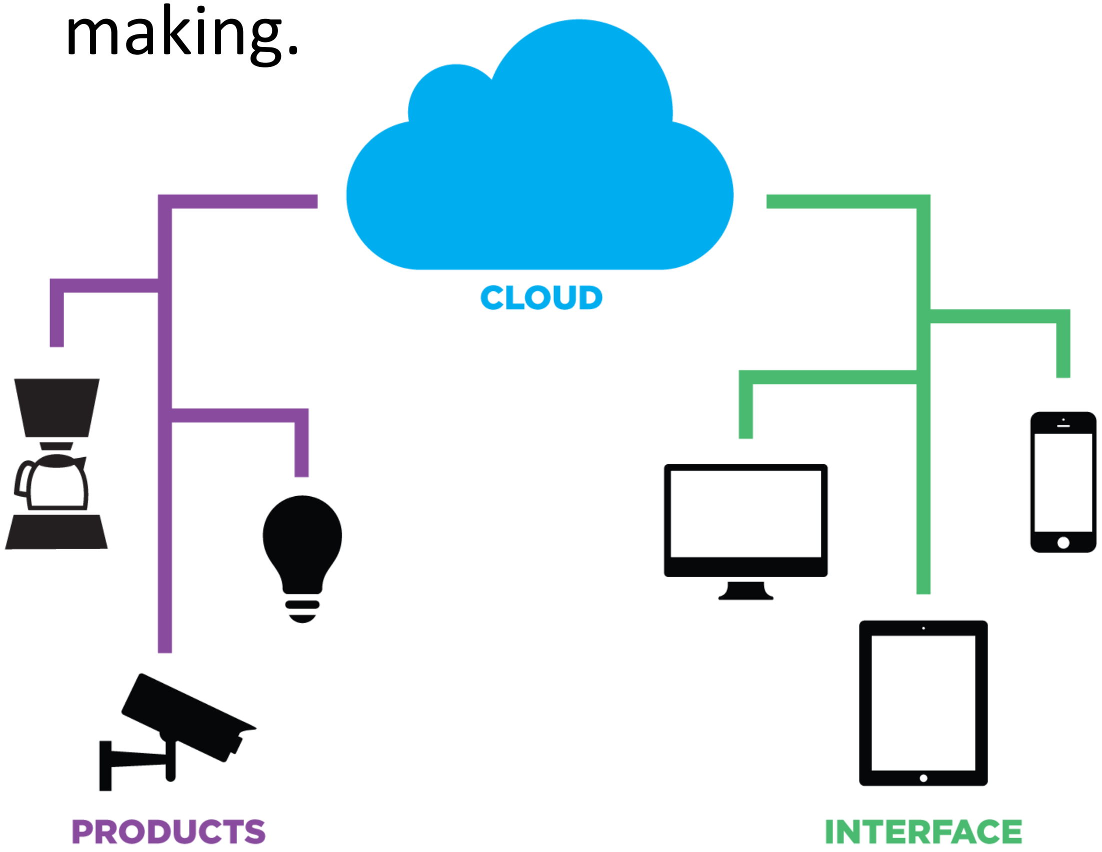
Secure Data Provenance in Cloud-centric Internet of Things via Blockchain Smart Contracts

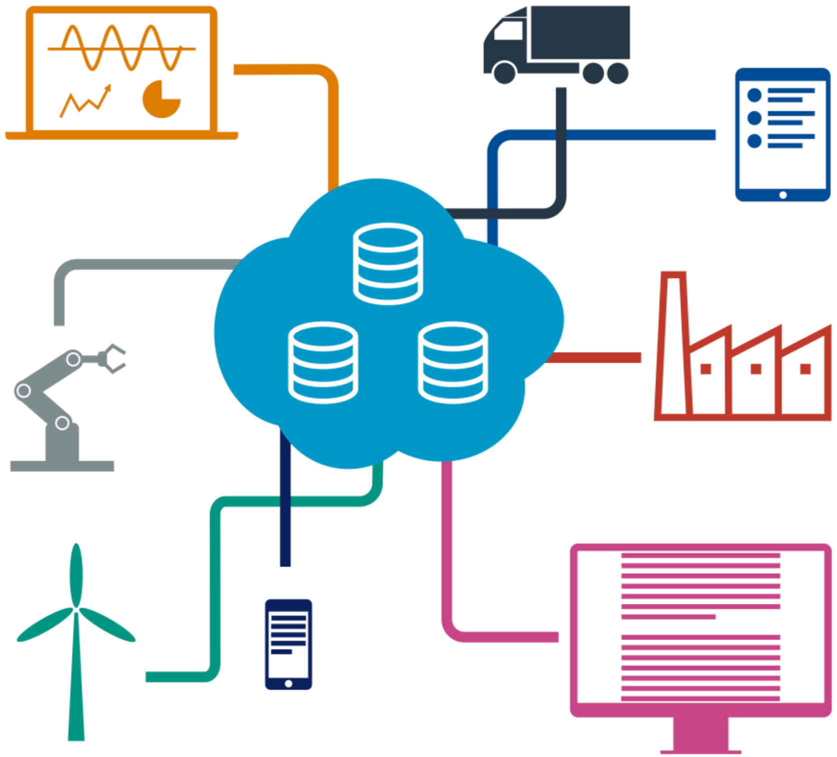
Saqib Ali (PhD)
Post-doc fellow

Supervisor:
Dr. Guojun Wang, Pearl River Scholarship Distinguished Professor
Director of Institute of Computer Networks,
Vice Dean of School of Computer Science and Technology,
Guangzhou University, Guangzhou.

Use Case: IoT Devices

IoT devices harvest the data through sensors and transmit it over a wireless network towards a cloud storage for analysis and decision making.





Problems in the Collected Data

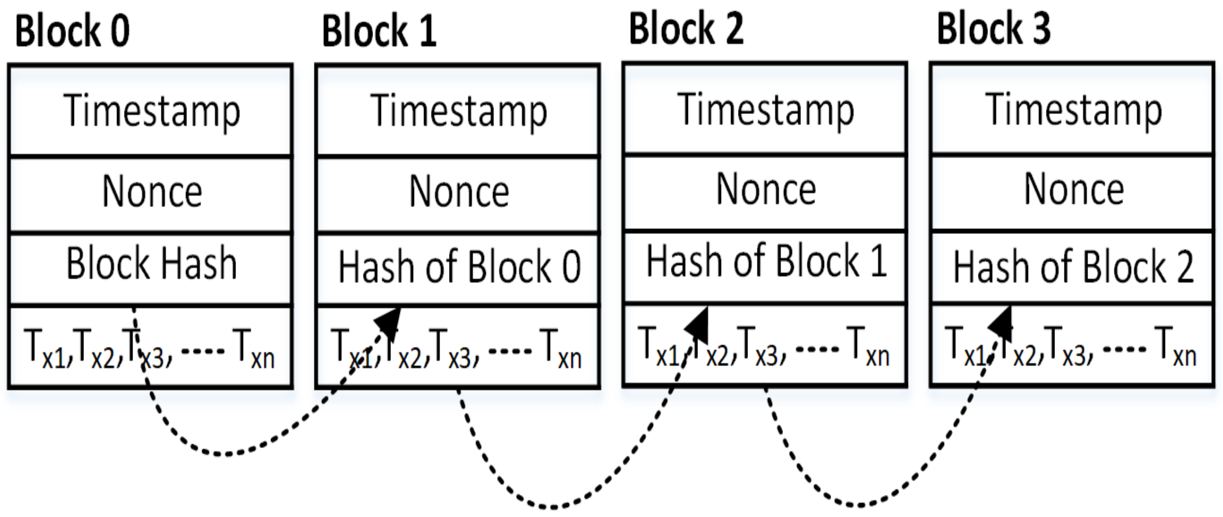
The collected data can be faulty due to:

- Environmental influence,
- Background noise,
- Faculty sensors,
- Dying battery,
- Device may generate biased or fake data due to security attacks (e.g., data modification, false information injection or fast sampling rate to drain the battery of the device).

