Project Documentation: Proxy Smart Contract with ProxyAdmin

1. Design Decisions

Proxy Contract (`Proxy.sol`)

Storage Slots

-> Two storage slots are used: one for the admin address and another for mapping function IDs to implementation contract addresses.
-> Storage slots are managed using the `StorageSlot` library.

Admin Functionality

-> The admin is designated during contract deployment and can be changed later by the current admin.
-> Admin is required for functions that modify the implementation contract or change the admin itself.

Implementation Storage

-> Function IDs are mapped to implementation contract addresses to keep track of which contract implements a specific function.
-> This allows for dynamic upgrades without modifying the proxy contract.

Fallback Function

-> The fallback function delegates the call to the corresponding implementation contract based on the received function ID.

Delegatecall

-> The `_delegate` function uses `delegatecall` to execute the implementation contract code in the context of the proxy contract, preserving storage and state.

ProxyAdmin Contract (`ProxyAdmin.sol`)

Ownership
-> The `ProxyAdmin` contract is owned by an address designated during deployment.
-> Only the owner can perform administrative actions on the proxy contracts.

Proxy Interaction
-> ProxyAdmin facilitates changing the admin of a proxy, upgrading the implementation contract, and deleting implementations.

-> Functions in `ProxyAdmin` interact with the respective functions in the `Proxy` contract using delegate calls.

2. Challenges Faced

Writing to Any Slot

-> Writing to arbitrary storage slots poses security risks. The `StorageSlot` library is used to manage storage slots securely.
-> Challenges involved understanding and implementing a secure way to read and write to storage slots dynamically.

Interaction with Implementation Contract from Proxy

-> Interacting with the implementation contract using `delegatecall` requires careful consideration of state variables and function signatures.
-> Ensuring that the implementation contract functions are compatible with the proxy contract's expectations.

3. How Challenges Were Addressed

Writing to Any Slot

-> Extensive research through articles and video tutorials helped understand the risks and best practices for writing to any storage slot.
-> Implementation of the `StorageSlot` library provides a secure way to manage storage slots without exposing vulnerabilities.

Interaction with Implementation Contract

-> Research and development involved trial and error to ensure that the `delegatecall` correctly executes the implementation contract code.
-> Rigorous testing was conducted to verify that state variables and function signatures between the proxy and implementation contracts were aligned.

Conclusion

The proxy smart contract and its admin contract provide a flexible and upgradable architecture. Challenges related to storage slot manipulation and implementation contract interaction were addressed through careful design decisions, research, and thorough testing. The resulting solution enables seamless upgrades and secure proxy contract management.