

# OpenCoin

---

*a protocol for privacy preserving electronic cash payments*

Version: 0.4 - draft (July 2022)

Copyright (2022) J. Baach, N. Toedtmann

This version of the protocol build on previous work by the following authors:

Jörg Baach

Nils Toedtmann

J. K. Muennich

M. Ryden

J. Suhr



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

For more information go to <https://opencoin.org>

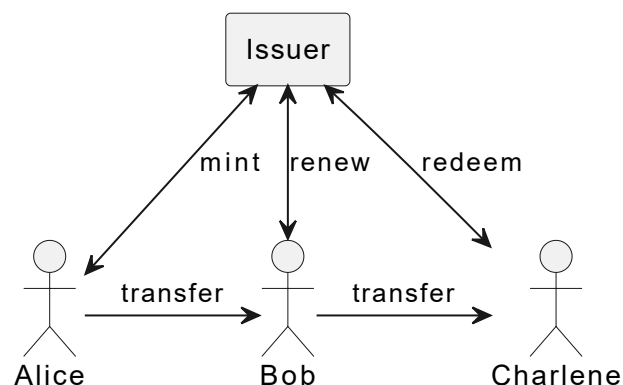
# Intro

---

We propose a protocol that allows cash-like payments in the electronic world. It is based on the invention by David Chaum<sup>1</sup>. The main focus are *untraceable* payments, which means that even though there is a central entity (called the issuer, something like a bank), this central entity can't see the transactions happening. This is good for the privacy of the users.

The focus of the project is the protocol. This means we standardize the way we exchange messages and their content in order to make electronic cash payments. But we don't deliver an implementation here. That is the scope of other project(s). OpenCoin is the foundation to build upon.

## How does it work?



This is a high level (but strongly simplified) image describing the basic system. We have three participants: Alice and Bob are normal users, while the Issuer is something like a bank, capable of minting coins. It also acts as an exchange for 'real-world' currency. At this high level it works as follows:

1. Alice asks the Issuer to **mint** coins. This is done in a special way using *blind signatures*, which means that the coins *can't be linked to her* later on.
2. Alice then **transfers** the coins to Bob. She can do that any way she wants, e.g. using WhatsApp, Email or any other system of her choice (also depending on what her client software supports). This could even be done by printing the coins and handing them over.
3. Bob then **renews** the coins. He swaps the coins he got from Alice for fresh coins. This way he protects himself against Alice "accidentally" using the coins

somewhere else. One can spend opencoins only once, so *double spending* needs to be ruled out, and this is done by immediately renewing received coins.

4. Bob might **transfer** the coins to yet another person, Charlene
5. Charlene decides to **redeem** the coins, meaning she asks the issuer to swap the opencoins for real-world money.

On **blind signatures**: at the core a coin is a serial number with a signature from the mint. In order to ensure that *a coin can't be traced back to the original client* we use blind signatures.

Imagine Bob hands in a coin that Alice had minted. In order to ensure the coin can't be traced back to Alice, the issuer has to sign the serial number without seeing it. In non-technical terms Alice puts the serial number in an envelope (along with carbon copy paper), and the issuer actually signs the envelope. Because of the carbon copy paper the signature presses through onto the serial number. Alice can then open up the envelope and has a signed serial, without the issuer ever seeing it.

## Who is it for?

OpenCoin (the protocol) allows the development of applications for electronic cash. So firstly OpenCoin is targeted at developers. These applications however should allow everyone to make and receive electronic payments. It still requires somebody to run the central issuer. This issuer would issue an OpenCoin based electronic money system. Because electronic money is quite regulated in Europe (and other countries), the issuer would be most likely a regulated electronic money provider or a bank. We think, that a central bank would be the best issuer, because central banks issue money anyhow. But nothing technical stops you from using OpenCoin for your private project <sup>2</sup>.

## Alternatives

Why don't just use one of the alternatives?

### Bitcoin / blockchain

Bitcoin (or blockchain in the more general form) is basically the opposite of OpenCoin: transfers have to happen within the system, they are visible to everybody, there is no central instance, there is no guaranteed value you can redeem the bitcoins for.

OpenCoin on the contrary makes the transfers invisible and untraceable, and has a central instance that is able to guarantee a value if you redeem the OpenCoin.

One could say that bitcoin behaves more like gold, while OpenCoin behaves more like cash.

## GNU Taler

[GNU Taler](#) is build around the same central idea as OpenCoin. It started later, and is more complete than OpenCoin. They differ in the way the take care of the [renewal step](#) and coin splitting. They also make more assumptions regarding the clients (e.g. clients having key identifying them), they have clearer roles (e.g. consumer and merchant) and by all of this hope to get around the inherent problems of untraceable transfers, e.g. taxability.

The trade-off seems to be that their system is harder is more complex and harder to understand. We also doubt that this complexities are necessary to reach the stated goals. We also doubt that the goals can really be reached, and also find that the systems documentation is quite hard to understand. This might be because they deliver implementations for all necessary software components, and are not really targeted at other implementations of they system.

Because of all this one could say that GNU Taler is less open to other developers.

