

CList

A personal learning and communications environment that runs in your browser

clist.mooc.ca

What is CList?

A personal learning and communications application that runs in your browser.

- Read feeds and search the web
- Annotate pages and follow the people you learn from
- Write once and publish to Mastodon, Bluesky, WordPress, and Blogger
- Chat and collaborate with others

CList runs in your web browser, not on a company's servers, so no single company controls it. It is built out of simple components so nobody can own it.

Read more: [Architecture](#)

What you do with CList

CList is for reading, writing, and connecting with others who are doing the same.

- Read feeds, accounts, and searches in one place
- Write posts and publish them to the platforms you use
- Save what you find into collections, and annotate pages as you read
- Chat with people you find, and edit documents together

Learning happens when you read, respond, and share with others, so CList brings those activities together in one place.

Read more: [Architecture](#)

Decentralized and distributed

CList has no central server. Its parts run in many places.

- Each person runs their own copy, hosted on any web server or on their own computer
- Services such as kvstore, collab and discussions can each run as many separate instances
- Your CList connects only to the service instances you choose
- Peer-to-peer chat connects browsers directly once two people have found each other

No central server is required, so no single operator can switch CList off or change its terms for everyone.

Read more: [Architecture](#)

The parts of CList

CList is made of separate parts, and any of them can be swapped out.

CList web client

Static files on any web host, or on disk

Annotation server

Stores your annotations; any instance works

Peer discovery (discussions)

Lets chat peers find each other

Third-party platforms

Mastodon, Bluesky, WordPress, Blogger, AI APIs

CList desktop app

The same files for Windows, macOS, Linux

Feed converter (opml2json)

Reads OPML lists and aggregates feeds

Collaborative editing (collab)

Real-time shared documents

Account server (kvstore)

Encrypted accounts, login, DID documents

Cross-domain proxy (proxyp)

Reaches APIs that block browser requests

Pastebin

Publishes collections as shareable links

Each part can be replaced, so no single operator is required. You can mix and match these services using any providers you wish.

Read more: [Architecture](#)

Your data belongs to you

Your data is yours, and you can move it between account servers.

Create

Register with clist.mooc.ca or another account server, or host your own.

Connect

Add the services you use, such as Mastodon, Bluesky, WordPress, and Blogger.

Use

Read, write, and chat from any CList instance. Your settings follow you.

Your data is encrypted with a key from your password, so only you can read it. You can move it to another account server, including one you host yourself.

In the near term we will be adding automated data migration to make the process of moving even simpler.

Read more: [Accounts](#)

Privacy and security

Your password and encryption key never leave your browser.

- Passwords and keys are encrypted in your browser before they reach the account server
- The account server stores only encrypted data
- Loading an item checks its annotation server, which sees the item's address and your IP
- Looking up a person's identity document reveals your IP and interest to its host
- Social platforms see your IP address and the API calls made with your own credentials
- The project publishes its privacy notes, including known gaps

CList keeps no reading history on a server, so you decide which services to connect and can replace any of them. A copy of your data on a server is unreadable without your password, so a server breach exposes much less.

Read more: [Privacy model](#)

Accounts

An account is the credential CList needs to read or publish to one service.

- Each account has a type, such as Mastodon, Bluesky, or OPML
- Account information is encrypted locally before it is stored
- Permission flags decide where an account is used: reading, posting, editing, and so on

Each account is a separate credential, so you can change or remove one service without affecting the others.

Read more: [Accounts](#)

Your identity

Your DID is a public identifier that belongs to you, not to any one platform.

- Your DID looks like `did:web:kvstore.mooc.ca:users:alice`
- The key pair is generated in your browser, and the private key is stored encrypted
- Your social accounts can be linked to your DID, so others can find them

A DID is an identifier you can keep even when you change keys or services, so your identity does not depend on one company's account.

In the Me panel, you choose which accounts appear in your public DID document. Chat messages are already signed with your key; signing published posts is future work.

