



Multiplayer Digital Garden Platforms and Tools

"Multiplayer" digital gardens are collaborative knowledge spaces (like wikis or networked note collections) that multiple users can contribute to, often with changes syncing in real-time. Below we explore examples – from commercial apps to open-source tools and prototypes – focusing on web-based platforms (many built with modern stacks like React/Next.js) that automatically update content for all collaborators. We note each platform's tech stack (if known), how it handles multi-user content sync (synchronous vs. asynchronous), cross-platform availability, and content-sharing capabilities.

Collaborative Digital Garden Platforms (SaaS Products)

These are hosted apps that provide team or community "digital garden" environments with seamless multi-user editing:

- **Notion** – A popular all-in-one workspace (web and apps) that can function as a collaborative digital garden. Notion's web app (built with Electron/JavaScript, likely React) supports real-time co-editing with multi-cursor presence similar to Google Docs. It offers flexible pages and databases that users can interlink, and any edits are instantly synced to others ¹. Notion is cross-platform (browser, Windows/Mac desktop, iOS/Android) and content can be **shared easily** – pages or entire workspaces can be shared via web links (public or with access control), and content can be exported (HTML, Markdown, PDF). It emphasizes **seamless collaboration** (multiple users can comment or edit concurrently) and bi-directional linking for building a network of notes ¹.
- **Microsoft Loop** – A newer collaboration app from Microsoft 365 (powered by the **Fluid Framework** CRDT technology). Loop provides a flexible canvas and **portable components** (like lists, tables, notes) that stay in sync across platforms ² ³. Multiple users can co-author Loop pages in real-time – changes are instantly visible to everyone ⁴. Uniquely, Loop components can be embedded in other services (e.g. a task list in a Teams chat or Outlook email) and any update propagates to all instances, enabling cross-app sharing. The tech stack is web-based (Fluid's reference implementation uses TypeScript/React). Loop is accessible via web and integrated into Microsoft Teams, Outlook, Word web, etc., making it **platform-agnostic** within the M365 ecosystem. It prioritizes **real-time sync** and easy sharing of content blocks across services ⁴ ².
- **Scrapbox (now Cosense)** – A web-based collaborative knowledge base (by Helpfeel Inc.) often described as a "networked notebook" for teams. It uses a **wiki-like** approach: every page is a freeform text note where links are created with simple syntax (e.g. [bracketed] words), and it shows backlinks and even two-hop link context. Scrapbox's interface is lightweight (tech stack is JavaScript; not explicitly stated but likely a modern SPA). Crucially, it supports **real-time co-editing**: multiple people can edit the same page simultaneously, with changes syncing instantly to everyone ⁵. This provides a very Google-Docs-like feel for collaborative note-taking. It's cross-platform via browser (with a mobile-friendly UI; no coding needed). For sharing, Scrapbox allows both **private and public projects** – teams can keep knowledge bases private or open them publicly for community collaboration ⁵. There's also Slack integration to broadcast edits and an API for

exporting data. In summary, Scrapbox enables an “always up-to-date” shared garden of interlinked notes with minimal structure, emphasizing **bi-directional links** and intuitive collaboration.

- **Coda** (honorable mention) – A web-based doc/spreadsheet hybrid that teams use for knowledge bases. Built on React/TypeScript, it supports real-time collaboration and cross-platform access (web, mobile). Coda pages can be interlinked and can contain databases, and it offers **pack integrations** to share data with other apps. Content updates in Coda are immediately reflected to other viewers (using an OT/CRDT sync under the hood). It’s not a classic “garden” with backlink graphs, but its flexibility and rich integrations make it useful for building living documents. Sharing is via links or exports (PDF, etc.), and it has an API for cross-service data flows.
- **Nuclino** (honorable mention) – A lightweight collaborative wiki for teams (marketed as a “collective brain”). It’s a React-based web app focused on speed and simplicity. Nuclino offers **real-time collaborative editing** (no save button; edits appear live to others) ⁶. Users create interlinked pages and can organize them in a graph, board, or hierarchical list view. It’s accessible on web and has desktop/mobile apps. For sharing, Nuclino lets you invite team members or share read-only links publicly. It’s akin to a modern Confluence or wiki with effortless multi-user sync and a clean UI.

