Scalus TxBuilder — Catalyst Fund 13 Project Close-out Report

Scalus Multiplatform Tx Builder API

Catalyst Project Link

Project Number

1300009

Project Manager

Alexander Nemish

Dates

- **Start:** 2025-01-20
- **Completion:** 2025-12-23

Project Description

Scalus Multiplatform Tx Builder API allows constructing Cardano transactions on both JVM and JavaScript platforms from a single Scala 3 codebase. This frees developers from implementing the same transaction building logic twice — once for server-side (JVM) and once for client-side (browser/Node.js) applications.

The project delivers a TxBuilder API following best practices from MeshJS, Lucid Evolution, and Cardano Client Lib, compiled via Scala.js to JavaScript and published as an NPM package with full TypeScript definitions.

Project KPIs • How We Addressed Them

Project KPI	Outcome
Cross-Platform Code Reuse Single codebase for JVM and JS	One Scala 3 codebase compiles to JVM JAR (Maven Central) and JavaScript ESM/NPM package
Transaction Type Coverage Support all Conway era transactions	100% of Conway CDDL transaction types implemented with CBOR serialization
Library Integration Integrate with existing JS/JVM libraries	Full integration with Lucid Evolution for wallet management, key derivation, and transaction signing

Project KPI	Outcome
Transaction Validation Verify against real mainnet transactions	100,000+ mainnet transactions successfully deserialized and re-serialized with byte-by-byte match
Script Budget Accuracy Match cardano-node execution metrics	Plutus script evaluation with exact budget match verified against cardano-node; 130+ conformance tests passed
NPM Package Publication Publish usable JavaScript package	Published scalus package v0.14.0 on NPM with TypeScript definitions
DocumentationComprehensive API documentation	ScalaDoc for all public APIs, TypeScript type definitions, README with usage examples

---

Key Achievements

1. **Single-Codebase Multi-Platform Build** — One Scala 3 codebase compiles to both JVM (JAR for Maven Central) and JavaScript (ESM bundle for NPM) via Scala.js, enabling true code sharing between server and client applications.

2. **100% Conway Transaction Coverage** — Full implementation of all Conway era transaction types defined in the CDDL specification, with comprehensive CBOR serialization support and 63+ property-based roundtrip tests.

3. **Lucid Evolution Integration** — Complete integration with Anastasia Labs' Lucid Evolution TypeScript library for wallet management, mnemonic-based key derivation, and transaction signing on the JavaScript platform.

4. **Plutus Script Evaluation on JavaScript** — Fee and script budget calculation working natively in JavaScript, validated against mainnet execution metrics and passing 130+ Plutus conformance tests.

5. **Transaction Balancing** — Full transaction balancing implementation including UTXO selection (UtxoPool), change output calculation, and automatic fee balancing — all working cross-platform.

6. **Published NPM Package** — Released scalus v0.14.0 on NPM with optimized JavaScript bundle, TypeScript type definitions, and comprehensive README documentation.

7. **Comprehensive Integration Testing** — 9 transaction types (payment, minting, stake registration, stake delegation, DRep registration, vote del-

egation, proposal submission, voting, native script minting) successfully tested against a real Cardano node using Yaci DevKit.

Key Learnings

- Scala.js enables true code sharing between server (JVM) and client (browser/Node.js) platforms with minimal platform-specific code, making cross-platform DApp development practical.
- Integration with existing JavaScript libraries (Lucid Evolution) provides a familiar developer experience while leveraging Scalus's type-safe transaction building.
- Property-based testing with ScalaCheck catches edge cases in CBOR serialization that would be missed by example-based tests alone.
- Early integration testing against real Cardano nodes (via Yaci DevKit) surfaces protocol-level issues before production deployment.

Next Steps

- **Plutus V4 Support** — Implement new V4 built-ins and cost models as they become available.
- **Mesh.JS Integration** — Extend JavaScript ecosystem reach with Mesh.JS SDK integration.
- **Enhanced Developer Experience** — More tutorials, examples, and documentation for transaction building patterns.

Final Thoughts

This project successfully delivered a truly cross-platform transaction building API for Cardano. By leveraging Scala 3 and Scala.js, developers can now write transaction building logic once and deploy it to both JVM servers and JavaScript clients without duplication. The integration with Lucid Evolution provides a familiar developer experience, while Scalus's type-safe approach catches errors at compile time rather than runtime. We look forward to continued adoption and feedback from the Cardano developer community.

Resources

- GitHub • https://github.com/scalus3/scalus
- NPM Package • https://www.npmjs.com/package/scalus

- Documentation • https://scalus.org
- API Docs • https://scalus.org/api/index.html
- Starter Project • https://github.com/lantr-io/scalus-starter
- Discord • https://discord.gg/ygwtuBybsy

Close-out Video

Video demonstration of TxBuilder API on JavaScript platform

The video demonstrates that mainnet transactions can be deserialized and serialized back on the JavaScript platform, and all types of transactions can be constructed and successfully submitted to a real Cardano node using Yaci DevKit.