Proposed Solution:

- Blockchain & Smart Contracts

i. Highly Trusted
ii. No need for the trusted 3rd party
iii. Protected
iv. Tampered proof
v. Data transparency
vi. Visibility
vii. Auditability



Distributed Ledger Technology (DLT)

A **distributed ledger** is a type of data structure which resides across multiple computer devices, generally spread across locations or regions.

Key Components

P2P network

Distributed computation

Cryptography

Consensus algorithm

Smart Contracts

Transactions

- The record of an event, cryptographically secured with a digital signature, that is verified, ordered, and bundled together into blocks, form the transactions in the blockchain.

- In the Bitcoin blockchain, transactions involve the transfer of bitcoins, while in other blockchains, transactions may involve the transfer of any asset or a record of some service being rendered.

- Furthermore, a smart contract within the blockchain may allow automatic execution of transactions upon meeting predefined criteria.

What are Smart Contracts?

i. Autonomous agents which execute automatically and independently on the top of the blockchain
ii. Visible to everyone & Fully trusted and scrutinised
iii. Can write or query the blockchain
iv. Invoked using transactions (ID + Data)
v. Logic in the smart contract is totally immutable
vi. All transactions to and from the smart contracts are fully traceable

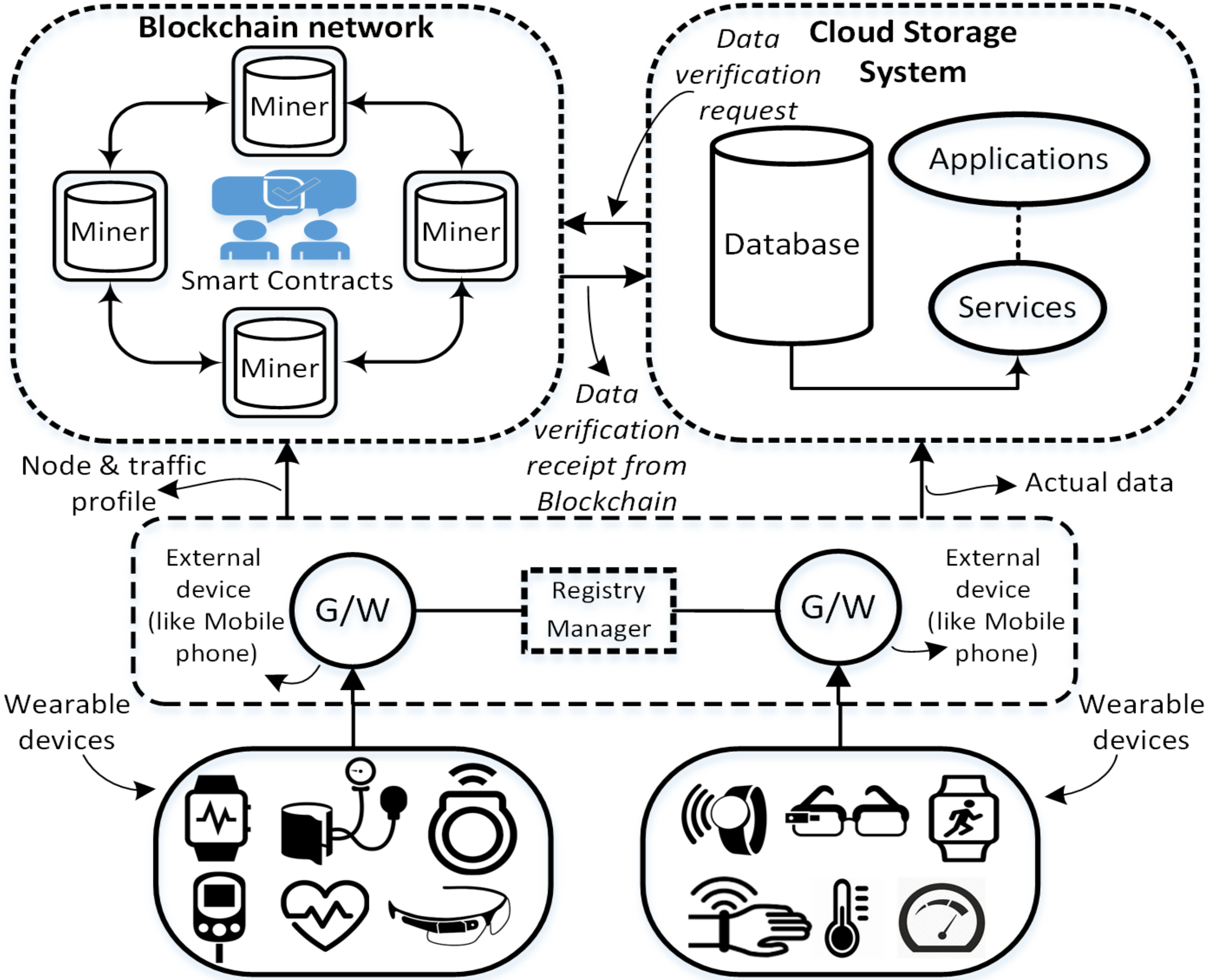
Components:

i. Unique address on the blockchain
ii. Private storage or State
iii. Associated code (where we can define our logic)

Caution

i. Once deployed, smart contracts cannot be reversed

Proposed Framework Using Blockchain & Smart Contracts



Contributions

1. We proposed a trustworthy **device registration and identity provenance smart contracts** via blockchain to ensure the device integrity and provenance in cloud-centric IoT network.

2. We stationed a comprehensive **data provenance smart contract** in the blockchain to guarantee the secure provenance towards the data stored in the cloud.

3. We deploy a **self-learning traffic profile provenance contract** in the network. The contract is used to certify the provenance to the device traffic profile while tracing the discrepancies in its traffic behavior.

