The Alexandria Review

Decentralized peer review and publication of academic articles.

By Cameron Rout (rout@) - Draft 1/25/2018

Overview

This protocol enables editors to submit academic papers for double blind peer review without the need for a third party intermediary. It provides proof of publication for articles and related comments while withholding the identity of authors through the review process and obscuring the identity of reviewers in perpetuity. The protocol also enables the trustworthy transfer of publication fees to the paper's reviewers and editors.

Recent news

- [Judge grants CrossFit's request to unmask anonymous peer reviewers](#)
- [Scholars seek open access in academic journal deal](#)
- [We must have open access to scholarly publishing](#)
- [South Korean Universities Make Deal with Elsevier](#)

Value propositions

- University libraries (and other institutions) can function as publishers of peer reviewed work without requiring third party intermediary (i.e. journals) to ensure a trustworthy review process.
- Academic institutions can enter into agreements directly with each other for the publication of work without a third party journal subscription.
- Authors can archive comments along with their work without disclosing the identity of the reviewers.
- The merit of institutions can be rewarded with full transparency into the peer review process.
- Publication fees and subscription revenues can be earned by university editors and peer reviewers rather than third party journals.

Use cases

Author

- submit work to be peer reviewed by an institution of their choice.
- Remain anonymous during the review and claim work publicly afterwards.
- respond to comments during the review without revealing their identity.

- have full visibility into all comments by reviewers on their work and can publish or archive them along with the article.

Editor

- accept or reject work on editorial standards.
- request review from colleagues, peers from other institutions and from institutions themselves.
- Request reviews from specific reviewers
- Request a number of reviews from a pool of reviewers
- Prove that articles were reviewed by a specific number of members of a specific group of public identities without revealing identities.
- set the criterion for acceptable peer review based on acceptance by reviewers.
- control pricing and access to the work they publish.
- maintain the right to be the sole publisher of works.
- request review from other institutions or individuals, who in turn may request review from other reviewers based on their own criteria for acceptance.

Reviewer

- submit comments to the author without knowing their identity.
- recommend whether a paper should or should not be published.
- determine the cost of their review based on complex criteria (topic, requestor, timeline, etc.)
- receive a share of the publication fee for reviewing the article without their participation being traced

Researcher / Student

- access works from a publication by subscription, pay-per-view, or open access.
- access revisions and comments of the review process of a paper
- participate in peer review process with amateur/semi-pro journals (student journals)

