

# xxRFP-2022-010

## Pirate Radio

This xx foundation request for proposals will fund the creation of Pirate Radio, an open-source tool that can broadcast audio to an audience anonymously over the xx network. This app is similar to other [internet radio](#) services in that the message is streamed as audio over the internet. However, it is more akin to traditional [radio broadcasting](#) in that both the broadcasters and listeners are anonymous (except where radio broadcasts are legally required to identify themselves). Also, Pirate Radio over the xx network is guaranteed to be encrypted.

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

Pirate radio is a stand-alone tool that facilitates the broadcasting and reception of streamed audio over xx network while maintaining strong anonymity for both the broadcasters and moderate anonymity for listeners.

The tool has three main parts: the listening app, the broadcasting app, and the server-side software. Proposals should discuss whether to build a single app or separate apps for listening and broadcasting taking into account which platforms it needs to run on. A broadcasting app may need to run on desktop operating systems while a listening app may only need to run on a phone or a web app.

The listening app should allow users to connect to a radio station by selecting it from a public list or loading a pirate radio channel from a file. Users should also be able to save and switch between their favorite stations easily. In addition, broadcasters can allow their station to be publicly available or only accessible by sharing a private radio channel. Consider possible methods of sharing a private channel (e.g. QR codes, a file).

The broadcasting app should provide the necessary tools needed to record and broadcast an audio stream.

The third tool, the server-wise software, is optional and an addendum to the core system. It will handle radio station discovery and will be a cMix client running on a server. Broadcasters can choose to list a station on the public server for users to find. All listing and discovery occur over cMix, but broadcasters can choose how much metadata they want to expose to make it easier for users to find them.

An additional optional feature to consider is the ability for a listener to anonymously send a message to a broadcaster.

Proposals should include a draft of a user interface for both the broadcasters and listeners.

## Architecture

Radio channels will use asymmetric [broadcast channels](#), which will allow a broadcaster to anonymously broadcast to many listeners. The generated channel file is used to connect to the station.

There must be a layer to handle the streaming of audio over the channel. This will involve significant compression—proposals should discuss which compression methods will be suitable for allowing human-voice to be audible and discernible while remaining within the bandwidth limitations of the network. Also take into consideration network failures (rate of 0.05%) and discuss methods of including redundancy in the audio transmission.

The server-side software will be a cMix client where anyone can register a channel to be a radio station. Communication with the server should be anonymous so that a broadcaster or listener cannot be associated with a station. To do this, use either [single-use messaging](#) or creating a new ID and using [connections](#).

## 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 and submit a final design for the listening app, broadcasting app, and server. 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 compressions and redundancy for the audio transmission.

**Phase 2: Demo**—Get a basic radio station functionality working in a barebones demo that demonstrates a one to many broadcast.

**Phase 3: Command-Line Tools**—Fully functional client-side command-line tool which fully exercises your broadcasting and listening system without a user interface—this does not need to include the server. The client-side command-line tool is the final library used by the final apps, with test coverage of at least 85% of the code base and an accompanying continuous integration/continuous deployment (CI/CD) styled testing script.

**Phase 4: Web App**—Build a web application that demonstrates the broadcasting and listening functionality.

**Phase 5: Radio Discovery**—Build a server client that a broadcaster can upload a radio station file to and that a user can browse and find new stations.

**Phase 6: Apps**—Build full fledged apps that run on phones and desktops.

## Submission Instructions

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

- FULL ADDRESS HERE

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.