Comparison of Key SaaS Platforms:

Platform	Tech Stack / Platform	Collaboration Model	Cross-Platform Access & Sharing
Notion	Web app (Electron, likely React); also native apps	Real-time multi-user editing (with live cursors) ¹ ; also async comments	Web, Windows, macOS, iOS, Android. Share pages via public link or invite; export to HTML/MD/PDF.
Microsoft Loop	Web (Fluid Framework – CRDT, built with TypeScript/React)	Real-time co-authoring of pages & modular components ⁴ ; syncs across M365 apps	Web and integrated in Teams, Outlook, etc. Loop components can be embedded in other apps, staying in sync ² .
Scrapbox (Cosense)	Web SPA (JavaScript; proprietary)	Real-time collaborative wiki-style editing (instant sync) ⁵	Web (desktop & mobile). Private or public projects; share via link. Slack integration and data export for sharing.
Coda	Web (React/TS) + cloud backend	Real-time editing of docs with embedded tables/apps	Web, iOS, Android. Share by link or invites; packs/API for integration with other services.
Nuclino	Web (React) + apps	Real-time wiki editing (instant save) ⁶	Web, Windows/macOS (apps), iOS/Android. Share read-only links or collaborate with team spaces.

Open-Source and Self-Hosted Collaborative Tools

For those who prefer open-source or custom setups, several projects enable multiplayer digital gardens with modern web technology:

- **Outline** – An open-source team knowledge base and wiki, built with Node.js and React (uses Prosemirror under the hood). Outline supports **real-time collaborative editing** of pages: multiple users (up to 100) can concurrently edit a document, with each character typed syncing live to others [7](#). It even shows live cursors with user names for presence [8](#). The stack includes a React frontend and a server that uses websockets/CRDT for sync. Cross-platform: Outline is accessed via web browser (responsive design for desktop/mobile), or it can be self-hosted on your own server. Sharing: it's primarily internal for teams (with user accounts and access control), but pages can be made publicly shareable. Data is stored in Markdown, so you can import/export easily. In short, Outline provides a **modern wiki** experience with Google-Docs-style live editing [7](#) and easy deployment for self-hosting.
- **HedgeDoc** (formerly CodiMD) – An open-source **collaborative Markdown editor** that can serve as a multiparty notes hub. It's a Node.js app with a React/Angular frontend (the exact UI framework has evolved) that lets users create and edit Markdown notes in real-time. **Every edit syncs live** to others viewing the same note (no manual refresh needed) [9](#) [10](#). This makes HedgeDoc useful as a collective digital garden for text-based content – you can interlink notes with Markdown links, and everyone sees updates immediately. It runs in the browser (no extra client software) and is self-hosted, so it's platform-agnostic (any device with a browser can use it). Collaboration can be synchronous (live editing) or asynchronous (users can also just open and edit notes at different times; version history is stored for conflict resolution). Sharing is straightforward: send someone the note's link and, with appropriate permissions, they can join and contribute or at least view. HedgeDoc notes can also be **published read-only** for public access. In summary, HedgeDoc provides Google Docs-like concurrent editing for Markdown notes [9](#) [10](#), making knowledge-sharing instant and easy on any platform.
- **AFFiNE** – A nascent open-source project (sometimes dubbed a “next-gen PKM [Personal Knowledge Management] OS”) that merges docs, whiteboards, and databases. AFFiNE is built with a **local-first** mindset – it uses **CRDTs for real-time sync** and offline editing. The tech stack is modern (written in TypeScript; uses React and **BlockSuite** for its editor component, and can run as a web app or local app via Tauri). Its **multiplayer content sync** is handled via CRDT-based algorithms rather than a central OT server [11](#) [12](#). This means multiple users can edit an AFFiNE document or canvas simultaneously, even if temporarily offline, and the data will merge without conflicts when reconnected. AFFiNE’s design is platform-agnostic: there’s a web app and self-hosted cloud option, plus the ability to run it as a desktop app that syncs. Cross-platform access is a goal (with end-to-end encryption and self-hosting for privacy). While still in development (as of 2025), AFFiNE aims to allow **synchronous co-editing** of rich content (text with formatting, diagrams, etc.) using CRDT to ensure eventual consistency [11](#) [12](#). It also supports **exporting** data (e.g. Markdown) so you can share or back up your knowledge base elsewhere. In essence, AFFiNE is an example of a modern React/TS web app embracing CRDTs to enable **multiplayer knowledge work** with offline-first capability.
- **Flowershow** – Not a real-time editor per se, but an open-source toolchain to publish a digital garden from Markdown. It's built on **React, Next.js, and Tailwind CSS** [13](#). Flowershow takes a folder or

repo of Markdown (designed to work seamlessly with Obsidian vaults) and produces a website. While editing is not done *within* Flowershow, it supports **asynchronous collaboration**: multiple people can contribute notes via Git (or any sync mechanism) and the site auto-builds to reflect new content ¹⁴. For example, a team could use Obsidian or VS Code to edit a shared vault; every git commit triggers an update so the published garden auto-updates when users add content. This is essentially a **platform-agnostic** approach (everyone works in their own editors/OS; the web build is platform-neutral). Sharing is inherent – the output is a website (which can be accessed on desktop or mobile browsers). Flowershow even offers a cloud service and an Obsidian plugin to simplify publishing. In summary, Flowershow leverages a modern web stack (React/Next) to enable **collaborative digital garden publishing**: it's not real-time sync like others, but it ensures an up-to-date shared garden through automation (great for version-controlled, async workflows) ¹⁴ ¹³.