# The OpenCoin protocol

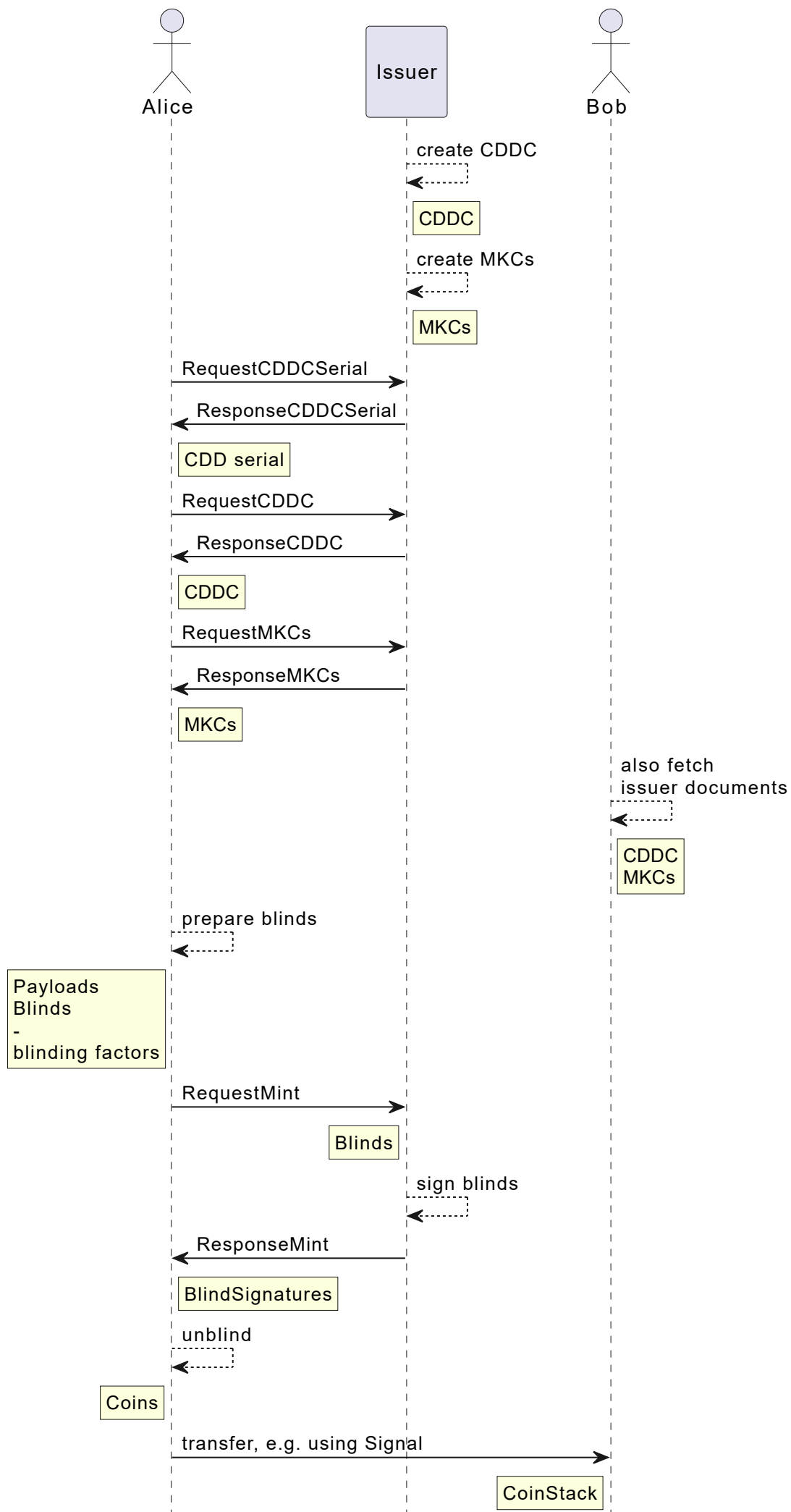
---

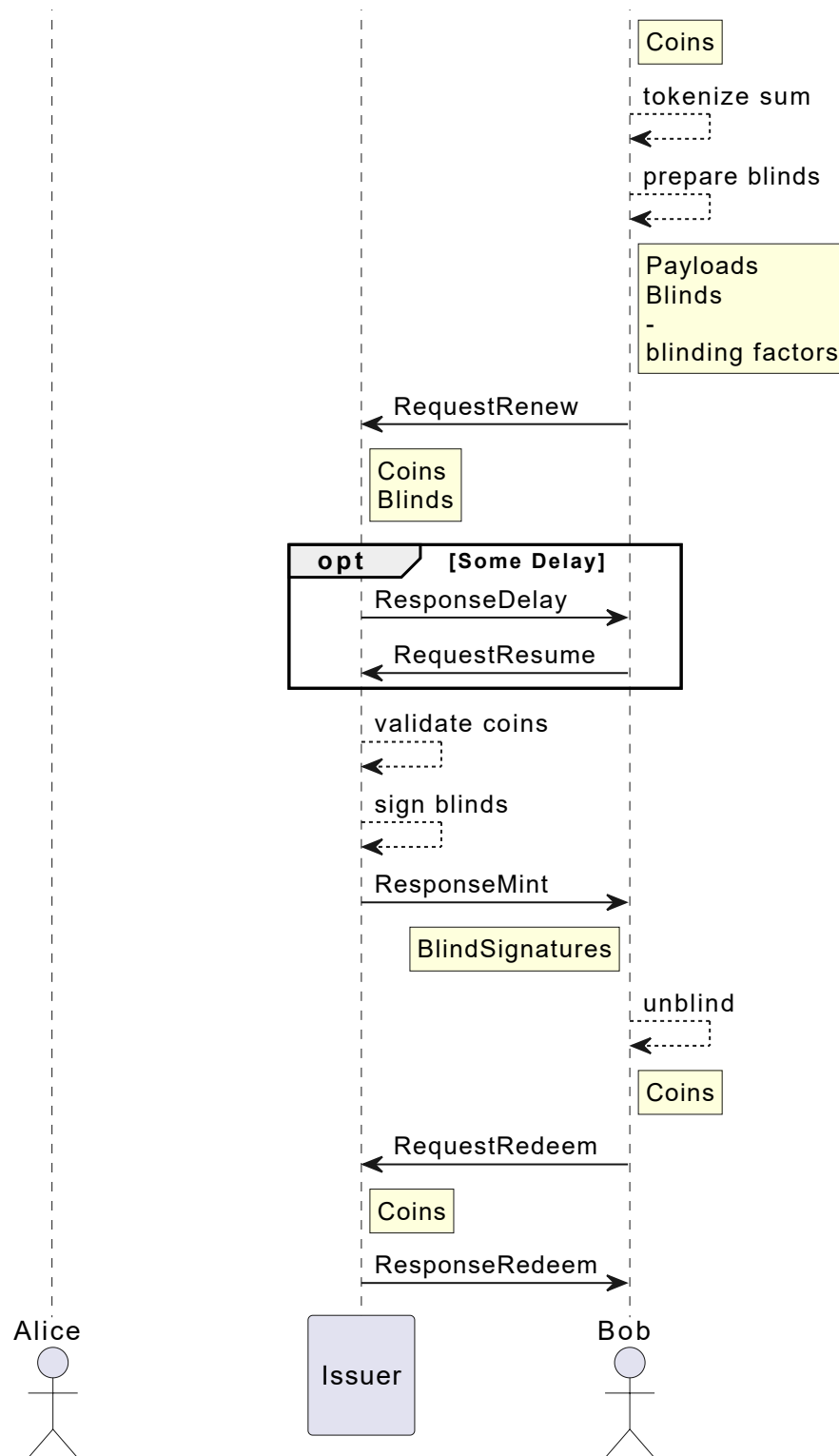
## Assumptions

The exchange of messages **MUST** happen over a **secure channel**. For HTTPS this means [TLS](#), but other channels like messengers provide their own. For an email exchange [GPG](#) would be recommended. Either way, it is the responsibility of the developer to take care of the transport security.

When requesting the issuer to mint or redeem coins some form of **authentication & authorization** is most likely required - the issuer needs to secure payment for the coins, or make a payment somewhere for redeemed coins. Because auth\* might already be provided by the transport layer, we don't include it in the OpenCoin protocol.

## Overview





## Description

This is a high level description of the actual steps, details follow in the chapters in [Details](#).

## Participants

**Issuer** can mint, refresh and redeem coins. This entity will probably have an account handling system (a.k.a bank) behind it for doing actual real-world payments. The issuer is trusted to handle coins and payments correctly, but is *not trusted* in regards of privacy - the target of OpenCoin is to protect the privacy of the transfers.

**Alice** and **Bob** are clients using OpenCoin. Technically they are software clients, and they represent the *users* of the system.<sup>3</sup> They need to be known by the customer in order to mint or redeem coins. Authentication could be required to renew coin. This would allow a "closed" system, in which accounts of the users could be monitored.

## Steps

### create CDDC

The issuer creates a pair of cryptographic keys (the currency keys), and signs a *Currency Description Document Certificate (CDDC)* with its secret key. This contains information about the currency, like denominations, urls but also the public key. This is the top document which establishes the trust in all other elements.

