**Hypothesis 1 Architecture – Asset Bank with Treasury Proxy**

This the architectural design for Hypothesis 1, essentially, since Asset Bank Struct can not hold a generic type (<T>) because it **must be initialised at the deployment of the smart contract (init() function call), therefore this my design hypothesis as follows:**

**Asset Bank Admin can initialise triggers Treasury initialisation**

**Handle User Asset Storage Functionality via Proxy**

**Treasury**

**Stores UID of the treasury, ID of the Asset Bank and Table that store key-value entries of user coin deposits of different types**

**Asset Bank Admin == Treasury Admin Initialiser**

**Call Init()**

**User Sends Receipt<T>**

**Receives**

**Receipt<T>**

**TX**

**Trigger State Updates in the Asset Bank**

**While Asset Bank signs of transfers**

**Accessible via Asset Bank Only (Private)**

**User Sends Coin<T>**

**Deposit()**

**Publicly Accessible (Shared)**

**Asset Bank**

**Stores UID, NFT\_COUNT and DEPOSIT\_COUNT**

**Contract Interface (Public View Functions)**

**TX**

**No Direct Access to Treasury Related Functions**

**NOTES:**

1. ***There are******no direct public interface functions to the treasury from user to treasury in my hypothesised design***
2. ***The Asset Bank is in existence in a 1 to 1 relationship with the Treasury***