unit

Peer-to-peer network for decentralized big data and anonymity

Main ideas of the Unit-chain

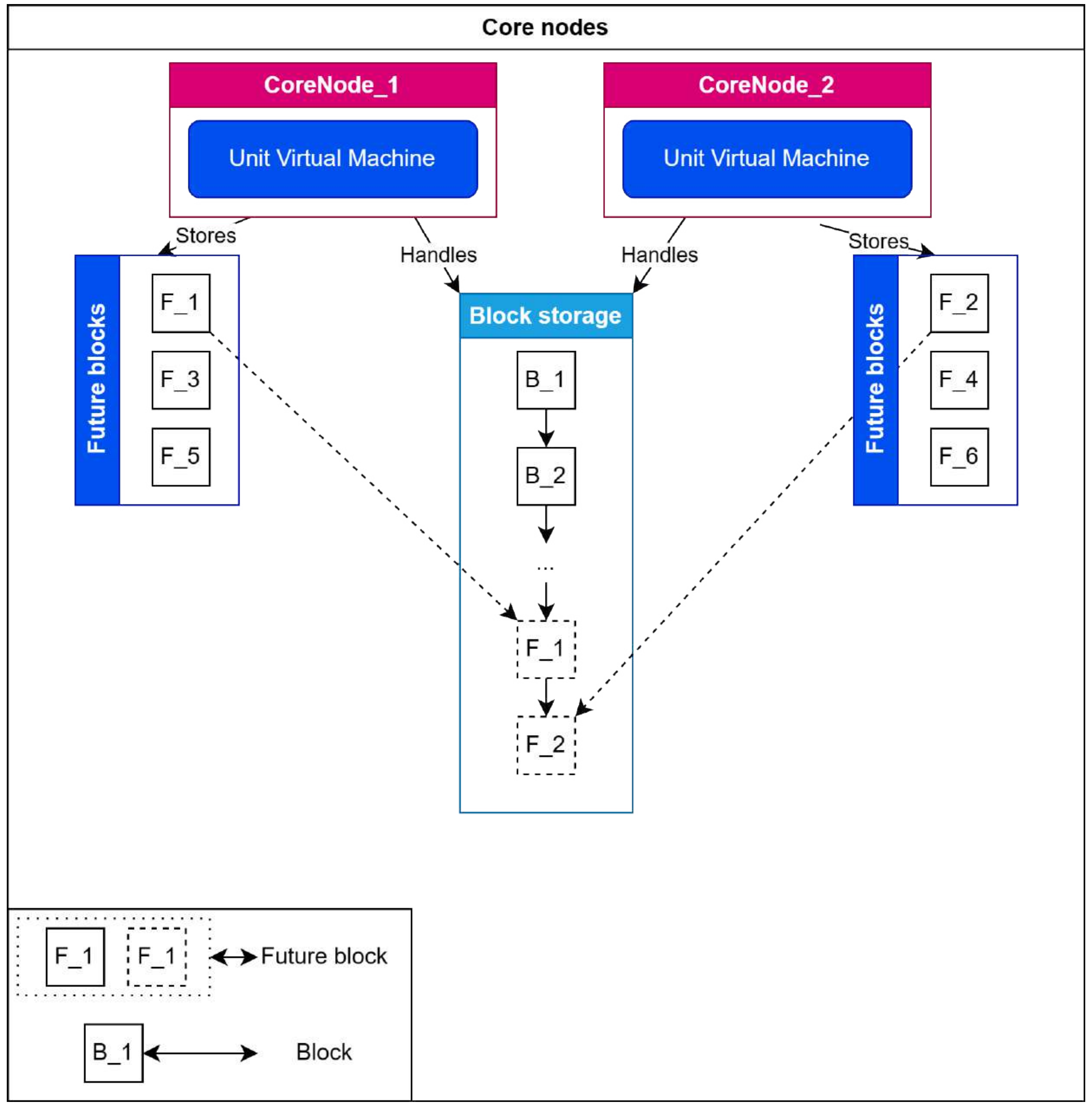
- Support two types of nodes: Core-nodes and Shard-nodes
- Virtual storage providing for smart-contracts and users
- Strong security of the user’s data
- Custom virtual machine for working with smart-contracts
- High speed and low latency

Core-node

- Core-nodes can work independently and fully support the network. They are capable of processing transactions and require high performance servers. Core-nodes also have the ability to provide a small amount of memory for smart-contracts. They have Future-blocks in "queue" to increase speed of the network.

- Future-blocks are written to the main thread of blocks only when there is a choice of the node that provides the block at the moment or the current block + 2.
  In order to reduce the latency of writing a transaction, it is necessary to know future blocks in advance.

