3.4 **Address, Keys and Wallets**

**Addresses:**

The fundamental technology used in Ethereum is also the **cryptography** (i.e. public, private key and hashing).

**Two Types of Account in Ethereum:**

1. Externally Owned Account (EOA) (**which contain the user account address**)
2. Contract Account (**CA also contain the address which we say reference or handle**)

The major difference between EOA and Smart Contract

**EOA have the Private key but Smart contract does not have the Private key.**

You need to have a **Private Key** to execute the **transaction** because the transaction is signed by the Private key.

**Ownership of Ether is also established by Private Key.**

That’s why EOA can initiate the transaction but Smart Contract cannot initiate any transaction. (It’s possible after the execution of transaction by EOA the smart contract run another transaction but it doesn’t have any authority through it can run an execution by itself.)

**Wallets:**

A wallet is a software application which provide the user interface to Ethereum.

The wallet control access to:

* User Money
* Managing Keys and address
* Tracking the balance
* Creating and Signing Transaction

Ethereum Wallets can also interact with contracts, such as ERC20 tokens (private currency).

**Wallets also provide the API’s (**js library known as web3 library through which the application become the decentralized and it can communicate with Ethereum blockchain**).**

**Wallet Technology Overview:**

Ethereum wallet is a key chain which contain the pairs of private and public key. Wallet does not store the Ether or Tokens but it is used to validate the transaction.

**3 Types of Wallets:**

1. Non-Deterministic Wallet (JBOK Wallet):

* Also known as Just a bunch of Keys (JBOK)
* Each key i.e. Private key is independently generated from a different random number.
* Keys are not related to each other’s.
* Anonymity is achieved but backup becomes hard

2. Deterministic Wallet (HD Wallet):

* All Private keys are derived from a single master key known as **Seed.**
* Keys are related to each other.
* Can be regenerated from the seed.
* Most common method, Tree like Structure (Hierarchical Deterministic)

3. Hierarchical Deterministic Wallet:

* Most advance form of Deterministic wallet
* Organize the keys more properly like Tree Structure.
* Have the advance functionality about keys like you can set some particular keys to only receive the ether not to send .
* BIPs-39 standardize this approach