

# BOXY

Modern File Sharing & Task Management



Rust + Actix | WebSocket | Vanilla JS

# WHAT IS BOXY?



## File Management

- Upload
- Rename
- Move
- Delete
- Search
- Sort



## Task Boards

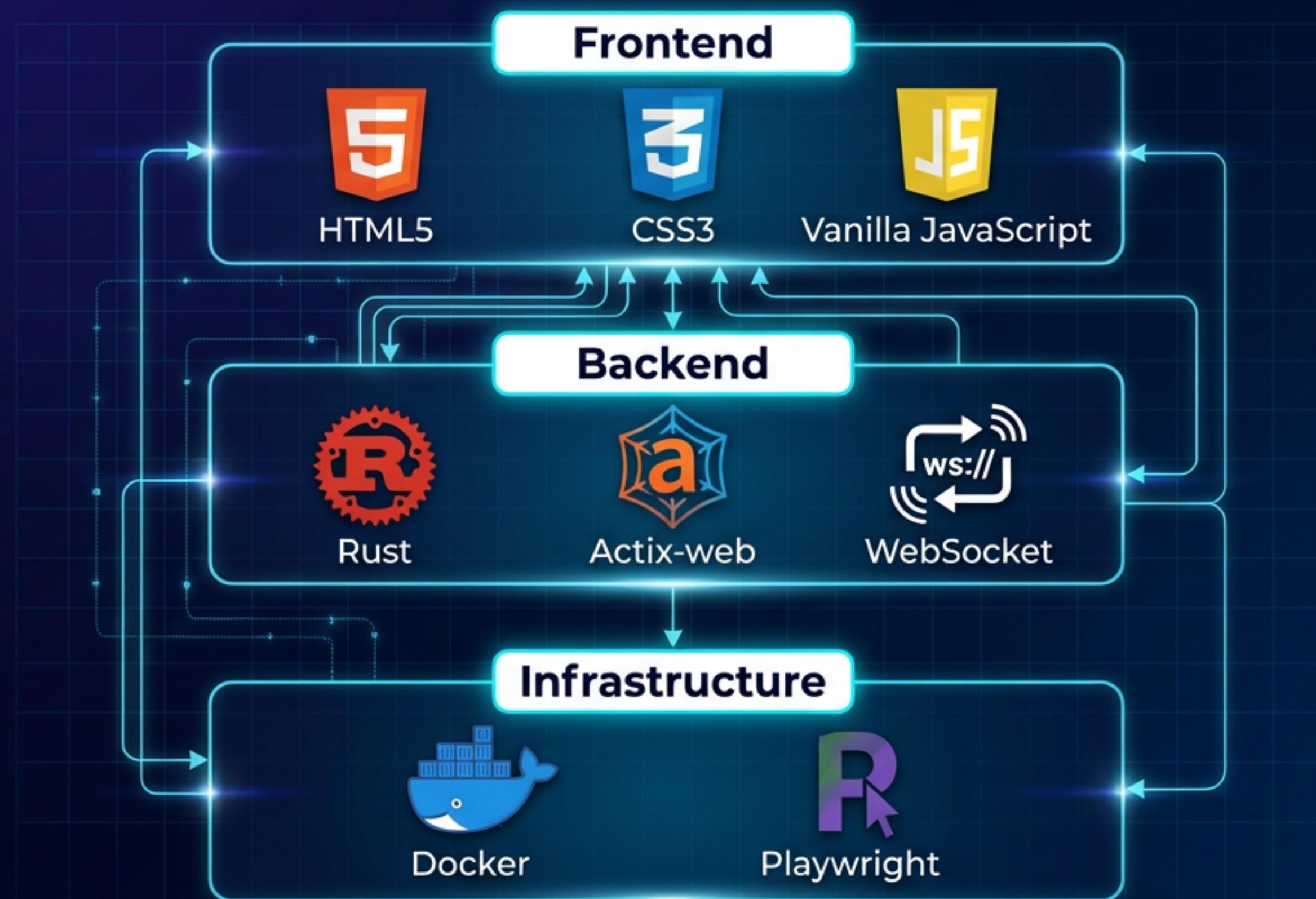
- Drag & drop
- Multiple boards
- `localStorage` persistence