Not mentioned in the CDDC but probably somewhere on the issuer website is the relation between opencoins and actual real-world money. Let's say the currency of an example issuer is called "opencent". The rule might be that one opencent is given out for one EUR cent, and redeemed for one EUR cent, effectively binding the opencent to the EUR.

### create MKCs

For each denomination in the currency separate minting keys are generated, and a *Mint Key Certificate (MKC)* for them as well. Those MKCs are signed the secret currency key. The mint keys are only valid for a defined period of time.<sup>4</sup>

### RequestCDDCSerial

This message asks for the current serial number of the CDDC. The currency description could change over time, maybe because urls have changed. Every time a new CDDC is created, with a new, increasing serial number. The clients need to make sure to always use the most current CDDC, but they can cache it, allowing them to skip the next step.



## ResponseCDDCSerial

This message contains the **CDDC serial**.

## RequestCDDC

This message asks for a CDDC. If no serial is provided, the message asks for the most current CDDC.

## ResponseCDDC

This message contains the **CDDC**

## RequestMKCs

With this message the client asks for the *Mint Key Certificates*. The client can specify specific denominations or *mint key ids*. An unspecified request will return all current MKCs.

## ResponseMKCs

This reply contains the **MKCs**

## prepare blinds

This step prepares a coin. In essence this is a **payload** with a serial number, which is later on signed by the issuer using a denomination specific mint key. The "envelope" [mentioned above](#) really means that the serial is blinded using a separate random secret **blinding factor** for each serial number. This factor is needed later on to "open up the envelope", reversing the blinding operation. Hence the client has to store the blinding factor for later on. As the blinding factor is individual for each serial number, a reference number is created to reference serial, blinding factor and blind.

