

Zadanie 3 – Hyperledger Fabric smart systém

Zadanie 3 je povinné a pracuje sa na ňom v dvojiciach! Tým pádom aj Late days sú jednotné v rámci dvojice (môže byť na výnimku trojica, ktorá má naviac spraviť aj GUI ako povinnú časť), berie sa nižšia hodnota od študenta, ktorý má menej voľných late days, teda výsledok $\min(\text{študent1}, \text{študent2}, \dots)$. [Prosím nahláste tímy do Teamsu, Channel Cvičenia a hore Files ak dochádza ku zmene oproti zadaniu 2.](#)

V rámci tohto projektu budete pracovať na biznis sieti medzi aerolíniami, cestovnou kanceláriou a zákazníkmi pomocou Hyperledger Fabric.

1. Hyperledger Fabric

Fabric je aktuálne najstabilnejší a jediný framework pripravený na produkciu biznis sietí v Hyperledger ekosystéme. Jeho hlavným cieľom je vytváranie privátnych blockchain sietí (permissioned), ktoré umožňujú známym identitám navzájom spolupracovať pomocou „**Membership Service Provider**“ (MSP). Taktiež poskytuje možnosť vytvárať kanály (channels), ktoré dovoľia iba špecifickej skupine účastníkov založiť separátnu účtovnú knihu (ledger) transakcií a zdieľať dáta. Vo Fabricu sa smart kontrakty volajú **chaincode**, môžu byť viazané iba na kanál a spúšťané na peeroch s cieľom vytvárať transakcie. Chaincode umožňuje aplikáciám interagovať s účtovnou knihou a aktualizovať stavy aktív (**assets**).

2. Scenár

EconFly a *BusiFly* sú dve aerolínie, ktoré raketovo rástli v posledných pár rokoch. Väčšina ich profitu prichádzala od rôznych cestovných agentúr a kancelárií, ako je napríklad *GladlyAbroad*. Posledný rok, ale nanešťastie, zaregistrovali náhly prepád v ich firemných záujmoch kvôli nespokojným zákazníkom a biznis partnerom. Niektoré výzvy, ktorým čelia sú stratené rezervácie letov, neplatné dáta pri letoch, náhodné zmeny rezervácií zákazníkom bez ich vedomia. Toto všetko je ovplyvnené používaním zastaraného komunikačného systému.

Ich finančné straty sú tak veľké, že obe aerolínie siahli po kompletnej výmene vedenia. Na pozície CTO (chief technology officer) sa dostali priaznivci moderných technológií a rozhodli sa vytvoriť biznis sieť s ambíciou poskytovať viac efektívnu B2B (biznis-to-biznis) spoluprácu a lepšie zákaznícky zážitok. S celým tým ošialom okolo technológie blockchain, ich jasný pohľad smeruje na využitie Hyperledger Fabric s cieľom poskytnúť vždy aktuálne informácie všetkým svojim zákazníkom v akomkoľvek čase.

Vy, ako jediní známi blockchain experti na celom svete, ste boli najatí *EconFly*, *BusiFly* a *GladlyAbroad*, samozrejme s veľmi atraktívnym platom, aby ste im pomohli implementovať a konfigurovať biznis sieť nazvanú *FlyNet*.

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 aerolína objedná požadované sedadlá aktualizovaním informácií v blockchaine.

Podme sa bližšie pozrieť na štruktúru letu a rezervácie, ale tiež na transakcie, ktoré ich ovplyvňujú v sieti.

2.1. Let

Let môže byť vytvorený **jedine** aerolínou a s danými parametrami. Taktiež aktualizácia letu môže byť vykonaná **jedine** aerolínou, ktorá tento let poskytuje. Kľudne si upravte dané atribúty podľa vašich potrieb.

