Just an idea right now, but I find it interesting and I didn't find anything similar. I'd like to get some opinions to see whether it makes sense and is worth working for.

1. Abstract.

The system allows Sellers can to make accounts, boost their reputation by donating some money to already reputable non-profit institutions, charities etc. (Receivers), and allow Customers to post Reviews.

Review posting is done in two steps. First is done by Seller, who generates Review Token and signs most of its data fields, except the rating, which is filled by Customer. Second step is done by Customer, who adds rating data signed by one-time use key provided by Seller.

Review Token is not present on chain, only finalized Review is, so the only on-chain transaction is initiated by the Customer, but it can be done without having to own/spend any on-chain assets by making use of account abstraction (ERC-4337 standard).

1. Definitions.

2.1. User types:

2.1.1. Sellers

- typically profit-oriented businesses or any other entities wanting to increase their reputation and visibility to end users.

2.1.2. Receivers

- public benefit organizations, who should be already trusted.

2.1.3. Customers

- they can post reviews even without owning any ETH, by utilising Review Tokens.

2.2. Seller Account

A contract deployed by Seller, which can hold assets and pay them to Receivers when a Review is posted. Implements methods necessary to post Reviews and browse them with Client Application, stores data described in 3.2.

2.3. Client Application

An external user interface, allowing to browse Seller's data and Reviews.

Overall Seller rating usually depends on total amount of donated fees and average Review rating weighted by fee paid.

Also chosen Receiver reputation can be a factor - there is some trust needed in this part, but it can be handled off-chain depending on Client Application and end user preferences.

Only objective data has to be stored on-chain, affect money flow and contract execution, subjective value of a Receiver only affects view/browse phase.

Client Application allows sorting and filtering Seller Accounts and their Reviews by any criteria, as well as utilising additional contract like account registry that can list Seller Account and give them unique names.

2.4. Review Token.

An off-chain set of data that allows Customer post Review, that will be paid by Seller (both Receiver donations and gas fees).

2.5. Review.

An on-chain set of data, appended to Seller Account reviews array when Cusomer adds it. Structure data described in 3.4., possible methods of adding in 4. and 5.

1. Data types.

3.1.

Object

- internally, it's just bytes, but fields defined as Object

have additional special purpose. They contain JSON encoded fields, which don't affect contract execution, but can be meaningful to client application. This will also allow to make some further improvement that don't require any changes on the contract execution layer.

3.2. Seller Account. (on-chain)

- Object

data - contain fields like name, description, keywords, location, contact info etc. Can be edited by the owner.

- address

owner EOA or contract. Accounts can be transferrable like NFTs.

- []Review

reviews - array containing all posted Reviews (defined in 3.4).

3.3. Review Token. (off-chain)

- sender, nonce, callGasLimit, verificationGasLimit, preVerificationGas, maxFeePerGas - fields from ERC-4337 Account interface, defined and signed by Seller

- uint12

expiryDate - after this time, posting Review using this Review Token is no longer possible

- Object

sellerData - additional data provided by seller, such as product details

- address

receiverCurrency - can be ETH if this field is left blank, or any ERC-20 token if it's its contract address.

- [](

addressaccount,

uint256amount)

receivers - array of Receiver accounts with amount paid to them from Seller Account.

- bytes

customerPublicKey one-time-use public key for a Customer to sign the Review.

- bytes

sellerSignature - signs all above fields (all Review fields except customerData rating and customerSignature)

- bytes

customerPrivateKey - one-time-use private key for a Customer to sign the Review. Unlike customerPublicKey, not included in posted Review structure.

3.4. Review. (on-chain)

- all fields of Review Token, except customerPrivateKey

- Object

customerData - overall rating, partial ratings, description etc.

- bytes

customerSignature - signs all fields

1. Example use case:

4.1.

Seller generates a Review Token, prints it as scannable QR code and includes it in product package.

4.2.

Seller sells product to the cutomer.

4.3.

Customer scans Review Token using Client Application, checks its validity, expiry date and other details.

4.4.

Customer writes a Review using Client Application.

4.5.

Client Application sends UserOperation with review data signed by customerPrivateKey.

Only at step 4.5, any transaction goes into blockchain and it’s only done if the token gets actually used by the Customer to post Review.

This way, Seller will not pay anything for tokens that were not used. So if not all Review Tokens are used, the costs of including tokens are, on average, lower than its value.

In addition Review Tokens have defined expiry date, so Seller don’t have to fear that some massive use of very old tokens, not used before, will deplete his account.

Tokens also include one-time private key to prevent frontrunning attacks. As the customerPrivateKey is received from the Seller, he could still do such attacks to replace negative feedback with positive one.

However, such actions are aren’t easy to do (Sellers don’t have a lot of control over bundler’s mempool, and the UserOperation is initiated by Customer) and, even if done, they are risky as the Customer can retaliate by sending another Review (using method from point 5.) with himself being the payer of fees, including signing it with same customerPrivateKey to prove it’s related to given token.

Although in general Customers don’t need to own ETH or even any account, some still can do and the Seller can’t know which ones are able and willing to do it.

1. Alternative method for posting Review.

This is second possible method of publishing Review, that can be done directly by anyone without having Review Token.

In this case, all payments (to Receivers and for gas) must be done directly by user who posts the Review - then the only required fields are: sellerData, customerData, receiverCurrency and receivers.

Sellers themselves, especially new ones, can use this to bootstrap their reputation. For Customers, it can be useful in case of Seller’s bad behavior related to Review Token itself, such as:

- frontrunning attack like described above, trying to replace bad feedback with good one.
- giving invalid, expired or already used token, or use them by themselves, not allowing Customer to do it later.
- not having enough assets on the account to pay fees.

This use case, unlike the one with Review Token, can be done as regular transaction, not ERC-4337 UserOperation.