UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Paralelni i distribuirani algoritmi i strukture podataka

ms Nebojša Horvat

Zimski semestar 2019/2020.

Studijski program: Računarstvo i automatika

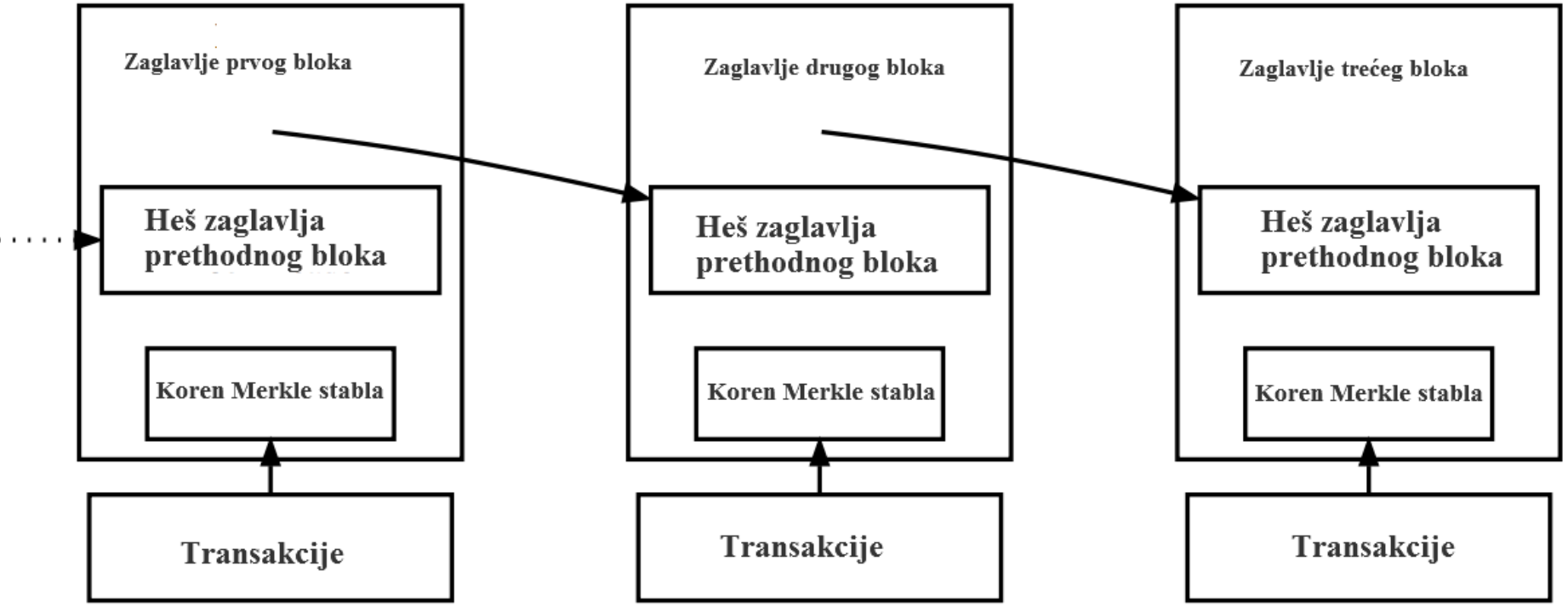
Modul: Računarstvo visokih performansi

Blockchain

