Ministry of Education and Science of Ukraine
NATIONAL UNIVERSITY OF KYIV-MOHYLA ACADEMY
Department of Informatics of the Faculty of Informatics

Creation of an atomic cross-chain escrow, which supports EVM, Solana and TON

Volodymyr Rastiehaiev

Thesis supervisor - Kyrylo Gorokhovskyi

2025

Terms

- Escrow is a financial arrangement where a third party holds and manages assets or funds on behalf of two parties involved in a transaction, releasing them only when agreed-upon conditions are met.
- Atomic escrow is a type of escrow mechanism that ensures a transaction between parties is completed entirely or not at all.

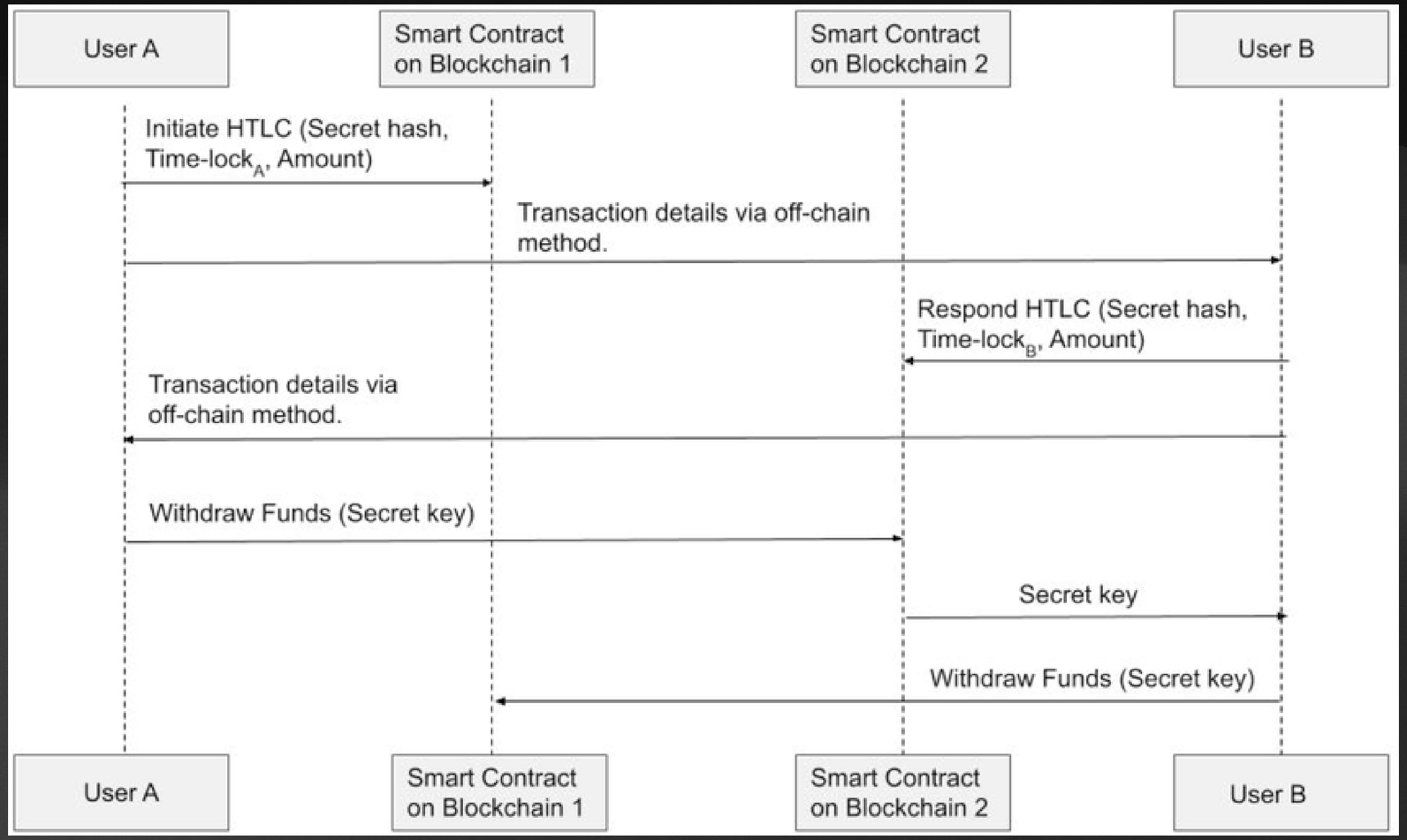
Objectives

- Addresses the interoperability issue.
- Research approaches.
- Create an HTLC-based atomic cross-chain escrow for Solana, TON and EVM.
- Develop contracts for all mentioned blockchains.