Read more: [Identity \(DID\)](#)

Where CList runs

The same application can run from three places.

Hosted site

clist.mooc.ca, in any modern browser.
The default, and the simplest way to start.

Desktop app

A download that starts a small local server on your computer and opens CList in your browser.

Saved files

Unzip the files and open index.html directly.
No install. Logging in to a remote Account Server is not yet supported this way.

The CList client runs on any web host or on your own computer, so you are not tied to one site.

In the future we'll provide a way to clone an entire server with all the CList service applications, so you can self-host everything yourself, or provide services to your school or institution.

Where your data lives

Choose an Account Server, or keep everything in Local storage.

Account Server

Encrypted data on kvstore.mooc.ca by default. Available from any device and any CList instance.

Local storage

Everything stays in this browser, and no server is involved. Collaboration, following, and chat need an Account Server.

You choose where your account data lives: a server you trust, or this browser only. No one option is forced on you.

Free and easy data and service migration is a priority area for future work.

Read more: [Accounts](#)

The CList multi-pane interface

Two main panes, with a source selector on the left and publishing controls on the right.

Left pane

Select the accounts and sources you read from.

Read pane

Read the items from the sources you selected. Send items and references from here to the write pane.

Write pane

Compose and edit. Items and references sent from the read pane appear here.

Right pane

Define where and how you publish or save what you have written.

Learning tools need to support both reading and writing, and connect both directly to the community of other people who are reading and writing too.

CList also runs on mobile; you swipe left or right to move from pane to pane.

Read more: [UI components](#)

The left pane

The left pane chooses what you read and which accounts CList uses.

- Select the accounts and sources you read from
- Open feeds, social accounts, searches, and people you follow
- Manage accounts and your identity, and reach Options, About, and login

Reading starts with choosing sources, so every source sits in one place where you can add, remove, or switch it at any time. CList already supports a variety of social service types, with more planned.

Read more: [UI components](#) · [Feed structure](#)

Adding feeds

Read any RSS feed, or import a whole list at once.

- Add a feed by pasting its URL
- Import an OPML file from another reader to add many feeds at once
- Feeds you add appear in the feed menu, where you switch between them

RSS and OPML are open formats, so your subscriptions are not locked into CList and can move to other readers.

Future plans include importing from more reader formats and keeping feed lists in sync across devices. A separate project, LocalRSS, is implementing this.

Read more: [Architecture](#)

Reading from social accounts

Connect a social account to read its timeline alongside your feeds.

Mastodon

Read your Mastodon timeline, and reply to posts.

Bluesky

Read your Bluesky timeline, and reply to posts.

RSS and OPML

Feeds from any site that publishes them.

Many people already have accounts on these platforms, so CList reads them where they are rather than asking everyone to move first.

This is just a sample of the services available in the fediverse. Others – everything from Pixelfed to Misskey – are planned for inclusion. It's an open question whether they'll be supported via their APIs or via the fediverse protocols.

Searching the web

Search without leaving CList.

- DuckDuckGo and Google searches are built in
- Google can also search for images (with limitations)
- For learners, access to an OER repository is included, with access to more planned

Search finds material to respond to without leaving CList, so reading and writing stay in one place.

Future search could add saved searches and search across your own annotations and collections.

Annotations

Attach a note to any web page or feed item.

- Annotate while you read, and your note appears on that page or item
- Annotations are stored on an annotation server, annotations.mooc.ca by default
- Each annotation is tied to your decentralized identity

A note written while reading is the moment a reader becomes a writer. Keeping it attached to its source keeps the thinking connected to what prompted it.

Read more: [Annotations](#)

Reading annotations

See what the people you follow have written about the same things.

- Annotations from people you follow appear on the items you read
- Open a thread to read the discussion and reply

Seeing what people you follow wrote about the same item turns reading into a shared conversation, not a solitary one.

