BTCR DID Resolver Specification

A DID Update from Rebooting the Web of Trust VI

By Kim Hamilton Duffy, Christopher Allen, Ryan Grant, and Dan Pape

This describes the process of resolving a BTCR DID into a DID Document. The draft reference implementation is available at https://github.com/WebOfTrustInfo/btcr-did-tools-js (see didFormatter.js). Note that not all steps described in this document are implemented yet.

See the BTCR playground for a live demonstration. The BTCR playground uses the draft reference implementation BTCR DID resolver.

INPUT: BTCR DID

The input to a BTCR DID resolver is a BTCR DID.

```
Format:  did:btcr:<specific-idstring>
Example: did:btcr:xkyt-fzgq-qq87-xnhn
```

RESOLUTION PHASE 1: CONSTRUCT "IMPLICIT" DID DOCUMENT

Terminology

- "Extended transaction reference": refers to our specific transaction reference customizations for the BTCR DID method spec Issue #1
- "txref-ext": abbreviation for above
- "Constructed" DID Document: what the resolver generates
- "Continuation" DID Document: a referenced DID document to be merged into the constructed DID document
  ⇨ spec

Goal

This phase constructs the "implicit" DID Document from Bitcoin transaction data.

Steps

0. Confirm `method` from the DID is `btcr`. Fail if not
1. From the BTCR DID, extract the extended transaction reference: this is the `<specific-idstring>` component of `did:btcr:<specific-idstring>` = `did:btcr:<TXREF-EXT(TX)>`
2. Extract transaction details from the txref-ext encoding:
   - txref-ext encodes these transaction details:
     - bitcoin network (mainnet, testnet, ..)
     - the transaction block height and position
   - Example: in the BTCR Playground note that `did:btcr:xkyt-fzgq-qq87-xnhn` resolves to:
     - network = testnet3
     - transaction id = 67c0ee676221d9e0e08b98a55a8bf8add9cba854f13dda393e38ffa1b982b833
     - blockheight = 1201739, position = 2
   - Reference implementation: https://github.com/WebOfTrustInfo/txref-conversion-js
     - Note that txref-ext deviates from txrefs Issue #1
       - The most significant difference at the moment is that the network prefix is removed. So for example, a txref of `txtest1-xkyt-fzgq-qq87-xnhn` converts to a txref-ext of `xkyt-fzgq-qq87-xnhn`.
       - For now, calling libraries handle this conversion by adding back the txref prefix. In the btcr-did-tools library, see util.ensureTxref
3. Look up transaction by height and position. Is the transaction output spent?
   - no: this is the latest version of the DID. From this we can construct the DID Document
   - yes: keep following transaction chain until the latest with an unspent output is found
4. Extract the hex-encoded public key that signed the transaction and update the DID document with default authentication capability
   - Populate the first entry of the `publicKey` array in the DID document.
     - The `id` will have a fragment of `#keys-1`, so that the full `id` is `did:btcr:<specific-idstring>#keys-1`. This is a BTCR method spec convention that `#keys-1` corresponds to the transaction signing key. We'll see in the next section that overriding this path in the supplementary DID document data is not allowed
     - Encode the key material according to the Koblitz Elliptic Curve Signature 2016 signature suite. Issue #5
   - Populate the first entry of the `authentication` array in the DID document, referencing the key above
   - Alternate representation note: a public key can be inlined if there is only one reference in the DID document (as opposed to the representation above, in which there is a `publicKey` array and a reference from `authentication`)
5. If the transaction contains an `OP_RETURN` field, populate the `serviceEndpoint` in the DID document. This is assumed to reference supplementary DID document data
   - Add an entry to the `service` section of the DID document
     - `type` is `BTCREndpoint`

- serviceEndpoint is the value in the OP_RETURN field, e.g. "https://github.com/myopreturnpointer"

6. Add SatoshiAuditTrail, which contains additional metadata available from the Bitcoin transaction.
   ➾ This is still being defined; Issue #3

If the transaction contained no OP_RETURN data (and therefore no serviceEndpoint was added), the resolution process is done. Otherwise, proceed to phase 2.

Output of Phase 1

The output of this resolution phase is referred to as the "implicit" DID Document; it is derived exclusively from Bitcoin transaction data.

If the transaction has no OP_RETURN data, then the service array would have no entries. The only default capabilities would be to authenticate with the transaction signing key.

Example: in the BTCR Playground note that did:btcr:xkyt-fzgq-qq87-xnhn Phase 1 output is:

```
{
    "@context": "https://w3id.org/btcr/v1",
    "id": "did:btcr:xkyt-fzgq-qq87-xnhn",
    "publicKey": [
        {
            "id": "did:btcr:xkyt-fzgq-qq87-xnhn#keys-1",
            "owner": "did:btcr:xkyt-fzgq-qq87-xnhn",
            "type": "EdDsaSAPublicKeySecp256k1",
            "publicKeyHex":
"0280e0b456b9e97eecb8028215664c5b99ffa79628b60798edd9d562c6db1e4f85"
        }
    ],
    "authentication": [
        {
            "type": "EdDsaSAPublicKeySecp256k1Authentication",
            "publicKey": "#keys-1"
        }
    ],
    "service": [
        {
            "type": "BTCREndpoint",
            "serviceEndpoint":
"https://raw.githubusercontent.com/kimdhamilton/did/master/ddo.jsonld"
        }
    ],
    "SatoshiAuditTrail": [
        {
            "chain": "testnet",
            "blockhash":
"0000000000000722ded9d85d67e145ba41c53ef2e8680f75540a08b885febba5",
            "blockindex": 2,
            "outputindex": 1,
            "blocktime": "2017-09-23T17:27:56.682Z",
            "time": 1499501000,
            "timereceived": "2017-09-23T17:27:56.682Z",
            "burn-fee": -0.05
        }
    ]
}
```

