@ecobridge.xyz/eco_os

the os to connect your devices to ecobridge.xyz through a standalone PC/VM in your network. Comes with TouchScreen support, to run your scan station or home dashboard.

readme.md for @ecobridge.xyz/eco_os

□□ EcoOS

> **A purpose-built, minimal Linux distribution for kiosk and digital signage deployments.**

EcoOS is a streamlined operating system that boots directly into a full-screen Chromium browser, managed by a powerful daemon with a built-in web UI. Perfect for digital signage, interactive kiosks, info displays, and any scenario where you need a locked-down, browser-based interface.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

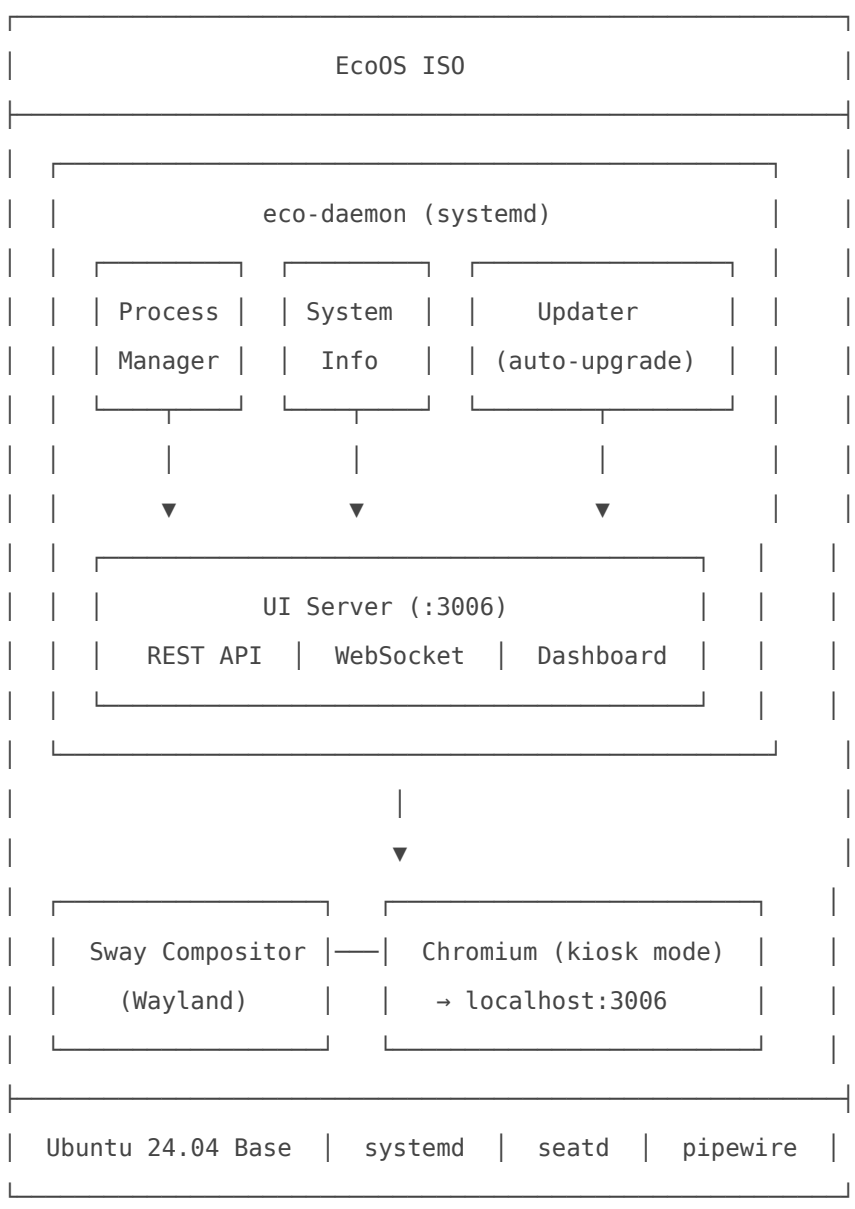
□ Features

- □□ **Zero-Config Boot** — Boots straight into a Wayland-based kiosk browser
- □□ **Sway Compositor** — Modern, tiling Wayland compositor with automatic fallback modes
- □□ **Chromium Kiosk** — Full-screen browser in locked-down kiosk mode
- □□ **Management UI** — Real-time system monitoring and control via web interface on port 3006
- □□ **Auto-Updates** — Daemon self-updates with smart stability checking
- □□ **System Monitoring** — CPU, memory, disk, network, GPU, and audio device stats

