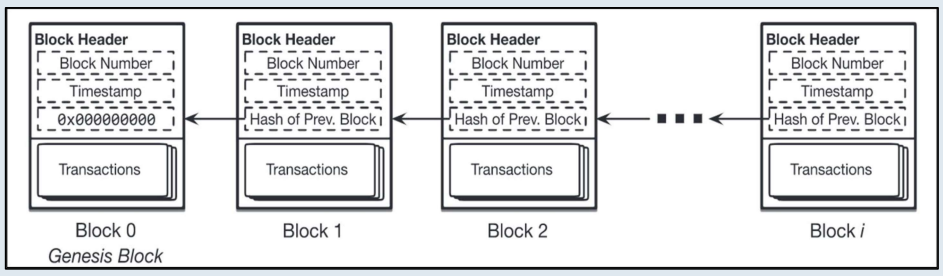
A biophysical observation model for field potentials of networks of LIF neurons

Ameya Kasture, Shaunak Damle, Siddhant Kulkarni, Sharwin Neema

{f20212058, f20212607, f20212606, f20211442}@goa.bits-pilani.ac.in

Introduction



- **Blockchain** brings **transparency**, and **security** in **HPC** environments. Also helps in tracking the **data provenance**.

Smart Contracts:

- **Fundamental** building block.
- Allow **transparent**, **traceable**, and **self-executing** decentralized transactions.
- **Challenging to update** once deployed.
- Prone to **security attacks**.

```
contract SavingsBank {
    mapping(address => uint256) public balances;

    function deposit() public payable {
        balances[msg.sender] += msg.value;
    }

function withdraw() public {
    uint256 bal = balances[msg.sender];
    require(bal > 0);

    (bool sent, ) = msg.sender.call{value: bal}("");
    require(sent, "Failed to send Ether");

    balances[msg.sender] = 0;
}
```

Motivation

Smart Contract Upgradeability

- **Upgrading a contract** means changing the business logic of a smart contract while preserving the contract's state.

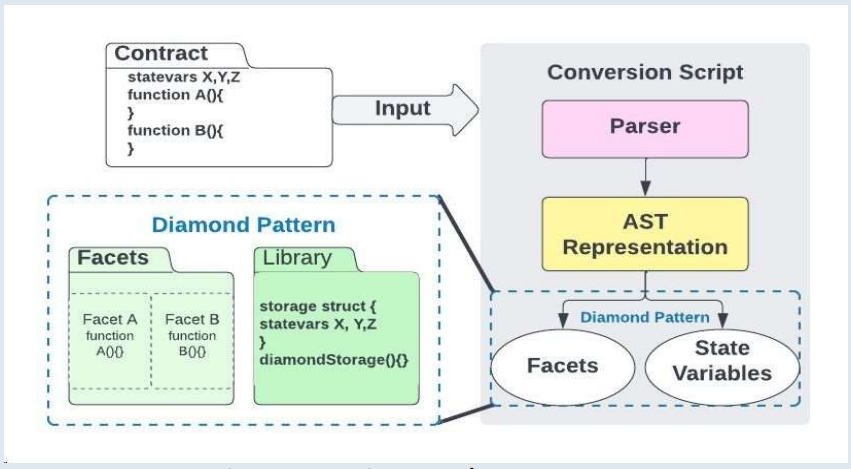
Method	Description	Advantages	Disadvantages
Contract Migration	**Migrating state** (i.e., data) from the **old contract** to a **new** instance of the contract.	• **Straightforward and safe** measure to **upgrade** a smart contract.	• **Time Consuming.** • Can incur **high gas-cost** to manually transfer storage and balance.
Data Separation	Creating **separate contracts** to store **business logic** and **state**.	• **Easier to implement** as compared to contract migration.	• Not ideal to **change contract address** with each **upgrade**.
Proxy Pattern	To **delegate function calls** from an immutable **proxy contract** to a modifiable **logic contract**.	• Easy to **change** the **logic** contract. • **Widely used.**	• **Complicated** and **can induce errors** if used incorrectly.
Diamond Pattern	**Delegate function calls** from a **proxy** contract to **multiple logic** contracts.	• **Easy** to **fix vulnerabilities** after deployment. • **Bypasses** the **24KB** smart contract **size.**	• Users must **trust developers** to not **modify contracts arbitrarily.**

We select **ERC-2535 Diamond, Multi-facet Proxy:**

- **Modular** smart contract, can be extended after deployment.
- Can handle more **complex architecture**.
- Bypasses the **24KB** contract size limit.

Plots Obtained

- **Develop a converter** that automatically converts an existing smart contract into a diamond pattern.



Auto-conversion to a Diamond pattern smart contract.

Results & Conclusion

Vulnerabilities	Normal Smart Contract			Diamond Smart Contract		
	CPU (%)	Memory (MB)	Disk (kB)	CPU (%)	Memory (MB)	Disk (kB)
Reentrancy	99.6	370	4	99.5	375	7
Bad randomness	99.7	378	5	99.7	268	7

- This work proposed an **automatic converter** to convert the immutable smart contracts to a mutable diamond pattern. The converted contracts show **minimal resource overhead.**

- In **future work**, we aim to develop a **framework** to **help developers** deploy the smart contract in correct form, **detect bugs,** and **fix** those in deployment to **reduce operational costs.**

References

1. Abdullah Al Mamun, Feng Yan, and Dongfang Zhao. Baash: lightweight, efficient, and reliable blockchain-as-a-service for hpc systems. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2021.
2. Abdullah Al-Mamun, Feng Yan, and Dongfang Zhao. Scichain:Blockchain-enabled lightweight and efficient data provenance for reproducible scientific computing. In 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, 2021.
3. Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. white paper , 3(37):2–1, 2014.

ACKNOWLEDGEMENT

A special thanks to Dr