Carnegie Mellon University
Information Networking Institute

TrustDER, SLAC

November 28, 2022

Keshava Srinivas
Nithin Ram Gomatam Vasudevan
Noora Alfayez
Pramod Illuri (Project Manager)
Prerit Pathak
Vaanya Gupta

Problem Statement

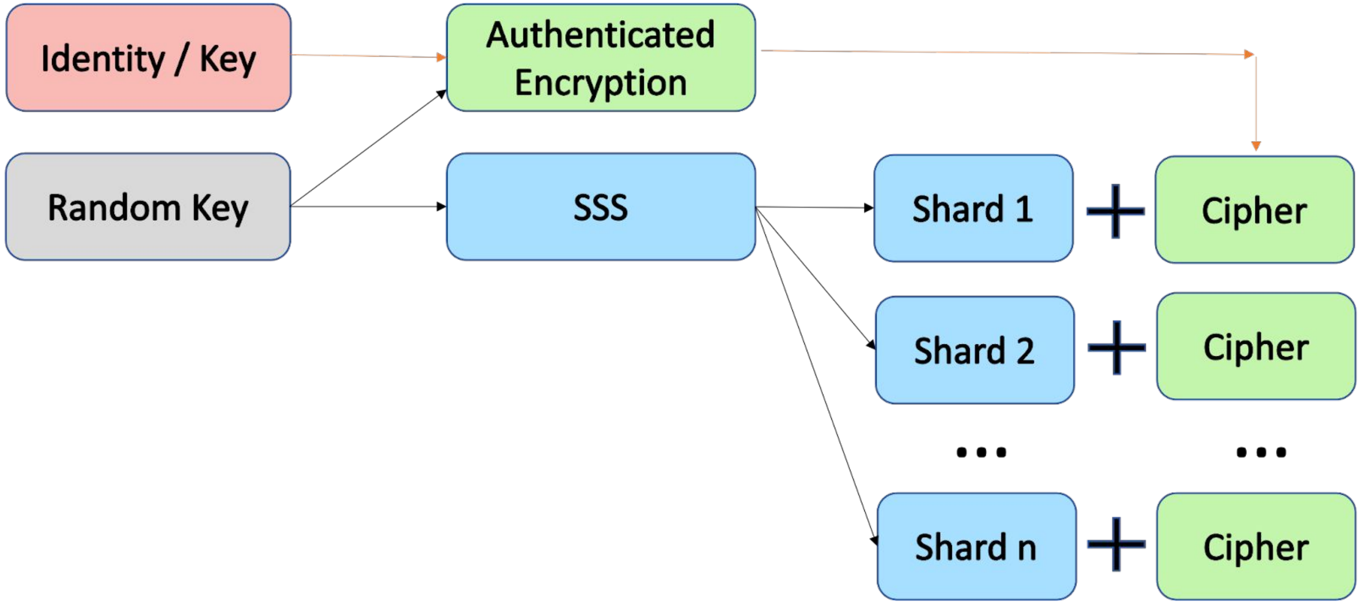
The goal of this project is to implement a software-based information theoretic approach that allows device identity to be verified by network participants.

This approach aims to be decentralized to prevent a single point of failure and be as effective as imprinting an identity in silicon.

Logical Components

The solution revolves around 3 main logical components:

- **The KeyMaker component:** The identity provisioning component that uses Shamir's Secret Sharing (SSS) to distribute shards of a random key to all the nodes in the network.