When you are signed in, you see your own annotations and those of people you follow, as long as they have shared them publicly. Nobody is presented with unwanted content – not even annotations. This is a core value of CList.

Read more: [Annotations](#)

Following

Follow people, and see what they annotate.

- Follow people from the person-add button beside their annotations
- Their annotations appear on the items you read
- Pass along an annotation you value by annotating it again – this is called flowing

You decide whose annotations you see. Following people, rather than a feed ranking, keeps the social side of learning under your control.

Future following could support lists or groups of people so you can follow by topic as well as by person.

Read more: [Annotations](#)

How to follow someone

Following is one click on an annotation you read.

- Open an item and look at the annotations on it
- Click the person-add button beside an annotator's name
- Their annotations now appear on the items you read
- To stop following, open Me, then Following, and click Unfollow

Your follow list is stored encrypted in your own account server, and only you can read it. The person you follow is not told. Only people with a DID identity who share their annotations publicly can be followed.

Read more: [Annotations](#), [Identity](#)

Sharing your DID in feeds

Your RSS or Atom feed can say who you are, so people can follow you.

- Add an author link to your feed that points to your DID
- Anyone reading the feed can find your DID and follow you from it
- The DID points to an identity document that you control
- The WordPress plugin and Mastodon profile fields can already publish a DID
- CList's feed reader does not read these links yet

This is planned as another way to find people to follow. The feed carries the DID, so you follow the person, not a platform account.

Read more: [Adding DID to RSS and Atom feeds](#)

The read pane

The read pane shows the items from the sources you select.

- Read the items from the sources you selected
- Annotate items, and see annotations left by people you follow
- Send items and references to the write pane

Reading and writing sit side by side, so taking something in and responding to it take one move, not a change of tool.

Today there's one basic type of read pane, showing a list of resources, but future read panes will allow for different types of display and data design.

Read more: [UI components](#) · [Feed structure](#)

How items are presented

Every source's items appear in the same object model, no matter what service they come from.

- Items appear as a list of entries in one standard format
- Each service supplies its own status actions, such as like, share, or reply
- If a service does not offer an action, CList does not show it

A common format means you read any source the same way, and the actions you see match what each service can actually do.

A standardized object model enables interoperability between various types of data, allowing multiple types of resource to be presented in the same environment.

Read more: [Feed structure](#)

Working with items

Every item can be read, annotated, and used as a source.

- Open an item to read it in the pane
- Load items into the write pane to reply to them or cite them
- Annotations left by others appear on the items you read

Each item carries its next steps (save, annotate, share), so responding is one click away. Reading, thinking, writing and sharing are all part of the same action.

Read more: [Feed structure](#)

The write pane

The write pane is where you compose and edit what you will share.

- Write in plain text, a rich-text editor, or a shared editor
- Load items from the read pane to reply to them or cite them
- Tags and references are recorded as you write

Writing next to your sources keeps credit attached to the words, so replies and citations are recorded as you write them.

Some future plans for the editor: multiple content formats, including diagrams and images; multi-page documents; version control.

Read more: [Editors](#)

Loading items into the editor

Anything you are reading can be loaded into the editor. You can load multiple items from multiple sources into the same document – this is convergence of CList.

- Load an item from the read pane into the write pane to reply to it or cite it
- Load several items into the same post, converging many sources into the single thing you are writing
- CList keeps a reference list of every item you load, with its URL, author, and service

Loading many items into the same document allows you to bring ideas together; it's not just a thread that breaks off into various unrelated topics.

CList keeps track of your sources for attribution. Future work will include integration with services like Zotero and automated bibliography creation.

Read more: [Editors](#) · [Convergence](#)

Writing and editors

Write in plain text, a rich-text editor, or a shared editor.

- Plain text for simple posts
- TinyMCE for rich formatting
- Collab, CList's own shared editor, for writing together in real time
- Etherpad, a separate shared editor, for writing together with others

Different writing tasks need different editors: plain text is quick and portable, rich text handles formatting, and shared editors let groups write together.