## Real-time Sync

- **WebSocket** broadcast
- Multi-client
- Auto-refresh

# Technology Stack



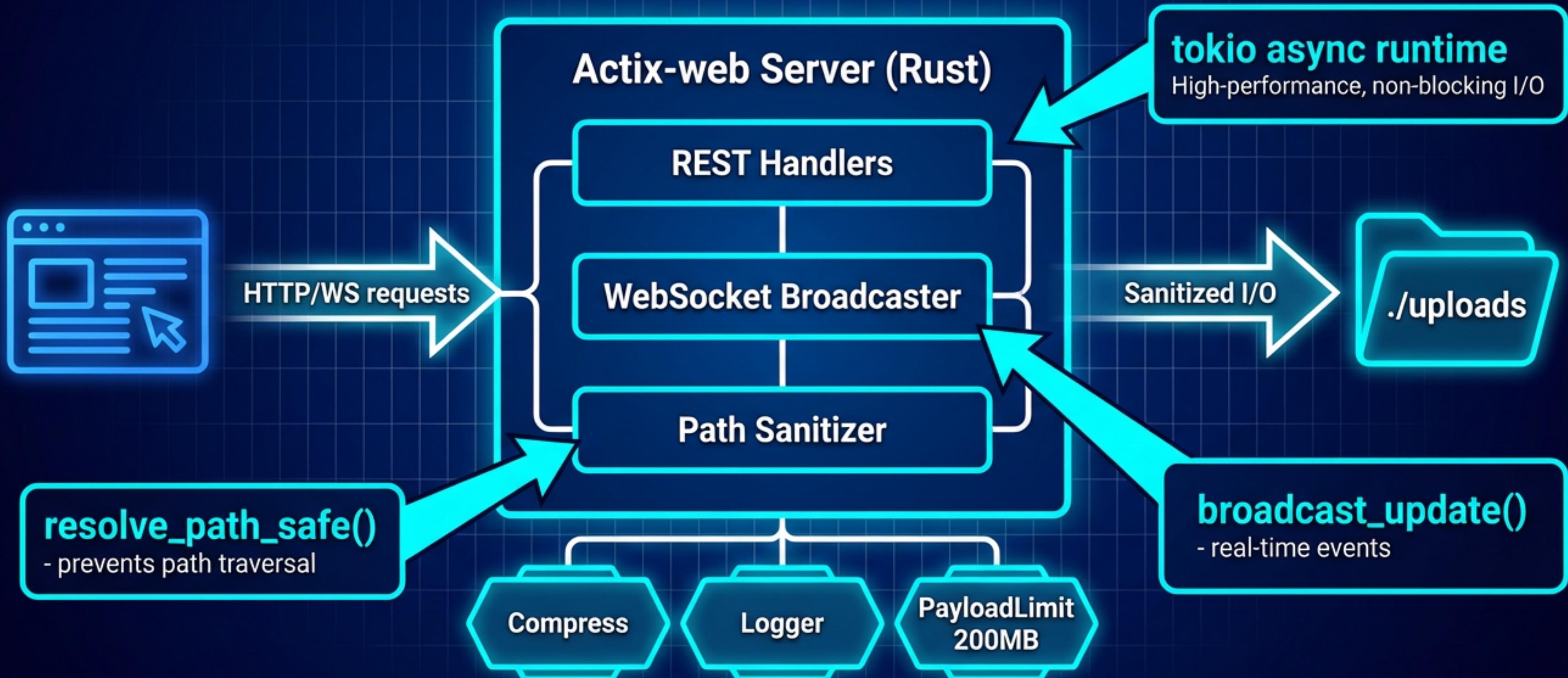
## Tech Details

- Rust
- **Actix-web** 4
- tokio async
- **WebSocket** /ws
- HTML/CSS/JS
- Docker
- **Playwright**

# Frontend Architecture



# Backend Architecture



# REST + WebSocket API



## REST Endpoints

Method	Endpoint
GET	/api/files
GET	/api/search
POST	/api/upload
POST	/api/folder
POST	/api/rename
POST	/api/move
POST	/api/delete
GET	/api/download
GET/POST	/api/content
POST	/api/newfile
GET	/api/health



