COSMEX Project Close-out Report

Project Identification

Field	Value
Project Name	COSMEX - Cardano L2 Order Book Exchange
Project Number	1100102
Project Manager	Alexander Nemish
Start Date	March 11, 2024
Completion Date	November 26, 2025
GitHub Repository	https://github.com/lantr-io/cosmex

KPI Analysis

Category Requirements

This project was submitted under the **Cardano Use Cases: Concept** category, which required demonstrating a novel use case for Cardano technology.

Requirement	How Addressed
Novel Cardano Use Case	Developed a Layer-2 order book exchange using star-shaped state channels—a specialized adaptation of Hydra Head protocol optimized for trading
Open Source	Full codebase published under open source license at https://github.com/lantr-io/cosmex
Documentation	Technical Whitepaper and Litepaper published with complete protocol specification
Proof of Concept	Working implementation demonstrated on Cardano Preprod testnet with verified transactions

Project KPIs

KPI	Status	Evidence
Smart Contract Implementation	Completed	CosmexValidator.scala - 4-phase state machine with 7 redeemers
PoC Exchange Server	Completed	WebSocket server with real-time order matching

KPI	Status	Evidence
On-chain Channel Operations	Completed	Verified transactions on Cardano Preprod testnet
Test Suite	Completed	Property-based and unit tests in `/src/test/scala/cosmex/`

Performance Summary

Key Achievements

1. State Channel Smart Contract

- Implemented 4-phase state machine: Open → SnapshotContest → TradesContest → Payout
- Ed25519 signature verification for snapshot authentication
- Balance preservation invariants enforced on-chain
- Trade execution validation with price-time priority

2. Off-Chain Exchange Implementation

- WebSocket API for real-time trading
- Limit order support (BUY/SELL) with automatic matching
- Multi-client channel management
- Snapshot signing protocol between clients and server

3. On-Chain Verification

- Alice's channel opening: tx/56f68c24...
- Bob's channel opening: tx/302c1584...

Impact

COSMEX demonstrates a viable path for Cardano L2 order book exchanges with:

- **Zero slippage**: Centralized matching without AMM price impact
- **MEV protection**: No front-running possible in private channels
- **Non-custodial**: Users retain cryptographic control; can always withdraw via contestation
- **Provable solvency**: On-chain state machine guarantees fund availability

Community Value

This project advances Cardano's L2 ecosystem by providing an alternative to Hydra that is:

- Simpler to deploy (star topology vs. full mesh)
- More flexible (users join/leave anytime without unanimous consent)
- Lower operational burden (no full node requirement for clients)

Documentation Evolution

Whitepaper

Version	Date	Key Changes
Initial Draft	Jan 2025	basic concepts: deposit/withdraw flow, contestation period, high-level design
Final v0.1	Nov 2025	Expanded with: detailed 4-phase state machine specification, Mermaid diagrams, security model, comparison with Hydra/AMMs, complete protocol flow, rebalancing mechanism

Litepaper

Version	Date	Key Changes
Initial Draft	Jan 2025	Draft overview emphasizing "provably solvent" and "RealFI" messaging
Final v0.1	Nov 2025	Restructured as technical summary: added "The Problem" and "The Solution" sections, technology overview, cleaner architecture diagram

---

Key Learnings

Technical Insights

1. **Script Size Optimization is Critical**: Early development required intensive optimization to fit the complex state machine within Cardano's script size limits. Initial validator was huge; through careful refactoring (custom ScriptContext, extracting methods, pattern optimizations) we reduced it to ~11730 bytes.

2. **Evolving with Scalus**: The project grew alongside the Scalus compiler, upgrading through 15+ versions (0.2-SNAPSHOT $\rightarrow$ 0.14). While this required ongoing adaptation to API changes, it allowed us to benefit from improved optimizations and the Plutus V3 migration support.

3. **Plutus V3 Migration**: Migrating from Plutus V2 to V3 required significant refactoring but enabled better performance and access to new Plutus features.

4. **State Channel Design Trade-offs**: The star-shaped topology (bilateral channels between each client and exchange) proved simpler to implement and reason about than full mesh networks, while still providing the core benefits of off-chain trading with on-chain security.