What is HTLC?

- HTLC (Hashed TimeLock Contract) is a smart contract used to enable secure, time-bound transactions between untrusted parties.
  It requires the receiver to provide a cryptographic proof (a hash) within a specified time to claim the funds.
  If the proof isn’t provided in time, the sender can reclaim the funds.

HTLC diagram



HTLC sequence diagram

2025

EVM

- EVM has two main types of accounts:
  - Externally Owned Accounts (EOA) – Controlled by private keys, used by users to send transactions and hold ETH or tokens.
  - Contract Accounts – Controlled by code, execute logic when triggered by transactions or messages.
- Smart contracts store both code and state on EVM.

Solana

- In Solana, accounts store data and are central to the execution of programs.
- There are two main types of accounts:
  - Data Accounts - These store persistent data and are used by programs to maintain state.
  - Program Accounts - These contain the compiled executable code and are marked as executable.
- On Solana, programs are stateless and require accounts that store state to be passed on invoke.

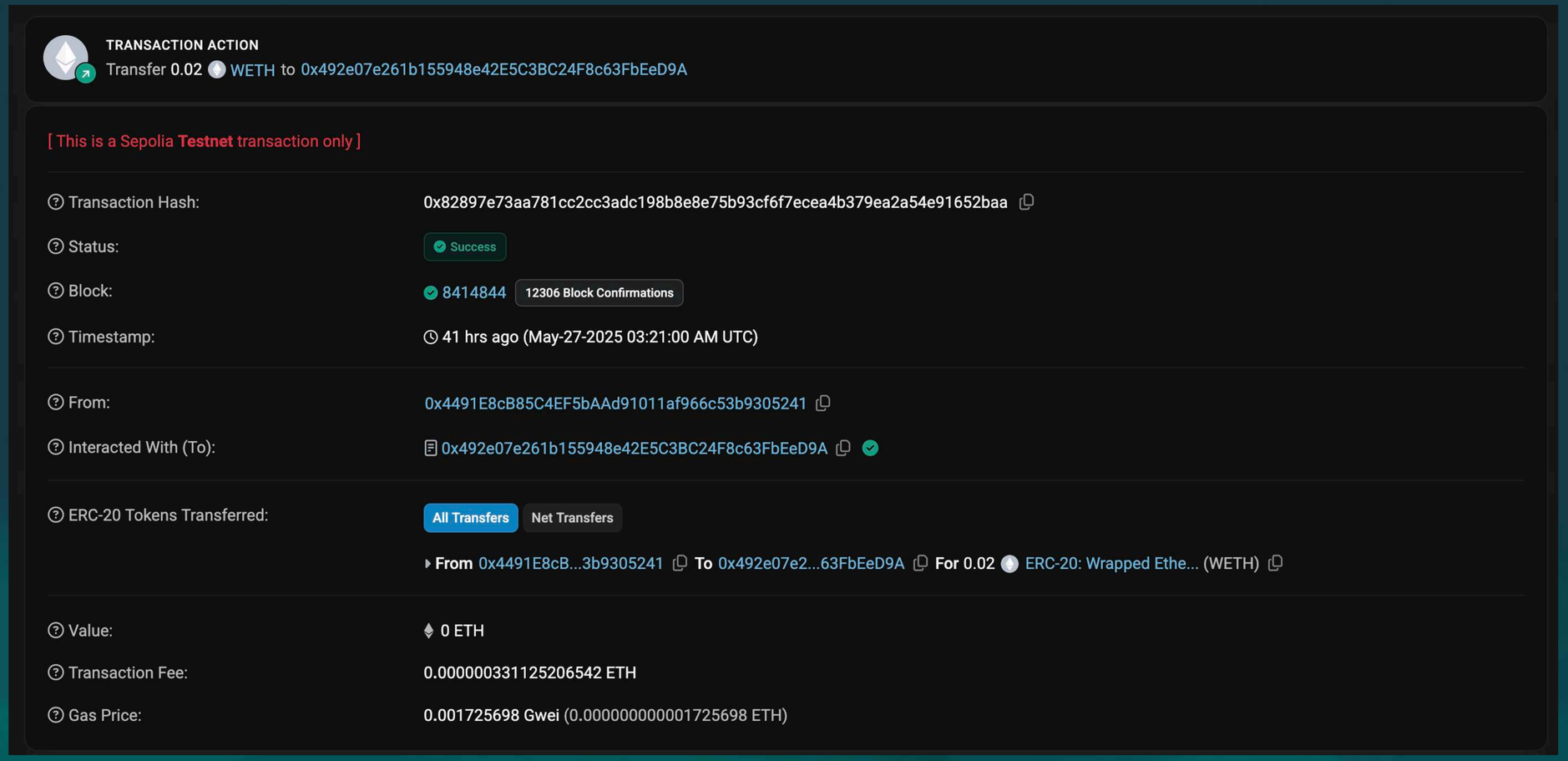
TON

- In TON, everything is a smart contract with its own code and persistent data.
- Unlike Solana and EVM, TON is asynchronous:
  - Actor Model: Each smart contract is an independent actor that communicates via messages, similar to concurrent systems.
  - Contracts execute only when they receive messages. All communication between contracts is asynchronous, enabling parallelism but making it important for developer to think about contracts interactions and architecture.

Technologies used

- For hashing algorithm, keccack256 was chosen as it is available in all of three blockchains and is needed to achieve compatibility with EVM.
- EVM:
  - Solidity language for smart contracts
  - Hardhat js framework for testing & deploying
- Solana:
  - Rust language for smart contracts & testing
  - Solana-sdk library for deploying smart contracts & blockchain interactions
  - Litesvm library for smart contract testing
- TON:
  - Tolk language for smart contracts
  - Blueprint js framework for testing & deployment