Features

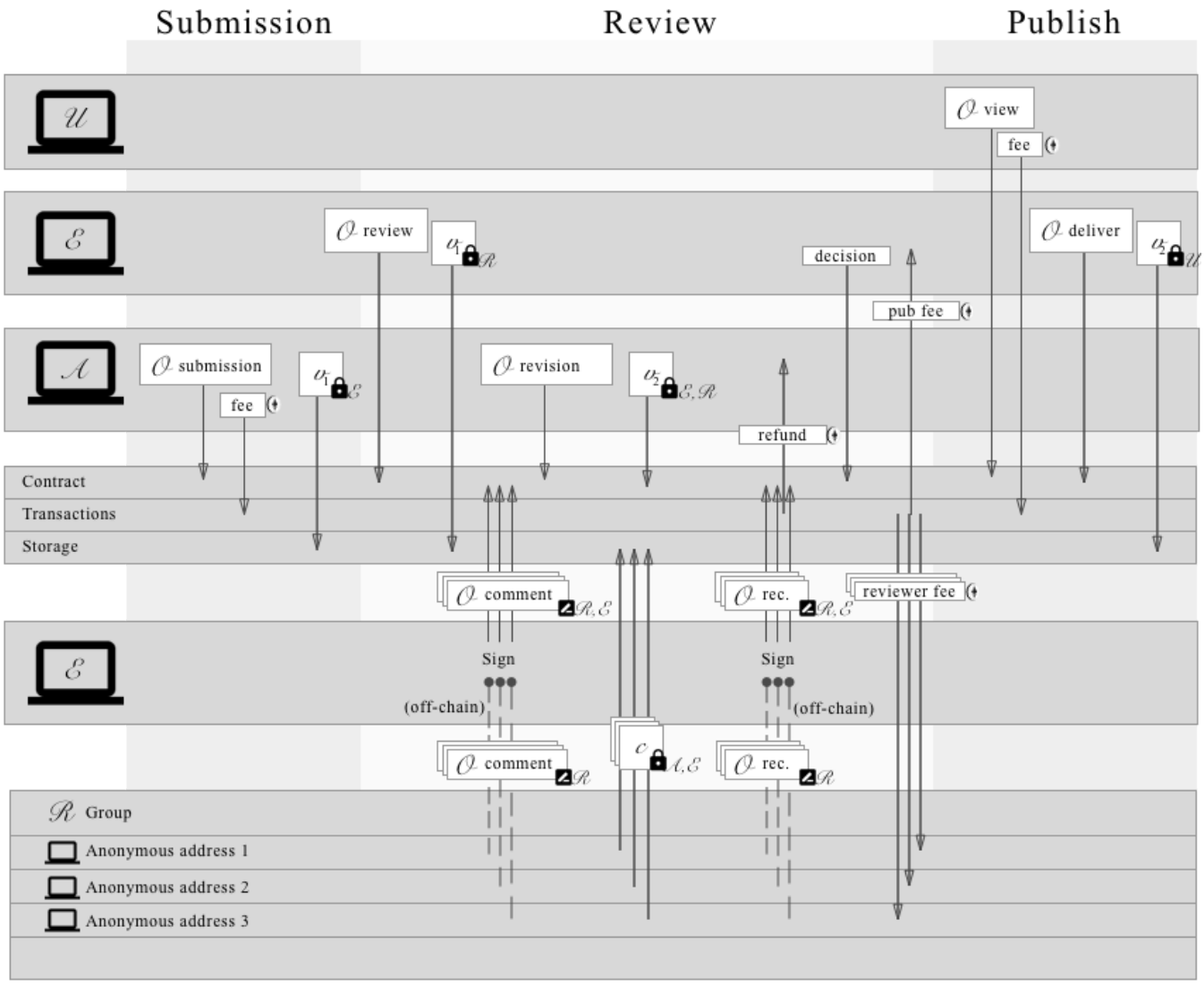
- Private submissions: articles are encrypted before storage with the public key of the reader. The unencrypted infohash of the content is always included for verification and common reference throughout the process by all parties.
- Public Registry: addresses can be registered publicly without compromising the double blind process. For example, authors can remain anonymous throughout the review and then claim their submission with a public identity. Publications can list their reviewers without disclosing which reviewer worked on which paper.
- Decentralized Storage: all materials are encrypted before storage in a decentralized repository such as IPFS, bittorrent or filecoin and referenced by infoHash.
- Variable Publication fees: Can be set independently for general submissions and for specific addresses. Fees are placed in escrow until the final criterion for publication or rejection is met. A guaranteed timeline can be provided such that if a decision is not reached in a certain timeline, the author can revoke their submission for a full or partial refund.
- Private Fee Structures:
    - **Open issue**: keeping fee structures private. Institutions may not want to know what discounts are being given different subscribers.
    - Options: encrypt fees by address with public key of recipient; request an encrypted quote off-chain; private bid/ask order system;