- `flightNr => ID (string);`
- `flyFrom => odkiaľ let začína (string);`
- `flyTo => cieľ letu (string);`
- `dateTime => deň a čas letu (string);`
- `availablePlaces => počet voľných sedadiel v lete (číslo).`

Na zistenie, ktoré lety jednotlivé aerolíne poskytujú, *flightNr* je zložené z unikátneho identifikátora aerolíne, kombinovaného s unikátnym náhodným číslom letu. Identifikátor pre EconFly je **EC** a pre BusiFly je **BS**.

Príklady:

Let vytvorený EconFly: **ID=EC001**, **value={BUD, TXL, 05032021-1034, 100}**

Let vytvorený BusiFly: **ID=BS015**, **value={MUC, LIS, 10042021-2157, 150}**

2.2. Rezervácia

Rezervácie na sedadlá letu môžu byť vytvorené **jedine** cestovnou kanceláriou. Rezervácia má dané atribúty. Kľudne si ich upravte podľa vašich potrieb.

- `reservationNr => ID (string);`
- `customerNames/[] => mená zákazníkov, ktorý si chcú rezervovať sedadlá (string/pole);`
- `customerEmail => emailová adresa zákazníka, na ktorú mu prídu letenky (string);`

- `flightNr => let, v ktorom sa rezervuje (string);`
- `nrOfSeats => koľko sedadiel sa ide rezervovať (číslo);`
- `status => stav rezervácie (string).`

Status rezervácie je na začiatku nastavený na „Pending.“ Akonáhle je rezervácia spracovaná aerolíniou, status sa zmení na „Completed.“

Príklad:

Žiadosť o rezerváciu jedného sedadla na let BS015: **ID**=R08, **value**={bob, BS015, 1, „Pending“}

2.3. FlyNet chaincode

Aplikácie môžu interagovať so sieťou a aktualizovať stav pomocou chaincode funkcií. Pozrime sa na príklady transakcií vo FlyNet chaincode.

- `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 vrátiť error. Parameter `flightNr` nevstupuje ako argument funkcie, musí byť vygenerovaný automaticky.
- `getAllFlights()` (môžete spraviť aj `getAllReservations()`)
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 **bud'** cestovnou kanceláriou **alebo** konečným zákazníkom za účelom vybrať si konkrétne 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. Čo to sú „letenky“ a ako budú vyzerat' je úplne na vás.

3. Poznámky

- Navrhните si vlastnú sieť a architektúru, bude to súčasťou hodnotenia ako ste sa nad tým zamysleli a koľko ste si určili *orderer organizations*, *peers*, *organizations*. Zároveň túto informáciu zdôvodnite v dokumentácii.
- Každá *peer organization* by mala mať minimálne jedného jedného používateľa.
- Certifikáty môžete generovať buď pomocou cryptogen alebo deploynutím certifikačnej authority (CA).
- Použite také množstvo channelov, aké uznáte za vhodné, ale zdôvodnite to v dokumentácii.
- Chaincode môžete písať v akomkoľvek [podporovanom jazyku](#).
- Klientské aplikácie môžete implementovať v akomkoľvek [podporovanom SDK](#). Existuje aj [SDK pre Python](#), ale je podporované iba na staršej verzii Fabricu 1.4, čiže treba si rozbehať starší Fabric. Ak chcete môžete, ale je to na vašej zodpovednosti a berte to ako projekt v inkubátore.
- Grafické rozhranie **nie je** nutné, ale je súčasťou bonusu. Spúšťanie a beh programu pomocou CLI sú možné tiež.
- Synchronizovať Fabric stačí na localhoste, ale synchronizácia online je bonus.
- Rozbehaný Explorer vám môže pomôcť pri debugovaní a ukazovaní transakcií.

4. Ako začať

- Prepokladám, že máte nastavený environment docker, Fabric.
- Ak by ste mali problémy s Fabric verziou, môžete downgradenúť na akúkoľvek 2.X (experimentálne aj 1.X) verziu, ktorá vám bude viac sedieť, napr.

```
curl -sSL https://bit.ly/2ysb0FE | bash -s 2.2.0
```

- Máte rozbehanú test-network, resp. nejakú podobnú Fabric sieť a vyskúšali ste si createChannel, deployCC, ideálne aj asset-transfer-basic.