The **blinds** contains the reference, the blind to be signed, and the mint key id for the denomination or value of the coin.

## RequestMint

This message hands in the **blinds** created in the step before, asking for the blind to be signed.

Most likely the issuer has authenticated the client. The mint key id tells the issuer what denomination to use for the signing operation. This will allow the issuer to deduct a payment for the minting operation (outside OpenCoin).

The message also carries a `transaction_reference` (a random number), in case there is a delay in the minting process. The client can then later on ask again for the signatures to be delivered using the same `transaction_reference`.

## sign blinds

The issuer uses the secret minting key for the desired operation to sign the blind, creating the **blind signatures**.

## ResponseMint

This message contains the **blind signatures** for the blinds.

## unblind

The client will unblind the signature using the before stored secret blinding factor. This gives the client the signature for the serial number, and both together give the **coin**.

## CoinStack

When sending coins multiple coins can be combined into a **CoinStack**. This CoinStack can also have a "subject", maybe containing an order reference - the reason the CoinStack is handed over in the first place.

The transfer of the CoinStack is out of scope of the OpenCoin protocol. We imagine multiple ways: using a messenger like Signal, using email or using the Browser. A CoinStack can also be encoded using a QR code, and maybe printed out and sent using normal postal mail.

Anyhow, the point of this step is that Alice transfers a CoinStack to Bob. And because she is a fair user, she will delete all coins that were contained in the CoinStack on her side.

**tokenize sum**

**prepare blinds**

**RequestRenew**

**ResponseDelay**

**RequestResume**

**validate coins**

**RequestRedeem**

**ResponseRedeem**

**Problems**

- Tax
- Money laundering
- Blackmail and other crime

# Details

---

## Cryptographic operations

- hashes
- signatures

## Building blocks

Elements of messages, but never used standalone

## CDDC

### Description

FOO	BAR	BLA
1	two	three

### Example

```
1 {
2   "cdd": {
3     "additional_info": "",
4     "cdd_expiry_date": "2023-07-08T15:32:04.103469",
5     "cdd_location": "https://opencent.org",
6     "cdd_serial": 1,
7     "cdd_signing_date": "2022-07-08T15:32:04.103469",
8     "currency_divisor": 100,
9     "currency_name": "OpenCent",
```

```

10     "denominations": [1, 2, 5],
11     "id":
12     "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
13     "info_service": [
14         [10, "https://opencent.org"]
15     ],
16     "invalidation_service": [
17         [10, "https://opencent.org"]
18     ],
19     "issuer_cipher_suite": "RSA-SHA512-CHAUM86",
20     "issuer_public_master_key": {
21         "modulus":
22         "9f4e001f979e03a9c755694f936bc68930a67813e1d8b3afb4d9de408088522d551ee
23         c8babcc2ef99fc1f2814d49aa0c3e497f05d77fcd932192c742caf0adaf",
24         "public_exponent": 65537,
25         "type": "rsa public key"
26     },
27     "protocol_version": "https://opencoin.org/1.0",
28     "renewal_service": [
29         [10, "https://opencent.org"]
30     ],
31     "type": "cdd",
32     "validation_service": [
33         [10, "https://opencent.org"],
34         [20, "https://opencent.com/validate"]
35     ]
36 }

```

[Source](#)

## Mint Key Certificate (MKC)

## Description

## Example

```
1  {
2    "mint_key": {
3      "cdd_serial": 1,
4      "coins_expiry_date": "2023-10-16T15:32:04.103469",
5      "denomination": 1,
6      "id":
7        "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
8      "issuer_id":
9        "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
10     "public_mint_key": {
11       "modulus":
12         "8e1d32a23f1f596c6103c1efab9d63076e69e0fdce512fdf3b6989fc4acd30a1316ef
13         ffd8d6528c29256188aa6b149aebb03874cc8147383b913c8bd6f5e2869",
14       "public_exponent": 65537,
15       "type": "rsa public key"
16     },
17     "sign_coins_not_after": "2023-07-08T15:32:04.103469",
18     "sign_coins_not_before": "2022-07-08T15:32:04.103469",
19     "type": "mint key"
20   },
21   "signature":
22     "5f0efbe98fee5610ec911297e32ea1d7676d894dc9528e435da8ac31b2d76c5cc7e2f
23     44bf239b508a386850e87103bdf7cad493c375c1047070a9770c512edc",
24   "type": "mint key certificate"
25 }
```

## Source

## Blind

## Description

### Example

```
1 {
2   "blinded_payload_hash":
3     "3cc947ea0e017dab676ec5a674c464ed1eb3490c17dcb58dcd71c3106795e4c66da4b9
4     c8705f80b60bdf5d006b4ebdf59ef5a956c3a453f57d2085f549f68c5e",
5   "mint_key_id":
6     "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
7   "reference": "a0",
8   "type": "blinded payload hash"
9 }
```

[Source](#)

## Blind Signature

### Description

### Example

```
1 {
2   "blind_signature":
3     "5058415a5201b393843c0a86d20288be80ab45b60eaf269b43bea8a56a761d0deabde0
4     4da5b5538fb25f43c169e568c21575080375ef55e597165f5a538dc0e",
5   "reference": "a0",
6   "type": "blind signature"
7 }
```

[Source](#)