Logical Components

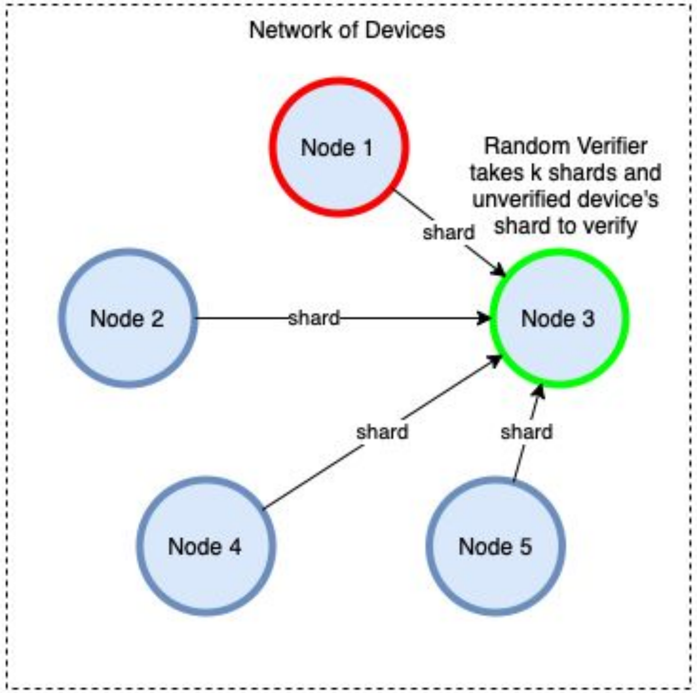
The solution revolves around 3 main logical components:

- **The KeyChecker component:** Blockchain smart contracts that perform the verification through lagrange interpolation using K shards from subset of the N peer nodes on the network along with the newly added unverified node.

$$\ell_j(x) = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}$$

$$= \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m}.$$

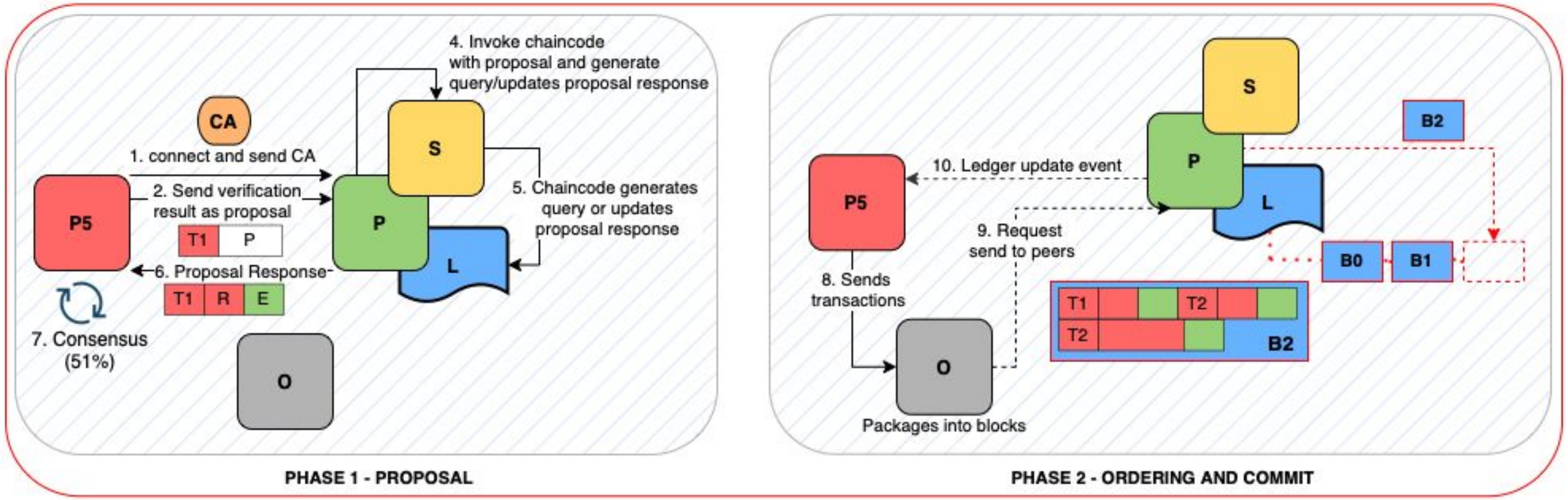
$$L(x) = \sum_{j=0}^{k} y_j \ell_j(x).$$