- □□ **Live Logs** — System journal and daemon logs accessible from the UI
- □□ **Hardware Support** — Input devices, speakers, microphones detection and display
- ⚡ **Rapid Recovery** — Auto-restart of crashed services within seconds

---

□□ Architecture



□□ Development

Prerequisites

- **Docker** (for ISO building)
- **pnpm** (package manager)
- **Deno** v2.x (for daemon development)
- **QEMU** (for testing)

Project Structure

```
eco_os/
├── ecoos_daemon/          # Daemon source (Deno/TypeScript)
│   ├── mod.ts             # Entry point
│   └── ts/
│       ├── daemon/        # Core daemon logic
│       │   ├── index.ts       # EcoDaemon class
│       │   ├── process-manager.ts  # Sway/Chromium management
│       │   ├── system-info.ts # Hardware detection
│       │   └── updater.ts     # Auto-update system
│       ├── ui/            # Web UI server
│       └── utils/         # Utilities
├── isobuild/              # ISO build configuration
│   ├── Dockerfile         # Build container
│   ├── config/            # live-build config
│   └── scripts/           # Build scripts
├── isotest/               # QEMU test scripts
└── .nogit/                # Generated artifacts (not in git)
    ├── iso/               # Built ISO
    ├── vm/                # QEMU files
    └── screenshots/       # VM screenshots
```

Commands

```
# Build the full ISO (auto-rebuilds daemon first)
pnpm run build

# Test ISO in QEMU virtual machine
pnpm run test
```

```
# Take screenshot of running VM
pnpm run test:screenshot

# Stop the QEMU VM
pnpm run test:stop

# Clean all build artifacts
pnpm run clean

# Daemon development (watch mode)
pnpm run daemon:dev

# Bundle daemon to standalone binary
pnpm run daemon:bundle

# Type-check daemon code
pnpm run daemon:typecheck
```

□□ Management UI

The daemon exposes a management interface at `http://localhost:3006` (or the device's IP on port 3006).

Dashboard Features

Panel	Description
Services	Status of Sway compositor and Chromium browser
CPU	Model, core count, real-time usage
Memory	Used/total with visual progress bar
Network	Interface names and IP addresses
Disks	Mount points, usage, and capacity
System	Hostname, uptime, GPU info
Controls	Restart browser, reboot system buttons
Updates	Version info, available updates, upgrade controls

Panel	Description
Input Devices	Keyboards, mice, touchscreens
Audio	Detected speakers and microphones
Logs	Daemon logs and system journal viewer

API Endpoints

Endpoint	Method	Description
`/api/status`	GET	Full system status
`/api/logs`	GET	Daemon logs
`/api/reboot`	POST	Reboot the system
`/api/restart-chromium`	POST	Restart the kiosk browser
`/api/updates`	GET	Update information
`/api/updates/check`	POST	Check for new updates
`/api/upgrade`	POST	Upgrade to specific version
`/ws`	WebSocket	Real-time status updates

□□ Update System

EcoOS features a smart auto-update mechanism:

1. **Hourly Checks** — Daemon polls for new releases every hour
2. **Stability Period** — New releases wait 24 hours before auto-upgrade (prevents deploying unstable releases)
3. **Seamless Upgrade** — Downloads new daemon binary, replaces, and restarts service
4. **Manual Override** — Force immediate upgrade via UI or API
5. **Version Tracking** — UI auto-reloads when daemon version changes

---

□□ Testing

QEMU Virtual Machine

```
# Start VM (creates disk, boots ISO)
pnpm run test

# Take screenshots to monitor progress
pnpm run test:screenshot

# Screenshot loop (every 5 seconds)
pnpm run test:screenshot:loop

# Stop VM
pnpm run test:stop

# Clean and restart fresh
pnpm run test:clean && pnpm run test
```

Serial Console

For debugging without graphics:

```
socat - UNIX-CONNECT:.nogit/vm/serial.sock
# Login: ecouser / ecouser
# Root: sudo -i
```