RESOLUTION PHASE 2: POPULATE DID DOCUMENT WITH SUPPLEMENTARY DID DOCUMENT DATA

Steps

7. Retrieve the continuation DID document from `serviceEndpoint.BTCREndpoint` and extract the portions of `type` "DIDDocument". Issue #6
   - If URL doesn't exist, ERROR
8. Verify the continuation DID Document Issue #2
   - If the content is in an immutable store:
     - full `ids` are not required (but a fragment is? -- Issue #2)
     - a signature is not required
   - Otherwise:
     - `ids` must be fully specified
     - signature is required
     - resolver must check signature
9. Merge in continuation DID document entries (keys, authorizations, etc -- as appropriate) into the constructed DID document. Additive only!
   - Merge items that are part of the DID specification (`publicKey`, `authentication`, `service`) into the constructed DID document by appending their entries to the arrays of the matching term
   - If any new `ids` are already used in the constructed DID document, ERROR
   - For immutable stores, merge `id` fragments with the DID value into the constructed DID document
   - Append unknown terms to the constructed DID Document Issue #6
10. Repeat steps 7-9 for additional referenced continuation DID documents
    - Issue #4
11. Proposed but not shown here: wrap the DID document in resolver envelope with additional metadata

Output of Phase 2

This resolution phase returns a final constructed JSON-LD DID Document to caller, which can use the keys to authenticate data such as the signature on a verifiable claim, or perform other application tasks.

Let's assume the supplementary DID document (from the OP_RETURN data) is stored in an immutable store and contains the following `didDocument`.

```
{
  ...
  "didDocument": {
      "@context": "https://w3id.org/did/v1",
      "publicKey": [{
        "id": "#keys-2",
        "type": "RsaVerificationKey2018",
        "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
      }],
      "authentication": [{
        "type": "RsaSignatureAuthentication2018",
        "publicKey": "#keys-2"
      }],
  ...
}
```

Note that the **id** is not known yet, because the transaction referencing this supplementary document has not occurred.

Example: in the BTCR Playground the final output for `did:btcr:xkyt-fzgq-qq87-xnhn` is:

```
{
      "@context": "https://w3id.org/did/v1",
      "id": "did:btcr:xkyt-fzgq-qq87-xnhn",
      "publicKey": [
        {
            "id": "did:btcr:xkyt-fzgq-qq87-xnhn#keys-1",
            "owner": "did:btcr:xkyt-fzgq-qq87-xnhn",
            "type": "EdDsaSAPublicKeySecp256k1",
            "publicKeyHex":
"0280e0b456b9e97eecb8028215664c5b99ffa79628b60798edd9d562c6db1e4f85"
        },
        {
            "id": "did:btcr:xkyt-fzgq-qq87-xnhn#keys-2",
            "type": "RsaVerificationKey2018",
            "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n",
            "owner": "did:btcr:xkyt-fzgq-qq87-xnhn"
        }
      ],
      "authentication": [
        {
            "type": "EdDsaSAPublicKeySecp256k1Authentication",
            "publicKey": "#keys-1"
        },
        {
            "type": "RsaSignatureAuthentication2018",
            "publicKey": "#keys-2"
        }
      ],
      "service": [
         {
             "type": "BTCREndpoint",
              "serviceEndpoint":
"https://raw.githubusercontent.com/kimdhamilton/did/master/ddo.jsonld"
         }
      ],
      "SatoshiAuditTrail": [
        {
            "chain": "testnet",
            "blockhash":
"0000000000000722ded9d85d67e145ba41c53ef2e8680f75540a08b885febba5",
            "blockindex": 2,
            "outputindex": 1,
            "blocktime": "2017-09-23T17:27:56.682Z",
            "time": 1499501000,
            "timereceived": "2017-09-23T17:27:56.682Z",
            "burn-fee": -0.05
        }
      ]
```

ADDITIONAL CREDITS

Lead Author: Kim Hamilton Duffy

Other Authors: Christopher Allen, Ryan Grant, and Dan Pape

About Rebooting the Web of Trust

This paper was produced as part of the Rebooting the Web of Trust VI design workshop. On March 6th to 8th, 2018, over 40 tech visionaries came together in Santa Barbara, California to talk about the future of decentralized trust on the internet with the goal of writing 3-5 white papers and specs. This is one of them.

Named Sponsors List: Sovrin Foundation, PTB Holdings

Workshop Credits: Christopher Allen (Founder), Joe Andrieu, PMP (Producer and Facilitator), Shannon Appelcline (Editor-in-chief), Erica Connell (Event Coordinator), Claire Rumore (Graphical Recorder), and The Narrative Loft (Venue)

Thanks to our other contributors and sponsors!

What's Next?

The design workshop and this paper are just starting points for Rebooting the Web of Trust. If you have any comments, thoughts, or expansions on this paper, please post them to our GitHub issues page:

https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2018/issues

The next Rebooting the Web of Trust design workshop is scheduled for for Fall 2018. If you'd like to be involved or would like to help sponsor these events, email:

> ChristopherA@LifeWithAlacrity.com