4. Finally, we outline the security analysis & implementation details of the proposed framework using private blockchain i.e., Hyperledger Fabric

Types of Contracts used in the Proposed Architecture

i. **Device Registration Contract**
   - This global contract maps device identification strings to their blockchain address identity (equivalent to a public key).
   - Policies coded into the contract can regulate registering new identities or changing the mapping of existing ones.

ii. **Device Provenance Contract**
   - Using device signature & device metadata

iii. **Data Provenance Contract**
   - Using Data hash, timestamp & device Signatures

iv. **Traffic Profile Provenance Contract**
   - Using traffic profile

Algorithm 1 Device Registration Contract

Input: transaction tx

$tx \leftarrow$ tx_type, reg_type, device ID, publickey, gateway ID, & other device metadata

$tx\_type \leftarrow$ device_record

$reg\_type \leftarrow$ new **or** update

Output: tx_{id} & bc_{id} {successful transaction tx to the blockchain}

if (reg_type = new) **then**
  $deviceID \leftarrow bc_{id}$ {unique blockchain address (equivalent to a public key) is assigned by a Membership Service Provider}
  $tx \leftarrow set(tx \cup bc_{id})$ {create a new asset}
else
  if (reg_type = update) **then**
    $query(tx)$ {query the blockchain to get the device registration information}
    $tx \leftarrow set(updated\_tx)$ {update an existing asset}
  end if
end if
$PutState(tx)$ {transaction tx is placed in a Write set as a data-write proposal. However, it does not affect the ledger}
$WriteState(tx)$ {transaction tx is validated and successfully committed to the ledger}
return tx_{id}, bc_{id}

Algorithm 2 Device Provenance Contract

Input: transaction tx

$tx \leftarrow$ tx_type, publickey (device), & device_metadata

$tx\_type \leftarrow$ device_proven

Output: $True$ **or** $False$

```
tx' ← query(publickey and tx_type) {query the blockchain to get the device metadata}
if (tx' = null) then
   throw {could not find the device}
else
   if (device'.sig = device.sig) then
      if (tx'.metadata = tx.metadata then
         return TRUE {device identity is proven}
      else
         return FALSE {device identity is NOT proven}
      end if
   else
      throw {device signature mismatch}
      return FALSE
   end if
end if
```

Algorithm 3 Data Provenance Contract

Input: transaction tx

$tx \leftarrow$ hash(data), tx_type, publickey (device), & timestamp (data collected)

$tx\_type \leftarrow$ valid_record

Output: $True$ **or** $False$

```
tx' ← query(publickey and timestamp) {query the blockchain to get the transaction record}
if (tx' = null) then
  throw {could not locate the record}
else
  if (device'.sig = device.sig) then
    if (tx'.hashvalue = hash(data)) then
      return TRUE {record is proven}
    else
      return FALSE {record is NOT proven}
    end if
  else
    throw {signature mismatch}
    return FALSE
  end if
end if
```

Algorithm 4 Traffic Profile Provenance Contract

Input: transaction tx'

$tx' \leftarrow$ publickey (device), tx_type, timestamp, traffic_profile with feature vector f, & threshold vector t for feature vector f in tx'

$tx\_type \leftarrow$ traffic_proven

Output: tx, $traffic\_proven$ ($True$ **or** $False$)

{True: discrepancy detected & False: no discrepancy}

for all $k : k \in$ set of recent traffic profiles tx_{n-1} **do**
  generate correlation model C for the feature vector f in tx_{n-1} {learning from the previously stored data in the blockchain}
end for
for all features f in tx' **do**
  if $(\delta(tx'.f, C) > t)$ **then**
    $discrepancy\_status \leftarrow TRUE$
    $traffic\_proven \leftarrow FALSE$
  else
    $discrepancy\_status \leftarrow FALSE$
    $traffic\_proven \leftarrow TRUE$
  end if
end for
$tx \leftarrow tx' \cup traffic\_proven$
$PutState(tx)$
$WriteState(tx)$ {transaction tx is validated and successfully committed to the ledger}
return tx, $discrepancy\_status$

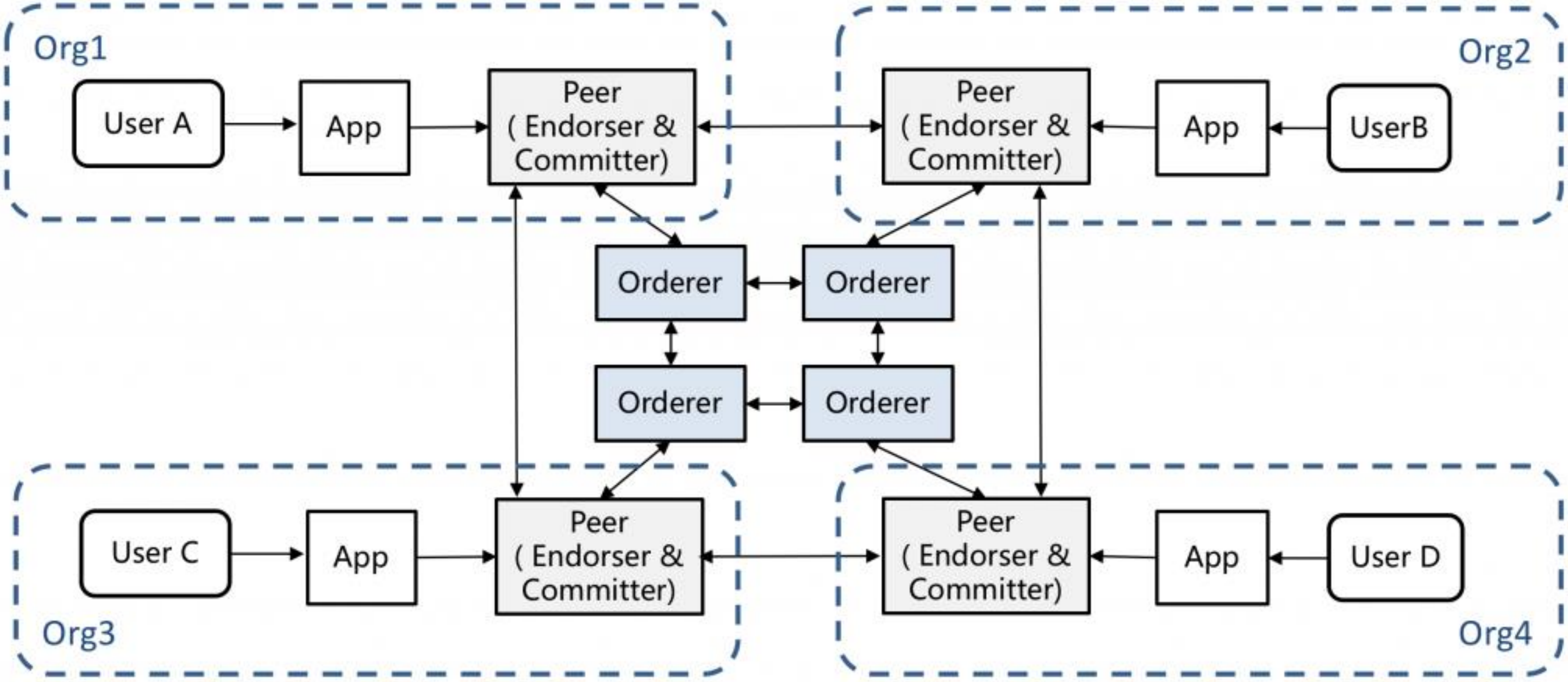
Significance

i. Trusted data collection
ii. Protected
iii. Tampered proof
iv. Data transparency and auditability