- Decentralized Download fees: Downloads are paid for with a double incentive scheme. Since the smart contract cannot safely encrypt documents and provide keys to the downloader, the protocol must rely on the editor dapp to provide access and the editor dapp to provide proof of receipt.
    - The downloader puts twice the fee in escrow and the publisher puts a fee in escrow as well upon accepting the request, this incentivizes both parties to complete the transaction truthfully. The publication sends an encrypted order for the download to the reader. Both parties are incentivized to complete the download transaction and the net result is the fee transferring to the selling party. The publication can rotate the same funds to fulfill multiple download requests to avoid high liquidity requirements.
- Group Subscriptions: Users can use a group signature to prove their membership in a subscription group (i.e. students of a university) and qualify for free/discounted downloads.
- Subscriptions: addresses can qualify for free downloads as long as their address delivers a minimum amount every period.
- Fiat Pricing: Contracts use oracles to set prices in stable, local fiat.
- Anonymous submissions: Authors generate a new “throw-away” address for the submission transaction for a double-blind review. Once the paper is accepted or rejected, the author can claim the submission by signing a non-anonymous version of the paper with the throw-away address and stake the claim with the publicly known address.
- Publishable history: The author has full control over which comments and revisions are included with the final publication or archive.
- Reviewer anonymity: there are two methods of review: pool and individual.
    - **Pool requests** are sent to a group address and members use their private key to the group key to accept the request.
    - **Individual requests** require a different method. The editor sends an encrypted request for review to all members of the reviewer group. Some of the orders are false and some are true, indicating which are being invited and which are not without revealing to the faculty members which other members were selected. A reviewer with a true order can use or create an anonymous (stealth) account and sign the order with a group ring signature to prove it came from a faculty member before sending to the blockchain.
- Anonymous review: All fee transfers and communication (on-chain and off-chain) with the reviewer use the verified anonymous address generated for the submission.
    - **Open Issue**: susceptible to tracking payments after they have received the anonymous fee.
    - How anonymous does this need to be? Do we need to use zCash/snarks?
- Anonymous comments: Comments from reviewers and editors are encrypted with the public key of the other participants (editor, author, reviewer) before seeding. The party submitting each comment communicates the infohash information on the blockchain for discovery.
- First-to-publish: Timestamp of the infohash and acceptance is immutable on the blockchain along with the time of receipt of original submission.
- Reviewer fees: The reviewer fee is attached by the author in escrow to the editor for each submission. Fees for review are sent in escrow from the editor to the reviewers. The required rate for review can be run as an anonymous auction in that if no reviewer accepts the transaction, the fee can be increased by the requestor until the required number of reviewers participate.
- Institutional reviews: The publication can require peer review from another institution for submissions. In this case, the submission is sent to a specialized publication that is intended only for peer reviews. For example, University A requires at least one review from University B or C in addition to their own peer review process in order to publish. Each university has a publication that allows submissions from the other two universities known as the review journal. University A submits the paper for publication in the review journal of the other two universities, if it is published, then the requirement is met. In this