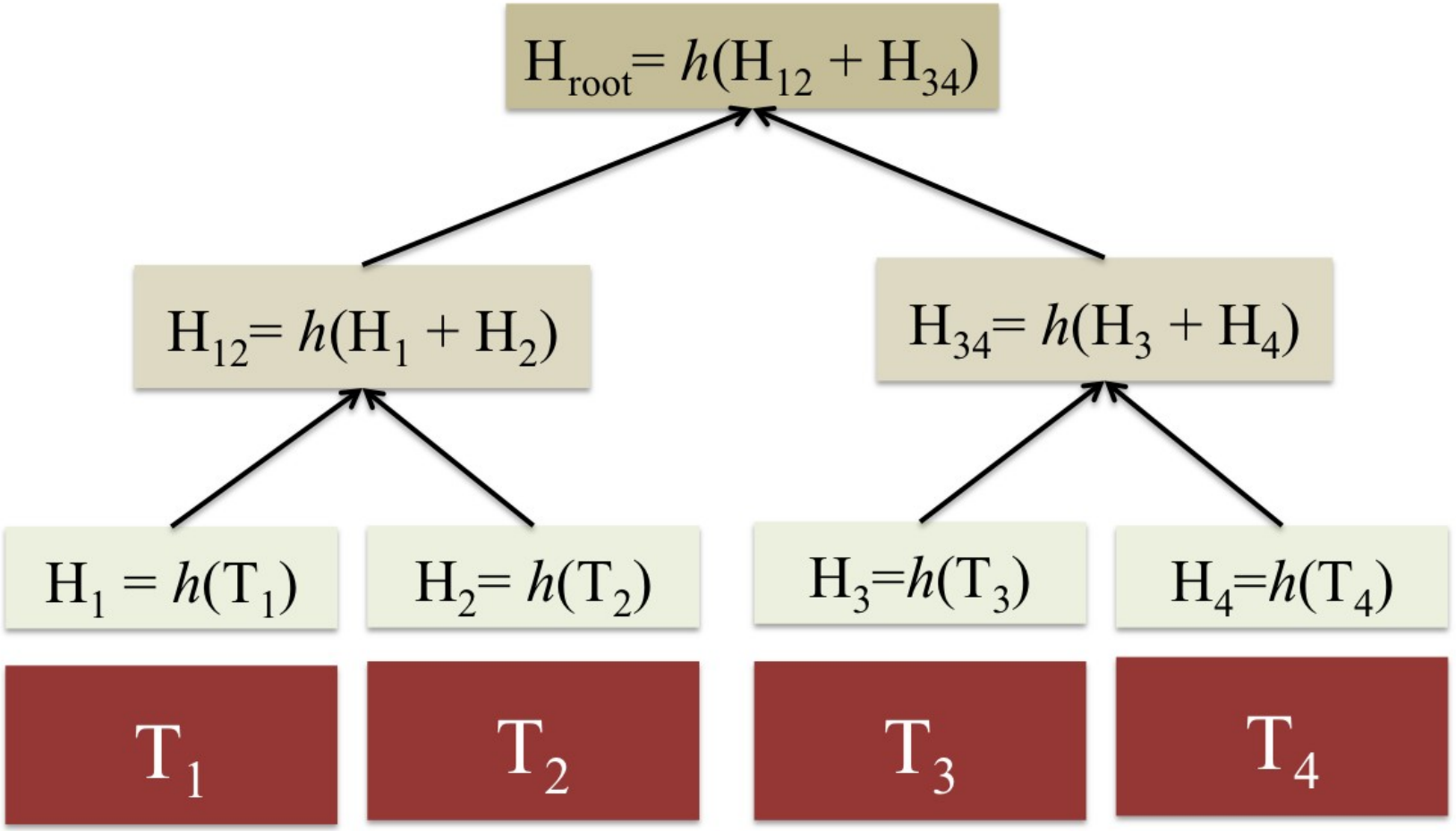
Šta je blockchain ?

- Distribuirani i deljeni dnevnik transakcija
- Upis podataka moguć postizanjem konsenzusa
- Ne može se izbristati transakcija
- Pametni ugovori
  - Bitcoin
  - Ethereum
  - Hyperledger Fabric
  - Ripple
  - ...

Šta je blockchain ?



Šta je blockchain – Merkleovo stablo ?