- **Wiki.js / BookStack** (honorable mentions) – These are popular open-source wiki platforms (Wiki.js is Node/JavaScript; BookStack is PHP) that teams use for knowledge bases. They allow multiple users to edit content (with permissions and revision history), though **not in true real-time**. Typically one user edits a page at a time (others see the update after save). Wiki.js has a modern UI and supports Markdown editing; BookStack has WYSIWYG and Markdown options. Both are web-based and mobile-friendly. They emphasize **asynchronous collaboration**: e.g. users working on different pages or at different times, rather than concurrent editing in one page. Cross-platform: accessible via any browser; self-hosted. Sharing: pages can often be made public, and both have export or API options. While they don't have live cursors or instantaneous sync, they're robust solutions for a multi-user digital garden with a bit more structure (and can be extended – e.g. Wiki.js has plugins and an API for integrations).

Comparison of Key Open-Source Tools:

Tool	Tech Stack	Collaboration Sync	Cross-Platform & Sharing
Outline	Node.js backend, React frontend ⁷	Real-time collaborative editing (Prosemirror/CRDT-based); multi-cursor live updates ⁷	Web UI (self-host or cloud). Mobile-friendly. Share pages publicly or keep team-private; data in Markdown for export.
HedgeDoc	Node.js, web frontend (JS)	Real-time markdown editing; all participants see updates instantly ⁹ ¹⁰	Web (no client app needed). Self-hosted. Share note links for co-editing; can publish read-only or export notes.
AFFiNE	TypeScript (React) + CRDT (local-first) ¹¹	Real-time sync via CRDT (supports offline edits and merges) ¹² ; suitable for synchronous or async use	Web app or local app (Windows/Mac/Linux via Tauri). Self-host cloud or run offline. Plans for end-to-end encryption. Content export (MD, etc.) available for sharing outside platform.

Tool	Tech Stack	Collaboration Sync	Cross-Platform & Sharing
Flowershow	Next.js (React) static site generator ¹³	Asynchronous – content updated via git commits, auto-deployed to web (not live simultaneous editing) ¹⁴	Web output (responsive site). Any OS/editor for writing (markdown files). Easy public sharing (published site); integrates with Obsidian for one-click publish.
Wiki.js	Node.js, Vue/Angular frontend	Async editing (concurrent edits locked or merged via revisions)	Web UI (desktop/mobile). Self-host. Public or private wikis; supports content export and REST API.
BookStack	PHP (Laravel), MySQL	Async editing (one user/page, with edit locks)	Web UI (desktop/mobile). Self-host. Role permissions for sharing internally; public view possible per page. Export to PDF/HTML.

Experimental and Notable Research Projects

Beyond mainstream products, there are innovative prototypes and concepts exploring “multiplayer” knowledge spaces:

- **Federated Wiki** – An experimental wiki platform by Ward Cunningham (wiki’s inventor). Instead of real-time co-editing, it enables *forkable* collaboration. Each user runs their own wiki, but they can copy (“fork”) pages from others and edit in their space ¹⁵. The system then shows references and updates so knowledge can propagate across a network of wikis rather than residing in one central repository. Federated Wiki (built in CoffeeScript/Node) thus supports a **collaborative garden** where each participant maintains autonomy (their own site) while sharing content with the federation. Changes don’t sync instantly to others, but anyone can pull in updates or alternative versions – creating a tapestry of linked yet personal wiki pages. This approach contrasts with single-site wikis by embracing multiple viewpoints (each fork is a perspective) ¹⁶. Federated Wiki runs in the browser and is platform-neutral. It’s a compelling prototype of asynchronous, distributed collaboration, showing that “multiplayer” knowledge-building can happen via federation as well as live editing.
- **Local-First + CRDT-based Prototypes** – Recent research in *local-first software* explores multiplayer document editing without central servers. For example, the **PushPin** project by Ink & Switch is a peer-to-peer collaborative corkboard app using CRDTs (Automerge) under the hood ¹⁷ ¹⁸. PushPin lets multiple users add and arrange notes on a board, syncing directly over P2P connections. Even if users go offline, no data is lost – when they reconnect, the CRDT merges changes (ensuring eventual consistency). The tech is typically **React with CRDT libraries** and uses frameworks like Electron or web tech for the UI. While PushPin is a prototype, it demonstrates how a digital garden or bulletin board can be truly distributed: **each user’s device stores the data**, and updates propagate via networking, not a central server ¹⁷ ¹⁸. Similarly, frameworks like **Yjs** and **Automerge** have been used in demos of collaborative note editors (for instance, the *Peritext* CRDT for rich text by Ink & Switch). These research efforts inform real products (like AFFiNE and Loop mentioned above) by proving that **multiplayer sync can be fast, offline-friendly, and conflict-free**.