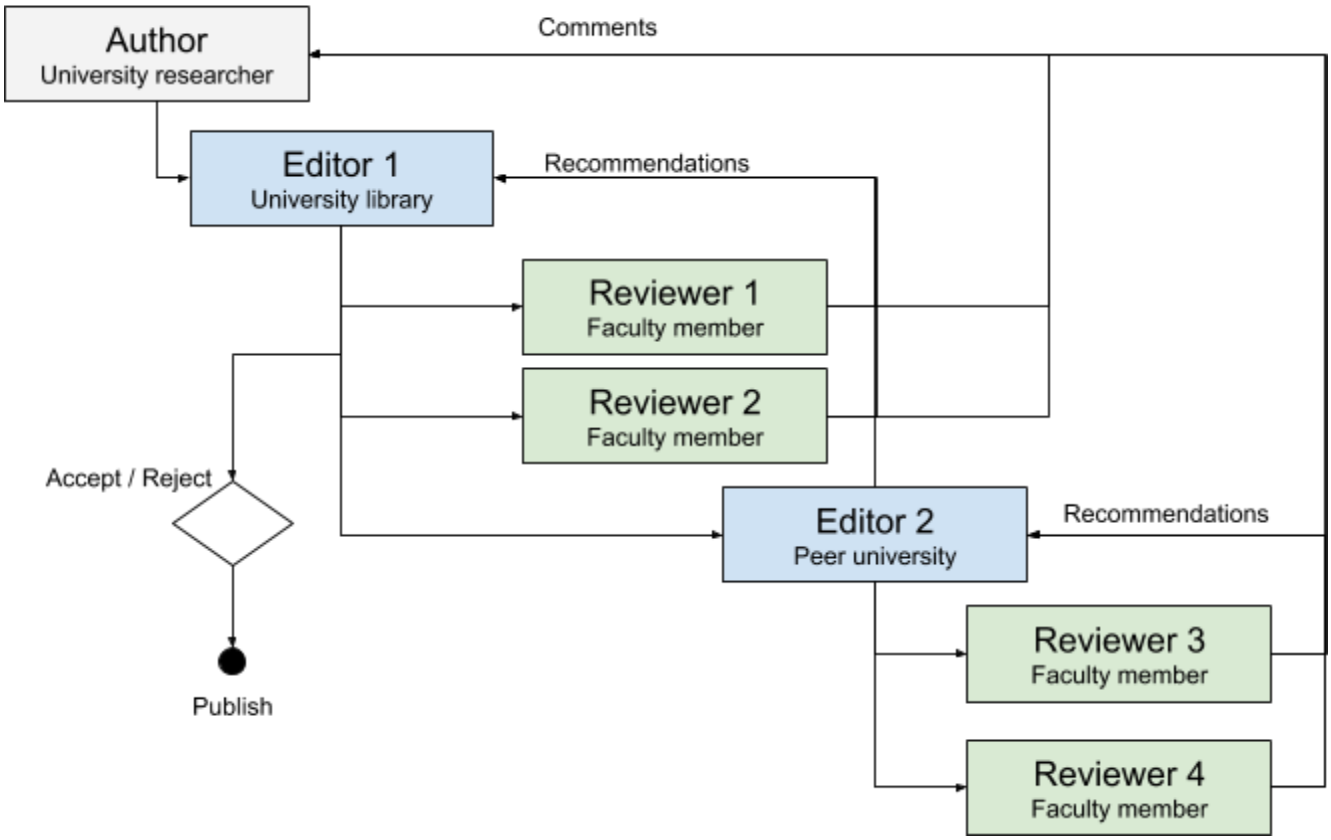
way, the ivy league may decide to require each paper to be reviewed by another institution while the author from Yale does not need to worry about bias from Harvard reviewers.

- Pay to download: Editors accept orders to view documents and fulfil them by encrypting a file with the subscriber's public key and broadcasting the infohash to the blockchain with an acceptance of the request. The file is then submitted to the storage network (i.e. bittorrent).

Simplified overview of the decentralized publication process



Example of recursive review



Open issues

IP Tracking

The protocol does not handle IP obfuscation. The process is only truly anonymous if performed via TOR/VPN.

Payment tracking

Using ETH for fees exposes anonymous reviewers to doxxing via payment tracking. It may be necessary to obfuscate payments with zk-snarks or use zCash for payment.

The case for a native token: ALX/Alexandrians

Does this protocol require a native token? The current proposal is for all fees to be paid in ether, while using oracles for Fiat pricing. Here are some options for how a native token could be used.

1. Fee token: use native token for fees instead of ether.
2. Tokenized submissions: Instead of a fee system, publications can issue an indivisible token that permits an author to request publication. The token is refunded if the request is rejected and the author can sell it on the market or use it again. The token is consumed if the review request is accepted.

Oraclized Reviewer Network

Reviewer network in the current proposal relies on public identities for credibility. An editor can prove that a review was provided by one of a group of public identities without revealing which identity reviewed a particular paper. However, if it is possible for reviewers to prove their credibility anonymously, then reviewers can act as market driven oracles using ZAP.

As long as credibility is based on authorship of published articles, then anonymous reviewers would have to also remain anonymous when publishing. However, if credibility can be established from the review process itself, then merit can drive demand for a reviewer without connecting her to her published work.