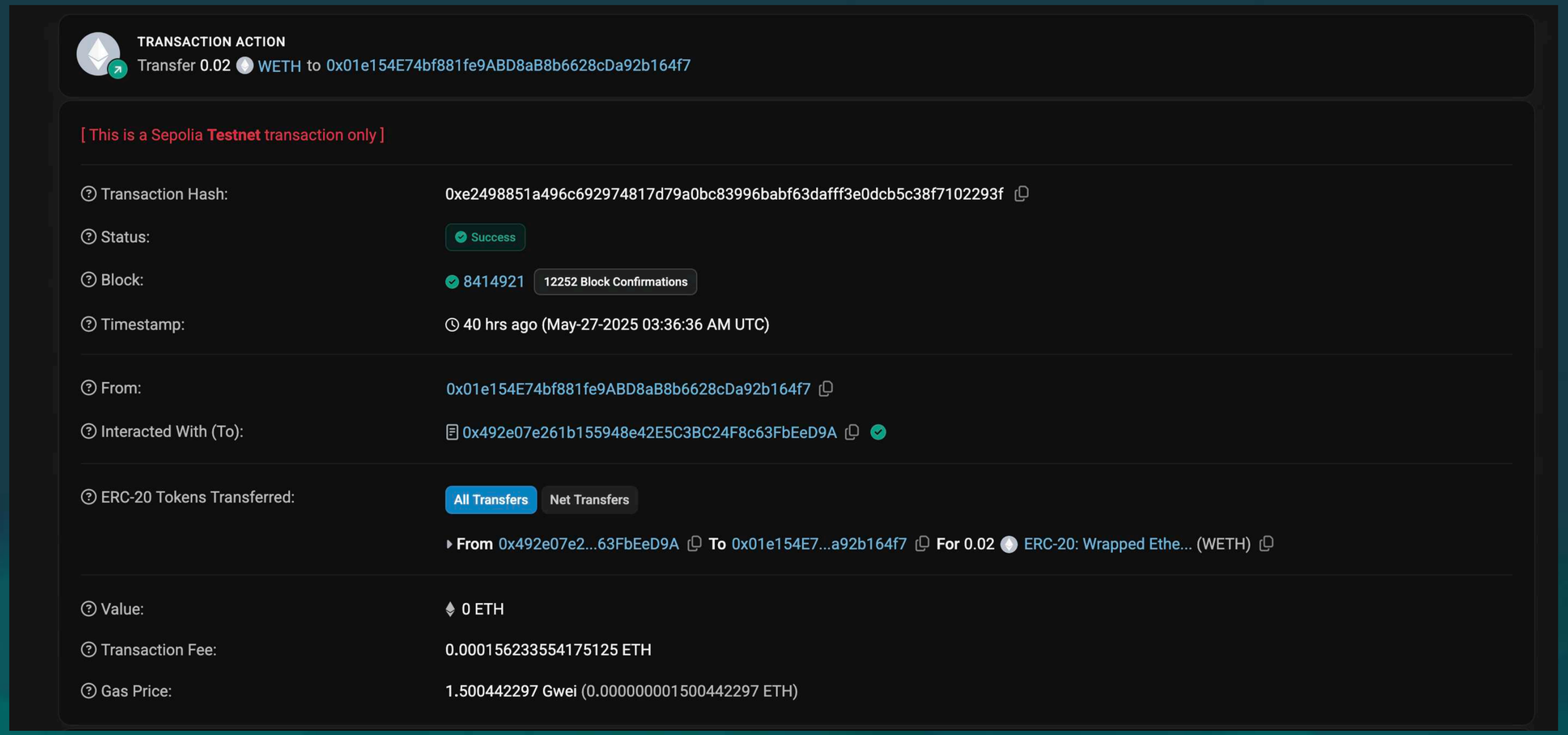
EVM



createTrade transaction on Sepolia testnet

2025

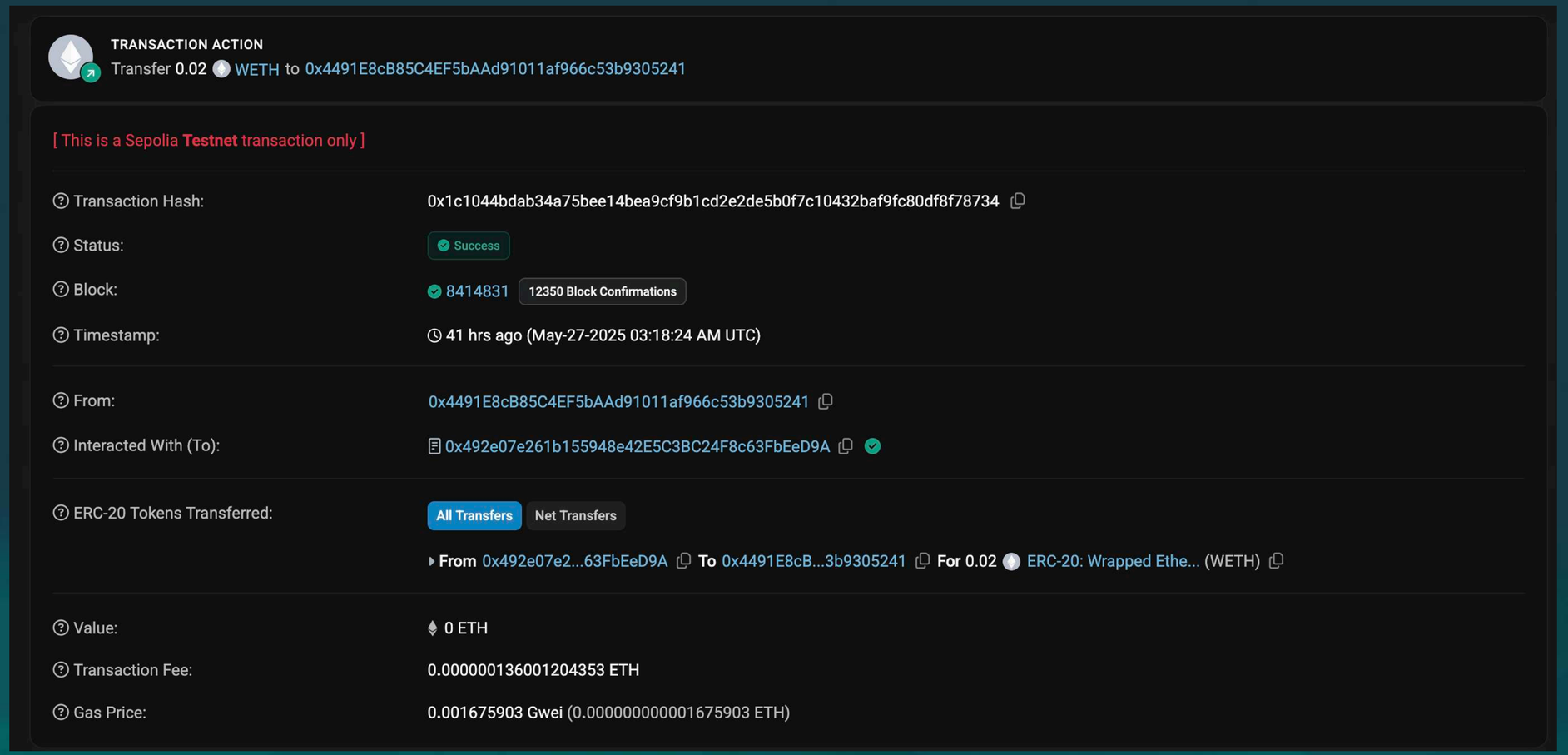
EVM



Withdraw transaction on Sepolia testnet

2025