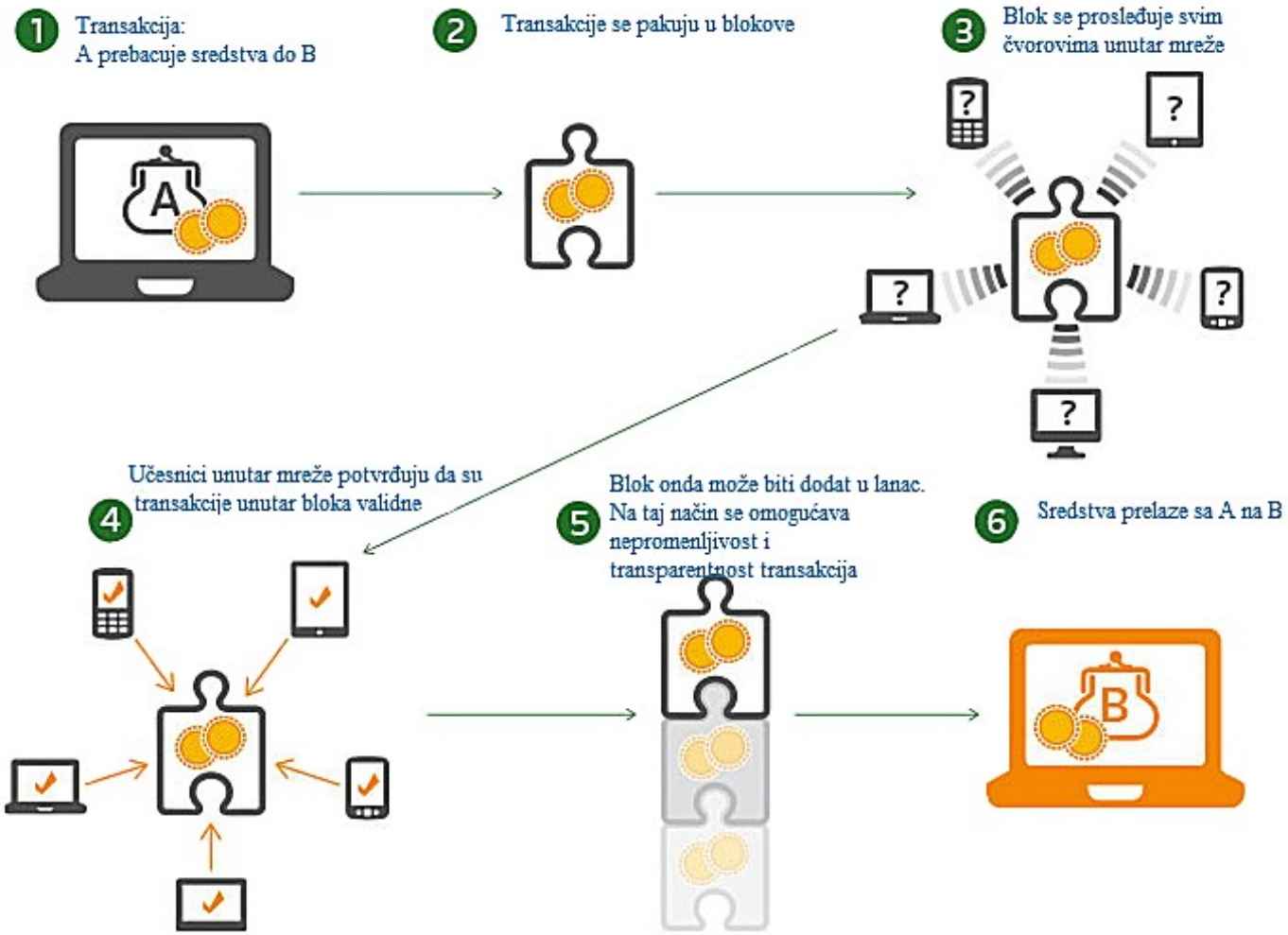


Javne i privatne Blockchain mreže

- Javne
  - Učesnici najčešće anonimni
  - Veliki broj učesnika
  - Slobodan upis/čitanje
  - Konsnzus se postiže najčešće pomoću računarske moći
  - Potrebno davati neku vrednost kako bi se dešavale transakcije
- Privatne
  - Brze
  - Manji broj učesnika
  - Učesnici poznati
  - Konsenzus se može postići na razne načine

- Javna mreža Bitcoin
- Privatna mreža Hyperledger Fabric

Tok transakcije



Hyperledger Fabric

Hyperledger

- Hyperledger je kolaborativni projekat otvorenog koda stvoren sa ciljem da se blockchain tehnologija unapredi i primeni u različitim sektorima industrije.

