# DotID: A Decentralized Identifier Implementation Based on Blockchain

Mempool&DotWallet team
www.dotid.com

**Abstract.** DotID is an implementation of Decentralized Identifier (DID)[1] based on blockchain that complies with W3C standards.[2] It's designed to meet the requirement of information independent storage, identifier distributed generation, identifier distributed management, user privacy anti-leakage and behavior traceability. According to the DotID scheme, a user can generate any number of DID identities and corresponding verifiable credentials (VC)[3]. The system fulfills the requirement of issuance, revocation and verification for DID and VC. Through DotID, we can trace user's behavior while protecting their privacy, which is critical in nowadays identifier system.

## 1. Decentralized Identifier

### 1.1 Introduction to Decentralized Identifier

Decentralized Identifier (DID) represents a verifiable distributed digital identity. This kind of digital identity does not need to be issued by a unique authority, and a user can have multiple verifiable digital identities to implement identifier distributed issuance, distributed verification, distributed revocation and distributed storage.

### 1.2 DID & DID Document

The DID identifier will be parsed into a DID Document through particular entity. Document contains DID identifier and DID detailed information (holder, issuer, issuance time, expiration time, etc.). It is the only explaining document of a single DID.

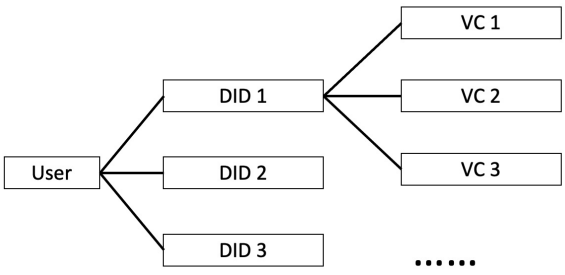
## 2. Verifiable credentials

### 2.1 Introduction to Verifiable Credentials

Verifiable credentials is a specific attribute of credential given to DID based on encryption algorithm and digital signature. Each DID identity can be given a specific attribute by the credential, and its accuracy can be verified by cryptographic methods quickly. The statement in the credential can represent different subjects, such as a driver's license, a degree, or a membership card, etc.

### 2.2 Relationship Between DIDs and Verifiable Credentials

A user can have multiple distributed identities, such as student, programmer, engineer, etc. Correspondingly, an identity can have multiple verifiable credentials, such as a student's identity can have a degree certificate, student ID card or driver license's verifiable credentials.



## 3. Dotid Decentralized Identifier Implementation

### 3.1 DotID DID Unified Format

The format of DotID consists of the following three parts. The **did** field indicates that this string is a decentralized ID, **dotid** indicates that this ID is issued by DotWallet, and **<TXID>** [4] is the unique identifier of this DID. Through the combination of the above three fields, we can determine the globally unique decentralized identifier, and can verify the correctness of the corresponding DID Document from a third party.

*did:dotid: <TXID>*

### 3.2 DID Document Format

DID Document is the only explanation document of a specific DID, and the **id** field needs to contain it. The **subject** field indicates the identity represented by this DID, which can be a student, engineer, teacher, etc. The **authentication** field contains the issuer's DID, issuer's public key and encryption and decryption methods. When verification is required, only the Document and the issuer's signature are sent to the verifier for verification to confirm whether it is legal.

```
{
 "did_document":
      {
        "@context":"https://www.w3.org/ns/did/v1",
        "subject":"example_subject",
        "id":" did:dotid: <TXID>",//user's did
        "authentication":{
                "type":"dotwalletVerificationKey2020",
                "controller":"did:dotid:<TXID>",//issuer's did
                "publicKey":" example_pubkey"
               }
       }
 }
```

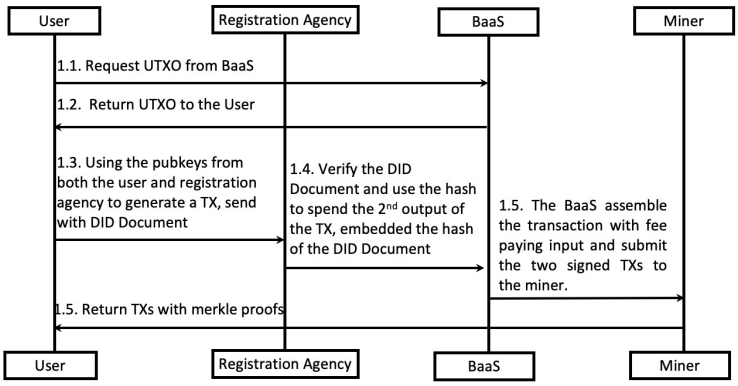
### 3.3 Why Do We Design Like This?