One option could be to use a rating system that allows peers to respond to the quality of reviewer comments before the review becomes public. Using a consensus model to determine if a review is based on domain-specific requirements.

Separating the review and publication processes

Is it necessary to handle both the review process and the download/distribution process? It may make sense to do the review process first and then build out the discovery and download use cases after there is a sufficient repository of published information for certain publications.

Dapp Features

- Broadcast: ensures comments and submissions are properly encrypted and announces torrents in a discoverable format.
- Notifications: scans the blockchain for requests for review, comments, errors and updates on submissions.
- Consume: pulls files from bittorrent by infoHash and decrypts comments and articles.
- Anonymity: generates throw-away addresses and group keys.
- Protocol: Interacts with the smart contract to escrow/transfer fees, accept requests for review, and accept or reject submissions.

Detailed dapp requirements by user journey

Submission process

Author submits article

- BE: Index the blockchain for relevant publications
- UI: Select public address of publication from index
- UI: Review terms of submission
- UI:: Identify local file for submission
- UI: Select action: Submit paper
- UI: Confirm submission
- BE: Generate anonymous submission address
- BE: Generate infoHash of unencrypted file
- BE: Encrypt submission file with editor's public key
- BE: Generate infoHash of encrypted file
- BE: Sign escrow transaction for publication fee
- BE: Generate and sign submission order
    - Submission ID (Unencrypted infoHash)
    - Editor's address
    - Publication ID
    - Submission address
    - InfoHash of encrypted file
    - Signed escrow transaction
- BE: Generate .torrent file from encrypted submission
- BE: Submit submission order to blockchain

- BE: Broadcast .torrent to network.

Editor reviews article

- BE: Scan blockchain for orders to the editor's address
- BE: Verify order and escrowed fee
- BE: Download .torrent by encrypted infoHash magnet
- BE: Decrypt file and verify unencrypted infoHash
- BE: IF file and transaction are NOT valid.
    - Reject order (off-chain)
    - ELSE: Notify editor of submission
- UI: Review decrypted submission

Editor requests resubmission

- UI: Write editorial comments
- UI: Select action: request resubmission
- UI: Confirm request for resubmission
- BE: Generate infoHash of unencrypted comments
- BE: Encrypt comments with public key of author
- BE: Generate infoHash of encrypted comments
- BE: Generate and sign Resubmission Request Order
    - Submission ID
    - Comments' encrypted infoHash
    - Comments' unencrypted infoHash
- BE: Generate .torrent file from encrypted comments file
- BE: Broadcast .torrent to network

Author responds to comments from editor

- BE: Scan blockchain for orders referencing the unencrypted infoHash
- BE: Verify order signature from editor
- BE: Download .torrent of comments by infoHash magnet
- BE: Decrypt comments file and verify by unencrypted infoHash
- BE: Notify author of request for resubmission
- UI: Review comments
- UI: Revise paper
- UI: Identify revised paper on local system
- UI: Select action: resubmit paper
- UI: Confirm selection
- BE: Generate infoHash of unencrypted resubmission file
- BE: Encrypt revision with editor public key
- BE: Generate infoHash of encrypted resubmission file
- BE: Generate and sign resubmission order
    - Editor's address
    - Submission address
    - Submission ID
    - Previous revision ID
    - New revision ID (InfoHash of encrypted resubmission file)
    - Signed transaction
- BE: Generate .torrent file from encrypted resubmission

- BE: Submit resubmission order to blockchain
- BE: Broadcast resubmission .torrent to network.

