

WDK Module

Title	WDK Module
Publish Date	Mar 24, 2026
Submission End Date	Jun 24, 2026
Reward	1'500 USDt

Problem Summary

What is the problem, deficiency, or gap which will be addressed by this project?

While WDK offers built-in modules for swap, bridge, and lending, the ecosystem still lacks coverage in areas such as additional swap protocols, cross-chain bridges, authentication methods, analytics, on/off-ramp integrations, and new blockchain or protocol support. Each missing module forces developers to build custom integrations, fragmenting the ecosystem and slowing adoption.

Impact / Reason Why

What is the metric or goal this bounty aims to improve?

New modules expand WDK's capability surface, making it more competitive and complete as a wallet development framework. Each module added reduces custom work for every subsequent team building on WDK.

Contributes to goals:

- Expand capabilities
- Improve existing tools
- Maximize developer satisfaction (NPS)

Scope Items

What important items are in scope for the project?

- Design, development, and testing of one or more new WDK modules following the existing module architecture and conventions
- Modules may cover areas such as: additional swap protocols, cross-chain bridges, lending platforms, authentication methods (social login, hardware keys), analytics and reporting, on/off-ramp providers, or new blockchain support
- Each module must follow WDK's modular architecture, implement the appropriate manager interfaces, and be compatible with Bare runtime
- Complete TypeScript definitions and JSDoc documentation
- Automated tests written in <https://github.com/holepunchto/brittle>
- A short demonstration video (2–5 minutes) showing the module in action

Scope Exclusions

What important items are out of scope for the project?

- Modifications to WDK core architecture or manager interfaces (modules must work within the existing framework)
- Full wallet applications (modules are standalone components)
- Backend infrastructure beyond what the module requires
- Production deployment or maintenance commitments

Deliverables

What are the outputs that must be submitted?

- Public GitHub repository with complete module source code
- Documentation: module overview, installation, configuration, usage examples, and API reference
- Automated test suite with <https://github.com/holepunchto/brittle>
- Demo video or example script demonstrating module functionality
- Pull Request to Tether GitHub repo

Acceptance Criteria / Definition of done

What needs to be achieved for this project to be considered completed?

- Module follows WDK's established architecture and naming conventions (e.g., wdk-protocol-, wdk-wallet-)
- Module is compatible with Bare runtime and works within the WDK Manager pattern
- TypeScript definitions are complete and accurate
- Automated tests achieve meaningful coverage of core functionality
- Documentation is clear enough for external developers to install and use the module independently
- All submitted code must follow clean code best practices, ensuring readability, modularity, meaningful naming, and maintainability.

Milestones

What are the intermediary checkpoints necessary to track project progress?

ID	Description	Deliverables	Reward
M1	Module Design & Proposal	Module specification, interface design, compatibility analysis	20% of grant
M2	Implementation & Testing	Working module with automated tests	50% of grant
M3	Final Delivery	Documentation, demo, PR submitted	30% of grant

Success Indicators / Key Results

What are the key results that help us measure if this project was a success?

- Module published to npm and integrated into WDK documentation
- 10+ projects using the module within 6 months
- Module passes WDK team code review and quality standards

Applicants Requirements

What requirements that applicants must meet?

- Experience developing JavaScript/TypeScript libraries or SDK modules
- Familiarity with WDK's module architecture and the create-wdk-module CLI tool
- Domain expertise relevant to the proposed module (e.g., DeFi protocols for swap/bridge modules, authentication standards for auth modules)

Reward & Payment Schedule

What are the payout structure and conditions?

Payment split across milestones: 20% upon design approval, 50% upon implementation, 30% upon final delivery and acceptance. Paid in USDT.

Resources & Links

What are additional useful links and resources for applicants?

- WDK Documentation: <https://docs.wallet.tether.io>
- create-wdk-module CLI
- WDK Module Architecture: <https://docs.wallet.tether.io/sdk/get-started>