Network of Devices
Node 1
Random Verifier takes k shards and unverified device's shard to verify
shard
Node 2
shard
Node 3
shard
shard
Node 4
Node 5

Logical Components

The solution revolves around 3 main logical components:

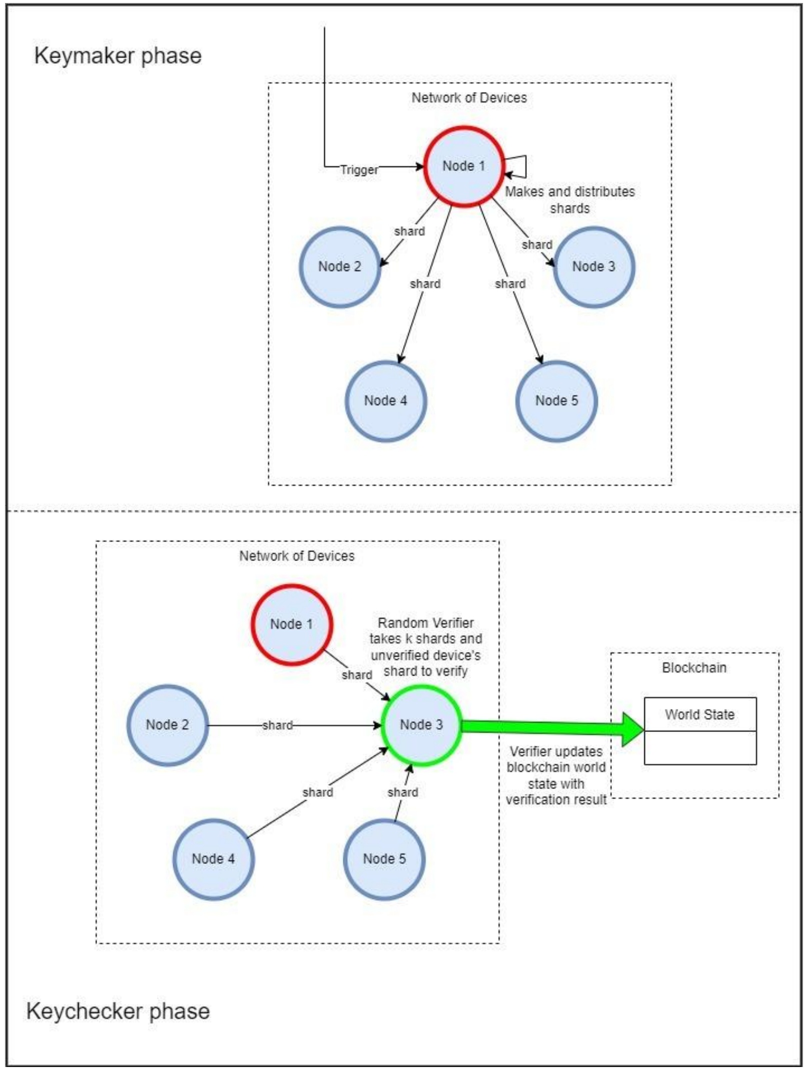
- **The Blockchain network:** The decentralized network that maintains the device verification status through the transactions log (worldstate) and manages function calls for social verification through smart contracts (chaincode).



