Project Initiation Document - MidnightZK Off-Ramp SDK: ADA↔Web2 Payments (Cash App, Wise)

1. Executive Summary

An open-source, non-custodial SDK that enables ADA holders to privately off-ramp ADA into real-world fiat payment platforms including **Cash App, Wise, and Revolut**, without relying on centralized exchanges or custodial intermediaries.

The project combines **Cardano**, **Midnight zero-knowledge proofs**, and a **developer-friendly SDK** to preserve user privacy, ensure user-controlled funds, and enable seamless ADA-to-fiat settlement.

This Project Initiation Document defines the project scope, governance, risks, and a **16-week execution timeline aligned to Catalyst Milestones 1–4**.

---

2. Problem Statement

ADA and Cardano Native Token holders currently lack a **non-custodial, open-source, privacy-preserving** method to off-ramp funds into mainstream fintech platforms.

Existing solutions depend on centralized exchanges that introduce:

- Custodial risk
- High fees and withdrawal delays
- Mandatory KYC and privacy leakage
- Limited developer extensibility

This restricts ADA's real-world usability and adoption.

---

3. Project Objectives

Primary Objectives

- Deliver a **non-custodial ADA off-ramp SDK**
- Enable **direct fiat payouts** to Cash App, Wise, and Revolut
- Preserve privacy using **Midnight zero-knowledge proofs**
- Support integration by wallets, dApps, and merchants
- Release all outputs under the **MIT License**

Secondary Objectives

- Enable extensibility for additional payment rails
- Provide production-grade developer documentation
- Drive real-world ADA usage beyond centralized exchanges

---

4. Project Scope

In Scope

- Project Initiation & SRS documentation
- Technical Architecture Design
- SDK development and zk contracts
- ZK-based payee privacy verification
- Sandbox and testnet deployments
- Developer documentation (GitBook)
- Community testing and marketing

Out of Scope

- Custody of user funds
- Fiat liquidity provision

- Regulatory licensing as a financial institution
- Centralized exchange integrations

---

5. Deliverables by Milestone

Milestone	Key Deliverables
Milestone 1	Project Initiation Document, SRS v1.0, Risk Register
Milestone 2	Technical Architecture Document, diagrams, API designs
Milestone 3	SDK implementation, ZKP mechanism, testnet deployment
Milestone 4	Final testing, documentation, marketing, closure reports

---

6. 16-Week Project Timeline

Weeks 1–3: Project Initiation & Requirements (Milestone 1)

Activities

- Finalize project scope and governance
- Define functional and non-functional requirements
- Document user stories and developer flows
- Identify compliance and privacy parameters
- Perform risk assessment and mitigation planning

Outputs

- Project Initiation Document
- Software Requirements Specification (SRS v1.0)
- Risk Register
- Public GitHub repository (MIT-licensed)

Weeks 4–7: Technical Architecture Design (Milestone 2)

Activities

- Design high-level on-chain and off-chain architecture
- Define ADA → fiat data flows
- Specify ZK proof generation and verification workflows
- Design escrow smart contract interaction models
- Define API integration designs for Cash App, Wise, and Revolut

Outputs

- Technical Architecture Document (TAD)
- Architecture and data-flow diagrams
- API integration specifications
- Smart contract interaction models

Weeks 8–12: Development & Internal Testing (Milestone 3)

Activities

- Implement core SDK modules
- Develop ZKP-based payee privacy mechanism
- Deploy escrow smart contracts to Cardano testnet
- Integrate sandbox APIs for payment platforms
- Execute internal end-to-end testing and fixes

Outputs

- Working SDK and smart contracts
- Public Cardano testnet deployment references
- Internal testing report including:
  - ≥90% transaction success rate
  - Proof generation and verification performance metrics
  - Summary of issues identified and fixes applied
- Demo video of simulated ADA-to-fiat transaction

Weeks 13–16: Final Testing, Documentation & Project Closure (Milestone 4)

Activities

- Apply final improvements and performance optimizations
- Conduct final testing and validation
- Publish comprehensive developer documentation (GitBook)
- Prepare final project and close-out reports
- Launch SDK marketing and community testing campaign
- Track engagement and adoption metrics

Outputs

- Final SDK release (public GitHub repository)
- Final testing and release document (PDF)
- GitBook developer documentation
- Final close-out report and close-out video
- Recorded demo walkthrough video
- Social media posts, screenshots, and engagement analytics

---

7. Governance & Transparency

- Public GitHub repository from project start
- MIT-licensed code and documentation
- Open issue tracking and tagged releases
- Regular updates via X/Twitter, Discord, and Cardano Forum
- Milestone-based reporting to Catalyst

---

8. Risk Management (Summary)

Risk	Impact	Mitigation
Regulatory ambiguity	Medium	Privacy-by-design, modular compliance
API changes	Medium	Adapter-based integration
ZKP complexity	High	Experienced Midnight ZK engineers
Adoption risk	Medium	Strong documentation and outreach

(Detailed Risk Register included in SRS)

16-Week Project Gantt Overview (Role-Based)

Legend

- **PM** – Project Manager
- **TL** – Technical Lead
- **ZK** – Zero-Knowledge Engineer
- **DEV** – Development Team

High-Level Gantt Chart (Weeks 1–16)

Role / Phase	W1–3	W4–7	W8–12	W13–16
Project Manager (PM)	████	████	████	████
Technical Lead (TL)	████	████	████	███
ZK Engineer (ZK)	███	████	████	██
Dev Team (DEV)	██	███	████	███

Phase Breakdown by Role

Weeks 1–3: Project Initiation & SRS (Milestone 1)

- **PM:** Project planning, scope definition, milestone coordination

- **TL:** Requirements validation

Weeks 4–7: Technical Architecture Design (Milestone 2)

- **PM:** Milestone tracking, documentation coordination
- **TL:** Architecture ownership, design reviews, security considerations
- **ZK Engineer:** ZK proof workflows and verification paths
- **Dev Team:** API design, contract interaction modeling

Weeks 8–12: Development & Internal Testing (Milestone 3)

- **PM:** Sprint management, progress reporting
- **TL:** Code reviews, architectural enforcement
- **ZK Engineer:** ZK implementation, proof performance optimization
- **Dev Team:** SDK development, smart contracts, testnet deployment

---

Weeks 13–16: Final Testing, Documentation & Closure (Milestone 4)

- **PM:** Final reporting, Catalyst close-out, coordination
- **TL:** Final validation and release oversight
- **ZK Engineer:** Final proof validation and fixes
- **Dev Team:** Bug fixes, documentation support, demo preparation