Core-node scheme

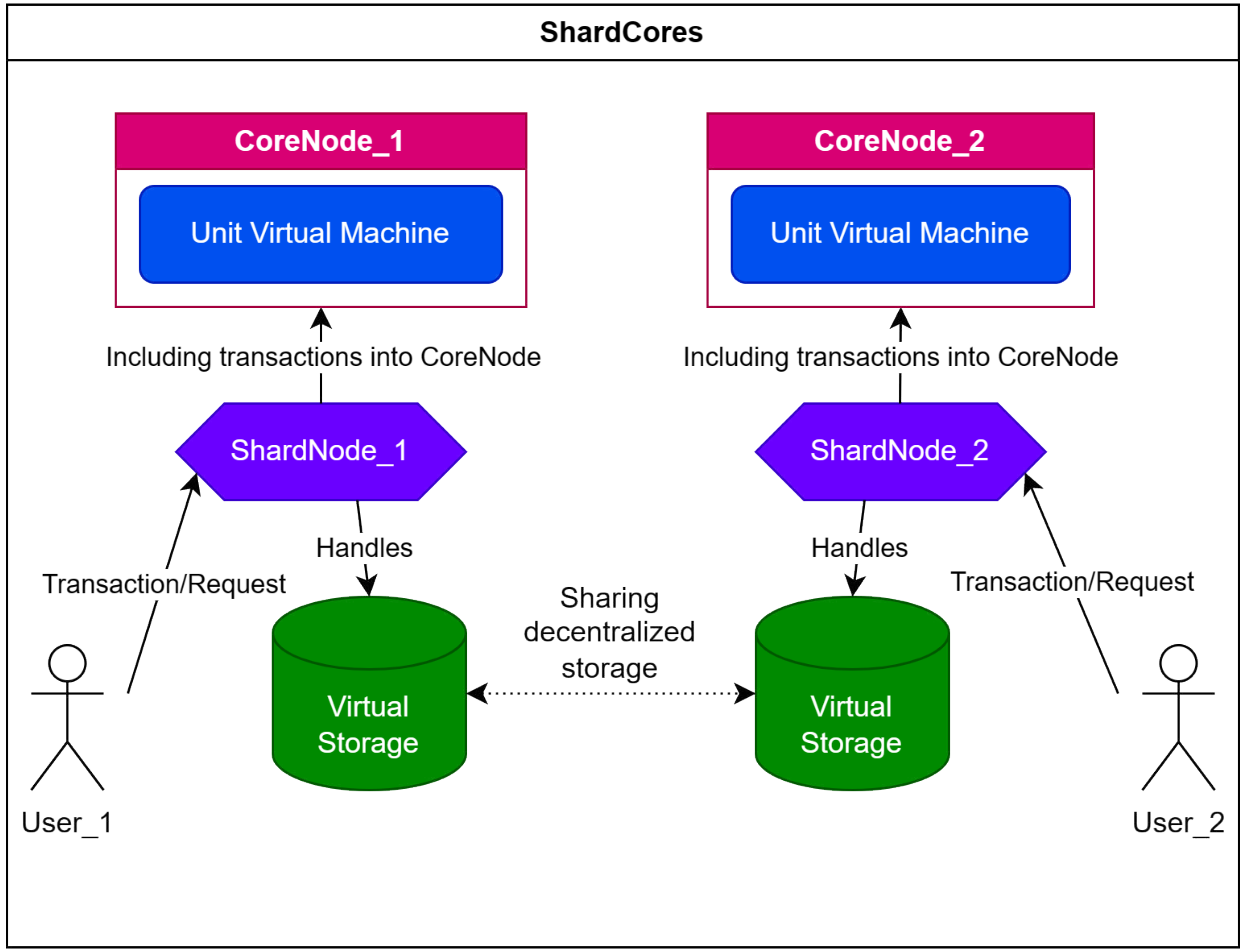


Shard-node

- Shard-nodes are supporters for Core-nodes in transactions verification. Shard-nodes can verify transactions, but cannot support the network. Main idea of Shard-nodes is providing huge amount of virtual memory. Memory allocates under a single consensus algorithm, as well as access to it. Memory is “stacking” and you can get access to it with payable transaction on node’s address (example of address: UNTuv79aE8caXefoNRs9XXfQyigAr7n).

- Core idea of decentralized storage is insipred by BitTorrent and IPFS. Implementaion of it with blockchain provides flexibility, for example for decentralized social networks and huge decentralized storages. All data is secured by quantum-resistant cryptographic algorithm and it is impossible to get access to virtual storage without permissions. Every memory cell will have its own private key to view and modify data in it.