Implementation

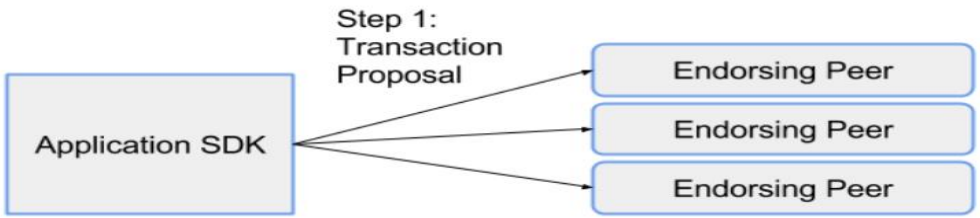
- Hyperledger Fabric

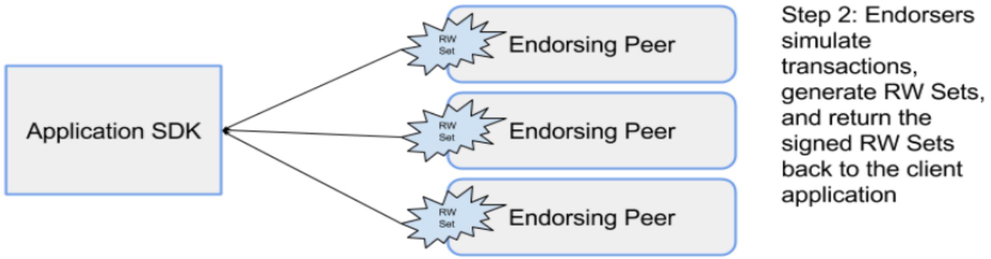
Hyperledger Fabric

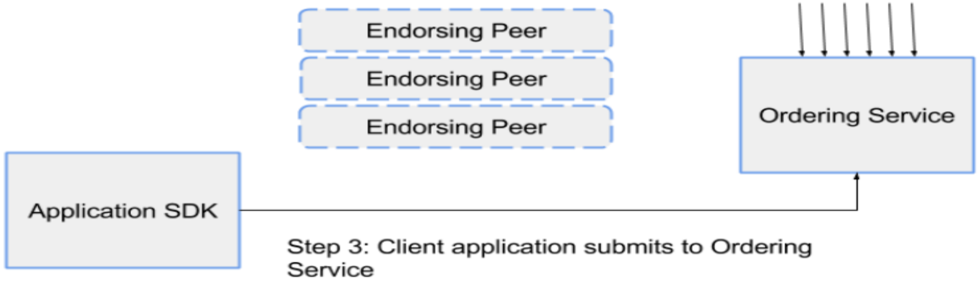


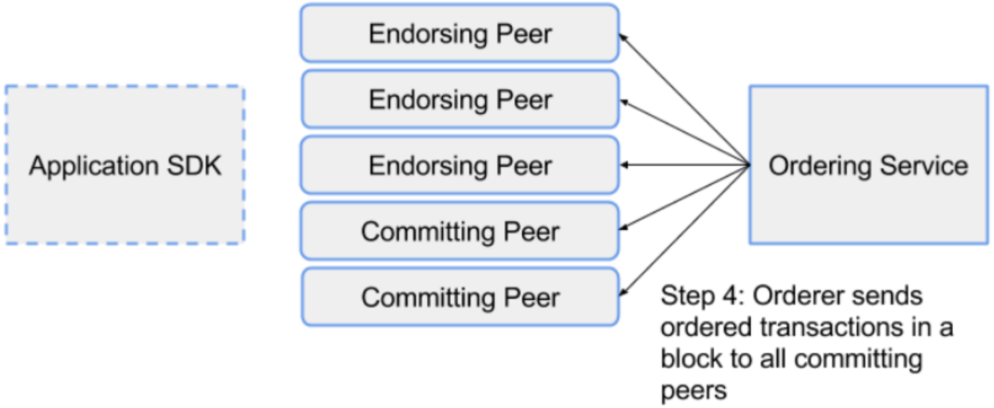
Fabric 1.0 deployment model. Some arrows are omitted for simplicity

