Technical Description of Trading on POT

1. Introduction

POT is a decentralized platform (DEX) on the TON blockchain that allows users to trade futures on any tokens using
liquidity provided by liquidity providers (LP). Unlike synthetic solutions, POT uses real liquidity: when positions are
opened, tokens are sold or bought on the DEX, and upon closing, the reverse operation is executed.

2. Trading Mechanics

2.1. Opening a Position

- The user selects an asset, leverage, and trade direction (long or short).
- The platform takes liquidity from LPs and:
- **If long:** Buys the asset on the DEX.
- **If short:** Sells the asset on the DEX.
- The transaction is confirmed through the TON API and executed at the market price.

2.2. Closing a Position

- When closing a trade:
- **Long:** The asset is sold on the DEX, and the profit/loss is returned to the user.
- **Short:** The asset is bought on the DEX, and the profit/loss is returned to the user.
- The platform automatically balances LP liquidity.

2.3. Liquidations

- If the asset price reaches the liquidation level, the system **automatically executes liquidation through the DEX** to
cover losses.
- Liquidations are executed partially (partial liquidation) to reduce market impact.
- The liquidation price is verified through the TON API, which updates data every few seconds. However, delays may
occur, especially under high network load, affecting liquidation accuracy. In such cases, the system utilizes additional
price data sources (e.g., DEX oracles) to ensure the accuracy of the exchange rate.

3. Risks and Solutions

3.1. Risk of Low Liquidity on the DEX

Problem: If the DEX lacks sufficient liquidity, large trades can cause slippage.
Solutions:
- Limits on order size based on liquidity availability.
- Splitting large orders into smaller parts to minimize price impact.
- Use of algorithmic orders.

3.2. Front-running and MEV Attacks

Problem: Bots can predict and front-run large orders, manipulating the price.
Solutions:
- Delayed execution for large orders.
- TWAP (time-weighted average price) instead of market price for liquidations.
- Use of hidden orders.

3.3. TON API Delay

Problem: If the API delays price updates, liquidations may be executed inaccurately.
Solutions:
- Use multiple price sources (DEX API + oracles).
- Check the latest API update before executing trades.

3.4. Slow Liquidations

Problem: If liquidations execute too slowly, LPs may incur losses.
Solutions:
- Partial liquidations instead of full liquidation.
- Use of a reserve liquidity pool for instant execution.
- Automatic liquidation execution on an alternative DEX in case of liquidity shortages.

3.5. Bad Debt

Problem: If an asset price drops too quickly, LP liquidity may be insufficient to cover losses.
Solutions:
- Insurance fund (2-3% of fees).
- Auto-Deleveraging (ADL) loss redistribution among large positions.
- Automatic increase of margin requirements (MR/MMR) during periods of high volatility.

3.6. High DEX Fees

Problem: Frequent trades result in high gas fees.
Solutions:
- Batching orders.
- Optimization of smart contracts to reduce fees.
- Switching to lower-cost DEXs when necessary.

3.7. LP Liquidity Imbalance

Problem: If most traders are long or short, the liquidity pool may become unbalanced.
Solutions:
- Dynamic funding (the greater the imbalance, the higher the funding payments).
- Limits on one-sided positions.
- Additional incentives for LPs (bonuses for replenishing scarce liquidity).

4. Conclusion

POT uses real liquidity by selling and buying assets on the DEX, making the platform transparent and decentralized.
However, this approach introduces risks related to liquidity, manipulation, execution speed, and commission costs.
Implementing adaptive risk management mechanisms, oracles, dynamic funding, and insurance funds will make the
platform more resilient to market fluctuations.

5. Next Steps

1. Development of a liquidity management algorithm for large orders.
2. Integration of multi-oracles for price verification.
3. Implementation of a buffer liquidity pool to accelerate liquidations.
4. Optimization of smart contracts to reduce fees.
5. Testing of dynamic funding mechanisms.