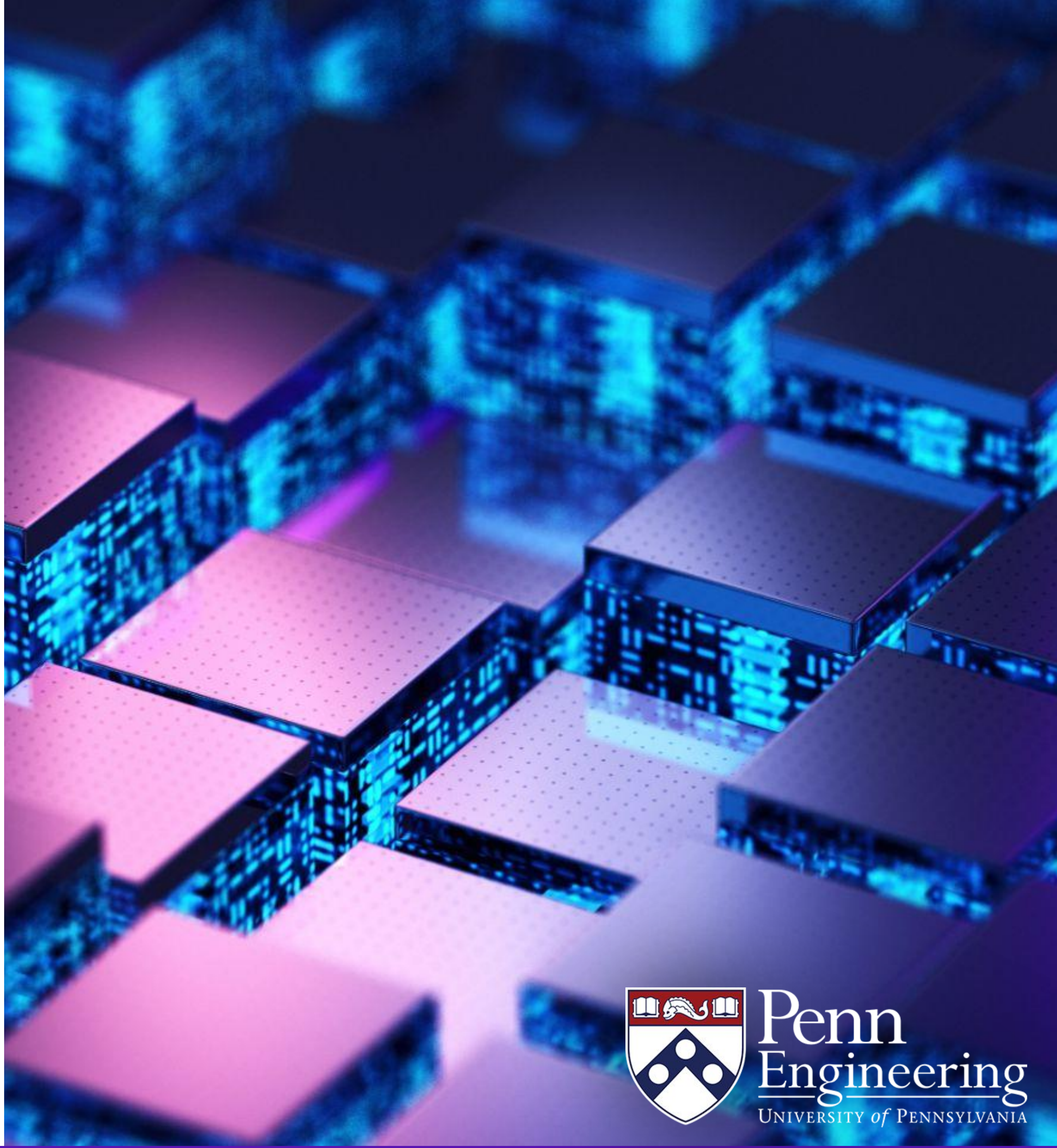