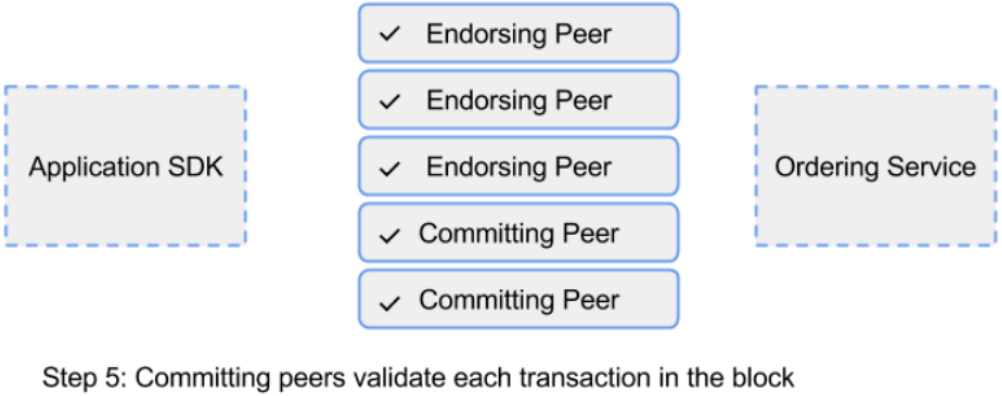
Transaction Flow

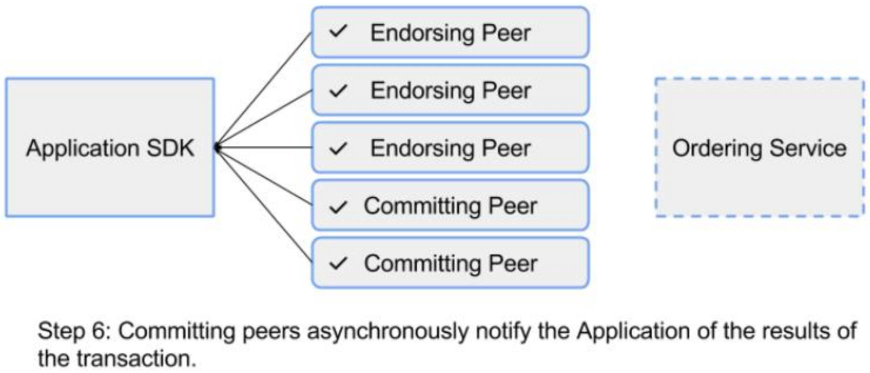




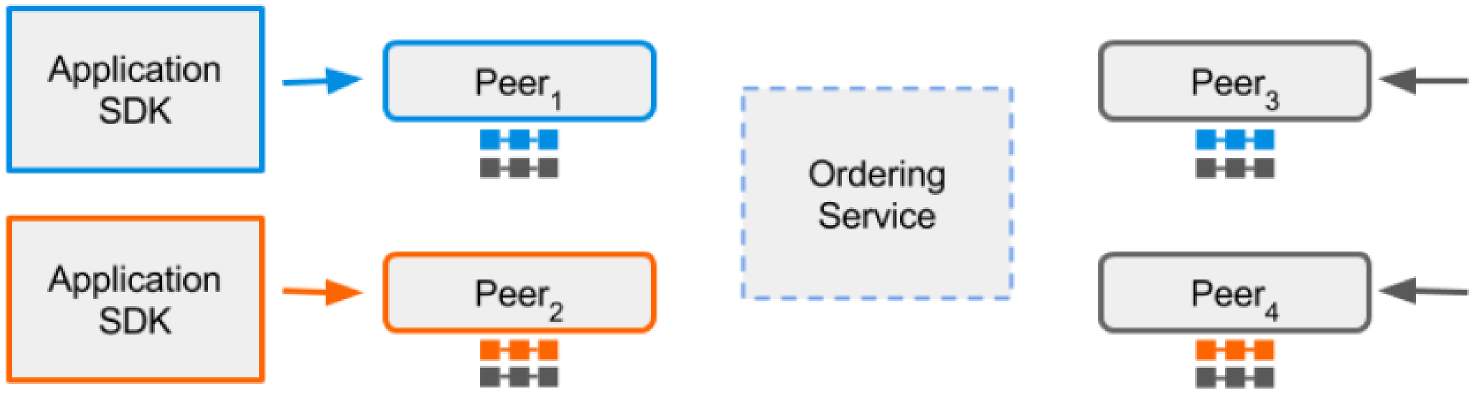


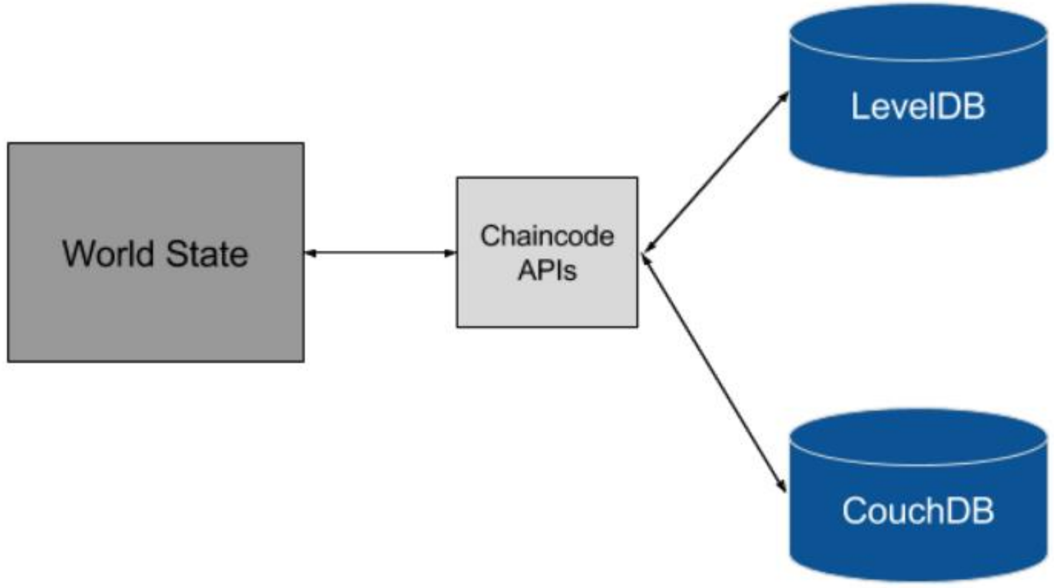


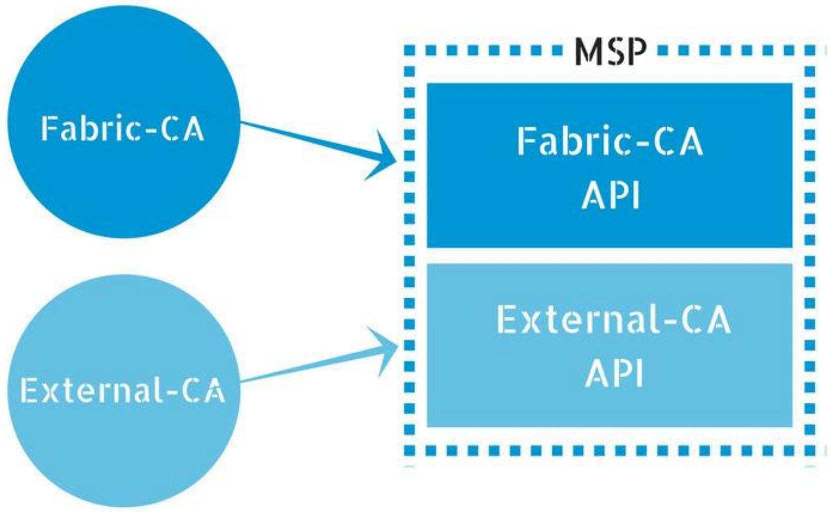




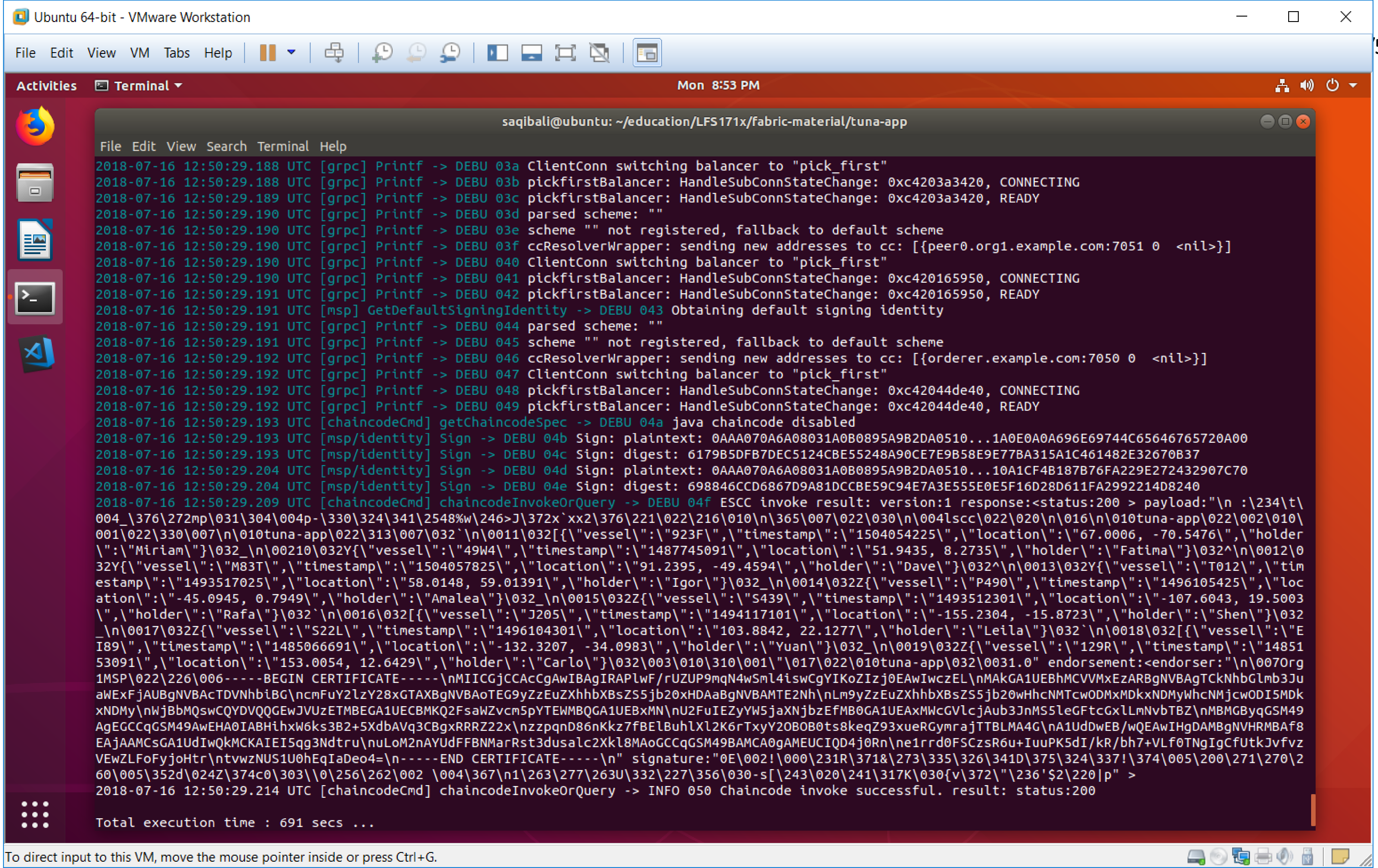
Channels, World State Database, & MSP







5)



Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Activities
Terminal
Mon 8:53 PM
saqibali@ubuntu: ~/education/LFS171x/fabric-material/tuna-app
File Edit View Search Terminal Help
2018-07-16 12:50:29.188 UTC [grpc] Printf -> DEBU 03a ClientConn switching balancer to "pick_first"
2018-07-16 12:50:29.188 UTC [grpc] Printf -> DEBU 03b pickfirstBalancer: HandleSubConnStateChange: 0xc4203a3420, CONNECTING
2018-07-16 12:50:29.189 UTC [grpc] Printf -> DEBU 03c pickfirstBalancer: HandleSubConnStateChange: 0xc4203a3420, READY
2018-07-16 12:50:29.190 UTC [grpc] Printf -> DEBU 03d parsed scheme: ""
2018-07-16 12:50:29.190 UTC [grpc] Printf -> DEBU 03e scheme "" not registered, fallback to default scheme
2018-07-16 12:50:29.190 UTC [grpc] Printf -> DEBU 03f ccResolverWrapper: sending new addresses to cc: [{peer0.org1.example.com:7051 0 <nil>}]
2018-07-16 12:50:29.190 UTC [grpc] Printf -> DEBU 040 ClientConn switching balancer to "pick_first"
2018-07-16 12:50:29.190 UTC [grpc] Printf -> DEBU 041 pickfirstBalancer: HandleSubConnStateChange: 0xc420165950, CONNECTING
2018-07-16 12:50:29.191 UTC [grpc] Printf -> DEBU 042 pickfirstBalancer: HandleSubConnStateChange: 0xc420165950, READY
2018-07-16 12:50:29.191 UTC [msp] GetDefaultSigningIdentity -> DEBU 043 Obtaining default signing identity
2018-07-16 12:50:29.191 UTC [grpc] Printf -> DEBU 044 parsed scheme: ""
2018-07-16 12:50:29.191 UTC [grpc] Printf -> DEBU 045 scheme "" not registered, fallback to default scheme
2018-07-16 12:50:29.192 UTC [grpc] Printf -> DEBU 046 ccResolverWrapper: sending new addresses to cc: [{orderer.example.com:7050 0 <nil>}]
2018-07-16 12:50:29.192 UTC [grpc] Printf -> DEBU 047 ClientConn switching balancer to "pick_first"
2018-07-16 12:50:29.192 UTC [grpc] Printf -> DEBU 048 pickfirstBalancer: HandleSubConnStateChange: 0xc42044de40, CONNECTING
2018-07-16 12:50:29.192 UTC [grpc] Printf -> DEBU 049 pickfirstBalancer: HandleSubConnStateChange: 0xc42044de40, READY
2018-07-16 12:50:29.193 UTC [chaincodeCmd] getChaincodeSpec -> DEBU 04a java chaincode disabled
2018-07-16 12:50:29.193 UTC [msp/identity] Sign -> DEBU 04b Sign: plaintext: 0AAA070A6A08031A0B0895A9B2DA0510...1A0E0A0A696E69744C65646765720A00
2018-07-16 12:50:29.193 UTC [msp/identity] Sign -> DEBU 04c Sign: digest: 6179B5DFB7DEC5124CBE55248A90CE7E9B58E9E77BA315A1C461482E32670B37
2018-07-16 12:50:29.204 UTC [msp/identity] Sign -> DEBU 04d Sign: plaintext: 0AAA070A6A08031A0B0895A9B2DA0510...10A1CF4B187B76FA229E272432907C70
2018-07-16 12:50:29.204 UTC [msp/identity] Sign -> DEBU 04e Sign: digest: 698846CCD6867D9A81DCCBE59C94E7A3E555E0E5F16D28D611FA2992214D8240
2018-07-16 12:50:29.209 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> DEBU 04f ESCC invoke result: version:1 response:<status:200 > payload:"\n :\234\t\
004_\376\272mp\031\304\004p-\330\324\341\2548%w\246>J\372x`xx2\376\221\022\216\010\n\365\007\022\030\n\004lscc\022\020\n\016\n\010tuna-app\022\002\010\
001\022\330\007\n\010tuna-app\022\313\007\032`\n\0011\032[{\"vessel\":\"923F\",\"timestamp\":\"1504054225\",\"location\":\"67.0006, -70.5476\",\"holder
\":\"Miriam\"}\032_\n\00210\032Y{\"vessel\":\"49W4\",\"timestamp\":\"1487745091\",\"location\":\"51.9435, 8.2735\",\"holder\":\"Fatima\"}\032^\n\0012\0
32Y{\"vessel\":\"M83T\",\"timestamp\":\"1504057825\",\"location\":\"91.2395, -49.4594\",\"holder\":\"Dave\"}\032^\n\0013\032Y{\"vessel\":\"T012\",\"tim
estamp\":\"1493517025\",\"location\":\"58.0148, 59.01391\",\"holder\":\"Igor\"}\032_\n\0014\032Z{\"vessel\":\"P490\",\"timestamp\":\"1496105425\",\"loc
ation\":\"-45.0945, 0.7949\",\"holder\":\"Amalea\"}\032_\n\0015\032Z{\"vessel\":\"S439\",\"timestamp\":\"1493512301\",\"location\":\"-107.6043, 19.5003
\",\"holder\":\"Rafa\"}\032`\n\0016\032[{\"vessel\":\"J205\",\"timestamp\":\"1494117101\",\"location\":\"-155.2304, -15.8723\",\"holder\":\"Shen\"}\032
_\n\0017\032Z{\"vessel\":\"S22L\",\"timestamp\":\"1496104301\",\"location\":\"103.8842, 22.1277\",\"holder\":\"Leila\"}\032`\n\0018\032[{\"vessel\":\"E
I89\",\"timestamp\":\"1485066691\",\"location\":\"-132.3207, -34.0983\",\"holder\":\"Yuan\"}\032_\n\0019\032Z{\"vessel\":\"129R\",\"timestamp\":\"14851
53091\",\"location\":\"153.0054, 12.6429\",\"holder\":\"Carlo\"}\032\003\010\310\001\"\017\022\010tuna-app\032\0031.0" endorsement:<endorser:"\n\007Org
1MSP\022\226\006-----BEGIN CERTIFICATE-----\nMIICGjCCAcCgAwIBAgIRAPlwF/rUZUP9mqN4wSml4iswCgYIKoZIzj0EAwIwczEL\nMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3Ju
aWExFjAUBgNVBAcTDVNhbiBG\ncmFuY2lzY28xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZS5jb20xHDAaBgNVBAMTE2Nh\nLm9yZzEuZXhhbXBsZS5jb20wHhcNMTcwODMxMDkxNDMyWhcNMjcwODI5MDk
xNDMy\nWjBbMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEWMBQGA1UEBxMN\nU2FuIEZyYW5jaXNjbzEfMB0GA1UEAxMWcGVlcjAub3JnMS5leGFtcGxlLmNvbTBZ\nMBMGByqGSM49
AgEGCCqGSM49AwEHA0IABHihxW6ks3B2+5XdbAVq3CBgxRRRZ22x\nzzpqnD86nKkz7fBElBuhlXl2K6rTxyY2OBOB0ts8keqZ93xueRGymrajTTBLMA4G\nA1UdDwEB/wQEAwIHgDAMBgNVHRMBAf8
EAjAAMCsGA1UdIwQkMCKAIEI5qg3Ndtru\nuLoM2nAYUdFFBNMarRst3dusalc2Xkl8MAoGCCqGSM49BAMCA0gAMEUCIQD4j0Rn\ne1rrd0FSCzsR6u+IuuPK5dI/kR/bh7+VLf0TNgIgCfUtkJvfvz
VEwZLFoFyjoHtr\ntvwzNUS1U0hEqIaDeo4=\n-----END CERTIFICATE-----\n" signature:"0E\002!\000\231R\371&\273\335\326\341D\375\324\337!\374\005\200\271\270\2
60\005\352d\024Z\374c0\303\\0\256\262\002 \004\367\n1\263\277\263U\332\227\356\030-s[\243\020\241\317K\030{v\372\"\236'$2\220|p" >
2018-07-16 12:50:29.214 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 050 Chaincode invoke successful. result: status:200
Total execution time : 691 secs ...
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
 Store path:/home/saqibali/.hfc-key-store
Successfully loaded admin from persistence
Assigned the admin user to the fabric client ::{"name":"admin","mspid":"Org1MSP","roles":null,"affiliation":"","enrollmentSecret":"","enrollment":{"sig
ningIdentity":"addc3b90966cbab5468e9a06a029e55a2a72845fc7359dbce60f40396c9b2987","identity":{"certificate":"-----BEGIN CERTIFICATE-----\nMIICAjCCAaigAw
IBAgIUCxv+G28gebCNbKO0doLHo7XEjCAwCgYIKoZIzj0EAwIw\nczELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNh\nbiBGcmFuY2lzY28xGTAXBgNVBAoTEG9yZ
zEuZXhhbXBsZS5jb20xHDAaBgNVBAMT\nE2NhLm9yZzEuZXhhbXBsZS5jb20wHhcNMTgwNzA4MDc1NzAwWhcNMTkwNzA4MDgw\nMjAwWjAhMQ8wDQYDVQQLEwZjbGllbnQxDjAMBgNVBAMTBWFkbWlu
MFkwEwYHKoZI\nzj0CAQYIKoZIzj0DAQcDQgAE68nxLGBALHrPrgSohZhWx3GsbDspu/+09a9Dc/rn\nDoChIIJNLAORxAX4bbc/BWatHVWVsWdzTAcqRXduzt69fKNsMGowDgYDVR0PAQH/\nBAQDA
geAMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFO+HgGBWA8jSXOZ10ur3R6Ds\n2NCdMCsGA1UdIwQkMCKAIEI5qg3NdtruuLoM2nAYUdFFBNMarRst3dusalc2Xkl8\nMAoGCCqGSM49BAMCA0gAMEUC
IQDPRIEr89N4VxS6mRLXChKbVo1nULdlvZXJaGWp\nSM8VHgIgatIi7bQLazxC+E/x+OUVgCJ8qYtIaXAb6cCEBzuTDrY=\n-----END CERTIFICATE-----\n"}}}
```

