Proposal: Secure and Decentralized Chat and News Portal Based on the ZCash Blockchain

Motivation

Mass access to direct uncensored information and free media coverage of news events has long been an appealing idea. While restriction in freedom of speech may vary from country to country, a completely independent and safe news culture is yet a goal to be attained.

A secure and peer-driven news hub would encourage citizens across the world to engage in real-time issues that really matter to them, both on a local and on a global level, to challenge agenda setting and news presentation in classical media and to become an active part of public opinion shaping.

We propose a platform without censorship (e.g. caused by conflict of interest) to anonymously post news articles by sending a sequence of ZCash transactions to shielded addresses provided by the news page, by that protecting the identity of writers and at the same time making information permanently available through a decentralized structure.

To overcome article size limitations due to the 512-byte memo field length, we also propose a simple tool/library for data fragmentation over a number of n shielded ZCash transactions. We also plan to store media information on IPFS and embed the permanent links into articles published on the ZCash blockchain.

The encoding / decoding scheme will be implemented in a reference library for other people to use. It will also ensure that anybody is able to decode the news articles from the ZCash chain, so there is no SPOF in form of the news-website.

Also, the whole system will be designed in such a way that the news website itself will be fully redundant. We will develop and provide a local toolbox which enables everybody with a ZCash client to fully interact with the on-chain news portal.

Scope

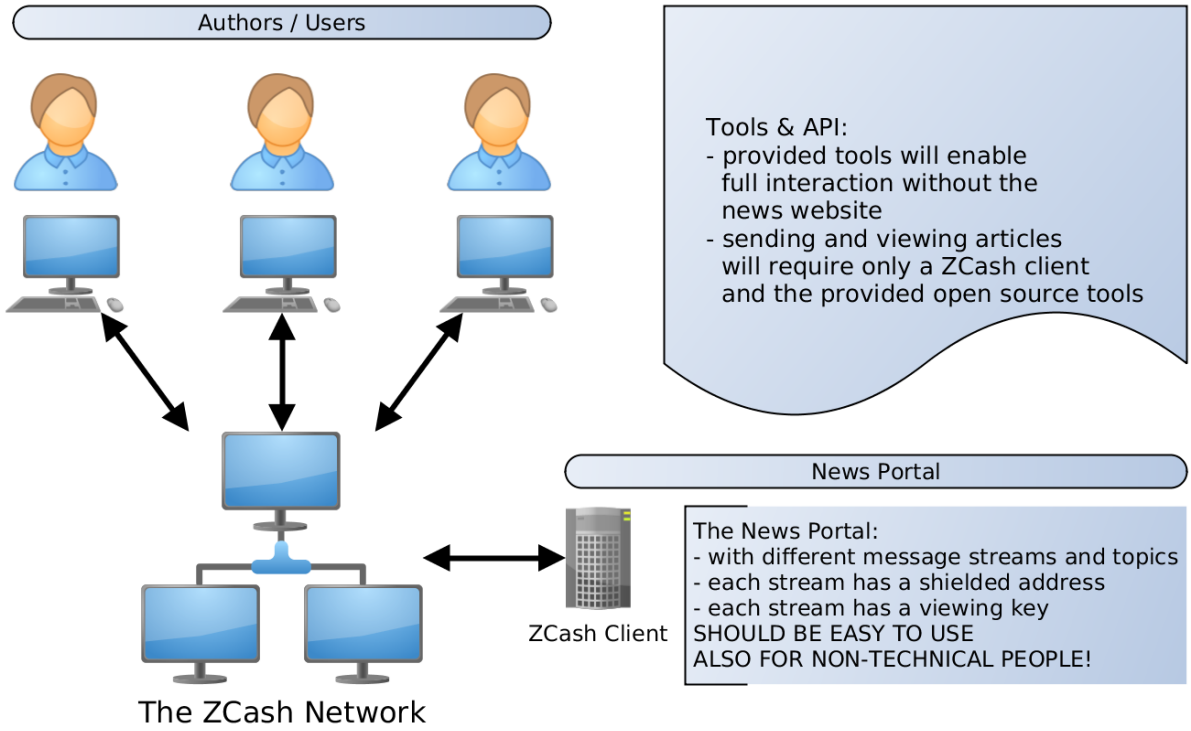
- We will build a secure and anonymous news/message platform using the ZCash blockchain as infrastructure, so people can freely post large news-articles as well as short twitter-like messages
- ZCash payments and donations to authors will be integrated
- A website, a local toolkit and an easy UI for reading and creating messages will be provided
- The platform will support features of social-network platforms (sharing of "tweets", up-/downvoting, quoting and replying to messages)
- A trust mechanism based on public keys will ensure anonymous user/author rating
- Using ZCash viewing keys, access to topics/tweets/articles can also be granted to a specific set of users, creating private/encrypted message areas

