ChatGPT

Open-Source Components

- **Visual/UI Libraries:** Standard open-source UI frameworks (Bootstrap, Material-UI, Tailwind, Vuetify, etc.) and dashboard templates (e.g. Creative Tim's Light Bootstrap Dashboard – free MIT-licensed templates built with React/Bootstrap [1]) can be used to create polished member and unit dashboards. For charts and graphs, libraries like Chart.js (MIT), D3.js (BSD), Vis.js/vis-network (MIT), or Cytoscape.js are available to render timelines, bar charts, network graphs, and other visualizations. These can be integrated into React/Vue components or server-rendered pages. For example, Nextcloud/Deck (an open-source Kanban app) provides a board-based task interface [2] that could be embedded or themed as part of the UI.

- **Task/Project Management:** Open-source Kanban and project-management tools can serve as models or backends. Popular options include **Taiga** (Python/Django + AngularJS, AGPL-3.0) [3], **Kanboard** (PHP, MIT) [4], **Wekan** (Node/Meteor, MIT), and **Taskcafe** (Go, MIT). These provide drag-and-drop task boards with features like lists, labels, due dates, checklists, and comments. They can be self-hosted (e.g. via Docker) and extendable; for example, Taiga can import Trello/Jira boards, and Wekan/Docker images support easy setup [3] [4]. Another approach is a *workspace app* like Nextcloud with the Deck plugin, which provides an integrated task board and collaboration suite [2]. These tools often include basic project tracking, file attachments, and comment threads tied to cards.

- **Voting/Decision & Commenting:** For democratic decision-making and comments, **Loomio** stands out. Loomio is open-source and designed for group decision support: it offers threaded discussions and poll/proposal features to facilitate inclusive votes [5]. It is widely used by cooperatives [6] and preserves meeting records. Other options include **Discourse** (open-source forum software, GPLv2) or **Flarum** (PHP forum) with polling extensions for threaded discussion and votes. For simple polls, lightweight apps (e.g. DemocracyOS or custom Django/Flask modules) could also be integrated. In all cases, comments can be stored in the app's database (as in Taskcafe or Kanboard) or in a dedicated forum module.

- **Authentication & Access Control:** Standard open solutions can handle logins and roles. In a Python stack, Django's built-in Auth provides user accounts, groups, and permissions out-of-the-box [7]; Django REST Framework can add JWT or token auth for API access. FastAPI can use libraries like `fastapi-users` or OAuth2 flows. For stricter RBAC/ABAC, systems like **Casbin** (multi-language access control library, Apache 2.0) or **django-guardian** (object-level permissions) are available. Alternatively, an external identity provider (Keycloak – open-source OIDC, or Ory Kratos/Keto) can manage SSO and fine-grained permissions. All these options are open-source/MIT/GPL-compatible.

- **Data Visualization/Graphs:** To display membership and unit relationships (e.g. a network graph of members, or floorplan layouts), JavaScript libraries like **Sigma.js**, **Cytoscape.js**, or **vis-network** (MIT) allow interactive network/graph rendering. For charts and dashboards, **Chart.js** (MIT), **Plotly/Dash** (MIT), and **Apache ECharts** (Apache 2.0) can produce bar charts, pie charts, timelines, etc. Python-based tools (Plotly, Bokeh, Altair) can generate charts server-side or via APIs. For example, Plotly

Dash (Python) can be used to build a standalone dashboard app with rich graphs. These libraries can be embedded in a frontend or served via API endpoints.

Full-Stack Architecture Options

We compare several open-source stack patterns, focusing on Python backends, Docker Compose deployment, and n8n integration:

Stack	Backend (Python)	Frontend	Details
Django Monolith	Django (with DRF)	Django templates + JS	Full-stack Django has built-in admin, ORM, auth [7]. Good for rapid development with batteries included (admin UI, forms, auth, migrations) [7]. Can serve HTML/JS directly or expose a REST API. Real-time via Django Channels (ASGI+Redis). Deploy via Docker Compose (Django + Postgres + Redis + Celery). Easy to integrate plugins for tasks/votes (e.g. django-vote, django-comments).
Django + SPA	Django REST Framework	React/Vue SPA	Django (or FastAPI) serves JSON API, frontend as single-page app (e.g. React + Material-UI/Vuetify). Decouples UI and backend. Django provides models, API, auth; frontend handles interactive UI. Real-time via WebSockets (Channels with token auth) or polling. Docker Compose with separate service for `node` (Dev server) or prebuilt assets, plus backend and DB.
FastAPI + SPA	FastAPI (async)	React/Vue SPA	FastAPI (Python) for API – very fast and supports ASGI (built-in WebSockets, async tasks) [7]. Use Pydantic models for validation and auto-generated OpenAPI docs. Frontend as in SPA case. Requires choosing auth solution (e.g. OAuth2/JWT with PyPI packages). Docker Compose: FastAPI app (Uvicorn), DB, (optional Redis/Celery). Good for high-performance API and realtime endpoints.
Headless CMS + Apps	Nextcloud (PHP, AGPL)	Nextcloud UI/Deck	Leverage existing co-op tools: Nextcloud provides user management, file storage, and Deck (Kanban) for tasks [2]. Write a Nextcloud app/plugin if needed (in PHP) for custom views (unit layout, voting). Real-time features limited, but Nextcloud Talk can add chat. Easy Docker Compose with Nextcloud+DB+Redis, minimal coding if Deck meets needs.

