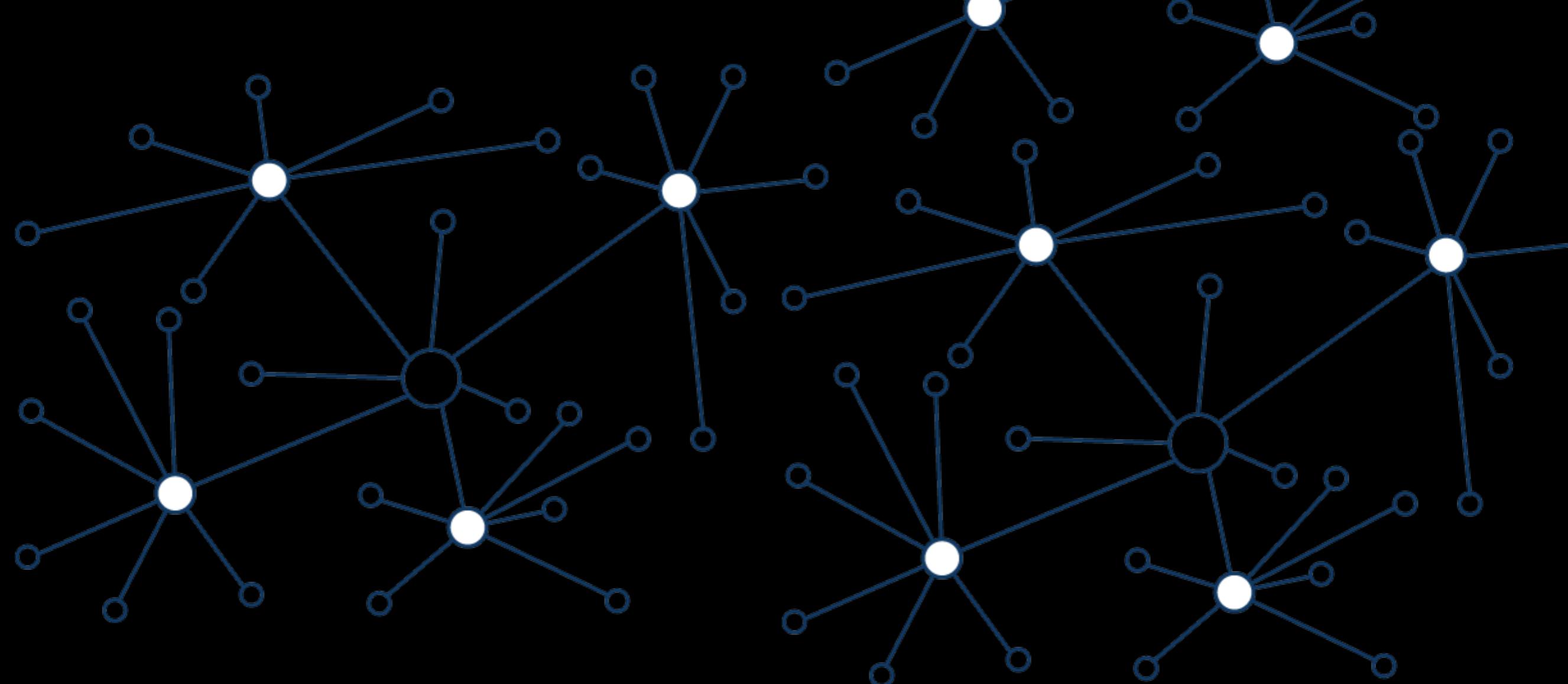
Run tab content:

- bet_win
- distributePrizes uint16 teamWinner
- kill
- checkPlayerExists address player
- contractInfo
- description
- get_lose_coefs 0: uint256: 1
- get_win_coefs 0: uint256: 1
- minimumBet 0: uint256: 1
- owner 0: address: 0xCA35b7d915458Ef540aDe6068dFe2F44E8fa733c
- playerInfo address
- players uint256
- totalBetsOne 0: uint256: 0
- totalBetsTwo

Дальнейшие улучшения



- Продвинутая система ставок
- Приятный front интерфейс
- Решение проблемы с oracle



Спасибо за внимание!