Technical Approach

The ZCash blockchain supports shielded transaction between shielded addresses based on zero-knowledge proofs [1]. Even though the blockchain itself is available to every peer in the p2p network, shielded transactions are stored in encrypted form. As a consequence, only users having a correct viewing key for the recipient shielded address are able to learn **some** information about the incoming transfer. Namely, the amount of coins received, and the content of an encrypted memo field passed along with the transaction [2]. However, the sending address, as well as the history of the coins sent is not disclosed. The memo field can be specified by the sender of any shielded transaction and enables transfer of a 512-byte payload. We consider the memo field as a secure vehicle for the applications and protocols we build on top.

Architecture Overview

The following will show the high-level architecture of our approach.



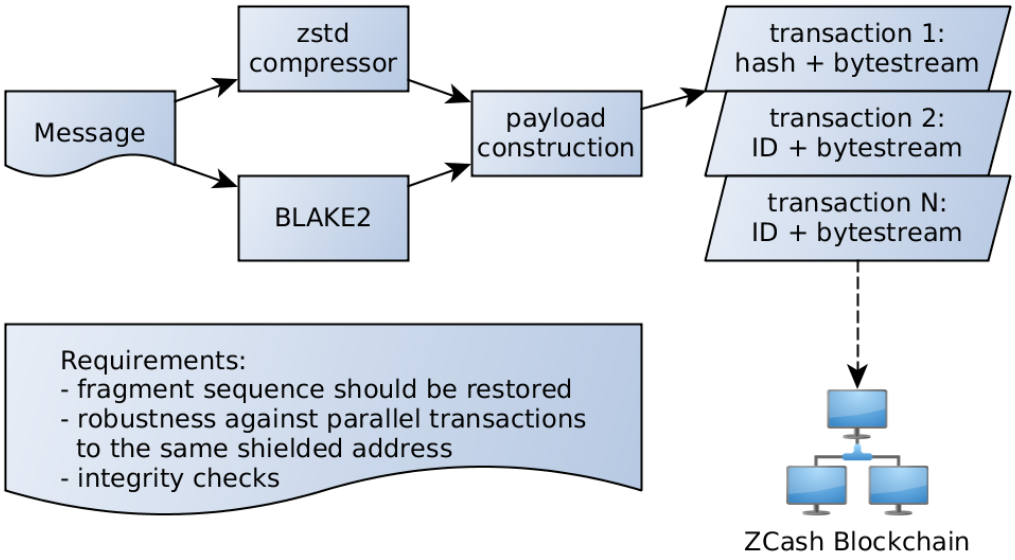
Core feature of our approach should be the redundancy of the news portal website itself. It will be merely a frontend to some encrypted area inside the ZCash blockchain. Therefore, we will provide all the tools necessary to interact with the on-chain news portal without ever having to visit the website at all (however, users can visit the website, if they like to and are permitted by their ISP). Just a local ZCash client and our provided tools should be sufficient. Fortunately, all this functionality can be implemented using ZCash viewing keys [2]. Also, for convenience, we plan to provide a JSON API for interacting with the website.