```
Store path:/home/saqibali/.hfc-key-store
Successfully loaded user1 from persistence
Query has completed, checking results
Response is  [{"Key":"1", "Record":{"holder":"Miriam","location":"67.0006, -70.5476","timestamp":"1504054225","vessel":"923F"}},{"Key":"10", "Record":{
"holder":"Fatima","location":"51.9435, 8.2735","timestamp":"1487745091","vessel":"49W4"}},{"Key":"2", "Record":{"holder":"Dave","location":"91.2395, -4
9.4594","timestamp":"1504057825","vessel":"M83T"}},{"Key":"3", "Record":{"holder":"Igor","location":"58.0148, 59.01391","timestamp":"1493517025","vesse
l":"T012"}},{"Key":"4", "Record":{"holder":"Amalea","location":"-45.0945, 0.7949","timestamp":"1496105425","vessel":"P490"}},{"Key":"5", "Record":{"hol
der":"Rafa","location":"-107.6043, 19.5003","timestamp":"1493512301","vessel":"S439"}},{"Key":"6", "Record":{"holder":"Shen","location":"-155.2304, -15
.8723","timestamp":"1494117101","vessel":"J205"}},{"Key":"7", "Record":{"holder":"Leila","location":"103.8842, 22.1277","timestamp":"1496104301","vesse
l":"S22L"}},{"Key":"8", "Record":{"holder":"Yuan","location":"-132.3207, -34.0983","timestamp":"1485066691","vessel":"EI89"}},{"Key":"9", "Record":{"ho
lder":"Carlo","location":"153.0054, 12.6429","timestamp":"1485153091","vessel":"129R"}}]
```

```
submit recording of a tuna catch:
[ '11', '28.012, 150.425', '498256368', 'Johon', '0239L' ]
Store path:/home/saqibali/.hfc-key-store
Successfully loaded user1 from persistence
Assigning transaction_id:  bad37fa295c4e77b00ee9754a21c7d3326fe73328ab56340d3f2ef6f2a1c8425
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
info: [EventHub.js]: _connect - options {}
The transaction has been committed on peer localhost:7053
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

Future Research Directions

Key Issues in Blockchain & Smart Contracts

1. Privacy
2. Scalability
3. Smart Contracts are not Intelligent

Parno, B., Howell, J., Gentry, C., & Raykova, M. (2013). Pinocchio: Nearly practical verifiable computation. *Proceedings - IEEE Symposium on Security and Privacy*, 238–252. https://doi.org/10.1109/SP.2013.47

Ames, S., Hazay, C., Ishai, Y., & Venkitasubramaniam, M. (2017). Ligero : Lightweight Sublinear Arguments Without a Trusted Setup. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS '17*, 2087–2104. https://doi.org/10.1145/3133956.3134104

1. Zero knowledge Proofs

Def. Zero-knowledge proofs are encryption schemes used to prove that you know something without revealing what it is.

For example, you can show without a doubt that you know the answer to a puzzle without actually disclosing the solution.

Types:

1. Interactive zero knowledge proofs
2. Non interactive zero knowledge proofs

Oded Goldreich and Yair Oren. Definitions and Properties of Zero-Knowledge Proof Systems. Journal of Cryptology. Vol 7(1). 1–32. 1994 (PS)

1. Interactive Zero Knowledge Proofs

Def. Interactive zero-knowledge proofs require interaction between the individual (or computer system) proving their knowledge and the individual validating the proof.

Actors in the system:

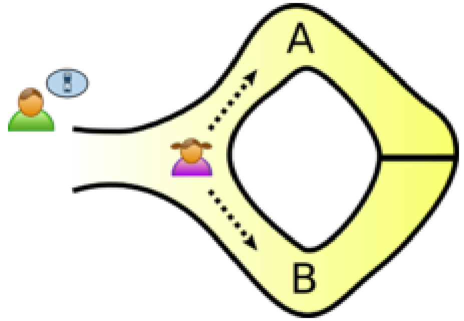
1. **Proof** - who claim some knowledge
2. **Verifier** - who verify the claim of the prover

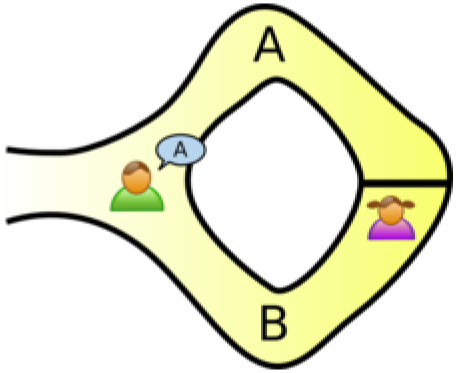
Key requirement: Interaction between the Prover & Verifier - in the form of the challenges such that the responses from the prover will convince the verifier if and only if the statement is true.