## WebSocket /ws

### Real-time Broadcast Events

- upload
- rename
- move
- delete
- edit

Bidirectional  
Communication

State  
Synchronization

# Security & Hardening

## Input Validation

- `resolve_path_safe()`
- blocks ..../ traversal
- prevents symlink escapes

## Resource Limits

- `MAX_SEARCH_RESULTS=100`
- 200MB upload limit

## Output Encoding

- `escapeHtml()`
- `escapeAttr()`
- backtick/dollar escaping



# FILE MANAGEMENT FEATURES

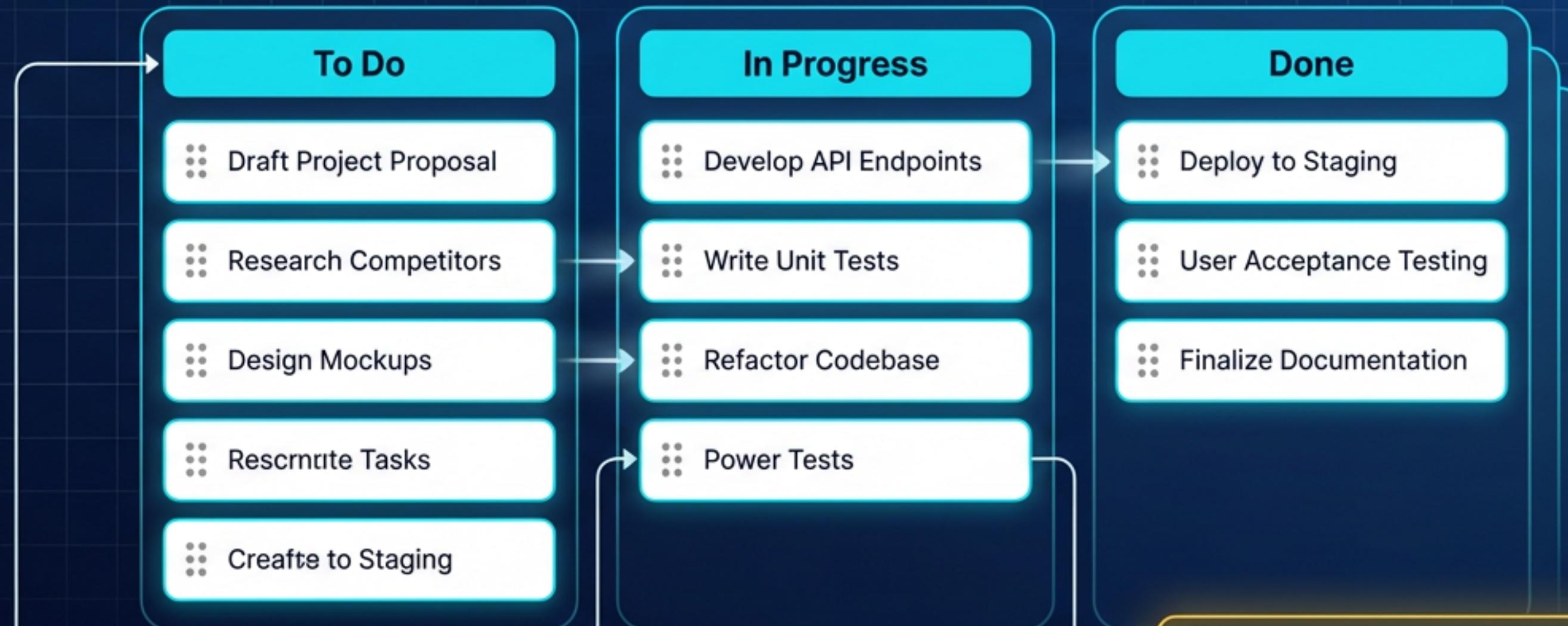
The diagram illustrates various file management features in a user interface. It includes a grid/list view toggle, breadcrumb navigation, a search bar, a sort dropdown, a drag-drop upload zone, thumbnail previews, and file actions on hover.

- Grid/List view toggle
- Breadcrumb navigation
- Search bar (recursive, max 100)
- Sort dropdown (name/size/type/date)
- Drag-drop upload zone
- Thumbnail previews
- File actions on hover: Download, Rename, Move, Delete, Edit

## ADDITIONAL FEATURES

- ▶ Folder upload support
- ▶ Paste from clipboard
- ▶ New file creation
- ▶ Edit text content