Message Fragmentation

Since we want to offer the most flexible way of sharing articles and do not want to impose any limitation to the maximum length of shared content, it is our goal to offer an opportunity of transmitting news articles of arbitrary length by splitting and separating them into multiple shielded transactions. We propose, and we will implement a procedure that will do the following steps (future changes possible):

- take an input message
- compute a hash of the message
- compress the original message with strongest compression parameters
- create a header containing the message hash, as well as IDs of the following transaction sequence
- append the byte-code of the compressed data in as many 512-byte blocks as necessary along with some sequence indicator or other meta information (to be clarified)
- send the full sequence of 512-byte blocks using the memo fields of shielded transactions

This procedure will enable sending large messages / news articles using the ZCash blockchain.



We will use the strongest possible compression parameters for zstd and we consider providing compression-dictionaries in a special topic on the ZCash chain, to reduce message size even further [5]. We will also let the authors decide to enable message compression or to compress message blocks separately (given as parameters to the fragmentation library), to prevent the possibility of compression-oracle attacks for third-party tools using the fragmentation library.

Transaction Time/Cost Estimation

As the actual amount of transmitted ZEC has no impact on our use-case, only transactions fees will have influence on the total amount of money spent for transmitting news articles. The time for sending depends on the speed of sending shielded ZCash transactions.

Example:
We take the first chapter of "Alice's Adventures in Wonderland" with the title "Down the Rabbit-Hole" which consists of 1350 ASCII characters. We execute the following command:

- *zstd -19 rabbit.txt*

After this we end up with a 710 byte compressed message length (using compression dictionaries, this can be reduced further). With an additional 32 byte from the BLAKE2 hash computation we get 742 byte. This can be transmitted (including meta information) in 2 ZCash transactions.
Let's assume a transaction cost of 0.0001 ZEC and an amount of 0.0001 ZEC to be transmitted in every transaction: 2 x 0.0002 ZEC = 0.0004 ZEC
With 1 ZEC equal to $250 USD during time of writing (2018/05/12), sending the message will cost $0.1 USD. With less transactions fees this can be reduced even more, however, $0.1 USD for a highly available and anonymous news article should be quite cost efficient. Also, some (small) cost associated with the service will discourage spam.
With the current ZCash reference implementation (1.1.1) and an Intel i7 4770K CPU we were able to send one transaction every 39 seconds on average. It would therefore take at most 1.5 minutes to send the given text stream. With the upcoming Sapling release, this should be even less of an issue [3].

Media Enrichment

Articles sent via ZCash transactions may contain immutable, permanent IPFS links to media files and syntax/tags for including them in the article; decoded by the website or the provided local toolbox. Storing of media content on IPFS ensures that it is always available and resiliently accessible through P2P network [4]. Like news articles stored on the ZCash blockchain, media files on IPFS will also serve the independent spirit of the world-wide web in full force. We will therefore elaborate procedures to combine IPFS content and ZCash news articles and check security implications of both combined approaches.

Web Frontend / News Portal

The website should be a "window" to the encrypted news-area within the ZCash blockchain. Therefore, it will be fully redundant, as all the functionality (submitting/reading of articles etc.) should also be possible using the local tools/libraries we provide. (For details, see 4. Software Packages)

Local Frontend / Local Library

If we just rely on a website for user interaction, this is a SPOF and the service could be easily restricted. However, by providing a public toolset for direct ZCash blockchain interaction, the service will be fully decentralized and very hard to limit.

Therefore, together with a local ZCash reference client installation, the toolset should provide all necessary features for submitting and reading articles, but also to sign and verify messages. Our goal is, that the main functionality of the news platform can be used without ever visiting the website for it. (For details, see 4. Software Packages)

Donation / Compensation for Authors

Every author will have a chance to send a private shielded address along with his article to receive donations for his work in a fully anonymous way. Therefore, using ZCash also directly as a payment method for donations, we provide an incentive for high-quality news and we support anonymous authors, since investigative journalism can sometimes be a risky activity.

Trust System

As news articles are submitted in a fully anonymous fashion, a moderation mechanism for articles on the website could be necessary (no moderation of the raw articles stored inside the ZCash chain is possible of course).