Stack	Backend (Python)	Frontend	Details
Low-Code Platform	Budibase (Node/ Svelte)	Budibase GUI	Budibase is an open-source low-code app builder (AGPL) [8]. It can connect to a database (e.g. Postgres) and build UIs via drag-drop. Users can design dashboards, forms, and listings for members/tasks with minimal coding. Runs in Docker. Less conventional for developers, but friendly for admins.

For each option, a Docker Compose stack is recommended: e.g. `backend`, `db (PostgreSQL)`, `redis` (for caching/Channels), `celery worker` (for background jobs/notifications), and optionally an `n8n` container for automation [9]. n8n can connect to the app's API via webhooks or REST to automate emails/ notifications or integrate with external services. For example, an environment variable `WEBHOOK_URL` is configured in n8n's Docker Compose for incoming hooks [9], illustrating how the app can expose webhook endpoints.

Proposed Architecture

Combining simplicity and extensibility, a recommended stack might be:

- **Backend Framework: Django** with Django REST Framework (or **FastAPI**). Django offers an admin interface, ORM, and auth out-of-the-box [7], which speeds development. FastAPI could be chosen if asynchronous features or performance are critical, as it excels at real-time (WebSockets) and generates API docs automatically [7].

- **Frontend:** A modern JS framework (e.g. **React** or **Vue**) using open-source UI kits (Material-UI, Bootstrap, Vuetify) for ease of use. This allows building interactive dashboards, charts, and forms. Alternatively, the built-in Django templating with a little JavaScript (using chart libraries) can suffice for a smaller team.

- **Database & ORM: PostgreSQL** (open-source) is recommended for reliability and complex queries. Use Django's ORM (or SQLAlchemy/Tortoise in FastAPI) to manage models (Members, Units, Tasks, Comments, Votes). For small deployments, SQLite could work but offers less scaling and multi-user robustness.

- **Authentication & Permissions:** Use the framework's auth system (Django Auth or a FastAPI auth library) for user accounts and role assignment (e.g. "member" vs "admin"). Django Groups/ Permissions can enforce who can edit tasks or view certain pages. For finer control, packages like django-guardian or Casbin can add object-level rules. All solutions are open-source (Django's auth is BSD/MIT, Casbin is Apache2).

- **Real-Time & Notifications:** Implement **WebSockets** for live updates (e.g. new tasks, comments, votes). Django Channels or FastAPI's WebSocket routes can push updates to clients. For push-style alerts or email reminders, a background worker like **Celery** (Redis broker, MIT) can send emails or notifications (e.g. send email when a vote passes or a task is assigned). A notification system (e.g.

django-notifications) can record events in-app. n8n can also monitor webhooks or poll the API to trigger SMS/Telegram/Matrix alerts, leveraging its many connectors [9].

- **APIs & Webhooks:** Expose a REST (or GraphQL) API for all data (members, tasks, votes, comments). This allows frontend and external tools to integrate. Provide webhook endpoints for key events (new task, vote cast, comment added) so n8n or other services can subscribe. As shown in the n8n Docker example, a `WEBHOOK_URL` can be configured for such hooks [9]. Comprehensive OpenAPI docs (FastAPI) or DRF docs facilitate automation.

- **Deployment:** Use **Docker Compose** to containerize each component: app server, database, cache, broker, and optionally front-end build. This ensures easy setup and portability. For example, n8n's recommended Compose file runs it alongside Traefik for networking [9]. A similar multi-container setup can run the Django+Postgres+Redis stack with single command.