Hyperledger Fabric - Uvod

- Platforma za razvoj distribuiranih rešenja koja se oslanja na blockchain tehnologiju.
- Modularna arhitekturu
  - dizajnirana tako da podrži različite implementacije komponenata
  - kako bi se prilagodila raznim potrebama poslovanja.
- Visok nivo skalabilnosti,
- Obezbeđuje sigurnost odnosno privatnost transakcija koja je neophodna u poslovnom okruženju.

Hyperledger Fabric – osnovni pojmovi

- Članovi
  - Peer (commiter, endorser)
  - Orderer – Ordering service
- Svojina (Asset) – materijalna ili nematerijalna
- Identiteti (MSP)
- Kanali – Privatne podmreže sa sopstvenim dnevnikom transakcija (Ledger)
- Konsenzus – jednostavno izmenljiv

Hyperledger Fabric – osnovni pojmovi

- Pametni ugovori= Chaincode
- Smeštanja podataka- > Ledger i World State
  - World state - > CouchDB, LevelDB
- Tok transakcije
- Smeštanja blokova
  - Vremenski
  - Na osnovu broja transakcija

Podešavanje okruženja Fabric mreže

- Pre podešavanja okruženja neophodnog za fabric mrežu potrebno je da imate instalirano:
  - Git
  - cURL
  - Docker (docker deamon bi trebalo da radi)
  - docker-compose

Podešavanje okruženja Fabric mreže

Elementi Fabric mreže (Orderer, Peer, CA...) se podižu kao posebni kontejneri

Svi primeri sa docker slikama se mogu instalirati kroz pokretanje skripte koja se može pronaći na boldovanoj adresi:

curl -sSLO **https://raw.githubusercontent.com/hyperledger/fabric/main/scripts/install-fabric.sh** && chmod +x install-fabric.sh

./install-fabric.sh --fabric-version 2.2.6

- Skipta će:
  - Klonirati hyperledger/fabric-samples repozitorijum
  - Instalirati izvršne i configuracione fajlove potrebne za pokretanje mreže na vašoj platformi
  - Instalirati sve potrebne docker slike

Podizanje Fabric mreže

- Nakon podešavanja okruženja fabric mreže potrebno je ući u **fabric-samples/test-network** folder
- Mreža se podiže kroz **./network.sh up** komandu
  - Komanda podiže mrežu koja se sastoji od dva peer-a i jednog orderer-a.
  - Kada ukucamo **docker ps** videćemo da su se pokrenula tri docker kontejnera

Podizanje Fabric mreže

- Komandom **./network.sh createChannel** pokrećemo:
  - Kreiranje kanal
  - Pridruživanje peer-ova kanalu
  - Podešavanje anchor peer-ova za svaku organizaciju

Na kraju, komanda **./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl go**

instalira i instancira chainCode (kod pametnih ugovora) na pokrenutoj mreži

Rad sa Fabric mreže

- Pre rada sa mrežom potebno je postaviti prvo sistemske variable:
  - export PATH=${PWD}/../bin:$PATH
  - export FABRIC_CFG_PATH=$PWD/../config/
  - export CORE_PEER_TLS_ENABLED=true
  - export CORE_PEER_LOCALMSPID="Org1MSP"
  - export CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt
  - export CORE_PEER_MSPCONFIGPATH=${PWD}/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
  - export CORE_PEER_ADDRESS=localhost:7051

Rad sa Fabric mreže

- Nakon toga moguće je incjalizovati podatke na ledgeru:
  - peer chaincode invoke -o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n basic --peerAddresses localhost:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses localhost:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"InitLedger","Args":[]}'
- Upit stanje koje se nalazi na ledgeru možemo da proverimo kroz komandu:
  - peer chaincode query -C mychannel -n basic -c '{"Args":["GetAllAssets"]}'