# Coin

## Description

## Example

```
1  {
2    "payload": {
3      "cdd_location": "https://opencent.org",
4      "denomination": 1,
5      "issuer_id":
6        "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
7      "mint_key_id":
8        "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
9      "protocol_version": "https://opencoin.org/1.0",
10     "serial":
11       "cbf41b41c80e85ff09edf7b6378b88aefaf14cf5697faabb9566573d34564faf",
12     "type": "payload"
13   },
14   "signature":
15     "44f6dc7319e1da0035b356141be002bb29972cf111c7f6efcb9ba9175ffaf861c07da
16     661e1984b2b85f824146b577cacf2d8354dbab87e1ed8ca7e233987c85b",
17   "type": "coin"
18 }
```

## Source

## Messages

## CDDSerial



## RequestCDDSerial

### Description

### Example

```
1 {  
2   "message_reference": 1,  
3   "type": "request cdd serial"  
4 }
```

### [Source](#)

## ResponseCDDSerial

### Description

### Example

```
1 {  
2   "cdd_serial": 1,  
3   "message_reference": 1,  
4   "status_code": 200,  
5   "status_description": "ok",  
6   "type": "response cdd serial"  
7 }
```

### [Source](#)

## CDDC

## RequestCDDC

### Description

## Example

```
1 {
2   "cdd_serial": 1,
3   "message_reference": 2,
4   "type": "request cddc"
5 }
```

## Source

## ResponseCDDC

## Description

## Example

```
1 {
2   "cddc": {
3     "cdd": {
4       "additional_info": "",
5       "cdd_expiry_date": "2023-07-08T15:32:04.103469",
6       "cdd_location": "https://opencent.org",
7       "cdd_serial": 1,
8       "cdd_signing_date": "2022-07-08T15:32:04.103469",
9       "currency_divisor": 100,
10      "currency_name": "OpenCent",
11      "denominations": [1, 2, 5],
12      "id":
13        "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
14      "info_service": [
15        [10, "https://opencent.org"]
16      ],
17      "invalidation_service": [
18        [10, "https://opencent.org"]
19      ],
20      "issuer_cipher_suite": "RSA-SHA512-CHAUM86",
21      "issuer_public_master_key": {
22        "modulus":
23          "9f4e001f979e03a9c755694f936bc68930a67813e1d8b3afb4d9de408088522d551ee
24          c8babcc2ef99fc1f2814d49aa0c3e497f05d77fcd932192c742caf0adaf",
```

```

22     "public_exponent": 65537,
23     "type": "rsa public key"
24 },
25 "protocol_version": "https://opencoin.org/1.0",
26 "renewal_service": [
27     [10, "https://opencent.org"]
28 ],
29 "type": "cdd",
30 "validation_service": [
31     [10, "https://opencent.org"],
32     [20, "https://opencent.com/validate"]
33 ]
34 },
35 "signature":
    "51287b6abca69924f8ab9ad121fb3a70fad0b4133d4c72e6e7142eb11ebdb770ac855
    b90301b124343ed97365dad40e62f0784fb7194f15e01b436d97d917318",
36     "type": "cdd certificate"
37 },
38 "message_reference": 2,
39 "status_code": 200,
40 "status_description": "ok",
41 "type": "response cddc"
42 }

```

[Source](#)

## MKCs

### RequestMKCs

### Description

### Example

```

1 {
2   "denominations": [1, 2, 5],
3   "message_reference": 3,
4   "mint_key_ids": [],
5   "type": "request mint key certificates"
6 }

```

[Source](#)

## ResponseMKCs

### Description

### Example

```

1 {
2   "keys": [
3     {
4       "mint_key": {
5         "cdd_serial": 1,
6         "coins_expiry_date": "2023-10-16T15:32:04.103469",
7         "denomination": 1,
8         "id":
9           "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
10        "issuer_id":
11          "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
12        "public_mint_key": {
13          "modulus":
14            "8e1d32a23f1f596c6103c1efab9d63076e69e0fdce512fdf3b6989fc4acd30a1316ef
15            ffd8d6528c29256188aa6b149aebb03874cc8147383b913c8bd6f5e2869",
16          "public_exponent": 65537,
17          "type": "rsa public key"
18        },
19        "sign_coins_not_after": "2023-07-08T15:32:04.103469",
20        "sign_coins_not_before": "2022-07-08T15:32:04.103469",
21        "type": "mint key"
22      },
23      "signature":
24        "5f0efbe98fee5610ec911297e32ea1d7676d894dc9528e435da8ac31b2d76c5cc7e2f
25        44bf239b508a386850e87103bdf7cad493c375c1047070a9770c512edc",