Future editors could add more formatting options and export directly to common document formats.

Read more: [Editors](#)

Collaborative editing

Write a document with others, in real time.

- Share a link for co-editing, or for read-only access
- Invite people from a chat, or start from a collection or recent chat messages
- See who else is in the document and where their cursors are

Shared editing lets groups draft together in real time. Read-only links let people review a document without changing it.

There is a stand-alone version of the collaborative editor so people without CList accounts can participate. The collaborative editor can also be used side-by-side with peer-to-peer chat, which you can use to invite people into the document.

Read more: [Architecture](#)

Tags and references

Keep track of where things came from.

- Tags help you organize the items you read and write
- A reference list records each source: its URL, author, and service
- The reference list travels with your post and drives replies and notifications when you publish

Recording the source of each post keeps credit visible and sends replies and notifications to the right place.

Possible next steps include a richer citation format and a view that lists everything a post refers to.

Read more: [Editors](#)

One format for every editor

Editors exchange content as HTML or plain text, so switching editors can change your formatting. The same object model is used for all edited documents.

- Each editor declares whether it works with HTML or plain text
- The same post can be sent to several publishing platforms
- Switching from rich text to plain text removes formatting

Not all editors support all formats. The editor shows what it can, and doesn't touch what it can't.

There is a rich source of innovation to be found in potential future CList editor design.

Read more: [Editors](#) · [Publishing](#)

Loading content and templates

Start a document from something you already have, or from a blank page.

- Load from a file, from a URL, or from Dropbox
- Load a collab document or one of your pages
- Load blank to start from nothing

If you have an AI account, Generate template drafts a starting structure for you.

Reusable templates are planned future work.

Read more: [Editors](#)

Multiple roles for content

One post can be a reply, a reference, and a notification at once.

Reply

Replying to a Mastodon or Bluesky post publishes your post as a threaded reply to it.

Notify

If WebMention is on, the sites you write about are notified automatically.

Reference

Each source's URL and author are carried along, so readers can trace where your post came from.

Real writing connects to other people's work. One post can reply, cite, and notify at once, so those connections are recorded automatically.

Again, it's a core value of CList to show you only content you actually choose to see. You define this by the people (or in the future, groups or other organizations) of people you follow.

Read more: [Convergence](#)

The right pane

The right pane decides where and how your content is published and saved.

- Publish to the platforms and accounts you choose
- Save to a file, or to a collection in your account
- Publish a collection as a shareable link

Publishing and saving are separate choices, so you can keep work private for as long as you like and share it when you are ready.

Future work could add scheduled publishing, so a post goes out at a time you choose.

Read more: [UI components](#) · [Collection publishing](#)

Publishing

You can publish one post to several platforms at once.

- Choose the publishing accounts for each post
- Supported platforms include Mastodon, Bluesky, WordPress, and Blogger
- If you save a long post, social networks will post a link to the full post in your blog

Writers reach several audiences on different platforms. Publishing to them all is a common practice, but cumbersome right now

Read more: [Publishing](#)

Hypothesis

Connect a Hypothesis account to read and write annotations alongside CList's own.

- Add a Hypothesis account under Accounts, with your username and your personal API key
- CList shows annotations from you and from people you follow who list a Hypothesis account in their DID
- When you choose Hypothesis as the destination, your new annotation is posted there

Hypothesis is an established open annotation service. Connecting to it means CList works with notes people already keep there, rather than starting a separate system.

Webmentions

Let the sites you write about know that you wrote about them.

- WebMention is opt-in and off by default. Turn it on in Options
- After a successful publish, each URL your post references is notified
- A failed notification never blocks your post

Webmentions are an open standard that tells a site you linked to it. It depends on the receiving service supporting and passing along your mention.

Read more: [Convergence](#)

Collections

Gather items into collections you can edit and share.