5. Referencie

- Hyperledger ecosystem: <https://www.hyperledger.org/projects>
- Hyperledger Fabric documentation: <https://hyperledger-fabric.readthedocs.io/en/release-2.2/>

Odovzdanie

Dokumentáciu a zdrojové kódy (conf files, chaincode, client application, frontend) implementácie odovzdá **jeden z dvojice** študentov v elektronickom tvare do AISu v určenom termíne vo formáte *.zip alebo *.tar. **Odovzdanie aspoň nejakej časti zadania je povinné!** Aj na toto zadanie je možné využiť Late days, pozor iba na algoritmus min(študent1, študent2). Všetky kódy prejdú kontrolou originality, v prípade vysokej percentuálnej zhody bude študentom začaté disciplinárne konanie.

Súčasťou odovzdania **musia byť aj vami napísané unit testy**, ktoré sú základným stavebným kameňom každého produkčného biznis blockchainu. Podmienkou je aj aktívne odprezentovanie programu študentami podľa individuálnej dohody po odovzdaní.

Súčasťou riešenia je aj dokumentácia, ktorá musí obsahovať najmä:

- a) formálne náležitosti ako sú titulná strana, číslovanie strán;
- b) navrhnutú architektúru a jej prepojenia;
- c) popísané implementované časti kódu, hlavne ak boli vami zmenené alebo vylepšené niektoré parametre popisované v tomto dokumente;
- d) voľbu implementačného prostredia – teda chaincode a aj klient app, príp. frontend;
- e) opis prostredia na testovanie a opis vybraných testov;
- f) iné záležitosti, ktorými sa študenti chcú pochváliť;
- g) záver a v ňom zhodnotenie vlastného príspevku a **reálne** percentuálne porovnanie podielu študentov na jednotlivých častiach a čo sme sa naučili;
- h) Prílohu – inštrukcie ako spustiť projekt (teda používateľská príručka).
- i) **Odpovede na otázky:** Podľa vášho názoru, je takéto blockchain-based riešenie najlepšia možnosť na vyriešenie daných výziev? Aké sú výhody a nevýhody používania technológie distribuovanej účtovnej knihy (distributed ledger technology) v porovnaní

s centralizovaným systémom na tomto konkrétnom prípade použitia? Napíšte krátke vysvetlenie a vaše názory.

Hodnotenie (bez minima)

Celé riešenie - max. 15 bodov, z toho:

- max. 3 body za navrhnutie architektúry (vrátane popísania v dokumentácii) – bude sa prihliadať na optimálne využitie jednotlivých prvkov Fabric siete a na optimálnu prácu s blockchainom;
- max. 7 bodov za celý projekt, jeho implementáciu, popasovanie sa s výzvami spoločností – ako úspešne sa vám darí interagovať s chaincodom a klientskou aplikáciou;
- max. 2 body za vytvorené testy na projekt;
- max. 3 body za výslednú dokumentáciu – 1 bod formálna stránka, 1 bod odpovede na otázky, 1 bod zvyšné časti.

Bonusy - dajú sa získať navyše k bodom za zadanie

- (V PRÍPADE TROJICE POVINNÉ) max. 4 body za grafické rozhranie, ktoré bude uľahčovať prácu s biznis sieťou – teda možnosť prihlásiť sa jednotlivým účastníkom, spravovanie ich pridelených funkcionalít, grafické zobrazenie lietadla aj s jednotlivými sedadlami, ...
- max. 4 body za upravenie localhost aplikácie na synchronizáciu v Cloude a/alebo medzi vašimi dvomi/tromi PC tak, že budeme vidieť update na jednej strane automaticky prešírený na ďalšie PC.
- max. 1 bod za pridanie nejakej zaujímavej feature (po konzultácii s cvičiacim);
- max. 1 bod za využitie nástroja Calliper na performance testy z pohľadu priepustnosti – teda test škálovateľnosti.