Optimistický dizajnový vzor v eUTxO modeloch

Diplomová práca

Bc. Michal Porubský
Ing. István Szentandrási, PhD.

Blockchain

Distribuovaná databáza

- Peniaze sú reprezentované ako tokeny
- Monetárna hodnota je nedôležitá
- Používatelia sú reprezentovaní dvojicou (súkromný kľúč, verejný kľúč)
- Samotný blockchain uchováva distribúciu tokenov; **kto vlastní koľko čoho**

UTxO model

Unspent transaction output model

- UTxO model je jeden zo spôsobov ako si pamätať distribúciu tokenov
- Základná jednotka, ktorá mení distribúciu je transakcia
  - vstupy, výstupy, podpisy, …
- Pamätáme si všetky transakcie, ktoré sa kedy stali
- Aktuálna distribúcia je **množina všetkých nepoužitých výstupov zo všetkých transakcií**
- Používaný napr. v Bitcoine, Cardane, Ergo, …

Smart kontrakty

Ako poskytnúť peniaze na špecifický účel tretej osobe.

- V UTxO modeli vie vytvoriť transakciu s konkrétnym UTxO len používateľ, ktorý vie súkromný kľúč
- Nechceme distribuovať súkromný kľúč
- Ďalší typ entity, tzv. smart kontrakty (validátor skripty)
- Tokeny vieme poslať na adresu validátor skriptu

Smart kontrakty

Ako poskytnúť peniaze na špecifický účel tretej osobe.

- Hocikto vie vytvoriť transakciu, **ak validátor súhlasí**
- Validátor je Haskell funkcia typu ***Datum -> Redeemer -> TxInfo -> Bool***
  - Rozhoduje, či súhlasí (***Bool***) s transakciou (***TxInfo***) na základe uloženého stavu (***Datum***) a pomocnej informácie (***Redeemer***)
  - Stav ukladá používateľ, ktorý transakčný výstup vytvára
  - Pomocnú informáciu predkladá používateľ, ktorý transakčný výstup používa

Smart kontrakty

Ako poskytnúť peniaze na špecifický účel tretej osobe.

- UTxO model s takýmito smart kontraktami = eUTxO model
- Validátory sú zväčša veľmi presne a jasne definované, auditované, open-source a povoľujú iba malú množinu operácií
- Dá sa im veriť

Centralizovaný stav

Prečo je to časté.

- Často nie je žiadané distribuovať stav na viacerých miestach
- Často to ani nie je také jednoduché (nie je možné)

Centralizovaný stav

Problém s konkurentným používaním smart kontraktu.

- Problém s viacerými ľuďmi vytvárajúcimi transakcie, čo sa vzájomne vylučujú
- Viac používateľov vytvára transakcie kde používajú ten istý UTxO na vstupe
- Lenže len jedna transakcia konzumujúca tento UTxO je možná
- Toto limituje interakciu s protokołom
  - Jedna za blok, čo je na Cardane ~20 sekúnd

Centralizovaný stav

Riešenie pomocou Agenta.

- Kvôli limitovanému kontextu validátorov je problém vymyslieť dizajn, ktorý toto elegantne rieši
- Preferovaný spôsob je pridanie novej entity, **Agent**a
- Iba Agent má schopnosť použiť UTxO s centralizovaným stavom
- Interakcia sa rozdelila na 2 kroky:
  - Používatelia vytvoria transakciu, ktorou dajú najavo čo chcú spraviť
  - Agent spracuje ich požiadavku

Centralizovaný stav

Riešenie pomocou Agenta.

- Rola Agenta má **privilegované** práva
- Validátory stále sú schopné kontrolovať správanie Agenta, ale nevedia skontrolovať úplne všetko
- V závislosti od konkrétnej aplikácie tak ide o rôzne závažné riziká
- Čo napríklad nie je možné skontrolovať:
  - Ignorácia požiadavok
  - Spracovávanie požiadavok v inom ako zamýšľanom poradí (ak na tom záleží)
  - …

Centralizovaný stav

Riešenie pomocou Agenta.

- Výzva riešení, ktoré idú týmto smerom je:
  - Spraviť Agenta čo najmenej privilegovanou rolou
  - Vytvoriť opatrenia, ktoré by trestali Agentove nečestné správanie a postihli ho. Teda nie prevencia, ale následok
- Téma diplomovky sa týka **druhého bodu** a opisuje **2 spôsoby** ako sa dá **trestať** Agentove **nečestné** správanie

Optimistický dizajnový vzor

Idea

- Dizajnový vzor sme nazvali Optimistický dizajnový vzor
- Optimisticky **dúfame**, že Agenti budú **čestní**, dáme privilegované právomoci **hocikomu** kto má záujem a splní isté **podmienky** a v najhoršom ho **potrestáme** a **odoberieme** mu privilegovanú rolu, **ak čestný nebol**.

