nostr

Notes and Other Stuff Transmitted by Relays

Basic protocol flow description
This NIP defines the basic protocol that should be implemented by everybody. New NIPs may add new optional (or mandatory) fields and messages and features to the structures and flows described here.

Events and signatures
Each user has a keypair. Signatures, public key, and encodings are done according to the Schnorr signatures standard for the curve secp256k1.

The only object type that exists is the event, which has the following format on the wire:

```
{
  "id": <32-bytes lowercase hex-encoded sha256 of the serialized event data>,
  "pubkey": <32-bytes lowercase hex-encoded public key of the event creator>,
  "created_at": <unix timestamp in seconds>,
  "kind": <integer between 0 and 65535>,
  "tags": [
    [<arbitrary string>...],
    // ...
  ],
  "content": <arbitrary string>,
  "sig": <64-bytes lowercase hex of the signature of the sha256 hash of the serialized event data, which is the same as the "id"
field>
}
```

To obtain the `event.id`, we `sha256` the serialized event. The serialization is done over the UTF-8 JSON-serialized string (which is described below) of the following structure:

```
[
  0,
  <pubkey, as a lowercase hex string>,
  <created_at, as a number>,
  <kind, as a number>,
  <tags, as an array of arrays of non-null strings>,
  <content, as a string>
]
```

To prevent implementation differences from creating a different event ID for the same event, the following rules MUST be followed while serializing:

- UTF-8 should be used for encoding.
- Whitespace, line breaks or other unnecessary formatting should not be included in the output JSON.
- The following characters in the content field must be escaped as shown, and all other characters must be included verbatim:
  - A line break (`0x0A`), use `\n`
  - A double quote (`0x22`), use `\"`
  - A backslash (`0x5C`), use `\\`
  - A carriage return (`0x0D`), use `\r`
  - A tab character (`0x09`), use `\t`
  - A backspace, (`0x08`), use `\b`
  - A form feed, (`0x0C`), use `\f`