

Web3User Permission Lab — Project Mindmap

Built prototype vs planned product • v0.2.0 • 22 October 2026

WEB3USER PERMISSION LAB

■■■ 1. PRODUCT

- ■■■ Goal: understandable, testable permissions for Web3 actions
- ■■■ Potential users: wallet users, dApps, wallet/account teams (hypotheses)
- ■■■ Output: ALLOW / DENY / UNKNOWN
- ■■■ Future UX: rule builder, explanations, policy history

■■■ 2. CORE ENGINE – BUILT PROTOTYPE

- ■■■ TypeScript: src/policy.ts
- ■■■ ERC-20 transfer(address,uint256), selector 0xa9059cbb
- ■■■ Exactly 68-byte calldata; validate ABI padding
- ■■■ Decode recipient and amount from calldata
- ■■■ Target + recipient allowlists; amount cap; validity; revocation
- ■■■ 34 tests passing; tsc check passes

■■■ 3. SECURITY – PARTIAL

- ■■■ Malformed/unsupported input → UNKNOWN
- ■■■ Amount cap is per-action, not cumulative
- ■■■ TODO: bind decision to immutable exact transaction
- ■■■ TODO: fail-closed enforcement integration
- ■■■ TODO: chain ID, sender, value, nonce, deadline, token identity
- ■■■ TODO: replay/concurrency controls + independent review

■■■ 4. WEBSITE – web3user.tech PLANNED

- ■■■ Home / mission / how it works
- ■■■ White paper (HTML + PDF) and version history
- ■■■ Roadmap / achievements / changelog
- ■■■ Security model / limitations / documentation
- ■■■ Safe simulated demo only at first
- ■■■ No wallet connection or seed/private-key collection

■■■ 5. HOSTING – OPTIONS, NOT CONFIGURED

- ■■■ Build static site locally first
- ■■■ Optional free provider preview URL
- ■■■ Candidates: Cloudflare Pages or GitHub Pages
- ■■■ Static paper needs no API, database, or smart contract
- ■■■ Keep custom domain DNS unchanged until approval

■■■ 6. TOOLS AND THEIR JOBS

- ■■■ TypeScript: policy implementation
- ■■■ Node.js + npm: local dev and dependency scripts
- ■■■ Vitest: unit/regression tests
- ■■■ tsc: type checking

- ■■■ npm audit: known dependency advisories at scan time
- ■■■ Git: local version history
- ■■■ GitHub: remote version control (repo private)
- ■■■ Markdown: docs and white-paper source
- ■■■ Later options: CI, ABI tooling, fuzz tests, static analysis
- 7. MILESTONES
- ■■■ M0-M3 COMPLETE: foundation, evaluator, tests, docs, private push
- ■■■ M4: exact transaction binding
- ■■■ M5: fail-closed enforcement interface
- ■■■ M6: chain-aware transaction context
- ■■■ M7: budgets, replay, concurrency
- ■■■ M8: adversarial/property testing
- ■■■ M9: independent review and controlled public demo
- 8. CONSTRAINTS
- ■0-first; avoid paid services for initial MVP
- No token, custody, real-money transactions, or on-chain deployment initially
- Keep GitHub private unless explicitly changed
- No claims of audit or production readiness without evidence

Status note: M0–M3 are supported by project outputs shared so far. M4–M9 are proposed work. Hosting providers and domain connection are not configured in this project.