Optimistický dizajnový vzor

Vytváranie Agentov

- Autentifikáciu Agentov realizujeme nám vlastným Agent tokenom. Dokážeme obmedziť ako sa používa
- Agent token vytvoríme pre Agentov, ktorí zložia dostatočne veľký collateral ako zálohu
- Agent token nedáme priamo Agentom. Docielime, aby vždy mohol byť len na **Agent skript** adrese
- V prípade, že disponujeme **dôkazom**, že daný Agent nebol čestný, tak mu vieme **odobrať** Agent token a zároveň mu vie **prepadnúť** zložený **collateral**

Optimistický dizajnový vzor

Aukcia - úvod

- Vlastník sa snaží predať jedinečný token (NFT) za čo najviac tokenov s hodnotou
- Používatelia môžu ponúknuť tokeny na výmenu za NFT (vytvárajú ponuky)
- Vyhráva kupca, čo ponúkne najviac tokenov
- Vďaka smart kontraktom je možné garantovať:
  - Predávajúcemu, že tokeny dostane
  - Kupcovi, že kúpi ak ponúkol najviac a Agent ponuku spracoval
- Agenti spracovávajú bidy

Optimistický dizajnový vzor

Získanie dôkazu priamo na blockchaine (možnosť 1)

- V prvom spôsobe nájdeme dôkaz priamo na blockchaine. V našom prípade s aukciou by to napríklad mohla byť **nespracovaná požiadavka** na kúpu NFT ak vieme (!) zaručiť kedy bola vytvorená a že mala byť spracovaná
- Dôkaz pozostáva z nespracovanej požiadavky, (doterajšieho) výsledku aukcie, zloženého collateralu a Agent scriptu
- Toto je ťažší spôsob, je tu veľa implementačných detailov
- **Toto sme implementovali**
- Nie je však použiteľný pre každú aplikáciu

Optimistický dizajnový vzor
Aukcia - detaily

- Skripty
  - Bid skript
  - Aukčný skript
  - Agent skript
- Tokeny
  - Bid validity token
  - Aukčný validity token
  - Agent token

Optimistický dizajnový vzor
Získanie dôkazu hlasovaním (možnosť 2)

- Druhý spôsob získania dôkazu je naopak **použiteľný vždy**
- Opiera sa o existenciu vlastníkov aplikácie
- Všetko čo sa na blockchaine stane je overiteľné hocikým, keďže všetky transakcie sú verejne dostupné
- Hocikto môže prísť s tým, že daný Agent sa správal nečestne
- Ľudia vlastniaci aplikáciu hlasujú o tom, či je to pravda alebo nie
- Hlasujú vytváraním transakčných výstupov na hlasovacom skripte

Optimistický dizajnový vzor

Získanie dôkazu hlasovaním

- Vyhodnotenie hlasovania je problém
- Hocikto si vie pozrieť výsledky, ale zabezpečiť čestnosť v **ohlasovaní výsledkov** nie je jednoduché
- Namiesto toho delegujeme tento problém na iný už inak vyriešený problém orákula
- Orákulum je schopné poskytnúť na blockchain transakčný výstup so stavom obsahujúcim výsledok hlasovania
- Výsledok volieb ohlásený orákulom použijeme ako **dôkaz** na odobratie Agent tokenu a prepadnutie collateralu

Prínos diplomovej práce

- Identifikovali sme problém, ktorý sa ukazuje ako častý
- Riešenie na problém nie je zatiaľ nikde v praxi dotiahnuté
- Navrhli sme 2 možné vylepšenia daného riešenia (dizajnový vzor)
- Demonštrovali sme ten ťažší z nich na Aukčnej aplikácii, ktorú sme sami naimplementovali

Ďakujem za pozornosť!

Priestor na otázky

Q: Typické zranitel'nosti

- Double satisfaction
- Loose conditions
- Resource DoS
- ...

Q: Odmena Agentov

Transakčné poplatky vedia byť veľké

- Vynechané pre jednoduchosť, v práci sme ich ale spomenuli
- Konštantný malý poplatok môže byť napr. súčasťou každého bidu

Q: Odignorovanie bidu tesne pred koncom

- Cool-down period
  - Uzatvoriť aukciu je možné až po nejakom čase po deadline
- Vytvárame priestor na zaručenú existenciu aukčného UTxO
- Tým pádom vieme odobrať Agentovi collateral

Note: Verifikácia

- Formálna verifikácia je študovaná
- Prakticky žiaľ zatiaľ nemožná, iba v extrémne limitovanej forme
- Takisto zoznam **všetkých** zraniteľností je problém získať (neexistuje)
- Model sa však dal viac sformálniť

Note: Off-chain hlasovanie

- Dve rôzne možnosti implementácie návrhového vzoru
  - Rovnaký cieľ, iné predpoklady a závislosti
- Implementácia ukazuje jednu z možností vo veľkom detaile
  - Tú komplikovanejšiu a zaujímavejšiu

Q: Hodnota NFT > collateral

- Nastáva ak:
  - Rapídne sa zvýši hodnota NFT
  - Cena podkladovej kryptomeny rapídne klesne
- Dostatočne krátke trvanie aukcie
- Rôzne vysoké collaterals, horný odhad predávajúceho
- Collateral v stablecoine (drží hodnotu voči USD)

Ďakujem za pozornosť!