```

```

20     "type": "mint key certificate"
21 },
22 {
23     "mint_key": {
24         "cdd_serial": 1,
25         "coins_expiry_date": "2023-10-16T15:32:04.103469",
26         "denomination": 2,
27         "id":
28         "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
29         "issuer_id":
30         "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
31         "public_mint_key": {
32             "modulus":
33             "bd9a22abc30d6641276df6f29b336fab051764a7ca98e272289b4c6e54e76ad24b756
34             d62ef559317be941339ed220dac695b354fb0646591906afa2d6d202b4f",
35             "public_exponent": 65537,
36             "type": "rsa public key"
37         },
38         "sign_coins_not_after": "2023-07-08T15:32:04.103469",
39         "sign_coins_not_before": "2022-07-08T15:32:04.103469",
40         "type": "mint key"
41     },
42     "signature":
43     "4af276841fd8c76cdbafe5d5be1beda3c52a343465d9662855a30187bf9b1054b649
44     39c21e3fe8fb65411c74a175867a2fe7ade4ef0728df217cb784e1f505e",
45     "type": "mint key certificate"
46 },
47 {
48     "mint_key": {
49         "cdd_serial": 1,
50         "coins_expiry_date": "2023-10-16T15:32:04.103469",
51         "denomination": 5,
52         "id":
53         "48c351a32976c61879165787ed83948b8c3703e9abdd80b701ba5ab3334a3fa9",
54         "issuer_id":
55         "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
56         "public_mint_key": {
57             "modulus":
58             "e613a15970970a017213682faf28c455e2ff0d4820e3e5f72eace9e0f0d875874674e
59             d32a41db3051455bfa34ba4bb4d43a8cbd539730f88cf4927ecf218e6e5",
60             "public_exponent": 65537,
61             "type": "rsa public key"

```

```

52     },
53     "sign_coins_not_after": "2023-07-08T15:32:04.103469",
54     "sign_coins_not_before": "2022-07-08T15:32:04.103469",
55     "type": "mint key"
56 },
57 "signature":
    "348f92ddc72ecf3451d0f7e94b48e65e60e96b6ba4db195b336a6b2f943d2636a65d3
    721fa2e0d9ad55f149ca8db36bc6b87ed3fe13474243ee1e6990361a7f4",
58     "type": "mint key certificate"
59 }
60 ],
61 "message_reference": 3,
62 "status_code": 200,
63 "status_description": "ok",
64 "type": "response mint key certificates"
65 }

```

[Source](#)

## Mint

### RequestMint

### Description

### Example

```

1  {
2    "blinds": [
3      {
4        "blinded_payload_hash":
    "3cc947ea0e017dab676ec5a674c464ed1eb3490c17dcb58dcd71c3106795e4c66da4b
    9c8705f80b60bdf5d006b4ebdf59ef5a956c3a453f57d2085f549f68c5e",
5        "mint_key_id":
    "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
6        "reference": "a0",
7        "type": "blinded payload hash"
8      },
9    ]

```

```

10     "blinded_payload_hash":
11     "1b795a0bed3bdc77d93cfcb0d8117474ba2f3aa6778c4b5cc131a99c5bdc9b4978174
12     32a05901af8e26ce56252383ee8d0932e8a5d550826cd8bb285f0cd489d",
13     "mint_key_id":
14     "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
15     "reference": "a1",
16     "type": "blinded payload hash"
17 },
18 {
19     "blinded_payload_hash":
20     "a0b66f65dd98a99d59b830078331ef81e5117144fd0427b9788b87311de4497b64f9d
21     4c065d365db3533514eba2256b455523e8491fb6de0e13e36ff38cb8528",
22     "mint_key_id":
23     "48c351a32976c61879165787ed83948b8c3703e9abdd80b701ba5ab3334a3fa9",
24     "reference": "a2",
25     "type": "blinded payload hash"
26 }
27 ],
28 "message_reference": 4,
29 "transaction_reference":
30 "ea8d3d8f2940daed8a60c5e30fee99dc8bcb7548aefc6e457141a0933d74465d",
31 "type": "request minting"
32 }

```

[Source](#)

## ResponseMint

### Description

### Example

```

1  {
2    "blind_signatures": [
3      {
4        "blind_signature":
5        "5058415a5201b393843c0a86d20288be80ab45b60eaf269b43bea8a56a761d0deabde
6        04da5b5538fb25f43c169e568c21575080375ef55e597165f5a538dc0e",
7        "reference": "a0",
8        "type": "blind signature"
9      }
10     ]
11  }

```

```

7      },
8      {
9          "blind_signature":
10             "31ba402ffda0f5dbd0dbaed4015ce3d837fb34dc14266fc2342ea9226d85ab5b23de4
11             a8274236340914f7d90a06188ede1f3b5f09f142d8202e0e8613ce9662",
12         "reference": "a1",
13         "type": "blind signature"
14     },
15     {
16         "blind_signature":
17             "d443048b88a474796d1066499c34c5d46bae5655330583695de7b39358964589f098b
18             1914ba1255efcda29f19a11682ae42f8f0f6ccfd14230a355847e4f90ac",
19         "reference": "a2",
20         "type": "blind signature"
21     }
22 ],
23 "message_reference": 4,
24 "status_code": 200,
25 "status_description": "ok",
26 "type": "response minting"
27 }

```

[Source](#)

## CoinStack

### CoinStack

### Description

### Example

```

1  {
2      "coins": [
3          {
4              "payload": {
5                  "cdd_location": "https://opencent.org",
6                  "denomination": 1,

