

Voting Open - Relay

Glossary:
 C - contribution split e.g. [A: 50, B: 100, C: 0, D: 75]
 (C) - encrypted contribution split e.g. [A: (Rca, Sca), B: (Rcb, Scb)...]
 V - vote e.g. [A: sqrt(50), B: sqrt(100), C: sqrt(0), D: sqrt(75)]
 (V) - encrypted vote e.g. [A: (Rva, Sva), B: (Rvb, Sv)...]

User signs message for USDC 2.1's receiveWithAuthorization():

- total value (225 USDC)
- their address (from addr)
- contract address (to addr)

BP signs message of:

- total value (225 USDC)
- user address (USDC from addr)
- nonce (transfer nonce)

only if user has completed KYC

Flow:

- Sign in** (User to MetaMask)
- Sign for USDC** (MetaMask to User)
- Basket:** [A: 50, B: 100, C: 0, D: 75] (User to API)
- Checks:**
 - Is user KYC'd?
 - Is contribution valid?
- Calculates:**
 - Encrypted contribution
 - Encrypted vote
 - Auth signature
- Send relay tx** (API to Infura relay)
- USDC +** (Infura relay to contract)
- Contribution event** (contract to Event Listener)
- Checks:**
 - bp has signed off
 - user has signed for USDC
 - user hasn't voted
- Adds 225 USDC to funding pool** (contract to Event Listener)
- tx confirmed** (Event Listener to API)
- tx confirmed** (API to DB)
- User addr, (C) (V) C V** (API to DB)

Each user's data stored as:

```

User i: {
  address: {},
  contribution: 225,
  votes: [
    {projectId: A,
      amount: (C_A) (=enc(50)),
      vote: (V_A) (=enc(sqrt(50))),
    },
    ...
  ],
  signature: {},
  status: ['pending', 'success', 'failed'],
}
  
```

