Digitálne meny a blockchain

**Hyperledger Fabric smart systém**

Roman Bitarovský, Emma Macháčová

**Meno cvičiaceho** : Viktor Valaštín

**Čas cvičení** : Streda, 18.00

**Dátum vytvorenia** : 01.05. 2022

Obsah

[1. Cieľ projektu 1](#_Toc102327361)

[2. Voľba implementačného prostredia 2](#_Toc102327362)

[3. Návrh architektúry a jej prepojenia 3](#_Toc102327363)

[4. Implementované časti kódu 4](#_Toc102327364)

[4.1. Asset Transfer 4](#_Toc102327365)

[4.2. Application 4](#_Toc102327366)

[5. Opis prostredia na testovanie a opis vybraných testov 5](#_Toc102327367)

[6. Užívateľská príručka 6](#_Toc102327368)

[6.1. CLI pre cestovnú kanceláriu 6](#_Toc102327369)

[6.2. CLI pre EconFly a BusiFly 6](#_Toc102327370)

[6.3. Spúšťanie aplikácie 6](#_Toc102327371)

[7. Odpovede na otázky 7](#_Toc102327372)

[8. Záver 8](#_Toc102327373)

1. Cieľ projektu

Cieľom projektu je implementovať a konfigurovať **biznis sieť nazvanú FlyNet**, a teda vytvoriť biznis sieť poskytujúcu viac efektívnu B2B (biznis-to-biznis) spoluprácu s využitím **Hyperledger Fabric**.

Spoločnosti vo FlyNet majú rôzne role. EconFly a BusiFly vytvárajú lety a objednávajú sedadlá a GladlyAbroad dopytuje lety a rezervuje sedadlá pre zákazníkov. Keď je prijatá rezervácia od cestovnej kancelárie, korešpondujúca aerolínia objedná požadované sedadlá aktualizovaním informácií v blockchaine.

Aplikácie môžu interagovať so sieťou a aktualizovať stav pomocou chaincode funkcií. Možnými transakciami sú:

* **createFlight(flyFrom, flyTo, dateTime, seats)**
  + Môže byť volaná jedine aerolíniou na vytvorenie nového letu s danými from, to, dateTime a počtom seats dostupných v lete. Potom let uloží do ledgeru a vráti správu o úspechu volania iba ak je volajúci tejto funkcie niekto z aerolínií. Inak by mala vrátit error. Parameter flightNr nevstupuje ako argument funkcie, musí byť vygenerovaný automaticky.
* **getAllFlights()**
  + Vráti všetky dostupné lety uložené v ledgeri.
* **getFlight(id)**
  + Vráti informácie o lete s daným id.
* **reserveSeats(flightNr, number)**
  + Môže byť volaná jedine cestovnou kanceláriou na rezervovanie number sedadiel na lete s id flightNr.
* **bookSeats(reservationNr)**
  + Volaná jedine aerolíniami na zmenu stavu rezervácie s reservationNr na „completed“ a aktualizovanie záznamu na ledgeri korešpondujúceho letu. Sedadlá sú objednané iba keď je požadované množstvo sedadiel stále dostupné na danom lete a keď flightNr rezervácie patrí aerolíniám, ktoré spracúvajú danú rezerváciu sedadiel.
* **checkIn(reservationNr, passportIDs[])**
  + Volaná buď cestovnou kanceláriou alebo konečným zákazníkom za účelom vybrať si konrétne sedadlá v lietadle a zaevidovanie cestovných dokladov na účely evidencie pasžierov v letoch. Cestovné doklady sú pár customerNames z rezervácie a čísla dokladu. Môžu byť dodané ako pole alebo ako zoznam, príp. aj ako string, kde sú jednotlivé záznamy oddelené čiarkou. Príklad záznamu: „Andrej Andrejev/OP123456, Bohuš Bohuš/PA123456“ Táto žiadosť je spracovávaná chaincode a zmení stav rezervácie na ledgeri na „Checked-In.“ Kontroluje, či sú žiadané sedadlá voľné a či sedia customerNames a reservationNr s rezerváciou. Vráti správu o úspechu volania a pošle email s „letenkami“ na emailovú adresu customerEmail z rezervácie.

2. Voľba implementačného prostredia

Projekt je implementovaný v prostredí HyperLedger Fabric (verzia 2.2.), so synchronizáciou na localhoste, ktorý je spúšťaný pomocou softvéru Docker (verzia 20.10.14.) a nástrojom Docker Compose (verzia 1.29.2.) v operačnom systéme MacOS.

Kód bol písaný v editore Visual Studio Code, v jazyku JavaScript. Projekt je spúšťaný a beží pomocou CLI.

Boli využité tiež node 16.5.0. a npm 8.5.5.