Existing Open-Source Projects

Some ready-made projects cover much of this functionality and could be forked or extended:

- **Loomio** – Collaborative decision-making (Ruby on Rails, open-source) with discussions and polling [5] [6]. It could be used "as is" for votes or integrated via API, though its UI is separate.
- **Nextcloud + Deck** – Self-hosted collaboration suite with file sharing and Kanban boards [2]. It handles members and tasks; a custom app or theming could add co-op-specific views (unit plans, meeting notes).
- **Taiga** – Python/Django project management (AGPL) [3]. Contains Kanban, Scrum boards, issue tracking. Can be forked to add cooperative-specific models.
- **Kanboard/Wekan/Taskcafe** – Lightweight Kanban tools (PHP or Node) for tasks, with commenting and basic access controls [4].
- **CiviCRM** – Nonprofit CRM (Drupal/WordPress integration, GPL) with membership tracking, contributions, events [10]. Not tailored to housing co-ops, but useful for membership data and mailing lists.
- **Discourse** – Forum platform (Ruby) with trust levels and polls. Could serve as a community forum/comment system.
- **BookStack or Wiki.js** – Wiki/documentation platforms (open-source) for institutional memory of bylaws, meeting minutes, etc.

While none of these is a perfect match, each covers parts of the spec and could be integrated. For example, one could use Loomio for voting, Nextcloud for file storage, and a Kanban tool for tasks, linking them through SSO and APIs.

Recommendations & Enhancements

- **Prefer Simplicity:** A single Python-based stack (Django or FastAPI) keeps development and ops simpler. Django's "batteries included" approach helps small teams avoid boilerplate [7].
- **Modular Design:** Build with a modular API so features (tasks, votes, comments) can be added or disabled. E.g., separate Django apps or FastAPI routers for each domain.

- **Low/No-Code Integration:** Consider using tools like **n8n** (open-source workflow automation) to glue services together (e.g. auto-email on new tasks) [9]. Alternatively, **Budibase** (open-source low-code platform) could be used to quickly build internal UIs without manual coding [11], though its AGPL license should be noted.
- **Realtime Updates:** Use WebSockets for immediate feedback (task moved, vote result) since state-of-art async frameworks (FastAPI) and Django Channels make this straightforward [7].
- **Responsive, Mobile-Friendly UI:** Choose a responsive framework (Bootstrap/Material) so non-technical users can access on phones/tablets. Provide intuitive dashboards (e.g., a calendar view of tasks or an interactive floorplan of units).
- **Scalable Notification Service:** For notifications beyond email (push, SMS, chat), leverage n8n's 100+ integrations. For example, n8n can listen to a "new announcement" webhook from the app and post it to Matrix/Slack/Telegram, using its ecosystem.
- **APIs for Extensibility:** Build REST/GraphQL APIs with thorough documentation so future tools (mobile apps, analytics) can hook in. FastAPI's automatic Swagger docs or DRF's browsable API are beneficial.
- **Continuous Updates & Community:** Use GitHub (MIT/GPL licensing) for the codebase to encourage forks and contributions from the co-op community. Keep dependencies (Docker images) up-to-date for security and stability.

By leveraging these open-source components and following a containerized, Python-centric architecture, the co-op tool can remain **maintainable** and **extensible**. Non-technical users will benefit from intuitive web interfaces (Kanban boards, polls, charts) while the developer team can adjust or enhance features without being locked into proprietary systems. All recommended software is open-source (MIT/AGPL/GPL licensed) and can be self-hosted via Docker Compose, fitting the co-op's needs for simplicity and transparency [3] [2].

Sources: Open-source project docs and comparisons (e.g. Loomio [5] [6], Django vs FastAPI [7], Kanban tool reviews [3] [4], cooperative software listings [10], and n8n documentation [9]).

---

1 20+ Stunning Free React Dashboard Templates 2025 - MUI
https://mui.com/store/collections/free-react-dashboard/

2 6 Top seven co-operative tech alternatives for working from home during Covid-19 - Co-operative News
https://www.thenews.coop/top-seven-co-operative-tech-alternatives-for-working-from-home-during-covid-19/

3 4 5 open source alternatives to Trello | Opensource.com
https://opensource.com/alternatives/trello

5 About us - Loomio
https://www.loomio.com/about/

7 Django vs FastAPI: Which is the Best Python Web Framework? | The PyCharm Blog
https://blog.jetbrains.com/pycharm/2023/12/django-vs-fastapi-which-is-the-best-python-web-framework/

8 11 Open Source Licenses
https://budibase.com/licenses/

9 Docker Compose | n8n Docs
https://docs.n8n.io/hosting/installation/server-setups/docker-compose/

10 Tools and Apps for the Cooperative Movement | ICA
https://ica.coop/en/tools-and-apps