Lesson 19 - Upgradeability patterns

Patterns overview

- Upgrading the immutable
- The simple Interface Pattern
- The problem with migrating storage
- The problem with logic changes
- Flexible and storage efficient solutions
    - Proxy contracts
    - Unstructured and transparent proxies
    - Storage patterns
    - Diamond patterns
    - Metamorphic contracts

References

https://mirror.xyz/0xB38709B8198d147cc9Ff9C133838a044d78B064B/M7oTptQkBGXxox-tk9VJjL66E1V8BUF0GF79MMK4YG0

https://medium.com/@shub.sharma350/upgradability-patterns-in-solidity-part-1-13e23ce1f144

https://medium.com/@shub.sharma350/upgradability-patterns-in-solidity-part-2-8a2e531d80f8

https://medium.com/@shub.sharma350/upgradability-patterns-in-solidity-part-3-cba09b164497

https://medium.com/@shub.sharma350/upgradability-patterns-in-solidity-part-4-99a2ae29876e

OpenZeppelin plugins and contracts

- Upgrade Plugin
- Upgradeable variants
- Building upgradeable contracts with OpenZeppelin
- Constructor and Initializers
- Upgradeable Libraries
- Creating contracts
- Unsafe operations
- Restrictions

References

https://docs.openzeppelin.com/upgrades

https://docs.openzeppelin.com/learn/upgrading-smart-contracts

https://docs.openzeppelin.com/contracts/4.x/api/proxy

https://docs.openzeppelin.com/upgrades-plugins/1.x/writing-upgradeable https://docs.openzeppelin.com/upgrades-plugins/1.x/proxies

https://docs.openzeppelin.com/contracts/4.x/upgradeable

Implementing a simple contract upgrade

- Running upgrades with hardhat
- Implementing a proxy pattern
- (Review) Delegate calls
- Extending storage
- Changing logic
- Understanding the risks
- Calling updates

References

https://docs.openzeppelin.com/upgrades-plugins/1.x/hardhat-upgrades

Homework

- Read the references
- Try out other upgrade patterns