```



```

7      "issuer_id":
      "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
8      "mint_key_id":
      "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
9      "protocol_version": "https://opencoin.org/1.0",
10     "serial":
      "cbf41b41c80e85ff09edf7b6378b88aefaf14cf5697faabb9566573d34564faf",
11     "type": "payload"
12 },
13     "signature":
      "44f6dc7319e1da0035b356141be002bb29972cf111c7f6efcb9ba9175ffaf861c07da
      661e1984b2b85f824146b577cacf2d8354dbab87e1ed8ca7e233987c85b",
14     "type": "coin"
15 },
16 {
17     "payload": {
18         "cdd_location": "https://opencent.org",
19         "denomination": 2,
20         "issuer_id":
      "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
21         "mint_key_id":
      "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
22         "protocol_version": "https://opencoin.org/1.0",
23         "serial":
      "e968db759e8392dccc40d7133bf20a8e63b6aa59eaa78ed2483f2e446f05ed33",
24         "type": "payload"
25     },
26     "signature":
      "61447b2796e6f550d370c07cb8413d965641873a702ec263c9313ced4d9fc3e0502e1
      239facaee97c3e6eae4510a45c246d48e5055f947599888eeaf1ee63ad87",
27     "type": "coin"
28 },
29 {
30     "payload": {
31         "cdd_location": "https://opencent.org",
32         "denomination": 5,
33         "issuer_id":
      "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
34         "mint_key_id":
      "48c351a32976c61879165787ed83948b8c3703e9abdd80b701ba5ab3334a3fa9",
35         "protocol_version": "https://opencoin.org/1.0",

```

```

36     "serial":
37     "baa9c0c17eda2048c1b5c48e368880db5813d8f369e1a1bd5cfa16f9fbdff6bb",
38     "type": "payload"
39 },
40 "signature":
41 "8496567f6aefd4bca8bbfd3b0536dfd36c0475186a6837fe0dcc15e28033b80cf0c8
42 567df877ffba1a78b8816aae249d00b0ab06feef8e9e8cc1642ef660085",
43 "type": "coin"
44 },
45 ],
46 "subject": "a little gift",
47 "type": "coins"
48 }

```

[Source](#)

## Renew

### RequestRenew

### Description

### Example

```

1  {
2    "blinds": [
3      {
4        "blinded_payload_hash":
5        "8b56ee70d433ac8290c3da1cfcf582f928aa303c8a1f9672d2ad5957e1cc2eadd72be
6        28482f90475ee6f56bb18345986232326e9645c3d247fcd32a56640ac8c",
7        "mint_key_id":
8        "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
9        "reference": "b0",
10       "type": "blinded payload hash"
11     },
12     {
13       "blinded_payload_hash":
14       "6b32c227b3035f2bc09e66d1d1344c304a58d8c68f9d8f1c5bacc128af6f0c492838a
15       805bdd97893ce6937582bccc3e1d4c199bc4e54bf37680e19290de69611",

```

```

11     "mint_key_id":
12     "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
13     "reference": "b1",
14     "type": "blinded payload hash"
15 },
16 {
17     "blinded_payload_hash":
18     "7a1223a28b47cb9aee1993a246f70d6bd91353a96ebc5a2e48a82971bb0ec7929fc15
19     0774cf35bcea6fe3f554fd3e26a75a73ecb0bcee59c69b7967fc4619f2",
20     "mint_key_id":
21     "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
22     "reference": "b2",
23     "type": "blinded payload hash"
24 },
25 {
26     "blinded_payload_hash":
27     "126afab640de4708437ee48b93d2d163da6f485ac734e1f0fb63028270237627eb6e7
28     0b0c1128906dc0cc77292530c42ee63843499b1a4633a4a6d3f79be683b",
29     "mint_key_id":
30     "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
31     "reference": "b3",
32     "type": "blinded payload hash"
33 }
34 ],
35 "coins": [
36 {
37     "payload": {
38         "cdd_location": "https://opencent.org",
39         "denomination": 1,
40         "issuer_id":
41         "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
42         "mint_key_id":
43         "ec98d90c8a418dd7a9671a4fd3be84ac3bcf14e29288d5dd2a1db687884f25f0",
44         "protocol_version": "https://opencoin.org/1.0",
45         "serial":
46         "cbf41b41c80e85ff09edf7b6378b88aefaf14cf5697faabb9566573d34564faf",
47         "type": "payload"
48     },
49     "signature":
50     "44f6dc7319e1da0035b356141be002bb29972cf111c7f6efcb9ba9175ffaf861c07da
51     661e1984b2b85f824146b577cacf2d8354dbab87e1ed8ca7e233987c85b",
52     "type": "coin"

```

```

41     },
42     {
43         "payload": {
44             "cdd_location": "https://opencent.org",
45             "denomination": 2,
46             "issuer_id":
47 "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
48             "mint_key_id":
49 "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
50             "protocol_version": "https://opencoin.org/1.0",
51             "serial":
52 "e968db759e8392dccc40d7133bf20a8e63b6aa59eaa78ed2483f2e446f05ed33",
53             "type": "payload"
54         },
55         "signature":
56 "61447b2796e6f550d370c07cb8413d965641873a702ec263c9313ced4d9fc3e0502e1
57 239facae97c3e6eae4510a45c246d48e5055f947599888eeaf1ee63ad87",
58         "type": "coin"
59     },
60     {
61         "payload": {
62             "cdd_location": "https://opencent.org",
63             "denomination": 5,
64             "issuer_id":
65 "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
66             "mint_key_id":
67 "48c351a32976c61879165787ed83948b8c3703e9abdd80b701ba5ab3334a3fa9",
68             "protocol_version": "https://opencoin.org/1.0",
69             "serial":
70 "baa9c0c17eda2048c1b5c48e368880db5813d8f369e1a1bd5cfa16f9fbdf6bb",
71             "type": "payload"
72         },
73         "signature":
74 "8496567f6aefd4bca8bbfd3b0536dfd36c0475186a6837fe0dccb15e28033b80cf0c8
75 567df877ffba1a78b8816aae249d00b0ab06feef8e9e8cc1642ef660085",
76         "type": "coin"
77     }
78 ],
79 "message_reference": 5,
80 "transaction_reference": "be209986090bf15fac94e32b04bf4b53",
81 "type": "request renewal"
82 }