# Tasks / Kanban Board



Create/Edit/Delete Boards

Drag Tasks Between Columns

Multiple Boards Support

**Important Notice:**  
localStorage ONLY - No server persistence. Clearing browser data resets all tasks.

# Testing & QA

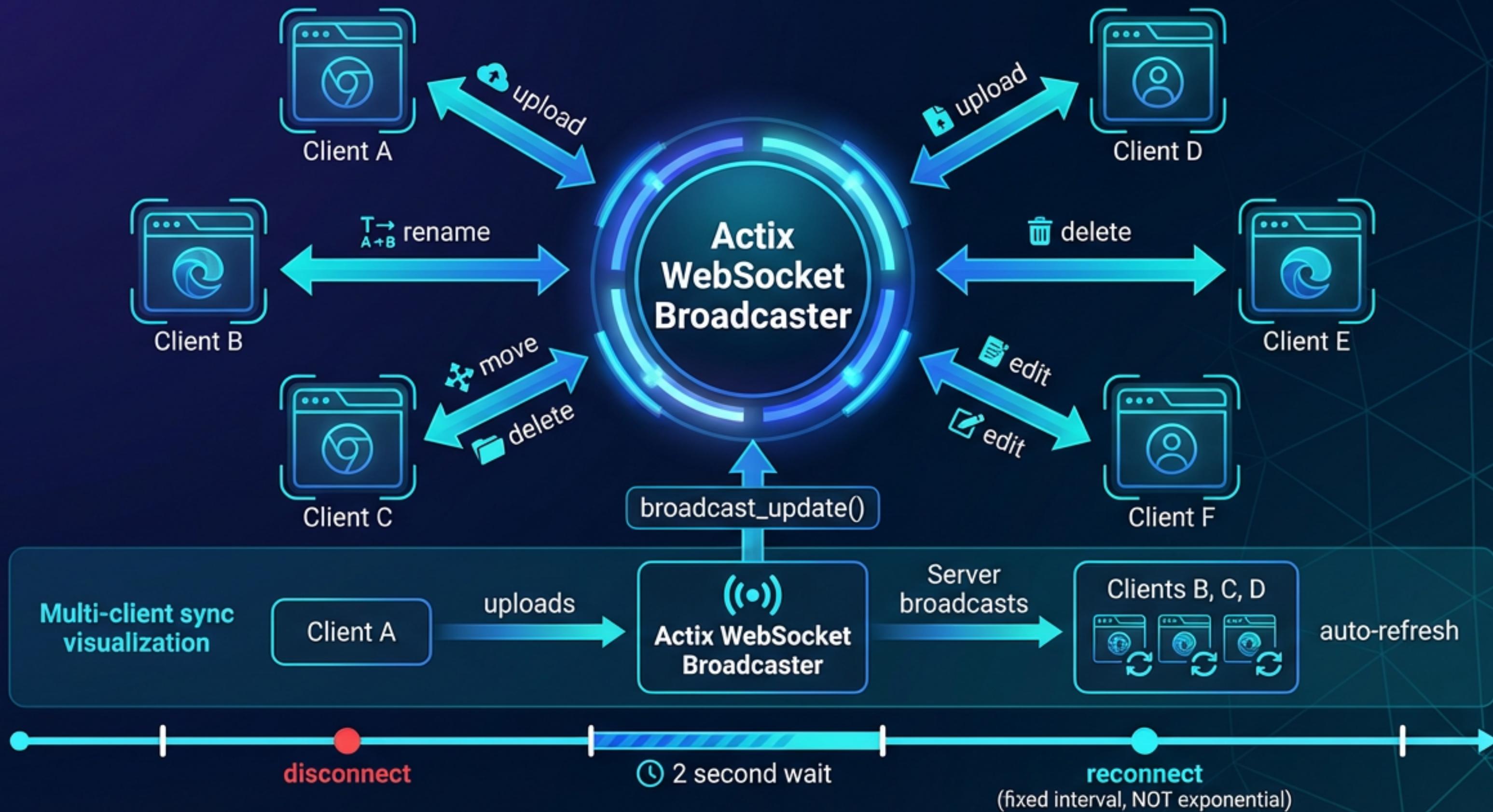


## Key Features:

- Headless Chromium
- Automated UI walkthrough capture
- 17 screens x 2 themes
- Reproducible via [docs/capture-ui-screenshots.mjs](#)

```
npx playwright install && npm run test:e2e
```

# Real-time WebSocket Model



# Deployment



```
docker build -t boxy .
```

```
docker run -p 8086:8086 -v uploads:/app/uploads boxy
```

```
docker compose up --build
```

## Environment Variables

```
BOX_PORT = '8086'
```

```
BOX_UPLOAD_DIR = '/app/uploads'
```

```
BOX_MAX_UPLOAD_BYTES = '10485760'
```

## Health Check Endpoint

```
GET /api/health
```