CA
1. connect and send CA
2. Send verification result as proposal
T1
P
P5
4. Invoke chaincode with proposal and generate query/updates proposal response
S
P
L
5. Chaincode generates query or updates proposal response
6. Proposal Response
T1
R
E
7. Consensus (51%)
O
PHASE 1 - PROPOSAL
S
P
L
B2
10. Ledger update event
P5
9. Request send to peers
8. Sends transactions
B0
B1
T1
T2
T2
B2
O
Packages into blocks
PHASE 2 - ORDERING AND COMMIT

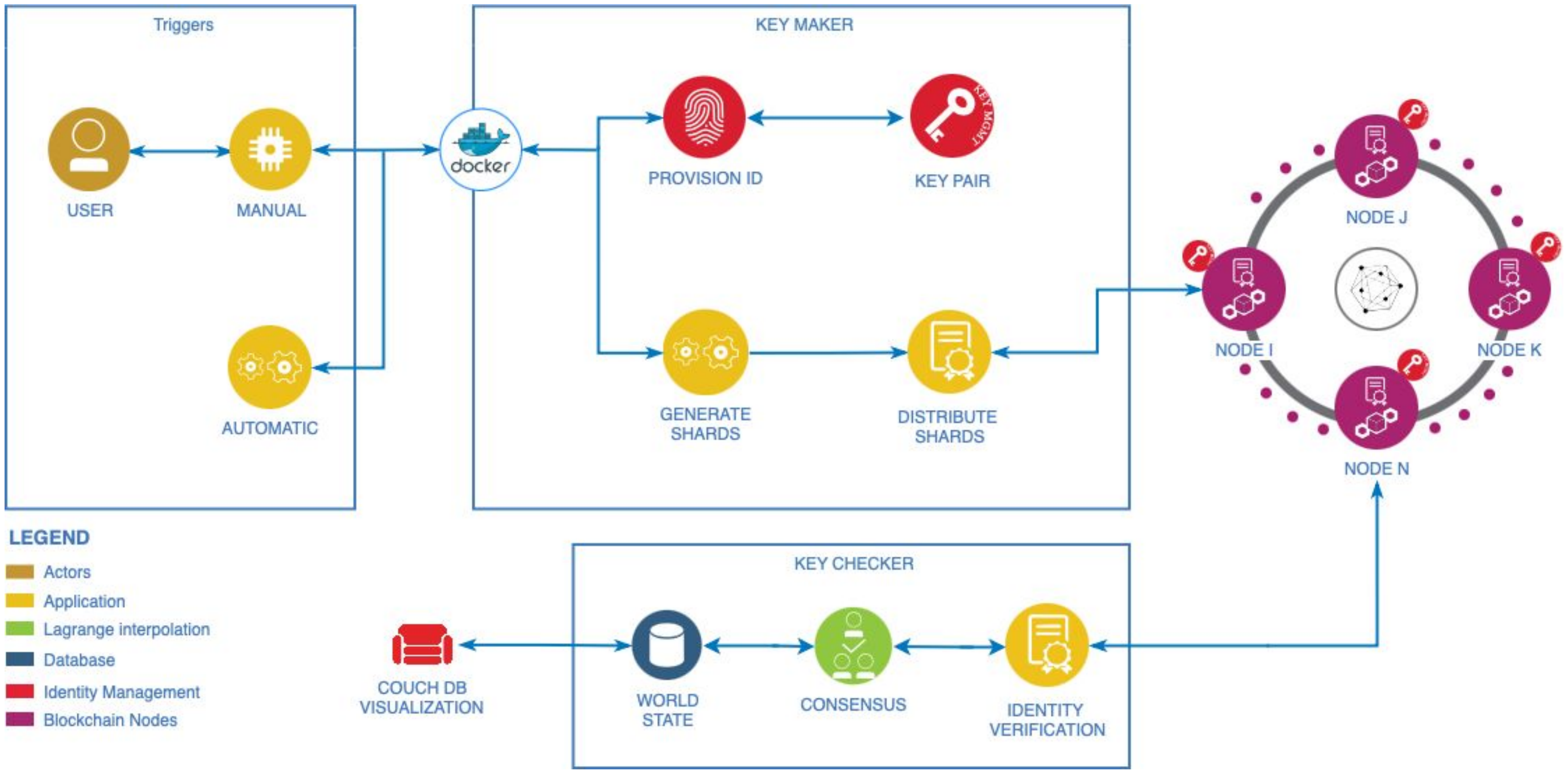
Reference Architecture

Reference architecture of the proposed distributed verification system and communication between each component.