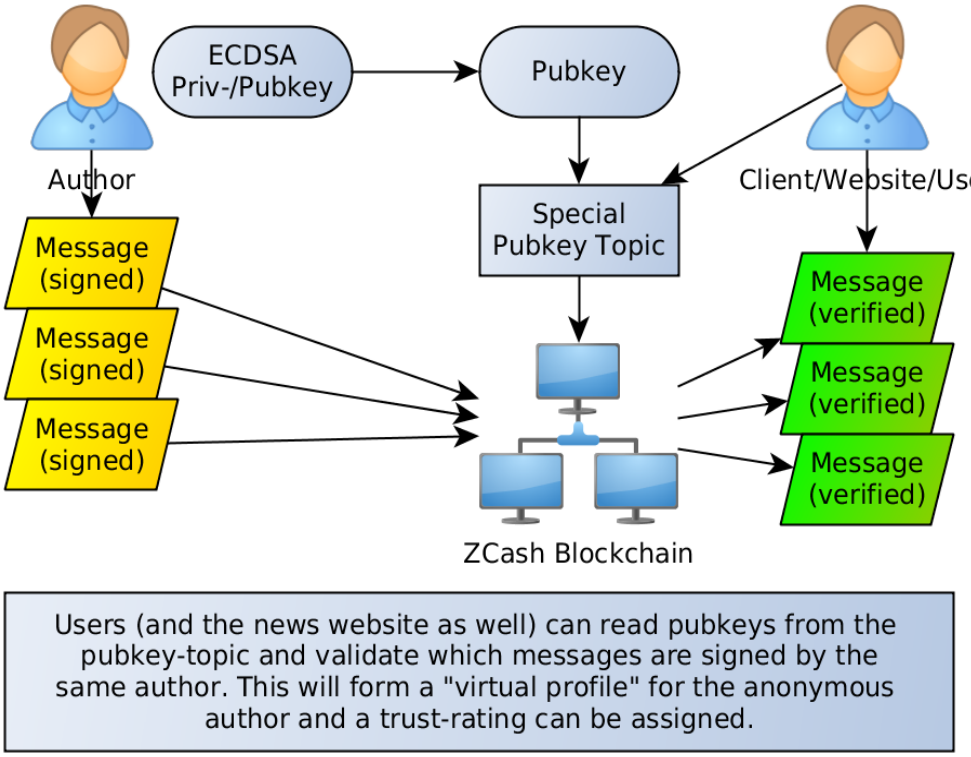
We will therefore investigate multiple reward mechanisms on the website:

- A simple up-/down vote mechanism where people can flag inappropriate content
- A pub-key approach, where anonymous authors can sign articles, so the website can verify which articles belong to the same anonymous identity. Based on this, a trust mechanism can be implemented. Example: As apparent from the article signatures, we know that anonymous user Alice has already published 40 news articles. This basically forms a virtual profile and all up-/downvotes are tracked for this profile. If now somebody claims to be Alice but is not in possession of her private key, this will be easily detected.
- A distributed moderation scheme, where peers randomly distributed over the world

(or a selected editorial board) will need to vote on messages of new(=untrusted) users to be shown on the page

- An algorithm that automatically filters inadmissible content with human dignity infringing pronouncements (e.g. using pattern matching, topic modelling, AI, …)

The nature and strength of these measures is in our view a major area of research for itself to ensure a balance between freedom of expression, fact based objective content and protection of human dignity.
Example of the optional signing procedure to assign authors a virtual profile for trust-tracking:



