

xxRFP-2022-014

Public Square (Private Twitter) Platform

This xx foundation request for proposals will fund the creation of an open-source tool for a Twitter-like platform, henceforth referred to as Private Twitter. This is intended to replicate the basic functionalities of Twitter, such as following, posting and replying to content, and possibly even re-tweet or like functionality.

The xx foundation is offering a competitive bidding process to complete this work, with independently compensated phases. The [selection process](#) will analyze each phase individually according to the response contents, the proposed compensation, and the proposed final deliverable dates and contents. Proposers are expected to propose for all phases.

Requirements

Private Twitter consists of a client-side app that leverages the decentralized broadcast functionality of the xx network to create streams of comments, and a server side component which receives, handles, and stores a directory of public comment streams.

Users should, at minimum, be able to create and reply to tweets, as well as follow other users as a feed. The proposal should outline additional functionality options such as image file submission, liking, and retweeting. Users should be able to post with a public identity that they can configure and connect with one another, possibly by making a clone of [User Discovery](#) that can be used for searching and connecting users to one another.

The identity of the sender communicating with the server will remain anonymous aside from their public identity. The server should be able to respond to the user with any errors that occur.

Proposals must discuss mechanisms by which repeated spam postings can be prevented and associated security implications. Furthermore, scalability and consistency are two important factors of any medium to large scale message board and thus there should be a plan outlined for supporting larger numbers of users.

Include designs for a UI for profile registration, browsing user feeds and posting/replying to tweets.

Architecture

When the user opens the app, it should establish an identity and automatically negotiate a relationship with the server, which is also a cMix client. The user should enter their profile information and attempt to register this data with the server as well as the User Discovery

mechanism of choice, likely via the [REST-like](#) package and backed by the [authenticated connection implementation](#) in order to [prove their identity to the server](#). When a registration request is received, the server should validate the data (e.g. username uniqueness) and return success or failure to the client.

Once the registration completes successfully, the client should create a new channel on the network and store its associated information - this will serve as the user's 'feed'. The client should then relay this channel information to the server, at which point the data will be stored so it can later be distributed to other users, allowing others to 'follow' one another by joining their respective channels.

The owner of each channel can send to it asymmetrically in order to create a 'tweet' that other users may view if they have joined the channel. Furthermore, other users can send to this symmetric channel in order to reply to tweets on one another's feed. Users that reply to another's feed may rebroadcast their reply on their own feed in order to notify their own followers.

Phases

You may propose your own phases, but the following phases are desired:

Phase 1: Proof of Concept—Implement the basic version of your proposed design using cMix channels and submit a final design for the client-side and server-side functionalities. This should include final versions of any cryptographic primitives and fully explained versions of all data structures and sub-protocols, as well as discussions on storing and serving forum data.

Phase 2: Command-Line Tools—Build a server client that can receive anonymous messages and serve content received over cMix to a user interface as a forum. Include a fully functional client-side command-line tool that exercises your proposed system without a user interface. The client-side command-line tool will be the final library used by the Android and iOS apps and have test coverage of at least 85% of the code base. Include an accompanying continuous integration/continuous deployment (CI/CD) styled testing script.

Phase 3: User Experience - Build the applications that users use. Each will have its own milestone

- **Android App**—Android app with the final user interface design using the library from Phase 3.
- **iOS App**—iOS app with the final user interface design using the library from Phase 3.

Submission Instructions

Proposers should submit their proposals, in English, to the following website:

- <https://xxfoundation.org/archive/xx-foundation-announces-the-xx-dapps-grant-program>

Note that proposals are divided into two parts: An anonymized technical proposal and a staffing proposal. The technical proposal will be posted online and should not contain any identifying information about your organization or staff. The staffing proposal will contain resumes and additional evidence for why you and your team are qualified to do the work you propose.