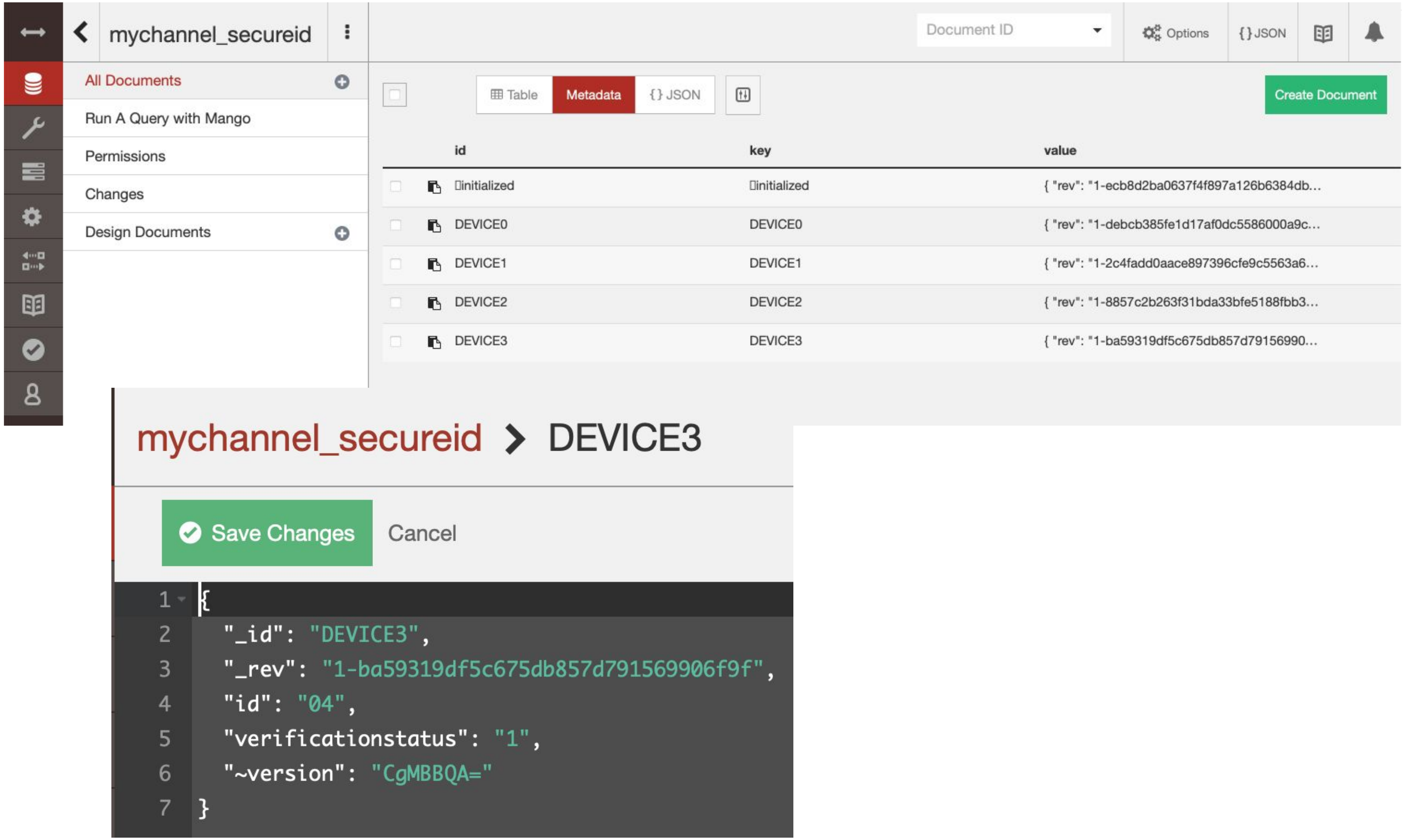


System Architecture

When a new node is added to the network, the world state is checked and the blockchain is triggered to perform verification. A verification of a node can also be triggered by other events such as device maintenance, replacement, or periodic