Shard-node scheme

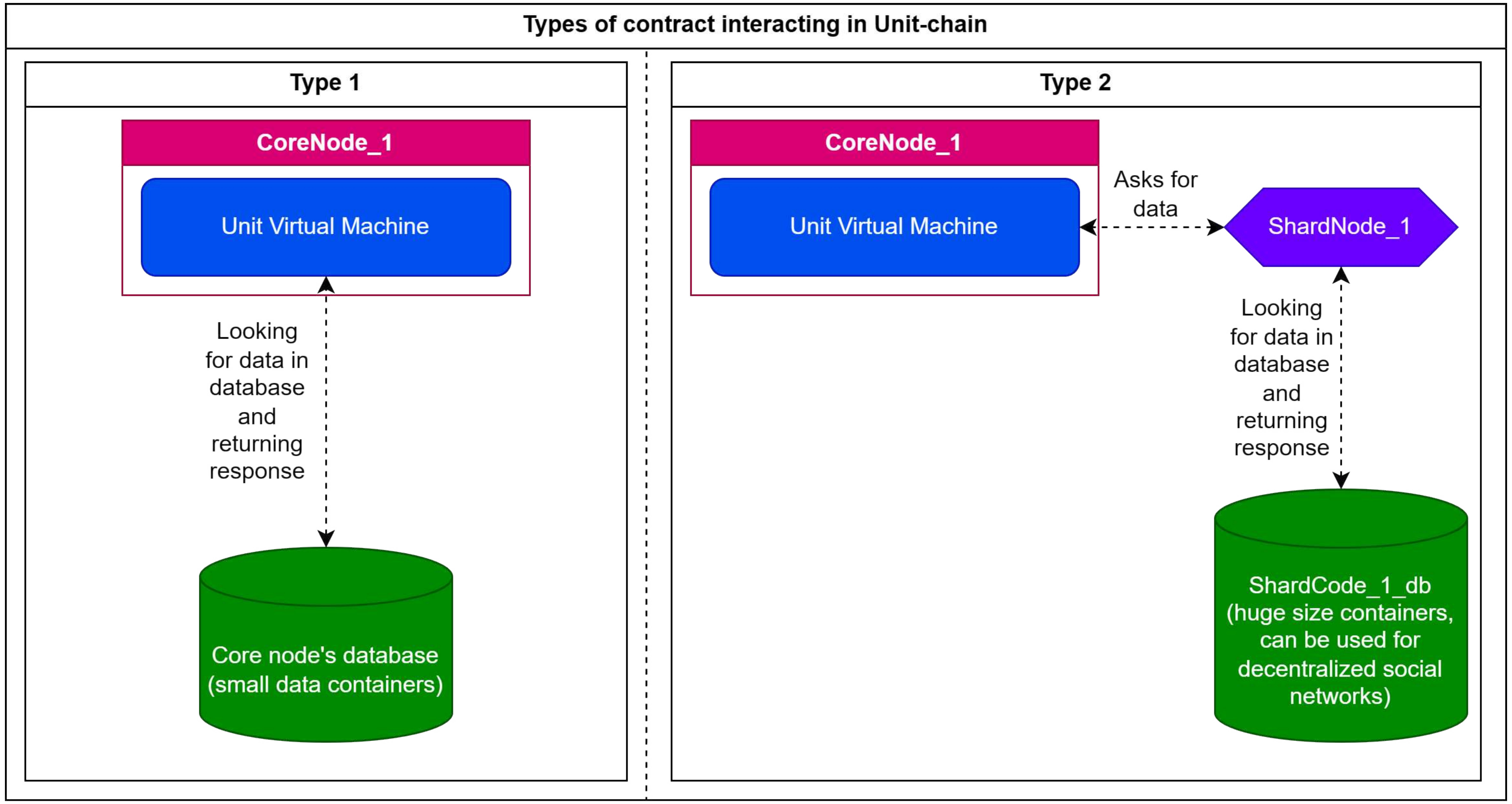


Unit-chain workflow

- Virtual machine have access to Core-node's memory to proceed transactions, inspite of that it's cost is higher than Shard-node's memory.

- It is nesessary to write event handler smart-contract to write data in Shard-node's memory. Event handler must include all logic to handle memory access and contracts should be linked to prevent unwanted access to smart-contract memory.

Unit-chain scheme


Types of contract interacting in Unit-chain
Type 1
CoreNode_1
Unit Virtual Machine
Looking for data in database and returning response
Core node's database (small data containers)
Type 2
CoreNode_1
Unit Virtual Machine
Asks for data
ShardNode_1
Looking for data in database and returning response
ShardCode_1_db (huge size containers, can be used for decentralized social networks)

Data security in Unit-chain

- Unit-chain supports BIP-32, BIP-39 and BIP-44. These protocols allow to create multiple crypto-wallets from only one easy to memory mnemonic phrase.

- Shard-node's data is encrypted with Advanced Encryption Standard (AES), wich provides highest possible secutiry from hacker attacks.