Calimero Documentation

Why Calimero Exists

The Internet was designed to be **peer-to-peer**. From its origins in **DARPA's research on packet switching** — a response to the fragility of circuit-switched networks like telephony — the Internet's architecture has always favored decentralization. Protocols such as **TCP/IP** and **SMTP** embody this spirit: open, resilient, and without a central authority. Calimero builds upon that same idea.

- **Calimero is not a blockchain.**
- **Calimero is an application layer** built on top of the network — a place for collaboration, computation, and coordination between peers.
- Where a blockchain would rely on **consensus**, Calimero uses **CRDTs (Conflict-free Replicated Data Types)** for distributed consistency without global agreement.

Calimero is the layer you reach for when you *don't* need the guarantees (or costs) of consensus — when local autonomy and asynchronous coordination are enough.

Build Self-Sovereign Applications with CRDT-Powered P2P Sync

Calimero is a framework for distributed, peer-to-peer applications with automatic conflict-free data synchronization, user-owned data, and verifiable off-chain computing.

Attribute	What it means
Local-first by default	Your data stays on your node; you control replication
DAG-based CRDT sync	Conflict resolution without coordination, resilient offline
Event-driven architecture	Real-time updates emitted across participating nodes
Encrypted P2P channels	End-to-end secure sharing between context members

Attribute	What it means
WASM runtime	Build applications in Rust or TypeScript, ship deterministic WebAssembly
Multi-chain integrations	Connect NEAR, Internet Computer (ICP), Ethereum, and Stellar for attestations

Calimero is a privacy-focused application layer for peer-to-peer collaboration. This site stays concise on purpose: each section orients you in a few minutes, then links directly to the canonical GitHub READMEs for full architecture and workflows.

Quick Actions

Start here	What you get
Launch a local network	Bootstrap `merod` + Merobox and observe a context end-to-end.
Build from a template	Scaffold a Rust or TypeScript + React app with `create-mero-app`.
Explore a reference app	Learn from maintained examples such as Battleships or Shared Todo.
Understand the architecture	See how contexts, nodes, state sync, and identity fit together.

Choose Your Path

If you are...	Go to...	Why
New to Calimero	Introduction	Philosophy, architecture snapshot, and repo map.
Shipping an application	Builder Directory	Toolchain checklist, dev loop, and SDK links.

If you are...	Go to...	Why
Evaluating existing apps	App Directory	Spot maintained demos with direct README links.
Securing deployments	Privacy · Verifiability · Security	Isolation model, identity delegation, auditability patterns.
Looking for tooling	Tools & APIs	Runtime, admin, SDK, and automation catalog.

Highlights

- **Contexts as private networks** — CRDT-backed state and scoped storage so teams can collaborate without global consensus.
- **Hierarchical identities** — Root keys delegate client keys per device, integrating with NEAR wallets.
- **Modular runtime** — `merod` orchestrates networking (libp2p), storage, and WASM apps with JSON-RPC/WebSocket surfaces.
- **Repository-first docs** — Detailed flows live in project READMEs such as `calimero-network/core` and `calimero-network/merobox`.