ETHEREUM BEHAVIORAL DESIGN PATTERNS – PROXY DELEGATION WITH ETERNAL STORAGE

Date: 12/05/2018

Brent Anthony Tudas

Sandra Alleine Blanca

Jaymar Dingcong

Robert Aries Dela Paz

Kimberly Mae Reyes

Patrick Oliver Palmero

INSTRUCTIONS

It is not adviceable to deploy this contract since this runs some assembly code that might cause your machine to malfunction.

```
assembly {
  let result := delegatecall(gas, _impl, add(data, 0x20), mload(data), 0, 0)
  let size := returndatasize
  let ptr := mload(0x40)
  returndatacopy(ptr, 0, size)
  switch result
  case 0 { revert(ptr, size) }
  default { return(ptr, size) }
}
```

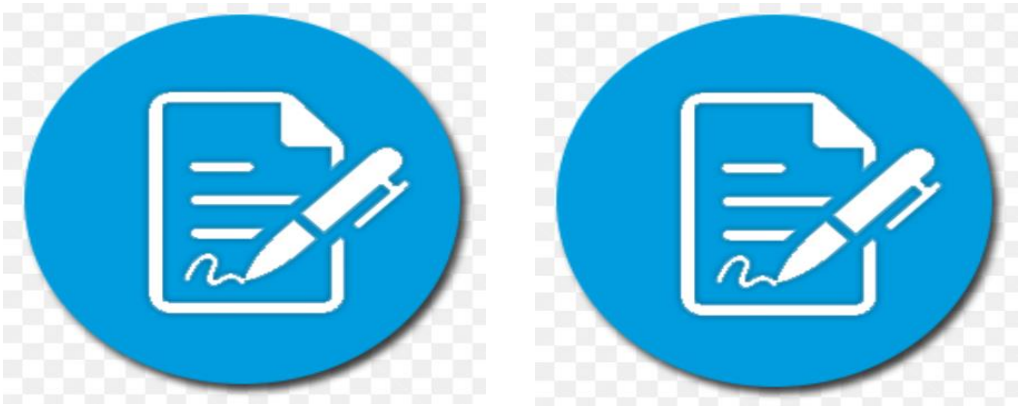
delegatecall(gas, _impl, add(data, 0x20), mload(data), 0, 0);

In above function delegate call is calling code at "_impl" address with the input "add(data,0x20)" and with input memory size "mload(data)",delegate call will return 0 on error and 1 on success and result of the fallback function is whatever will be returned by the called contract function.

Proxy Contract

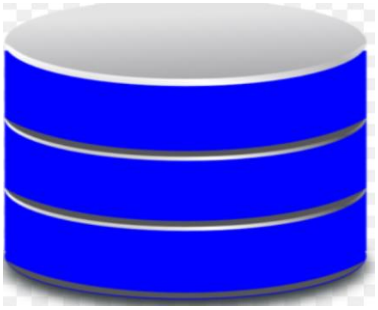
Contains the state which will be modified