- Create a collection and add items from your feeds or searches
- Edit titles, reorder items, and remove items
- Collections are saved to your account

A collection lets you gather and shape items before sharing them. Future work may involve pooled or shared collections.

Read more: [Collection publishing](#)

Publishing a collection

Turn a collection into a shareable link.

1. Open the collection
2. Choose Edit & Publish
3. Pick a format: RSS, OPML, JSON, HTML, Markdown, or Text
4. Publish, then share the link

Offering several formats (JSON, OPML, HTML) means people can use a collection in whatever tool they already have.

Read more: [Collection publishing](#)

Pages

A published collection is a public page that anyone can read.

- Publish a collection and it gets a public web address
- Pick RSS, OPML, JSON, HTML, Markdown, or Text
- My Pages lists everything you have published, with links to open each one
- Planned: publish any document from the editor as a page

A page is a file on a public server, not a private copy. Anyone with the link can read it, so do not publish anything you want kept private.

Read more: [Collection publishing](#)

Saving and loading files

Save your work to a file or to Dropbox, and load it back later.

Your computer

Save to a file on your own computer, and open it again whenever you need it.

Dropbox

Save to and load from your Dropbox. You grant CList access once.

Your work should be portable. Saving to a file you control, or to a service you already use, means it is not locked into CList.

Possible next steps include saving to more storage services and automatic backups of your saved files.

Read more: [Editors](#)

Peer-to-peer chat

CList supports direct peer-to-peer chat.

- A discussion board, discussions.mooc.ca, lists who is currently available
- PeerJS handles the connection between the two browsers
- After that, messages travel directly, not through a central server

Chat messages go directly between browsers, so no central server has to hold them. Open chat rooms can be advertised via a discussion board server.

There are many synchronous chat platforms that could play a similar role if integrated into CList, such as Discord and Matrix.

Read more: [Peer-to-peer chat](#)

The desktop app

CList can run on your own computer, without a web host.

Windows, Mac and Linux

Download the prebuilt executable file from the [GitHub release page](#).

Some people want CList on their own computer, without a browser tab or a hosted site. The desktop app runs the same files locally. The application includes a built-in server so it works seamlessly with all CList services.

Read more: [Desktop app](#)

Running from saved files

You can open CList straight from a folder on your computer.

- Your accounts are stored in this browser, not on a server
- Some features that rely on a remote server need the hosted site or the desktop app
- Back up your data before you clear the browser or move to a new computer

Data stored in the browser stays on your computer by default. The trade-off is that backups are your responsibility.

Not everything works when CList is run from plain HTML files on your desktop, though we're working to make it as full-feature as possible.

Hosting your own CList

CList is a set of static files, so you can host it anywhere that serves files.

- Copy the files to any web host, at any address
- Point CList at the kvstore and annotation services you trust
- Run your own kvstore to keep your credentials and DID on your own server

Because CList is static files, hosting it is inexpensive. Some features still rely on shared services, such as the discussions server for chat and the opml2json feed parser, so changes there can affect your copy.

At some point we will have a turnkey server setup so you can run the full suite of CList tools just by copying a prebuilt virtual server.

Read more: [Architecture](#)

Architecture

CList has no server of its own. Every feature that needs one uses a separate service.

- The application is static HTML, CSS, and JavaScript
- Services are federated: anyone can run one, and you choose which to use
- No required bottleneck: reading and publishing work without the optional services

With no server of its own, there is no central service to fail, change its terms, or be taken over. This is the core design choice of the project.

A priority of CList is to publish the architecture as a specification so other people can build compatible services.

Read more: [Architecture](#)

The services CList uses

Each service does one job. You can change any of them.

- kvstore: stores your encrypted accounts and your DID
- Annotations: stores your annotations
- Discussions: lists chat sessions so peers can find each other
- Collab: hosts shared documents
- opml2json: parses OPML feed files

Splitting the work into small services means each can be replaced, so you are not forced to trust one operator with everything.

Try it at clist.mooc.ca

Read more: [Architecture](#)