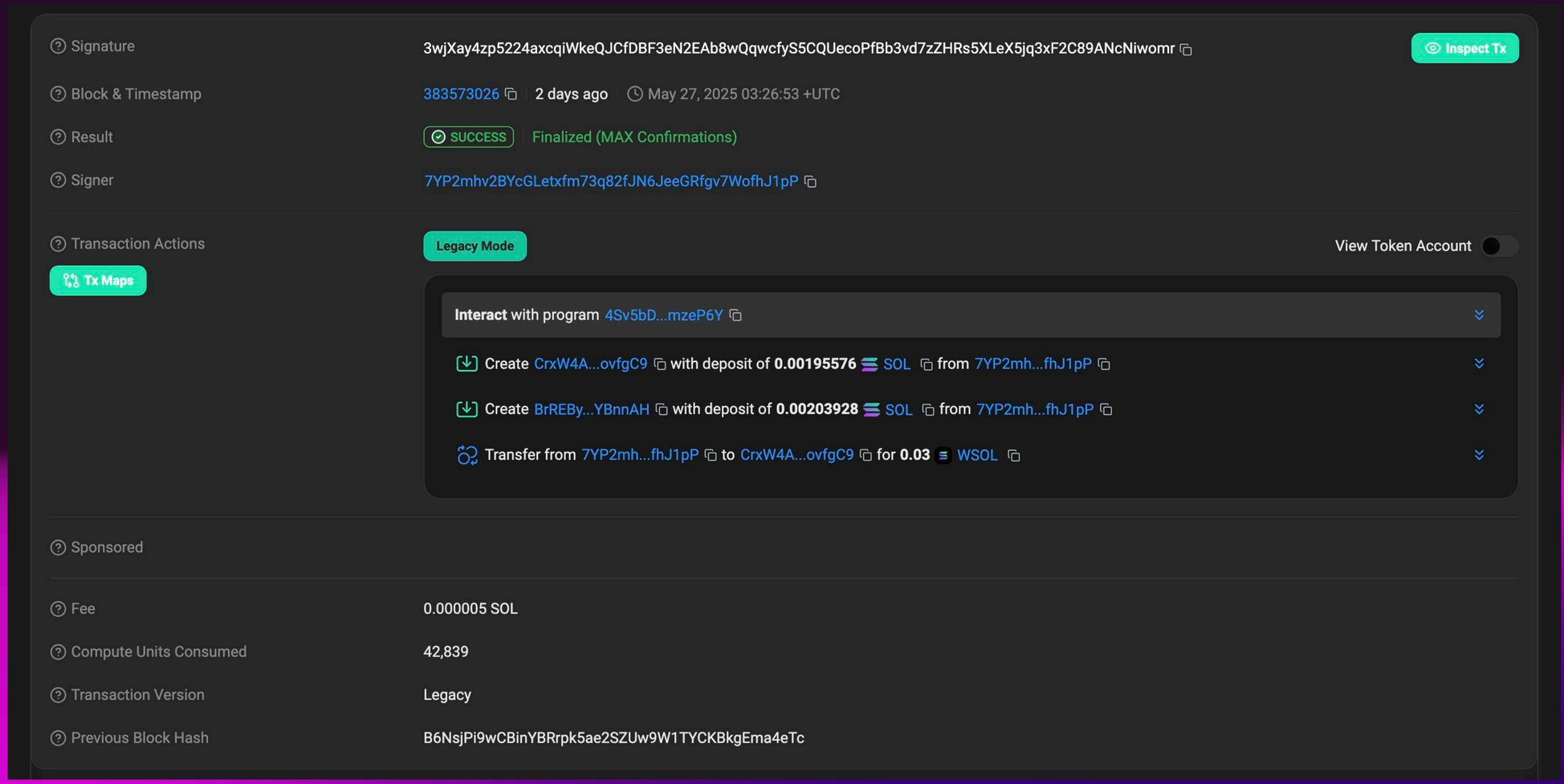
EVM



Refund transaction on Sepolia testnet

2025

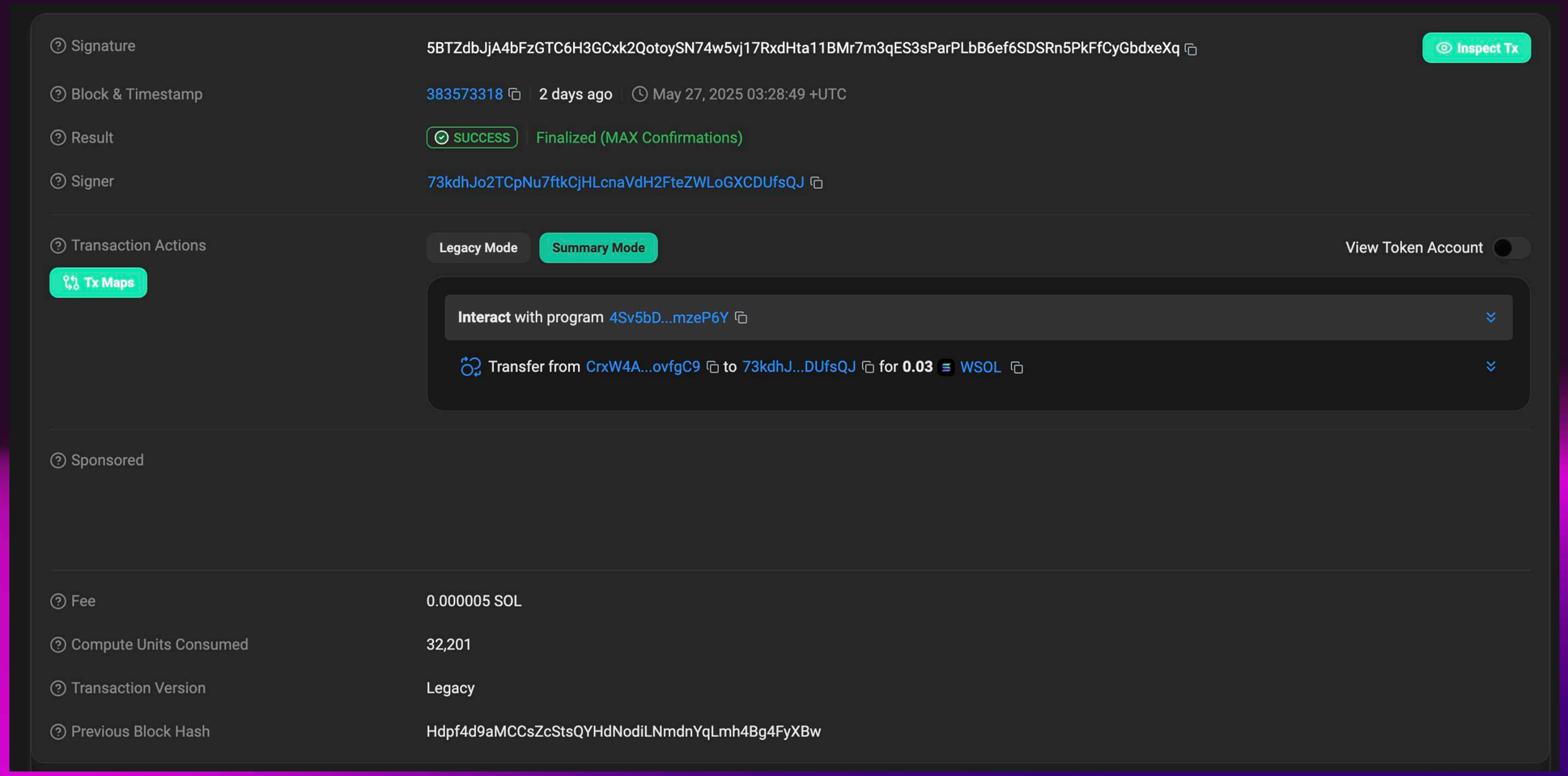
Solana



Deposit transaction on Solana devnet

2025

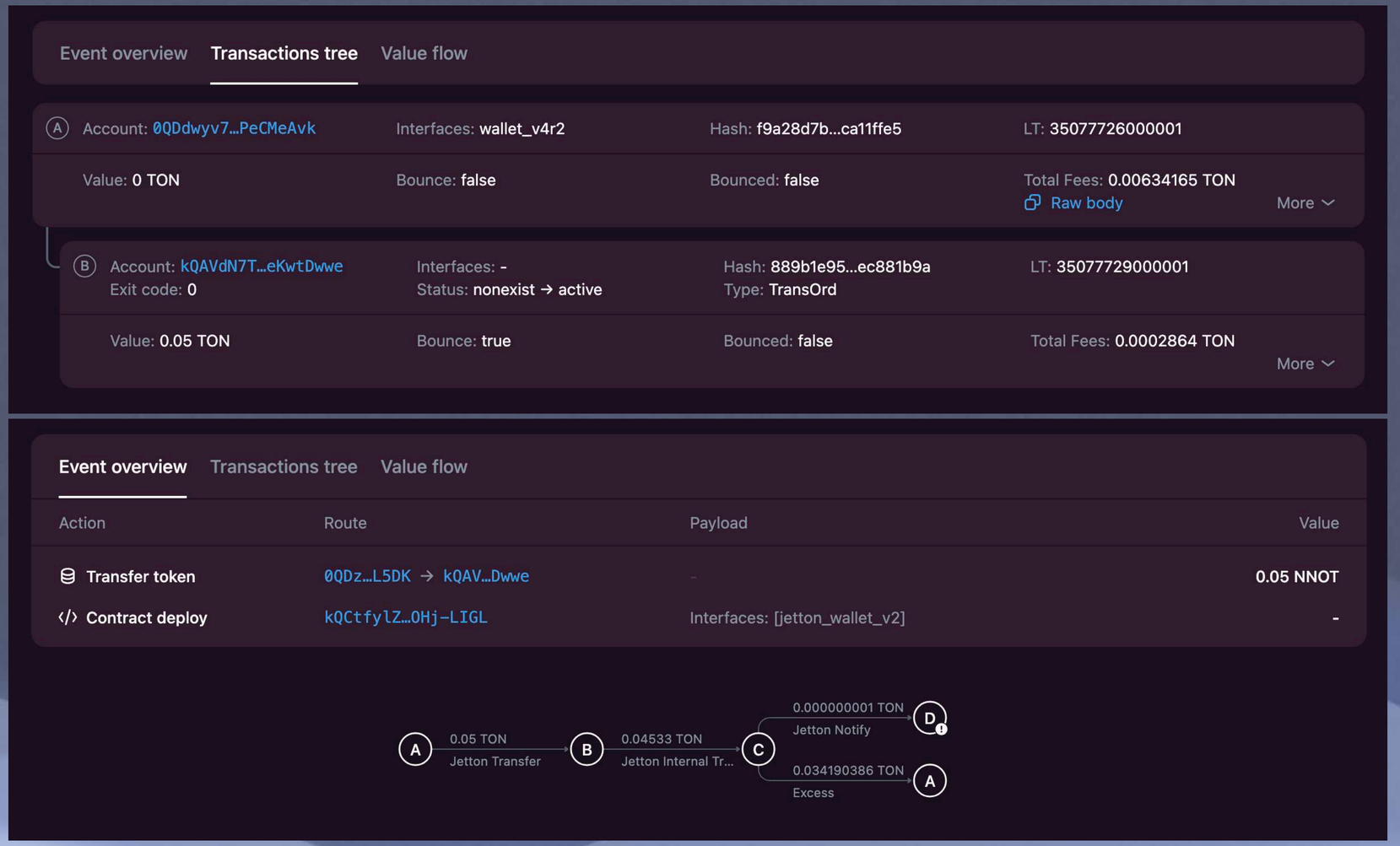