Integration Challenges

1. **Real Blockchain vs. Emulator**: Testing on actual Cardano testnet (preprod/preview) revealed issues not visible in emulator testing: clock skew handling, slot configuration differences, transaction confirmation timing, and UTxO indexing delays with providers like Blockfrost.
2. **Multi-Component Coordination**: Integrating the on-chain validator, off-chain server, WebSocket API, and blockchain provider required careful attention to state synchronization, especially during rebalancing and contested close operations.

Ecosystem Observations

1. **L2 Design Space**: Cardano's L2 landscape has room for specialized solutions beyond general-purpose protocols like Hydra. Application-specific state channels (like COSMEX for trading) can offer simpler deployment while maintaining security guarantees.

Challenges Encountered

Platform Choice and Scalus Evolution

The project was developed using Scalus, a Scala to Plutus compiler. While this enabled type-safe development and better tooling than raw Haskell/Plutus, it required continuous adaptation as the Scalus project evolved rapidly (15+ version upgrades from 0.2-SNAPSHOT to 0.14 over the project lifetime). Each major version brought API changes that required refactoring, though it also delivered improved optimizations and features.

Script Size Constraints

Early development revealed that Cardano's script size limits are a significant constraint for complex state machines. Initial implementation required intensive optimization work—reducing the validator from ~15200 to ~11730 bytes through techniques like custom ScriptContext structures, method extraction, and pattern optimization. This was time-consuming but essential for on-chain viability.

Protocol Design Iterations

The prototyping phase revealed significant complexity in establishing a secure channel opening and closing process. We explored three different solutions during

the design phase, evaluating trade-offs between security, simplicity, and user experience. This iterative design process, while time-consuming, resulted in a more robust protocol. A Hydrozoa L2-inspired approach (where both parties sign an opening deposit withdrawal transaction, similar to Lightning Network) remains a candidate for future protocol enhancements.

Security Edge Cases

Prototyping under real-world conditions exposed several edge cases—particularly around transaction contestations and client withdrawal flows that could potentially be exploited in malicious attacks. We invested significant effort to improve both security and stability for these scenarios, which contributed to the schedule extension but resulted in a more production-ready implementation.

Testnet Integration

Testing on actual Cardano testnet (preprod/preview) revealed many issues not visible in emulator testing:

- Clock skew between local time and blockchain slot time required buffer adjustments
- Different slot configurations between networks (preprod vs preview)
- Transaction confirmation delays and UTxO indexing lag with Blockfrost provider
- Large WebSocket message handling for signed transaction payloads

These challenges were addressed through iterative development and led to a Delivery Schedule Change Request, extending the project timeline while maintaining all original deliverables and objectives.

Final Thoughts

COSMEX successfully demonstrates that specialized Layer-2 solutions for Cardano are both feasible and practical. The project achieved all proposed deliverables: a working smart contract, proof-of-concept exchange server, verified testnet transactions, and comprehensive documentation.

The star-shaped state channel architecture proved to be a viable alternative to full Hydra for exchange use cases, offering simpler deployment while maintaining the core properties of non-custodial trading with on-chain security guarantees.

We believe this project contributes valuable knowledge to the Cardano ecosystem about L2 design patterns, and provides a foundation that could be extended to production use with additional engineering investment. The open-source codebase and detailed documentation should help other developers exploring similar solutions.

The experience reinforced that Cardano's extended UTxO model and Plutus smart contracts are well-suited for state channel protocols, and that the Scalus toolchain significantly improves developer productivity for complex validator development.

Documentation & Links

Source Code

- **Repository**: https://github.com/lantr-io/cosmex

Technical Documentation

- Whitepaper | PDF
- Litepaper | PDF

Demo

- **Trading Demo Video**: https://www.youtube.com/watch?v=A7XQOaOktvQ

Close-out Video

https://www.youtube.com/watch?v=sBm4BxZp5P4

Future Plans

- Mainnet deployment with production-grade infrastructure
- Integration with existing Cardano wallets
- Additional trading pairs and order types
- Community partnerships for liquidity provision