Zaustavljanje Fabric mreže

- Komanda **./network.sh down** će:
  - Zaustaviti i obrisati docker kontejnere čvorova i chaincode-a
  - Obrisati criptomaterijal organizacije
  - Briše channel artifacts i docker volumes (tako da je opet moguće pokrenuti mrežu kroz **./network.sh up** komandu)

Konfiguracioni fajlovi

- Podešavanje kripto materijala
  - Crypto-config-orderer.yaml
    - Specifikacija Orderer organizacija
    - Ordering service u produkciji, nikako solo
      - solo samo za razvoj
      - za produkciju – Kafka ordering, raft ...
  - Crypto-config-org1.yaml
    - Specifikacija organizacije 1
  - Crypto-config-org2.yaml
    - Specifikacija organizacije 2
- Sertifikati definišu organizacije
    - Daju identitete entitetima unutar Fabric mreže
- Organizacije logički grupišu peer-ove

Konfiguracioni fajlovi

- Sertifikat će se izgraditi kao hostname.domain
  - Odnosno orderer.example.com
  - Dodavenje novog uređivača = dodavanje novog Hostname-a unutar Specs sekcije
  - Dodavanje nove uređivačke organizacije obuhvata dodavanje cele sekcije (name, domain, specs)
- CommonName unutar specs sekcije može da “nadjača” pomenutu konvenciju imenovanja

Konfiguracioni fajlovi

- Definicija članova peer organizacija
  - Template
    - Count : 2 – kreiranje 2 peer-a za datu organizaciju
  - Imenovanje prati konvenciju
    - peer{index}.domain
      - Odnosno
        - » peer0.org1.example.com, peer1.org1.example.com itd.
  - Users
    - Count : 2
      - Dodatni sertifikati pored administratorskog
        - » Ako se broj korisnika zna unapred