CouchDB Visualization



	id	key	value
□	initialized	initialized	{ "rev": "1-ecb8d2ba0637f4f897a126b6384db...
□	DEVICE0	DEVICE0	{ "rev": "1-debcb385fe1d17af0dc5586000a9c...
□	DEVICE1	DEVICE1	{ "rev": "1-2c4fadd0aace897396cfe9c5563a6...
□	DEVICE2	DEVICE2	{ "rev": "1-8857c2b263f31bda33bfe5188fbb3...
□	DEVICE3	DEVICE3	{ "rev": "1-ba59319df5c675db857d79156990...

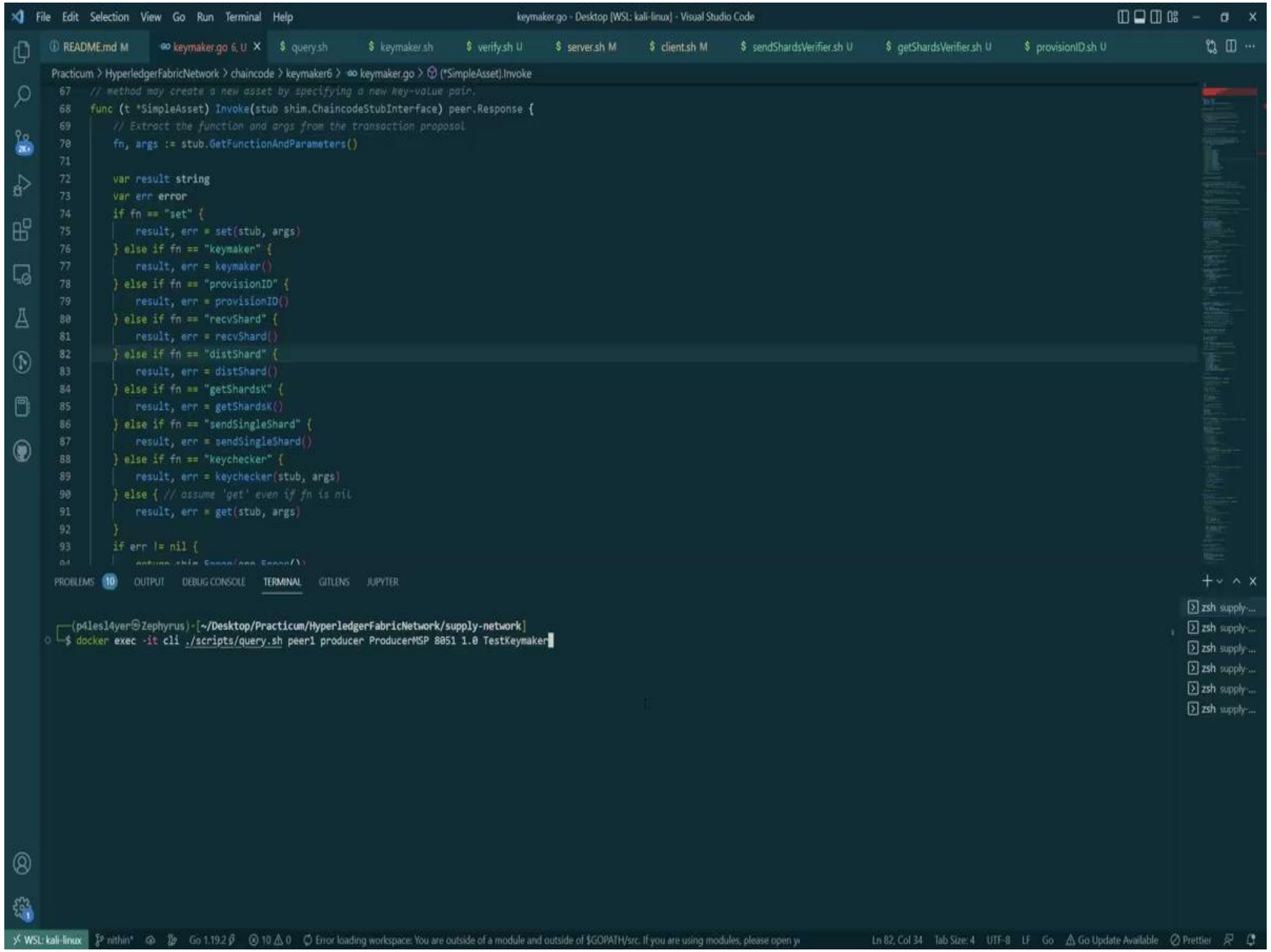
mychannel_secureid > DEVICE3

```
{
  "_id": "DEVICE3",
  "_rev": "1-ba59319df5c675db857d791569906f9f",
  "id": "04",
  "verificationstatus": "1",
  "~version": "CgMBBQA="
}
```