In order to meet the issuable, verifiable, and revokable requirements of a DID system, we use TXID as the identifier of the DID. Using TXID as an identifier has the following advantages:

1). TXID is globally unique, cannot be tampered but easy to trace
2). TXID determines the time of a DID creation naturally
3). User can change the public key without changing the identity
4). DID state can be determined by UTXO state quickly

### 3.4 Issuance of DID and DID Document

The user will provide a public key as the necessary information to generate the DID. At first user need to get UTXO from BaaS(Blockchain As A Server)[5], after providing the necessary identity information to the Registration Agency (Certificate Authority, CA), the CA will generate the first transaction TX, which consists of two outputs. The first output is composed of the CA and user's multi-signature, which is used as the proof of whether the DID is revoked. If this output is spent, it proves that this DID has been revoked by the user or CA and is no longer valid. The other output is the P2PKH of the CA public key encrypted script. After establishing a TX that meets this requirement, we will employ this TXID as the identifier of DID. Then generate a complete DID Document according to the information provided by the user, spend the second output of the first TX and insert the hash value of the DID Document into the output UTXO script. The procedure of DotID system is represented below.




The struct of two TXs is represented below.

| TXID 1 | |
|---|---|
| Vin from BaaS | vout0 |
| | OP_DUP OP_HASH160 <pubKeyHashUser> OP_EQUAL OP_IF OP_ELSE OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_ENDIF OP_CHECKSIG |
| | vout1 |
| | OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_CHECKSIG |

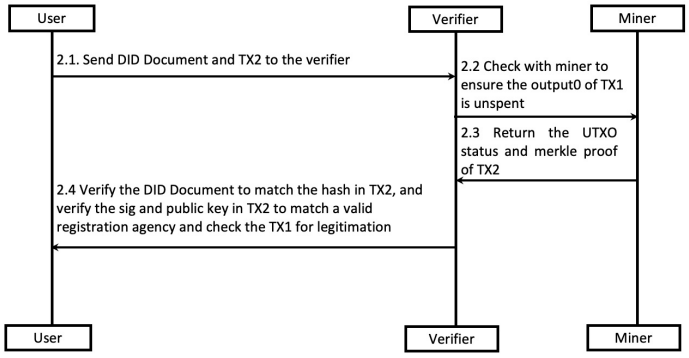
| TXID 2 | |
|---|---|
| TX1 Vout 1 | vout0 |
| | OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_CHECKSIG <DID DOCUMENT HASH> OP_DROP |

### 3.5 The Revocation of DID and DID Document

If the CA or the user would like to revoke an identifier, they only need to spend vout0 on TX1.

### 3.6 The Verification of DID and DID Document

The user submits TXID2 and the DID Document containing the CA signature to the verifier. The verifier will query TX1 based on the input of TX2, and check whether the output of TX1 is spent, if it is, it proves that the DID has been revoked by the user or CA, otherwise, the verifier will use CA public key on TX2 to verify the signature. If the verification result is true, the DID is proved to be valid. The procedure of DotID verification is represented below.




## 4. DotID Verifiable Credential Implementation

### 4.1 Verifiable Credential Implementation

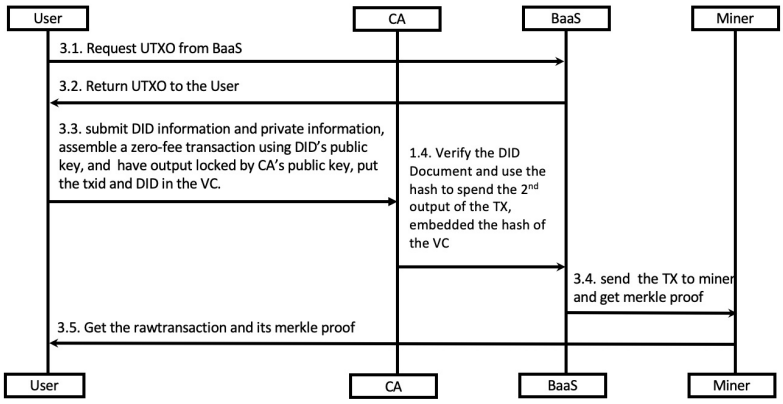
The VC assigns a certain attribute to a DID, and the **issuer** field represents the DID of the VC issuer. The **credentialSubject** field indicates the purpose of this VC, here we use coupons as an example. The **generated_txid** field represents the first TXID(TXID1) of two TXs in the complete steps of this VC on

the chain. This field is added to prevent users from generating multiple similar transactions but not submitting them to CA for registration, causing the verifier to be unable to confirm whether the VC has been revoked. The proof field contains the verified public key, signature and other cryptographic content, which is used to verify the correctness of this VC.

```
{
    "vc":
        {
            "@context":"https://www.w3.org/2018/credentials/v1",
            "type":["example_type"],
            "issuer":"did:dotid:<TXID>",//issuer's DID
            "generated_txid":"<TXID>"//first step's TXID
            "issuanceDate":"2020-01-01T19:73:24Z",
            "expirationDate":"2021-01-01T19:23:24Z",
            "credentialSubject":
                {
                    "example_member_card":
                        {
                            "holder":"did:dotid:<TXID>",//user's DID
                            "useage":"30% discount"
                        }
                },
            "proof":{
                "type":"example_ECDH",
                "created":"2020-01-01T19:73:24Z",
                "proofPurpose":"verify_example_member_card",
                "verificationMethod":"did:dotid:txid",//issuer's DID
                "signature":"example_sig" }
        }
}
```