Risks and External Dependencies

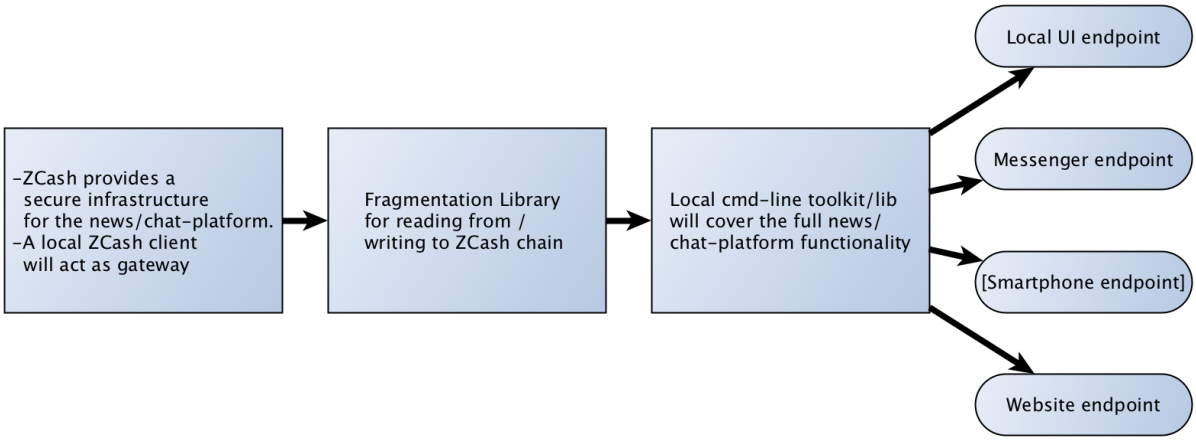
With today's technical condition of ZCash we see the project as feasible and straightforward. However, since the worldwide mobile internet usage exceeds the desktop one, better usability and scalability will only be achieved if ZCash transactions become mobile and the platform can be used device independently. The ZCash Sapling release will, however, definitely help a lot in this regard.

As stated above, in the second phase we plan to include IPFS into the news platform to be used in a natural way. As user protection is of utmost importance for this project and IPFS was not designed specifically for user privacy and identity protection, there might be some risks about this combination, which need to be evaluated down the road. However, this is not a critical point, as it is not affecting the project in general.

The biggest uncertainty for us is user adoption and the momentum of news/posts that can be published on the platform, as it offers completely anonymous and untraceable (and in the first step uncensored) posts of both text and media content on all possible topics and areas. Depending on usage and experience, both a trust system, additional usability and UI elements need to be added.

Software Packages

Packages Interaction



Packages Description

Package	***Description***	***Platform/OS/Dependencies***
Website Frontend	➢ Introduction of the service / education about functionality ➢ Main page with "trending / most interesting articles / chat topics / anonymous authors" ➢ How-to page with easy installation guide / security advice / download locations / API documentation ➢ WYSIWYG editor for writing and submitting articles / Settings for simultaneously sharing on other portals (e.g. Twitter) ➢ Signature page for author's public keys ➢ Topics- / chat- / article-interface to read news or chat messages and search for topics/articles/anonymous authors ➢ Up-/Downvotes, quotes, donations ➢ ZCash-Payment integrated virtual author profiles ➢ Social media connectors ➢ API for website interaction	Technologies: ➢ ZCash Client Software ➢ MySQL / PostgreSQL ➢ Python ➢ C++ 11 ➢ Flask ➢ Bootstrap ➢ AWS EC2-Instances ➢ BLAKE2 ➢ zstd-compression ➢ many more Supported Browsers: ➢ Chrome ➢ Firefox ➢ IE ➢ Safari ➢ etc. Supported OS: ➢ OS independent
Fragmentation Library	➢ As described under "Message Fragmentation", this package will transform arbitrary texts into 512-byte blocks to be sent via ZCash-transactions (compliant with ZIP 302 memo field specification). It will ensure message integrity using BLAKE2 hashing and sequence numbers. ➢ The interface should be generic to support arbitrary input data ➢ The package will include both: encoding and decoding part ➢ The package will include a library, as well as a simple binary cmd-line reference implementation to transform input files into 512-byte blocks and send them via the	Technologies: ➢ ZCash Client Software ➢ C++ 11 ➢ zstd-compression ➢ BLAKE2 hashing ➢ RPC communication Supported OS: ➢ Source should compile on Window, Linux, Mac, ARM ➢ Binary packages for Window, Linux, Mac, ARM