Delegate Contracts



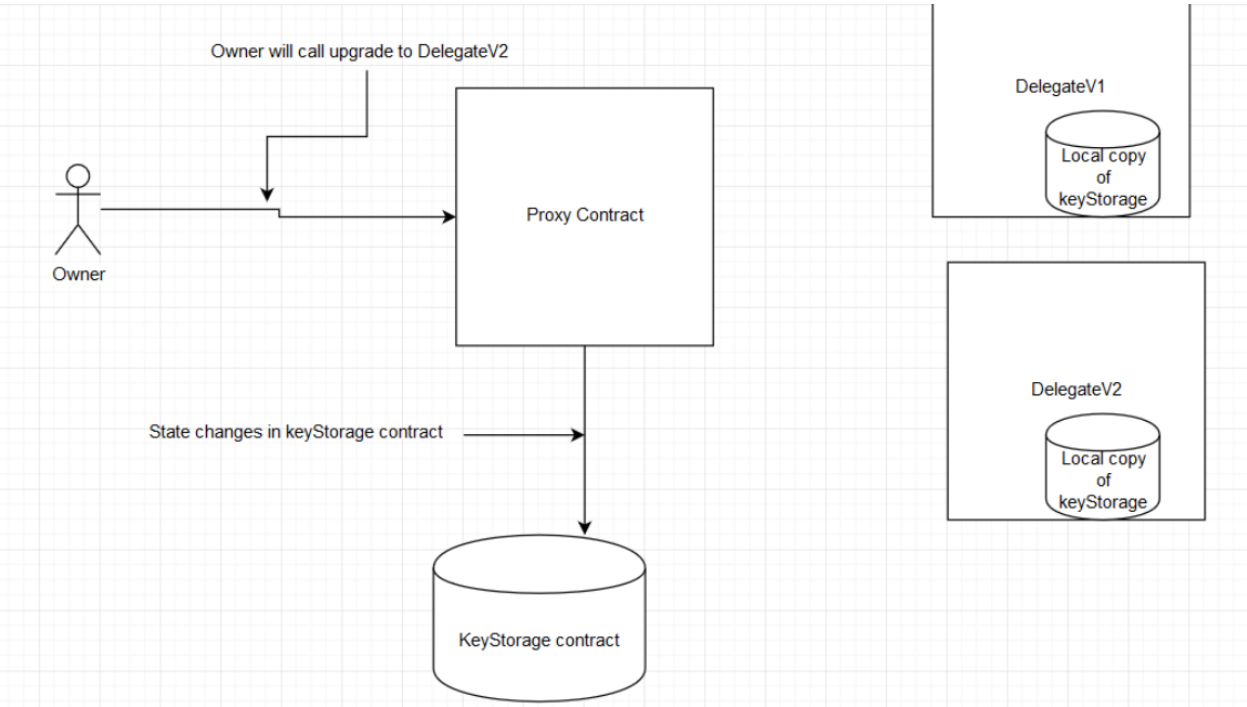
Contains the logic to be followed by the proxy contract. These can be continuously upgraded.

Key Storage Contract

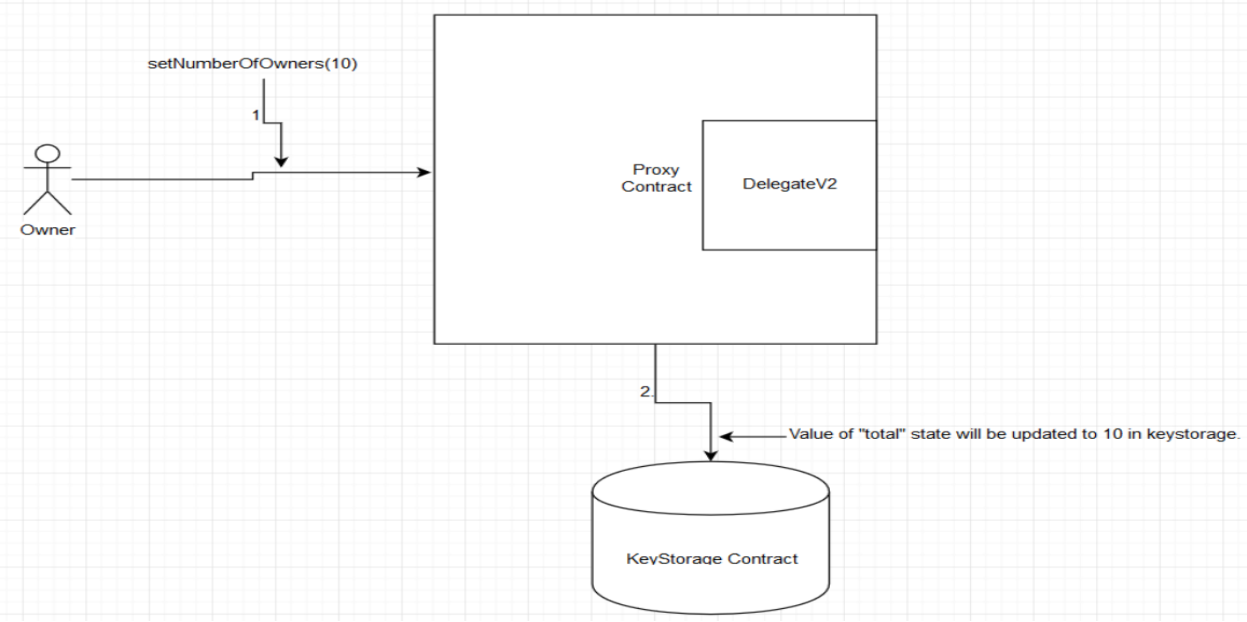


Common storage for all storage state variables which will be shared among all versions of smart contracts.

Workflow Diagram



Call upgrade to DelegateV2



References:

Upgrading Deployed Smart Contracts -

https://medium.com/quillhash/how-to-write-upgradable-smart-contracts-in-solidity-d8f1b95a0e9a/