### 4.2 Issuance of Verifiable Credentials

Similar to meet the issuable, verifiable, and revokable requirements like DID, we use similar UTXO structures to generate VC. The user will provide a public key as the necessary information to generate the VC. After providing the necessary identifier information to the CA, and then the CA will verify whether the user's DID is valid through the DID verification step. If it is valid, the first TX will be generated, which is composed of two outputs, the first output is a multi-signature composed of the public key of the CA and the public key of the user, which serves as the proof of whether the VC is revoked. If this output is spent, it proves that this VC has been revoked by the user or CA and is no longer valid. The other output is the P2PKH of the CA public key encrypted script. After establishing a TX that meets this requirement, CA will generate a complete VC document, which fill TXID1 in the generated_id field to bind this VC on-chain transaction. After that, spend the second output of the first TX and insert the hash value of VC in the output UTXO script. The procedure of DotID verifiable credentials system is represented below.



The struct of two TXs is represented below.

| TXID 1 | |
|---|---|
| Vin from BaaS | vout0 |
| | OP_DUP OP_HASH160 <pubKeyHashUser> OP_EQUAL OP_IF OP_ELSE OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_ENDIF OP_CHECKSIG |
| | vout1 |
| | OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_CHECKSIG |

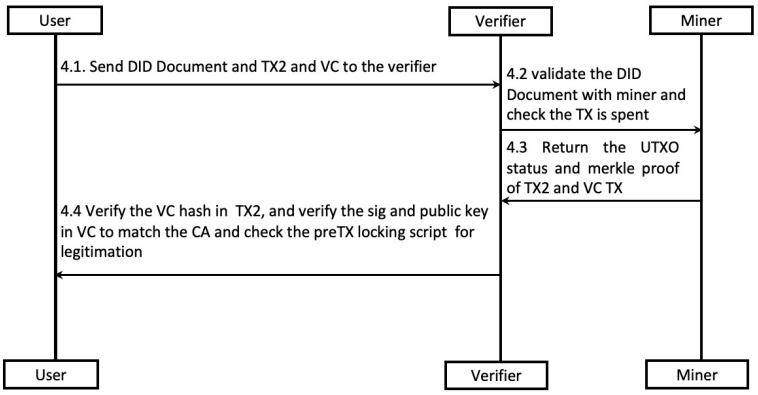
| TXID 2 | |
|---|---|
| TX1 Vout 1 | vout0 |
| | OP_DUP OP_HASH160 <pubKeyHashCA> OP_EQUALVERIFY OP_CHECKSIG <VC HASH> OP_DROP |

### 4.4 The Revocation of VC

If the CA or the user would like to revoke a VC, they only need to spend vout0 on TX1.

### 4.5 The Verification of VC

The user submits TX2/VC/DID Document to the verifier and the verifier will check DID's legitimation through DID verification procedure described above. At this time, the verifier needs to compare whether the generate_id on the VC is consistent with TXID1, if they are consistent, it can be determined that the TX is indeed the TX on the CA chain. After that, check whether the signature on the VC matches the public key on TXID2 and TX1's vout0 is unspent, if it is, we can confirm the validity of the VC according to the matching result.




## 5. Conclusion

We proposed a decentralized identifier solution named DotID. Using this protocol to establish decentralized identifier and verifiable credentials can implement the requirement of distributed issuance, distributed revocation, and distributed verification for identities. At the same time, both the DID Document and VC will be saved by the user, and all the information on blockchain is digital digest. The characteristics of the blockchain ensure that the data on the chain cannot be tampered with while protecting user's privacy. The user will only need to give the material to prove his identity whenever is necessary, and the verifier can verify the legality of it quickly.

## References


[1] Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M. and Holt, J., 2020. Decentralized identifiers (dids) v1. 0. Draft Community Group Report.

[2] https://en.wikipedia.org/wiki/World_Wide_Web_Consortium

[3] Arenas, R. and Fernandez, P., 2018, June. CredenceLedger: a permissioned blockchain for verifiable academic credentials. In 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) (pp. 1-6). IEEE.

[4] https://wiki.bitcoinsv.io/index.php/TXID

[5] https://en.wikipedia.org/wiki/Blockchain_as_a_service