- **Spatial and Gaming-Inspired Spaces** – Some experimental “knowledge garden” platforms take inspiration from multiplayer games or spatial apps. For example, **Miro** and **Figma** (while design-focused) have effectively created multiplayer canvases – multiple people editing a visual space in real time – which can include text, diagrams, notes, etc. There are also niche tools like **Kinopio** (a whimsical spatial notetaking app with multiplayer mode) and **Gather.Town** or **Muse** (allowing groups to arrange ideas in a virtual space). These aren’t traditional text gardens, but they show that the concept of a collaborative knowledge space can be extended to visual and spatial formats with real-time presence. Many of these use React or WebGL for frontends and websockets/CRDT for sync. They run in browsers and often on mobile, and they allow content export (e.g. export a board as an image or PDF, etc.) or linking out to external resources. The focus is on synchronous collaboration – e.g. all participants see each other’s cursors or avatars moving – adding an “**ambient co-presence**” aspect to digital gardening that static wikis lack.
-

In summary, a variety of modern platforms enable “multiplayer” digital gardening. Mature SaaS apps like Notion, Loop, and Scrapbox provide polished experiences with instant cross-user updates and easy sharing across devices. Open-source tools like Outline, HedgeDoc, and AFFINE leverage React and real-time sync technologies (operational transforms or CRDTs) to let teams co-edit knowledge bases on their own servers. And experimental projects from federated wikis to CRDT-powered peer-to-peer editors are expanding what’s possible, ensuring that no matter the stack – be it Next.js on a server or local CRDTs in a browser – users can **contribute simultaneously** and see a collective knowledge space evolve in real time. The common thread is a **modern web foundation (often React/JS)** combined with clever sync engines that make collaboration feel seamless. Each new note or edit “auto-updates” for everyone, cultivating a living knowledge garden that anyone (with permission) can tend, from any platform, often with the ability to publish or port that content to other services as needed.

Sources:

- Notion collaboration and features [1](#)
 - Microsoft Loop real-time sync and cross-platform components [4](#) [2](#)
 - Scrapbox (Cosense) real-time co-editing and project sharing [5](#)
 - Outline (getoutline.com) real-time wiki editing [7](#)
 - HedgeDoc collaborative markdown editing [9](#) [10](#)
 - AFFINE tech (React/CRDT) and multi-user design [11](#) [12](#)
 - Flowershow (Next.js-based Obsidian publisher) usage [14](#) [13](#)
 - Federated Wiki concept (forking for collaboration) [15](#)
 - PushPin (Ink & Switch) CRDT P2P collaboration research [17](#) [18](#)
 - Scrapbox description (Globe article) [5](#) and Nuclino help docs [6](#) for additional context.
-

1 Best PKM App For Sharing A Digital Garden 2024 | AFFiNE

<https://affine.pro/blog/best-pkm-app-for-sharing-a-digital-garden>

2 3 4 Microsoft Loop

<https://www.infiflex.com/microsoft-loop>

5 Scrapbox: A Modern Home for Ideas and Collaboration | Globe

<https://globe.ie/2025/09/14/scrapbox/>

6 Best Online Collaboration Tools for Teams in 2026 - Nuclino

<https://www.nuclino.com/solutions/online-collaboration-tools>

7 8 Collaborative editing

<https://docs.getoutline.com/s/guide/doc/collaborative-editing-GjkoCop1B7>

9 10 HedgeDoc - Ideas grow better together

<https://hedgedoc.org/>

11 12 Why AFFiNE Chose CRDT over OT to Build a Collaborative Editor | AFFiNE

<https://affine.pro/blog/why-affine-chose-crdt-over-ot-to-build-a-collaborative-editor>

13 Flowershow

<https://datahub.io/olarubaj/my-digital-garden>

14 Collective Digital Garden for Life Itself

<https://notes.lifeitself.org/Collective+Digital+Garden>

15 16 Federated Wiki - Wikipedia

https://en.wikipedia.org/wiki/Federated_Wiki

17 18 PushPin: Towards Production-Quality Peer-to-Peer Collaboration

<https://www.inkandswitch.com/pushpin/>