□□ Release Assets

Each release includes:

File	Description
`ecoos-vX.X.X.iso`	Full bootable ISO image (~2GB)
`eco-daemon-vX.X.X`	Standalone daemon binary for in-place upgrades
`SHA256SUMS.txt`	Checksums for verification

□□ Boot Menu Options

1. **Install EcoOS** *(default, auto-selects in 10s)* — Full installation to disk
2. **EcoOS Live** — Try without installing (runs from RAM)
3. **EcoOS Live (Safe Mode)** — Minimal boot for troubleshooting

---

⚙ Technical Details

Daemon

- **Runtime**: Deno (compiled to standalone binary)
- **Process Management**: Spawns and monitors Sway + Chromium
- **Backend Fallback**: Tries DRM first, falls back to headless/pixman
- **Auto-Recovery**: Restarts crashed services within 5 seconds
- **Logging**: Integrates with systemd journal

Kiosk Browser

- **Browser**: Chromium (official snapshots, not snap)
- **Flags**: `--ozone-platform=wayland --kiosk --no-first-run --disable-infobars`
- **Default URL**: `http://localhost:3006` (management UI)

System Stack

- **Base**: Ubuntu 24.04 LTS
- **Init**: systemd
- **Display**: Sway (Wayland compositor)
- **Seat Manager**: seatd
- **Audio**: PipeWire

License and Legal Information

use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

changelog.md for @ecobridge.xyz/eco_os

2026-01-10 - 0.5.0 - feat(ui,isotest)

Group disabled displays into a collapsible section and refactor display item rendering; start a background screenshot loop during isotest and improve test-run cleanup

- Refactored display rendering: introduced renderDisplayItem() and simplified updateDisplaysUI() to separate enabled/disabled displays
- Disabled displays are collapsed under a

```
summary showing count ("Disabled Displays (N)")
```

- Added a background screenshot loop in isotest/run-test.sh that runs screenshot.sh every 5 seconds and records SCREENSHOT_LOOP_PID
- Improved cleanup in isotest/run-test.sh to kill SCREENSHOT_LOOP_PID and ENABLE_PID if they are running

2026-01-10 - 0.4.15 - fix(isotest)

Improve robustness of SPICE display enabler: add logging, wait-for-port and URI parsing, retries and reconnection logic, stabilization delay before configuring, and verification/retry of monitor configuration

- Add immediate-flush logging helper for clearer background output
- Wait for SPICE TCP port (wait_for_port) and parse spice:// URIs before connecting
- Add stabilization delay before sending monitor config and track retry counts
- Add verify_and_retry to confirm configuration or retry up to configurable attempts
- Detect agent disconnects (VM reboots) and keep running to reconfigure on reconnect; add reconnect and periodic health checks

2026-01-09 - 0.4.1 - fix(release-upload)

clear progress timer on upload completion/error and add periodic upload progress reporting

- Clear the progress interval on response end and on stream/error to avoid leaking timers.
- Track bytesWritten (header + stream chunks + footer) to compute accurate progress percentages.
- Log upload progress (percent and MB) every 10 seconds for visibility.
- Handle stream errors by clearing the progress timer and rejecting with the error.

2026-01-09 - 0.4.0 - feat(displays)

add display detection and management (sway) with daemon APIs and UI controls

- Introduce DisplayInfo type in system-info.ts
- Add ProcessManager methods: getDisplays, setDisplayEnabled, setKioskDisplay (invoke swaymsg via runuser)
- Add daemon methods to expose getDisplays, setDisplayEnabled and setKioskDisplay with runtime/Wayland context and status checks
- Add UI server endpoints: GET /api/displays and POST /api/displays/{name}/(enable|disable|primary) and frontend UI to list and control displays (polling + buttons)
- Bump VERSION and package.json to 0.3.9

2026-01-09 - 0.3.8 - fix(ci(release-workflow))

use npx tsx to run release-upload.ts in the Gitea release workflow instead of installing tsx globally

- Removed 'pnpm install -g tsx' to avoid global installs in CI
- Replaced direct 'tsx' invocation with 'npx tsx' to run .gitea/release-upload.ts
- Reduces CI image footprint and avoids unnecessary global package installation

