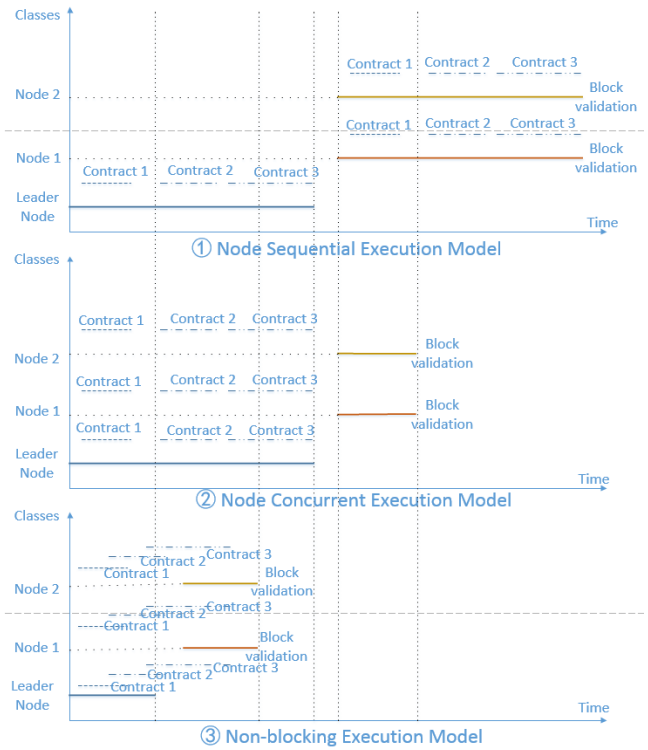
State synchronization in process-oriented chaincode

Lian Yu, Wei-Tek Tsai

Frontiers of Computer Science, DOI: 10.1007/s11704-017-6484-z

Problems & Ideas

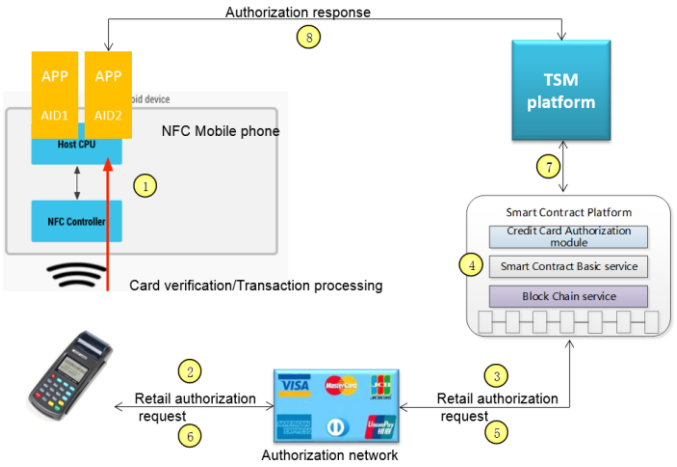
- Complex business processes have many participants, and need to address the issues not necessarily needed for cryptocurrency:
  - They need to control access permissions, such as *read* from or *write* into BCs while maintaining BCs properties. This needs to be true regardless if it is a public or private BC.
  - They have complex processes with multiple parties participating. Before an asset (money, stocks, bonds or others) can be placed into a chaincode (CC) for execution, its ownership needs to be cleared. Then a CC can execute the transaction, and store the results into BCs.
- CC needs to provide the following features:
  - Verify identities and certify transactions;
  - Provide an indelible record of transacting history;
  - Allow people to trade directly with each other without a trusted third party intermediary; and
  - Automated buy and sell transactions, whether B2B and B2C transactions.
- This approach has the following features:
  - There is a set of distributed authentication nodes called authnodes that communicate through a high-speed network instead of a P2P (peer-to-peer) network, and they synchronize with each other during the execution of a CC;
  - It has a voting mechanism to keep every node consistent with the states of business process regarding both the logical and temporal constraints;
  - It separates account BCs (ABCs) from transaction BCs (TBCs) to



Time of Block Building and Contract Executions in Three Models

Main Contributions

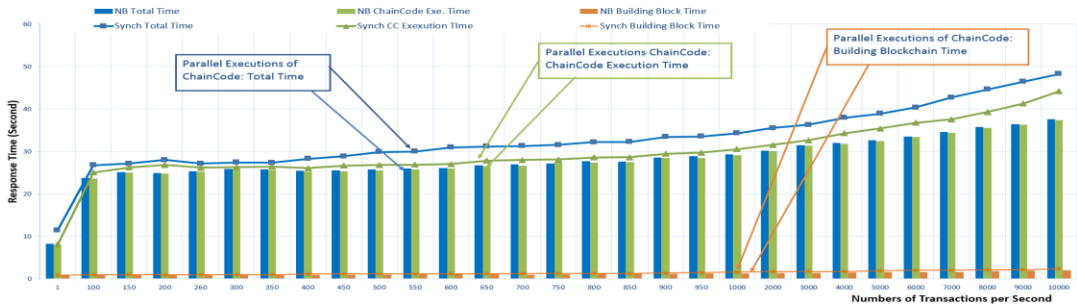
- As a common ledger, BCs store and record information that can be verified and audited by any nodes and users with the permissions, and this can effectively avoid the risk of the internal rats tampering with the information stored as shown in the following figure where the centralized system is replaced with CC platform on BCs.



HCE Payment Flow into Blockchains

The figure shows the comparison of response times between synch and asynch execution models. Horizontal axis represents the number of transactions per second sent to the system, and vertical axis represents the response time in seconds. Three lines with markers demonstrate the total response time, CC execution time, and building blockchain time, respectively, with synch execution model. Three vertical bars demonstrate the total response time, CC execution time, and building BC time, respectively, with asynch execution model.

It is observed that the time consumption of synch is larger than that of asynch. That is the CC execution time of synch is larger than that of asynch, the building block time of synch is larger than that of asynch, and the total time of synch is larger than that of asynch.



Response Time Comparison between Synch and Asynch Execution Models