3. Návrh architektúry a jej prepojenia

Sieť sa skladá z:

* Org1 – letecká spoločnosť EconFly
* Org2 – letecká spoločnosť BusiFly
* Org3 – cestovná kancelária
* myChannel1 – prepojenie organizácií

Ide o tri samostatné organizácie, a preto má byť každá samostatne reprezentovaná. Táto reprezentácie je tiež optimálna pre dodatočnú možnosť rozšírenia aplikácie a vyššiu bezpečnosť a decentralizáciu.

Počet kanálov je zvolený 1, preto že to bolo postačujúce pre našu jednoduchú implementáciu.

4. Implementované časti kódu

Projekt bol vytvorený upravením fabric-samples a asset-transfer-basic, a obsahuje tieto významnejšie časti kódu:

4.1. Asset Transfer

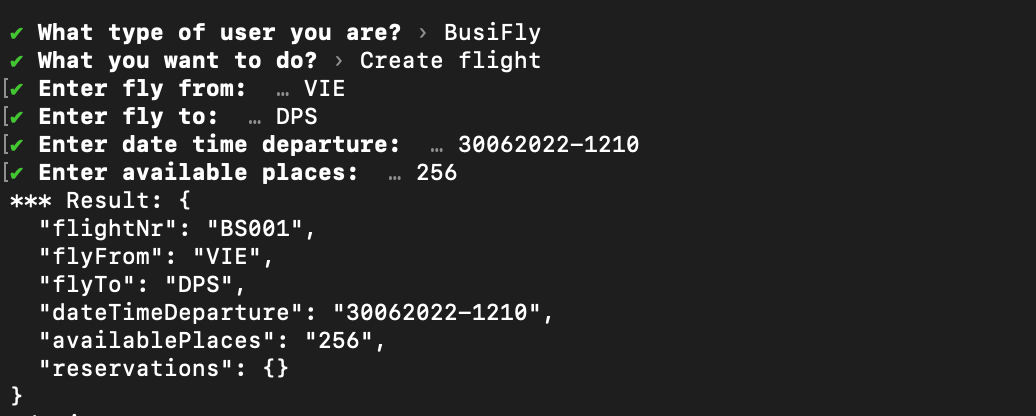
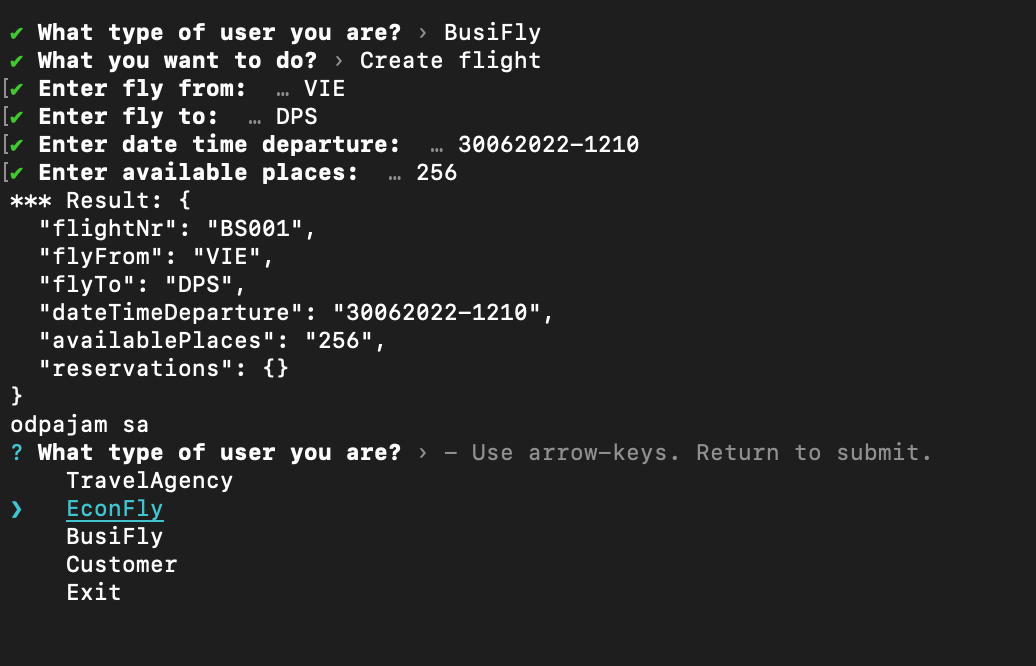
Teda chaincode, ktorý sa stará o manipuláciu údajov na blockchaine. Využíva na to nasledovné funkcie:

* ***createAsset*** *–* kontroluje, či funkciu volá organizácia. Potom získa organizáciu z kontextu (ctx), vygeneruje unikátne číslo letu, vytvorí názov letu a vloží deň všetky vstupné informácie. Napokon ho vloží medzi ostatné dáta.
* ***readAsset*** *–* funkcia vracia asset s daným flightNr, ktorý vyberie z chaincode state (ak existuje).
* ***updateAsset*** *–* aktualizuje už existujúci let za taký, ktorý má upravené parametre. Najskôr teda získa pôvodný let, a ten potom nahradí za nový a vloží ho medzi dáta.
* ***getAllAssets*** *–* vracia všety assety/lety z world state.
* ***bookSeats*** *–* umožňuje aerolinkám dokončiť rezerváciu cestovnej kancelárie. Najskôr kontroluje to, či je volaná aerolinkou. Potom kontroluje to, či let vôbec existuje, získa daný let, počet volných miest v danom lete, a potom iteruje polo všetkých rezervácií. Pri tých, ktoré majú status pending kontroluje to, či má dostatočný počet volných miest v lietadle na rezervovanie rezervácie, a ak áno zmení stav rezervácie na potvrdenú a odpočíta počet volných miest v lietadle. Na konci sa let aktualizuje s novými hodnotami.
* ***reserveSeats*** *–* umožňuje cestovnej agentúre rezervovať určitý počet sedadiel v lete. Najskôr kontroluje, či je volaná cestovnou agentúrou. Potom kontroluje, či daný let existuje. Potom vytvorí unikátne číslo rezervácie, vytvorí rezerváciu v stave pending, a aktualizuje let.

4.2. Application

Poskytuje celé rozhranie. Slúži ako klient pre používanie funkcionalít v chaincode. Poskytuje CLI rozhranie pre konečného usera, ktoré vyzerá nasledovne:





5. Opis prostredia na testovanie a opis vybraných testov

V rámci testovania boli vykonané nasledovné testy:

*doplniť*

6. Užívateľská príručka

Po spustení programu sa zobrazí CLI. Výber možností je na účely obmedzenia chýb umožnený metódou prompt(). Pohyb medzi jednotlivými možnosťami sa vykonáva pomocou arrow-keys a enter na klávesnici. CLI vyzerá nasledovne:

Ako prvé je potrebné vybrať si užívateľskú rolu (EconFly, BusiFly alebo cestovná kancelária). Ďalej sú užívateľovi prístupné funkcie podľa jeho roly:

6.1. CLI pre cestovnú kanceláriu

Cestovná kancelária môže vykonávať funkciu reserveSeats, ktorá slúži na rezervovanie určitého počtu sedadiel v konkrétnom lete zadanom pomocou flightNr.

Argumenty pre funkciu sa zadávajú v poradí: počet sedadiel, číslo letu.

6.2. CLI pre EconFly a BusiFly

Letecké spoločnosti môžu vykonávať tieto funkcie:

* createAsset
* getAllAssets
* readAsset
* updateAsset

Vstupy všetkých funkcií sú opísané v konzole.

6.3. Spúšťanie aplikácie

Ako prvé je potrebné spustiť sieť so vš. organizáciami a peermi, následne otvoriť channel a deploynuť chaincode. Vykonáva sa to týmito príkazmi:

cd

cd Desktop/DMBLOCK-zadanie-3/fabric-samples/test-network

./network.sh down

./network.sh up createChannel -c mychannel -ca

cd addOrg3

./addOrg3.sh up -c mychannel -ca

cd ..

./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-javascript/ -ccl javascript

7. Odpovede na otázky

Napíšte krátke vysvetlenie a vaše názory:

1. **Podľa vášho názoru, je takéto blockchain-based riešenie najlepšia možnosť na vyriešenie daných výziev?** 
   * Odpoveď: Je to dobré riešenie ale nie je to nutné – záleží na tom ako veľmi nedôveryhodný vzťah je medzi organizáciami, ale dalo by sa to dostatočne dobre vyriešiť centralizovanou databázou a klasickým informačným systémom.
2. **Aké sú výhody a nevýhody používania technológie distribuovanej účtovnej knihy (distributed ledger technology) v porovnaní s centralizovaným systémom na tomto konkrétnom prípade použitia?**

* Odpoveď: Nevýhodou by mohla byť prípadná latencia, zložitosť celého systému, problém tvorby zmien (pre zmenu/update v systéme musí nastať konsenzus), drahý vývoj, a výhodou je to, že sa organizácie môžu spoľahnúť na pravdivosť údajov.

8. Záver

V závere sme dospeli k tomu, že pre vývoj aplikácií a programovanie je niekedy (hraničiace so vždy) potrebné používať funkčný a dobrý operačný systém typu macOS a preto sme si jeden zaobstarali.

Ďalej sme zistili, že neexistuje funkčný framework na vývoj a využívanie privátnych enterprise blockchainov, a že Hyperledger Fabric nie je framework s ktorým plánujeme do budúcna pracovať.