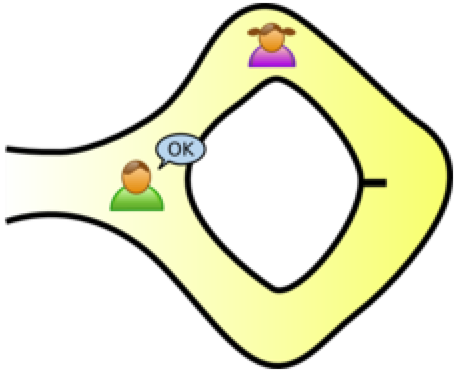
*Goldwasser, S.; Micali, S.; Rackoff, C. (1989), "The knowledge complexity of interactive proof systems" (PDF), SIAM Journal on Computing, Philadelphia: Society for Industrial and Applied Mathematics, **18** (1): 186–208, doi:10.1137/0218012, ISSN 1095-7111*

1. How Interactive Zero Knowledge Proofs work?

Example: Ali Baba Cave
Peggy (the **prover** of the statement) and Victor (the **verifier** of the statement).
Peggy has uncovered the secret word used to open a magic door in a cave.
Victor wants to know whether Peggy knows the secret word.
Peggy does not want to reveal her knowledge (the secret word) to Victor







*Quisquater, Jean-Jacques; Guillou, Louis C.; Berson, Thomas A. (1990). "How to Explain Zero-Knowledge Protocols to Your Children" (PDF). Advances in Cryptology – CRYPTO '89: Proceedings. **435**: 628–631.*

1. Issue with Interactive Zero Knowledge Proofs?

Advantage: An interactive zero-knowledge proof has the advantage that only the verifier can be absolutely convinced that the prover has the knowledge.

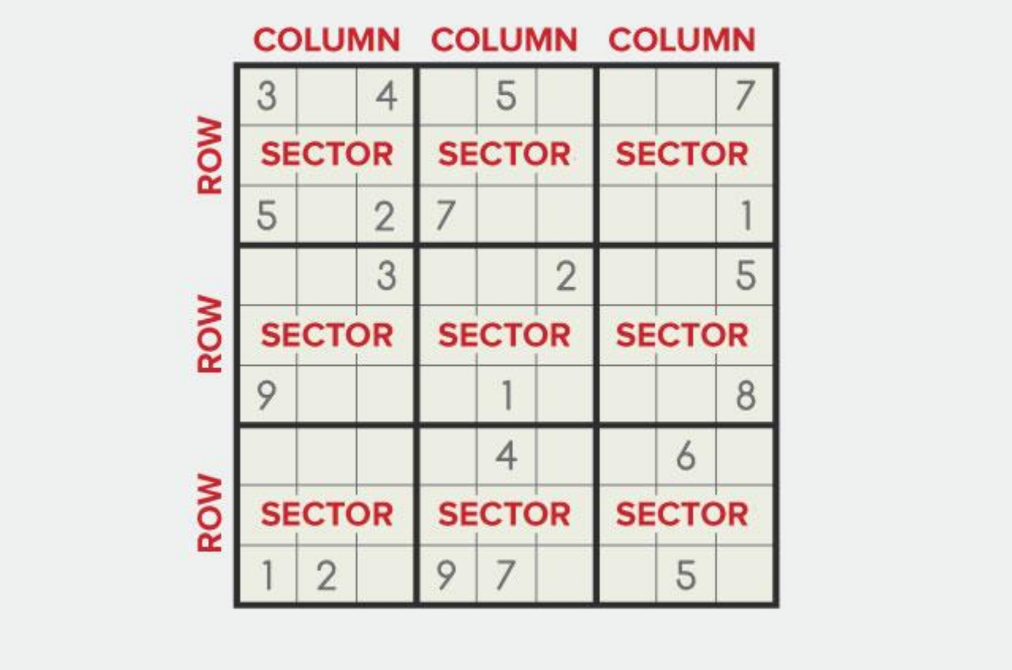
Disadvantage: If bystanders and observers can’t verify the claim, the prover then has to interact with every verifier independently—which takes time and is resource intensive.

2. Non-interactive zero-knowledge proofs

The reason for non-interactive zero-knowledge proofs is to allow a large number of observers to verify the proof efficiently.

In Blockchains: every block need to be computed and verified by every node in the network which raise the issue of scalability

Example: **Sudoku Puzzle**



Blum, Manuel; Feldman, Paul; Micali, Silvio (1988). "Non-Interactive Zero-Knowledge and Its Applications". Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988): 103–112. doi:10.1145/62212.62222.

Practical Applications of Zero Knowledge Proofs

	prover scalability (quasilinear time)	verifier scalability (polylogarithmic time)	Transparency (public randomness)	Post-quantum security
hPKC	Yes	Only repeated computation	No	No
DLP	Yes	No	Yes	No
IP	Yes	No	Yes	No
MPC	Yes	No	Yes	Yes
IVC+hPKC	Yes	Yes	No	No
ZK-STARK	Yes	Yes	Yes	Yes

Figure 2: Theoretical comparison of universal (NP complete) realized ZK systems.

References

[1] M. Díaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing,” J. Netw. Comput. Appl., vol. 67, pp. 99–117, 2016.

[2] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of Cloud computing and Internet of Things: A survey,” Futur. Gener. Comput. Syst., vol. 56, pp. 684–700, 2016.

[3] K. Lee, S. Kim, J. Jeong, S. Lee, H. Kim, and J. S. Park, “A framework for DNS naming services for Internet-of-Things devices,” Future Generation Computer Systems, North-Holland, 31-Jan-2018.

[4] D. He and S. Zeadally, “An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography,” vol. 2, no. 1, pp. 72–83, 2015.

[5] O. Novo, “Blockchain Meets IoT: an Architecture for Scalable Access Management in IoT,” IEEE Internet Things J., vol. 14, no. 8, pp. 1–1, 2018.

[6] S. Wilkinson and J. Lowry, “MetaDisk: Blockchain-Based Decentralized File Storage Application,” pp. 1–11, 2014.

[7] D. T. T. Anh, M. Zhang, B. C. Ooi, and G. Chen, “Untangling Blockchain: A Data Processing View of Blockchain Systems,” IEEE Trans. Knowl. Data Eng., vol. 4347, no. c, pp. 1–1, 2018.

[8] J. R. Douceur, “The Sybil Attack,” pp. 251–260, 2002.

[9] M. Castro and B. Liskov, “Practical byzantine fault tolerance and proactive recovery,” ACM Trans. Comput. Syst., vol. 20, no. 4, pp. 398–461, 2002.

[10] M. Castro, M. Castro, B. Liskov, and B. Liskov, “Practical Byzantine fault tolerance,” OSDI {’}99 Proc. third Symp. Oper. Syst. Des. Implement., no. February, pp. 173–186,

Q/A