Deviation from original plan

- Switching from curve fitting to Lagrange interpolation for shard verification
- Avoiding Cloud API Endpoints to make the solution decentralized

Demo Video



```go
// method may create a new asset by specifying a new key-value pair.
func (t *SimpleAsset) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
    // Extract the function and args from the transaction proposal
    fn, args := stub.GetFunctionAndParameters()

    var result string
    var err error
    if fn == "set" {
        result, err = set(stub, args)
    } else if fn == "keymaker" {
        result, err = keymaker()
    } else if fn == "provisionID" {
        result, err = provisionID()
    } else if fn == "recvShard" {
        result, err = recvShard()
    } else if fn == "distShard" {
        result, err = distShard()
    } else if fn == "getShardsK" {
        result, err = getShardsK()
    } else if fn == "sendSingleShard" {
        result, err = sendSingleShard()
    } else if fn == "keychecker" {
        result, err = keychecker(stub, args)
    } else { // assume 'get' even if fn is nil
        result, err = get(stub, args)
    }
    if err != nil {
        return shim.Error(err.Error())
```

```
┌──(p4les14yer㉿Zephyrus)-[~/Desktop/Practicum/HyperledgerFabricNetwork/supply-network]
└─$ docker exec -it cli ./scripts/query.sh peer1 producer ProducerMSP 8051 1.0 TestKeymaker
```

Error loading workspace: You are outside of a module and outside of $GOPATH/src. If you are using modules, please open y

Future Work

- Automate the triggers
- Optimize blockchain network implementation
- Implement protocol on raspberry pis
- Market adoption of decentralized software-based authentication lead to lower costs and lower power consumption

Flexibility in problem solving

- We were using the aws managed hyperledger fabric blockchain first, but we had to switch to our own custom implementation.
- We gained extensive knowledge about blockchain and its implementation.
- We were exposed to research projects and innovation.
- We benefited from networking opportunities with SLAC, Stanford, HyperLedger, and CyLab faculty and staff.
- We improved our leadership, teamwork and soft skills.

Acknowledgement

- **Mayank Malik** (Sponsor): His knowledge, enthusiasm, and support throughout the project helped us a lot.
- **Dr. Hanan Hibshi** (Faculty Advisor): Her constant encouragement and comments helped us to improve our project and professional skills.
- **Dr. Cynthia Kuo and Dr. Sujata Telang** (Practicum Professors): Their insights and teachings throughout the course helped us a lot.
- **Arjun Brar** (Secure Blockchain researcher @CMU): His mentoring and knowledge helped us to get guidance to get started and understand the blockchain technology.
- **Pramod Illuri** (Project Manager): His constant check-in's and reminders made it possible to succeed in our project timely.