Editor reviews resubmission

- BE: Scan blockchain for orders to the editor's address
- BE: Verify resubmission order
- BE: Download .torrent by encrypted infoHash magnet
- BE: Decrypt file and verify unencrypted infoHash
- UI: Review decrypted resubmission

Peer Review process

Editor submits paper for peer review from specific reviewers

- UI: Select public addresses of specific reviewers
- UI: Select decrypted version of paper on local system
- UI: Select action: Request Review
- UI: Enter/Confirm the reviewer fee
- UI: Confirm request for review
- BE: Encrypt the latest version with the public keys of the intended reviewer (one each)
- BE: Retrieve addresses of all members of reviewer's group
- BE: Generate and sign Request for Review orders for each member of the reviewer group
    - Submission ID
    - Publication ID
    - Revision ID
    - Editor's address
    - Infohash of encrypted submission
    - Signed escrow transaction for reviewer fees
    - Request for review: true / false
    - Request type: individual
- BE: Encrypt Request for Review with vk of each group member
- BE: Generate .torrent file of encrypted revision for each intended reviewer (one each)
- BE: Broadcast .torrent to network
- BE: Submit R4R orders to blockchain

Editor submits paper for peer review from pool of reviewers

- UI: Select reviewer group
- UI: Enter the number of reviewers requested from the pool
- UI: Select decrypted revision of paper on local system
- UI: Select action: Request Review
- UI: Confirm request for review
- BE: Identify public reviewer group key (RGK) of the specialist group on the correct topic
- BE: Encrypt latest version with the RGK
- BE: Sign escrow transaction to RGK for portion of submission fee
- BE: Generate and sign Request for Review order (R4R))
    - Editor's address
    - Submission ID
    - Submission address
    - Encrypted infohash
    - Revision ID (Unencrypted infohash)

        - Signed escrow transaction for reviewer fees
        - Request type: pool
- BE: Generate .torrent file from RGK encrypted version
- BE: Submit R4R order to blockchain
- BE: Broadcast .torrent to network.

Reviewer reviews R4R

- BE: Scan blockchain for R4R orders referencing the reviewer's address and group addresses
- BE: Verify order signature from editor
- BE: Decrypt individual R4R orders with pk and check if Request for Review is true
- BE: Download .torrent by infoHash magnet
- BE: Decrypt revision file and verify by unencrypted infoHash
- BE: Notify FE of request for review
- UI: Review paper

Reviewer accepts R4R (pool)

- UI: Select action: accept request for review
- BE: Generate new anonymous reviewer address
- BE: Accept reviewer fee by signing escrow transaction from RGK to anonymous reviewer address
- BE: Generate Review Request Response (RRR) order
    - New reviewer address
    - Submission ID
    - Submission address
    - Editor address
    - Accept request (true/false)
    - Signed escrow transaction
- BE: Sign R4R order with Group Ring Signature (GRS) proving they are a faculty member
- BE: Submit R4R to blockchain from anonymous reviewer address
- BE: Scan blockchain to determine if maximum number of acceptance orders have already been submitted.

Reviewer accepts R4R (individual)

- UI: Select action: accept request for review
- BE: Generate new anonymous reviewer address
- BE: Accept reviewer fee by signing escrow transaction from RGK to anonymous reviewer address
- BE: Generate Review Request Response (RRR) order
    - New reviewer address
    - Submission ID
    - Submission address
    - Editor address
    - Accept request (true/false)
    - Decrypted R4R order with value = TRUE (signed by editor)
        - Proves the address was generated by an intended reviewer
    - Signed escrow transaction to anonymous reviewer address
- BE: Sign R4R order with Group Ring Signature (GRS) proving they are a faculty member
- BE: Submit R4R to blockchain from anonymous reviewer address

Reviewer submits comments

- UI: Write comments

- UI: Select action: submit comments
- UI: Confirm action
- BE: Encrypt comments with public key of Author
- BE: Encrypt comments with public key of Editor
- BE: Generate Reviewer Comment order
    - Submission ID
    - Revision ID
    - Comment ID (Infohash of unencrypted file)
    - Editor infohash
    - Author infohash
    - Editor address
    - Submission address
- BE: Sign reviewer comment order with GRS
- BE: Generate .torrent file for Editor
- BE: Generate .torrent file for Author
- BE: Submit reviewer comment order to blockchain
- BE: Submit torrents to network

Editor verifies reviewer to author (passive)