Solana



Withdraw transaction on Solana devnet

2025

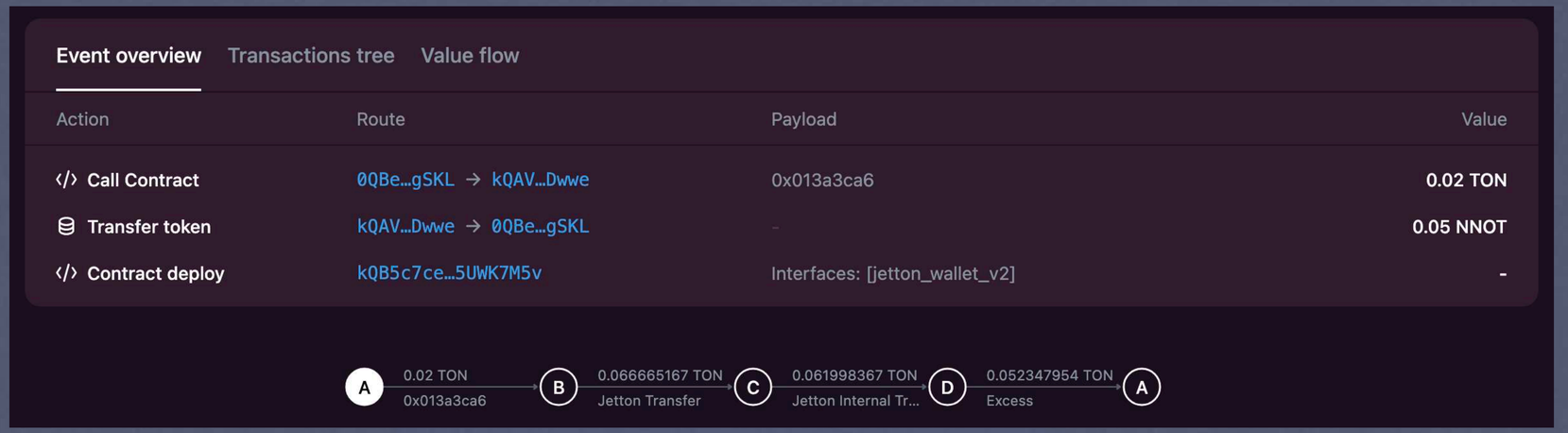
TON

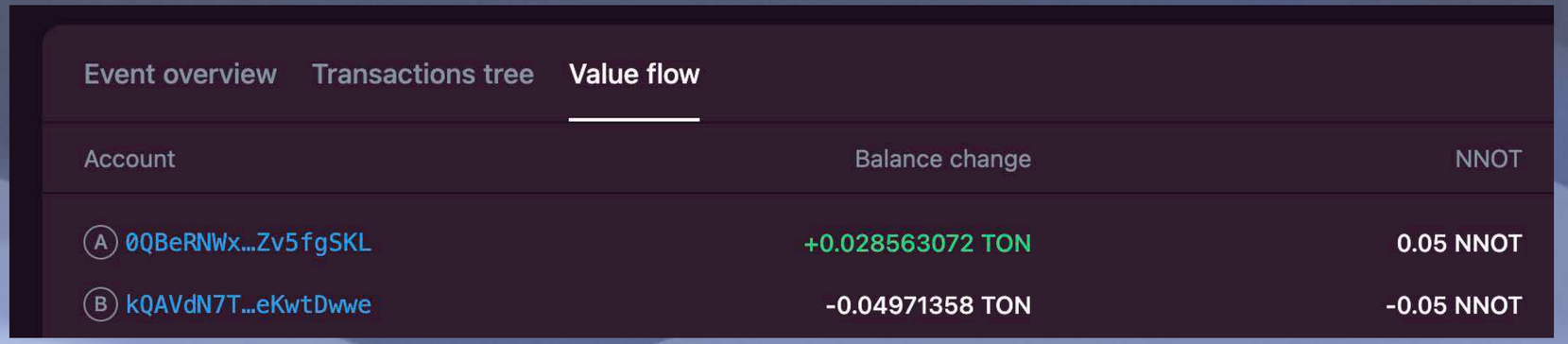


Contract deploy & deposit on TON testnet

2025

TON





Withdraw on TON testnet

2025

Conclusion

- Traditional methods often of cross-chain escrow often rely on trusted intermediaries or complex bridging mechanisms, introducing points of failure and security vulnerabilities.
- This work addressed this challenge by focusing on the design, development, and evaluation of an atomic cross-chain escrow system.
- As a result, the system was implemented to support specifically EVM, Solana and TON.
- This work successfully demonstrated the technical feasibility of such a system, providing a functional and evaluated solution.

Thanks for your attention