Konfiguracioni fajlovi

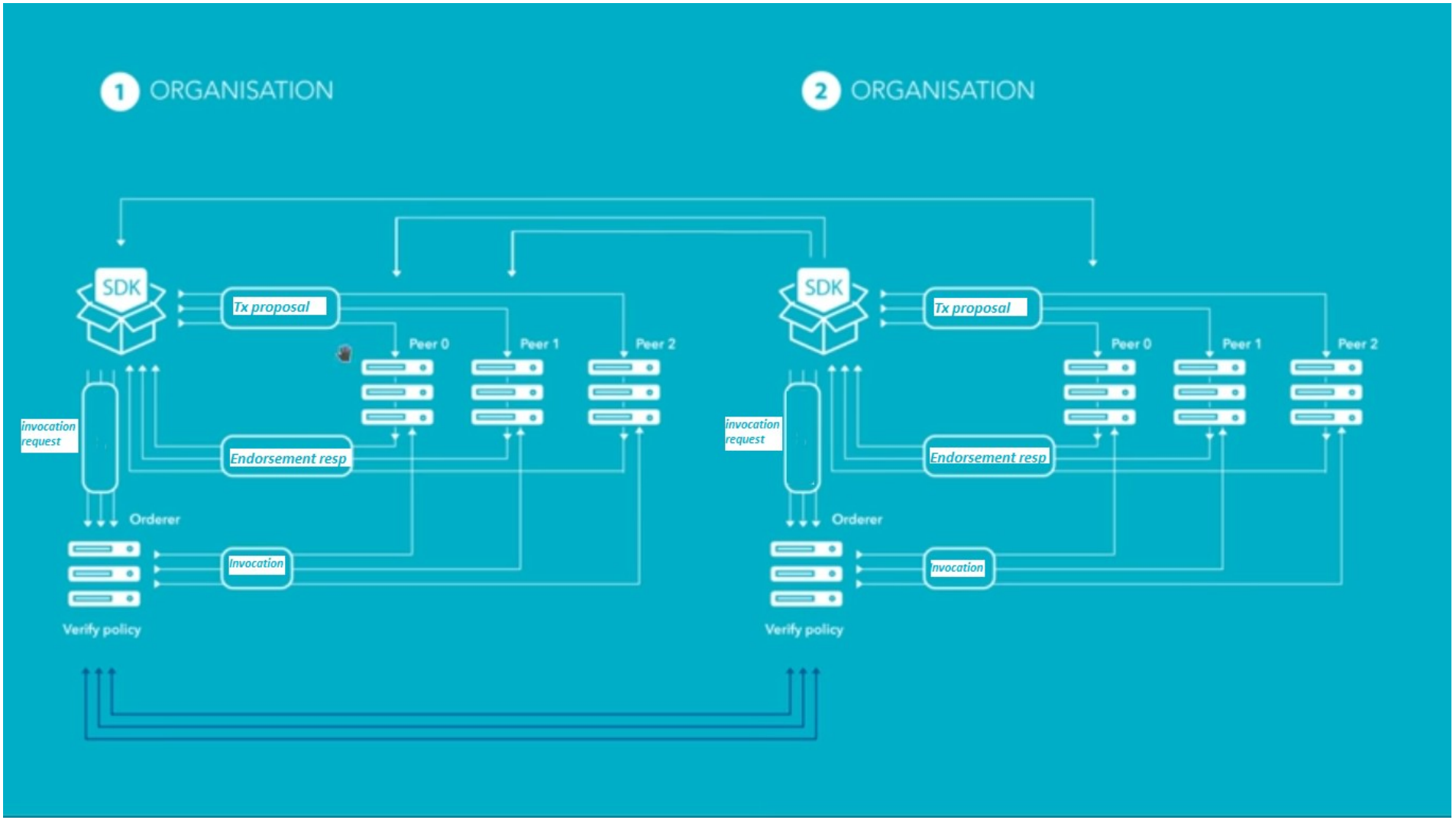
- Cryptogen alat generiše kriptografski materijal prema definiciji u crypto-config-<name>.yaml
- ./cryptogen generate
  - Kreira foldere:
    - ordererOrganizations
      - Example.com
        - » ca
        - » msp
        - » orderers
        - » ltd
    - peerOrganizations
      - org1.example.com
        - » ...
      - org2.example.com
      - ...

Sertifikat – PEM fajl

Certificate Information:

- **Common Name:** Admin@org1.example.com
- **Organization Unit:** client
- **Locality:** San Francisco
- **State:** California
- **Country:** US
- **Valid From:** December 13, 2018
- **Valid To:** December 10, 2028
- **Issuer:** ca.org1.example.com, org1.example.com
- **Serial Number:** 0ecf028ed6c055f8f337af152c6e770f

Tok transakcije



Tok transakcije

- Klijent inicira transakciju
  - SDK kreira predlog transakcije (Transaction proposal)
- Članovi prihvatioci (Endorsers) verifikuju transakciju
  - Izvršavaju transakciju nad podacima u ledger-u
  - Read & Write skupove podataka šalju nazad do SDK-a, zajedno sa potpisima – Proposal response

Tok transakcije

- SDK zatim proverava da li dobijeni odgovor ispravan
  - Da li sadrži potpise tako da zadovolji pravila prihvatanja transakcije (Endorsement policy)
  - Provere ovog tipa se vrše i na peer-ovima
- SDK se zatim obraća uređivačkom servisu (Ordering service) sa transakcijom koja sadrži sve prethodno pomenute podatke i ID kanala na kom se transakcija izvršava

Tok transakcije

- Uređivački servis poređa transakcije i sklopi ih u blokove
- Zatim se blokovi šalju do peer-ova koji validiraju blokove/transakcije
  - Transakcije se mogu označiti kao validne ili kao nevalidne
- Promene u validnim write skupovima se zatim izvrše nad trenutnim podacima na Ledger-u svakog peer-a na kanalu
  - Postoji VLedger(Validated ledger) u kom ne postoje nevalidne transakcije, sastavljen je od validnih blokova (vBlocks)
- Peer “ispaljuje” događaj kojim se klijent notificira o uspešnosti izvršenja transakcije