	ZCash-client	
Local cmd-line Toolbox	➢ Submitting / reading chat messages and articles ➢ Reading How-to-Page on ZCash chain ➢ Access to topics, creation of topics ➢ Up/-Downvotes ➢ Message signing ➢ Verification of articles / messages ➢ Requesting anonymous user profiles and popularity ratings ➢ Sending ZCash donations and payments to anonymous authors ➢ Convenience features: e.g. most popular articles, anonymous authors, newest activities etc.	Technologies: ➢ ZCash Client Software ➢ C++ 11 ➢ Python ➢ Fragmentation library Supported OS: ➢ Source should compile on Window, Linux, Mac, ARM ➢ Binary packages for Window, Linux, Mac, ARM
Local UI	➢ The local UI will focus on providing an easy interface to all the functionality provided by the "Local cmd-line Toolbox" ➢ It should have the look-and-feel of a messenger with supported features (e.g. syntax highlighting) for article/message formatting and special tags for embedding URLs, images, media content, signatures, etc.	Technologies: ➢ ZCash Client Software ➢ C++ 11 ➢ Python ➢ Local cmd-line Toolbox ➢ Messengers: e.g. Pidgin, Slack etc. ➢ UI and WYSIWYG packages: e.g. markdown, wxwidgets, QT Supported OS: ➢ Source should compile on Window, Linux, Mac, ARM ➢ Binary packages for Window, Linux, Mac, ARM

Proposed Steps for Increasing Platform Popularity

As the proposed platform will have news-platform functionality as well as the ability to post twitter-like short messages, we consider the following steps essential for making the platform virally popular:

Technical Actions:

- Giving articles/chat-messages color: emojis, URLs, images, media links
- Since ease of use is extremely important, a special platform independent client is necessary (stand-alone tool, and/or plugins to Slack/Pidgin and other messengers)
- The virtual profiles that we outlined above will improve user experience (up-/downvoting; donations, ZCash payments, trust ratings)
- Payment integration to send ZCash to those virtual authors, which incentivizes quality content and posts
- Additional twitter like features such as retweet/like/quote
- Automatic posting mechanisms to various sites: a „real“ twitter account that posts new topics and headlines from z-board; but also posts to other social media platforms
- Sharing of posts/articles on other platforms
- Strong focus on security, anonymity and availability (e.g. fully redundant website)
- API, RSS feeds, easy technical and non-technical documentation
- Easy integration into third-party tools (e.g. well documented and convenient local toolkit and python packages), so the provided service can be easily adopted by other applications that will run on top
- Private topics

Marketing:

- Referencing interesting web-articles in news-articles/chat-messages
- Posting interesting content
- Creating special topics (books, movies, recipes, tech papers, animals, etc.) that people find useful and which encourage conversations
- Convincing specialized target groups to actively use the platform e.g. Z-Cash developer community, especially the target groups for whom anonymity is crucial for their activity
- Sharing tweets/articles
- Giving conference talks and explaining the benefits (e.g. compared to twitter)
- Cross-linking between social media platforms
- Taking the statements: privacy, anonymity, availability, immutability and building a marketing strategy on top:
    - the platform could have topics which are an "immutable log“ of twitter messages, or news articles from other websites, thereby „saving fragile information from censorship-prone twitter into the immortal and secure ZCash chain“
    - “we protect information from censorship”, “freedom of news”, etc.

Team Background and Qualifications

Thomas S.: M. Sc. Computer Science (IT-Security, Machine Learning, Neuroinformatics and Cognitive Robotics), 7+ years as software Developer, Consultant and Manager in different positions in Big Data, Machine Learning and Blockchain technology

Kirill R.: M. Sc. of Finance Mathematics (Insurance and Finance Risk management, ECC, Optimization), 7+ years as Developer, Actuary, Product Manager, Online Marketer, responsible for Blockchain strategy in a large German insurance company

We have been dealing with blockchain technology and cryptocurrencies for more than 5 years. As we believe in innovation and disruptive power of technology, both of us quit our permanent jobs in companies and have been working full-time on our own projects since the beginning of the year. Together we have already launched two projects (currently in the live test stadium) - automated arbitrage trading on cryptocurrencies and advanced recurrent neural network architectures for automated trading on crypto exchanges.

Also, we are the authors of www.z-board.net (an anonymous chat platform based on the ZCash blockchain), which we built as a first prototype to tackle our much greater vision that we outlined in the current article.

Evaluation Plan

We will provide community updates on regular basis according to our schedule.

However, the best evaluation of this PoC can be done by testing and using our service and by posting articles. We look forward to a detailed feedback on platform functionality and we will gladly include this in our work.