```

[Source](#)

## Resume

### ResponseDelay

#### Description

#### Example

```
1 {  
2   "message_reference": 5,  
3   "status_code": 300,  
4   "status_description": "ok",  
5   "type": "response delay"  
6 }
```

[Source](#)

### RequestResume

#### Description

#### Example

```
1 {  
2   "message_reference": 6,  
3   "transaction_reference": "be209986090bf15fac94e32b04bf4b53",  
4   "type": "request resume"  
5 }
```

[Source](#)

# Redeem

## RequestRedeem

### Description

### Example

```
1  {
2    "coins": [
3      {
4        "payload": {
5          "cdd_location": "https://opencent.org",
6          "denomination": 2,
7          "issuer_id":
8            "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
9          "mint_key_id":
10            "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
11          "protocol_version": "https://opencoin.org/1.0",
12          "serial":
13            "b71b03576c31c733208775eb337849befdb904686f4dd6ce4e76fb779ba96c43",
14          "type": "payload"
15        },
16        "signature":
17          "687ec5d2dda17102deb8203e208aebdaa6764fc7c91fb45aeb7ddceed720f8749247
18          015c7f70ac651b21ed3ca409b6feff3ee97da174a65eae5e0c7a79ab756",
19        "type": "coin"
20      },
21      {
22        "payload": {
23          "cdd_location": "https://opencent.org",
24          "denomination": 2,
25          "issuer_id":
26            "b7520e474a9b80032c051cca120dc52048d3f334fc5b0d9e12c9aca155e90d3a",
27          "mint_key_id":
28            "8fea6f66fe0891e67c089b37459416f2bc285517b8c78fab4fb919e84d24fbba",
29          "protocol_version": "https://opencoin.org/1.0",
30          "serial":
31            "d6bca1c6a01fa6c3b5ffa4e4d9445fbbad9cf8888a0a36b0773b569f39cbf501",
32          "type": "payload"
33        },
34      },
35    ],
36  }
```

```
26     "signature":  
    "69d8287ea6e9ba2644c293fcc5089ac9099e8e7990debb4ea5bbc6204f2eb43571a2d  
    54e1e1f6fd7c65941e5e88fb91c02dc5ca8a0fc6ed907bf3f226d1b1e57",  
27     "type": "coin"  
28   }  
29 ],  
30   "message_reference": 7,  
31   "type": "request redeeming"  
32 }
```

[Source](#)

## ResponseRedeem

### Description

### Example

```
1 {  
2   "message_reference": 7,  
3   "status_code": 200,  
4   "status_description": "ok",  
5   "type": "response redeeming"  
6 }
```

[Source](#)

# Reference

---

## Appendix

### Scope

Having said all of the above, we scope the protocol and it's description in the following way:

**Targeted at developers** - developers should be enabled (and motivated) by the OpenCoin protocol to implement standard confirming software components and apps. However we hope that this documentation is also understandable for the interested user (or founder, investor, auditor, etc.)

**Just the protocol** - we don't deliver any ready to use implementations. This allows us to fully focus on the protocol, and keeps a separation to actual implementations.

**Easy to understand** - we try to avoid complexity. This affects the protocol itself as well as it's documentation. This means: if you, the reader, don't understand a sentence or a concept, please contact us. We will improve the description. Being easy to understand is one of the main goals of OpenCoin.

**Only the core** - lots of developments have happened since [we started](#). Take the example of messengers like Signal, Telegram or WhatsApp. They have opened new ways to transport messages, and they take care of identifying the communication partner. This especially means that message transport and authentication stays out of scope.

### History and old results

- Project history
- Project papers
- Crypto report
- Legal report
- Code bases (v1, sandbox, javascript implementation)

### Artifacts



Details of the documents in the [artifacts directory](#)

## JSON Schemata

### Ideas

- use .oc file ending
- oc over html
- opencoin.org as a web interface demo provider, that can handle .oc files

### Building blocks for writing

create CDDC

create MKCs

RequestCDDSerial

ResponseCDDSerial

RequestCDDC

ResponseCDDC

RequestMKCs

ResponseMKCs

prepare blinds

RequestMint

sign blinds

ResponseMint

unblind

transfer CoinStack, e.g. using Signal

tokenize sum

prepare blinds

RequestRenew

ResponseDelay

RequestResume

validate coins

RequestRedeem

ResponseRedeem

## Header

### Description

### Example

```
1
```

### Source

## Request

### Description

### Example

```
1
```

### Source

## Response

### Description

### Example

```
1
```

### Source

## Footnotes

---

- 
1. David Chaum, "Blind signatures for untraceable payments", Advances in Cryptology - Crypto '82, Springer-Verlag (1983), 199-203. ↩
  2. Please check with your lawyer if this is a good idea. ↩
  3. To keep the diagram simple we have left out Charlene who was mentioned above in "How does it work?". Bob does everything she does. ↩
  4. This is to minimize damage in case the mint keys get compromised. ↩