2026-01-09 - 0.3.7 - fix(daemon)

Point updater at the correct repository API (code.foss.global ecobridge.xyz/eco_os) and bump project/daemon versions to 0.3.6

- Updated repo API URL in ecoos_daemon/ts/daemon/updater.ts from 'https://code.foss.global/api/v1/repos/ecobridge/eco-os/releases' to 'https://code.foss.global/api/v1/repos/ecobridge.xyz/eco_os/releases'
- Bumped daemon version in ecoos_daemon/ts/version.ts from 0.3.4 to 0.3.6
- Bumped package version in package.json from 0.3.5 to 0.3.6
- Included rebuilt daemon binary at isobuild/config/includes.chroot/opt/eco/bin/eco-daemon (bundle updated)

2026-01-09 - 0.3.5 - fix(ci)

add Gitea release asset uploader and switch release workflow to use it; bump package and daemon versions to 0.3.4

- Add .gitea/release-upload.ts: streams assets to Gitea to avoid curl's 2GB multipart limit
- Update CI workflow (.gitea/workflows/release.yml) to run the TypeScript uploader via tsx
- Bump package.json and ecoos_daemon/ts/version.ts to 0.3.4
- Update bundled eco-daemon binary in isobuild/config/includes.chroot/opt/eco/bin/

2026-01-09 - 0.3.2 - fix(release)

bump package and daemon to v0.3.1, add project README, and fix Gitea release upload flag

- package.json version updated from 0.3.0 to 0.3.1
- ecoos_daemon/ts/version.ts updated to export VERSION = "0.3.1"
- Added comprehensive readme.md documenting the project, development and release workflow
- Fix .gitea/workflows/release.yml: use curl -T for uploading release assets instead of --data-binary
- Updated bundled eco-daemon binary in isobuild/config/includes.chroot/opt/eco/bin/ (new build artifact)

2026-01-09 - 0.3.0 - feat(daemon)

add automatic update mechanism (Updater), switch to system journal logs, and expose update controls in the UI

- Introduce Updater class: fetches releases from Gitea, computes auto-upgrade eligibility, downloads daemon binary, replaces binary and restarts service.
- Integrate updater into EcoDaemon: new methods getUpdateInfo, checkForUpdates, upgradeToVersion; run initial update check on startup and periodic auto-upgrade checks (hourly).
- Replace serial console reader with a journalctl-based system journal reader; rename serialLogs → systemLogs and update related logic and limits.
- UI/server: add API endpoints /api/updates, /api/updates/check and /api/upgrade; add an Updates panel to show current version, available releases, auto-upgrade status, and client-side actions to check and trigger upgrades; poll update info periodically.
- Version bump to 0.2.2 (package.json and ecoos_daemon/ts/version.ts).
- Build/workflow changes: release workflow now runs build step (Build ISO) and package.json build script adjusted for CI and updated Docker build/run handling.

2026-01-09 - 0.2.1 - fix(ci)

use GitHub Actions workspace for docker volume and add listing of build output directory for debugging

- Replace $(pwd) with ${{ github.workspace }} in docker run volume mount to work correctly in GitHub Actions runner
- Add ls -la .nogit/iso/ to list generated artifacts and aid debugging of release workflow

2026-01-09 - 0.2.0 - feat(daemon)

add serial console reader and UI tab for serial logs; add version propagation and CI/release workflows

- Start a background serial reader that reads /dev/ttyS0, retains up to 1000 lines and exposes serial logs via the daemon API
- Add a Serial Console tab in the management UI to view serial logs and a tab switcher; UI will auto-reload when daemon version changes
- Expose VERSION from ecoos_daemon and include it in status responses
- Bump package version to 0.1.3 and update daemon version constant
- Add .gitea workflows for CI (typecheck + bundle) and Release (build daemon, build ISO via Docker, upload releases to Gitea), and add a daemon:typecheck npm script; update test/clean scripts

2026-01-09 - 0.1.1 - initial project setup & minor update

Consolidated initial project creation and a follow-up update into the initial release (0.1.1).

- 2026-01-08: initial commit — project scaffold and first files added.
- 2026-01-09: minor update and version bump to 0.1.1 — small edits and housekeeping.