- BE: Scan the blockchain for reviewer comments to editor address
- BE: Verifies the order signature from RGK
- BE: Signs the order from the Editor private key
- BE: Submits signed order to blockchain

Author reads reviewer comments

- BE: Scan the blockchain for Reviewer Comment orders sent to submission address
- BE: Verify signature from editor
- BE: Download torrent using author infohash
- BE: Decrypt comment file and verify infohash
- BE: Notify FE of new comment
- UI: Review comments

Author responds to reviewer comment

- UI: Write response
- UI: Revise paper (optional)
- UI: Select action: respond to comment
- UI: Confirm action
- BE: Encrypt comments and revision with public key of reviewer
- BE: Encrypt comments and revision with public key of editor
- BE: Generate and sign Respond to Comment order
    - Submission ID
    - Previous revision ID
    - New revision ID (can be the same as previous if no change)
    - Comment ID
    - Editor infohash (comments)
    - Editor infohash (revision)
    - Reviewer infohash (comments)
    - Reviewer infohash (revision)

        - Reviewer address
- BE: Generate 4 .torrent files (comments and revision for editor and reviewer)
- BE: Submit Respond to Comment order to blockchain
- BE: Submit torrents to network

Reviewer submits recommendation

- UI: Select action: Recommend publication True / False
- UI: Confirm selection
- BE: Generate Publication Recommendation order
    - Submission ID
    - Recommendation (true/false)
    - Editor Address
- BE: Sign Publication Recommendation order with GRS
- BE: Submit Publication Recommendation order to EDITOR (offchain)

Publication process

Editor approves/denies publication

- BE: Scans the blockchain for submissions that have met the criterion for publication
- BE: Notify FE
- UI: Review comments, revisions, and addendums
- UI: Select action: approve/reject publication
- BE: Generate torrent of public metadata (abstract etc.)
- BE: Generate and sign Approve Publication order
    - Submission ID
    - Publication ID
    - Revision ID
    - Public metadata infohash
    - Approval status (true/false)
- BE: Submit signed order to blockchain
- BE: Submit public metadata torrent to network
- Blockchain: Verifies requirements of publication are met
    - IF TRUE: Executes signed fee transactions
        - Publication fee in escrow from author to editor
        - Reviewer fees in escrow from editor to reviewers
    - IF FALSE: sends refund in escrow to author

End user experience

User reviews abstracts of a publication

- UI: Find publication ID
- UI: Select action: review publication
- BE: Scan blockchain for Approve Publication orders by publication ID
- BE: Download public metadata torrent using infohash
- UI: Review public metadata

User requests to download an article (no subscription)

- UI: Select submission ID
- UI: Select Action: Download paper
- BE: Generate and sign Download Request order

        - Submission ID
        - Signed escrow transaction to editor for 2x the download fee to editor address
- BE: Submit Bid for Download order to blockchain

Editor fulfills a download request (no subscription)

- BE: Scan blockchain for Bid for Download orders
- BE: IF ACCEPT:
    - Generate and sign download confirmation order
    - Encrypt download file with requestor's public address
        - Published version of submission
        - Signed download confirmation order
        - High entropy nonce
    - Generate torrent of encrypted download file (containing confirmation)
    - Generate and sign Download order
        - Signed escrow transaction for 1X the download fee to requesting address
        - infoHash of encrypted file
    - Submit torrent to network
    - Submit Download order to blockchain
- BE: IF REJECT: send error to download address

User downloads an article

- BE: Scans blockchain for Download confirmation order
- BE: Download torrent using infohash
- BE: Decrypt downloaded file
- BE: Submit signed download confirmation to blockchain
- BE: Notify user
- UI: Read paper
- Blockchain: Confirm Download request order
- Blockchain: Confirm Download order
- Blockchain: Confirm Download Confirmation
- Blockchain: Send 2x download fee from user to editor
- Blockchain: Send 1x download fee from editor to user

Author: rout@

Draft complete for review: 1/25/2018