Also, quantitative KPIs for user behavior, such as number of visitors, average time spent on page, number of messages read, number of new articles per week etc. as well as qualitative KPIs such as user feedback on usability and functionality will be defined. We aim at a continuous monitoring concept for future developments.

Security Considerations

Our approach will of course use ZCash not solely as a currency. Thanks to decentralization, availability, integrity and most of all the privacy offered by the ZCash infrastructure, it is a secure basis to attack censorship and restrictions in freedom of speech. On the other hand, the solution can only be as secure as the underlying framework.

All tools will be designed in a way that makes the main news website fully redundant, so it cannot be a SPOF. Therefore, only a local ZCash client as well as our provided toolbox will be sufficient to interact with the news platform. We consider this a strong security benefit.

In addition to the underlying security aspect, it should be an easy to use solution, which can be used by masses not yet been exposed to Blockchain technology.

Implications of using IPFS media embeddings in posts need to be researched separately in phase 2.

Schedule

We have divided the full implementation into three stages (work-packages):

1. Technical Core Implementation and PoC (3-4 months)

- Message fragmentation over ZCash memo fields
- Publication of "fragmentation-library" as general-purpose package for data transfer over multiple ZCash transactions
- Backend development and server platform (e.g. creation of shielded addresses and viewing keys, database work, API, hardening, https, etc.)
- Website frontend for news articles, topics, how-to-page, signature-page
- Client tools with UI for reading and composing messages locally (Win, Linux, Mac)
- Procedure for users to sign messages and verify message integrity
- Topic creation via special ZCash transactions
- Compression dictionaries on special topic
- Website JSON API (e.g. retrieve all topics, get all messages, search for articles, etc.)
- Include shielded address for authors to receive donations

2. Usability for Authors: Advanced Editor for Writing and Posting Messages; Usability for Readers: Trust System and Customization Options for Displaying Relevant Articles (3-4 months)

- Implementation/parsing of special message tags to display images/links/code etc.
- Provide WYSIWYG editors for authors with embedded article functionality (e.g. images, links, IPFS content)
- Research interoperability and security of IPFS and the news platform and implement if reasonable
- Validate and implement trust system using ECDSA public keys to identify "anonymous virtual user entities" together with a trust-/rating-system on top
- Private chat/news areas using ZCash viewing keys
- Backend scaling
- Special article search features / sorting
- Social media connections (e.g. Twitter) to post headlines for trusted anonymous users
- Migrating to newer ZCash versions
- Platform and server maintenance
- Security auditing
- [Evaluate possibility of a smartphone app]

3. Additional Ways to Extract and Read Messages from the Blockchain to Ensure better Accessibility and Availability Regardless of Censorship and Local Restrictions

- To be defined, but depends on user adoption, previous development and can only be roughly estimated at this point

Budget and Justification

Since we have already spent some time on designing and testing the core function, we estimate around $38k to be sufficient to support our development for the first stage in the next three to four months.

To make the proposed solution attractive and user-friendly also for tech-averse users and in order to reach global distribution, we consider the second part as an integral part of this project. In our view, the second stage is also time-consuming and needs ideally permanent maintenance, improvement and development. For this we have to introduce additional developers and we estimate an effort of $42k for 3-4 months of work.

The third stage can be implemented later if we see that the platform finds adoption and is well used. Costs cannot be estimated at this point.

The work-packages are self-contained and can be processed in sequence. Therefore, if 80k (package 1 + 2) are a too large amount for funding, or are considered unfair to other Grant Program projects, we would like to only apply for funding of the first work-package (38k) during the current round, as we could submit the second work-package later to a next round in the future.

Direct Contact

Thomas S.:

- E-Mail: thomas.schmiedel.at.work@gmail.com

Kirill R.:

- E-Mail: kirill.rubinstein@gmail.com

References

[1] https://z.cash/technology/zksnarks.html
[2] https://blog.z.cash/viewing-keys-selective-disclosure/
[3] https://medium.com/@_garethtdavies/why-zcash-is-set-to-shine-in-2018-2e8c388f35fd
[4] https://ipfs.io
[5] https://github.com/facebook/zstd