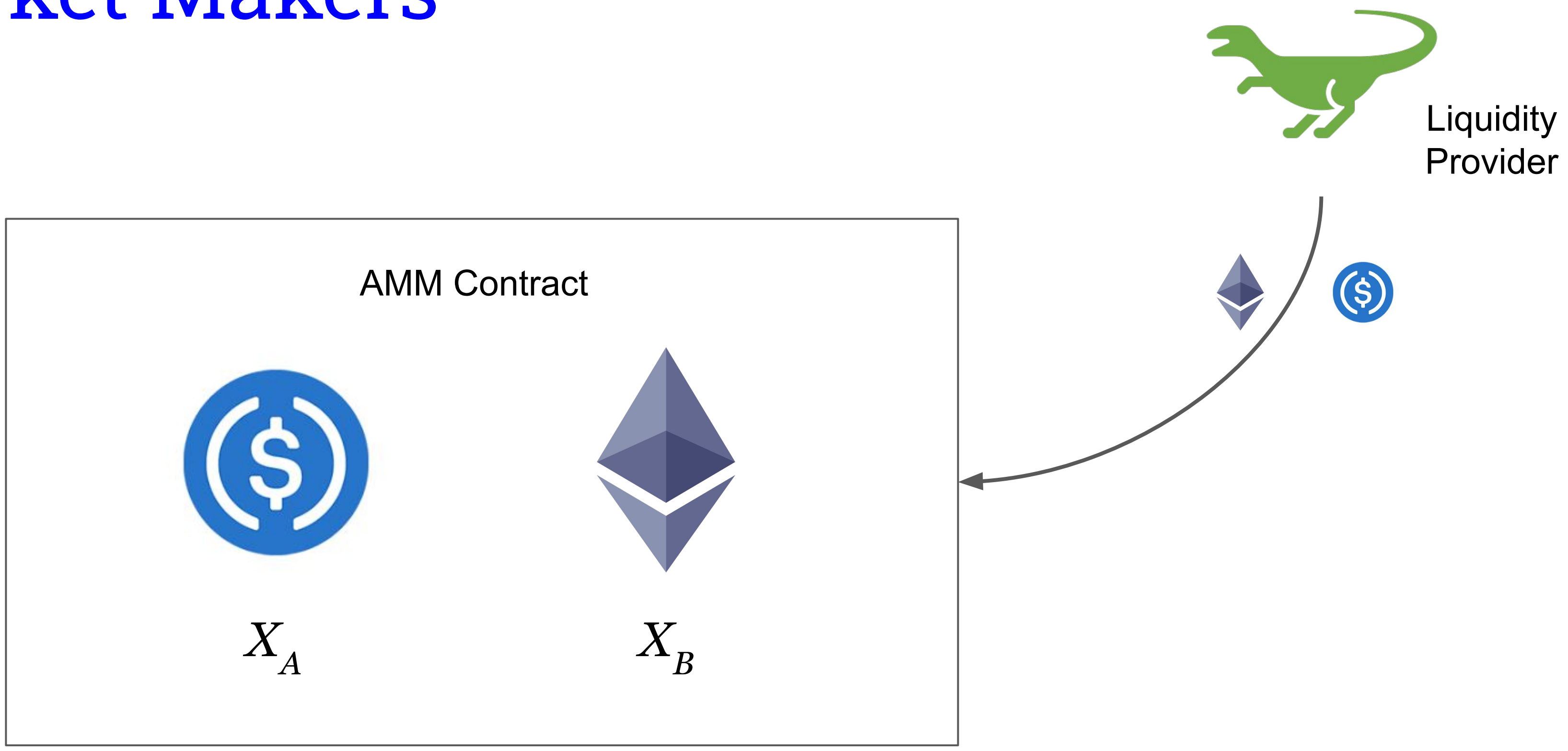
EAS 5830: BLOCKCHAINS

AMMs

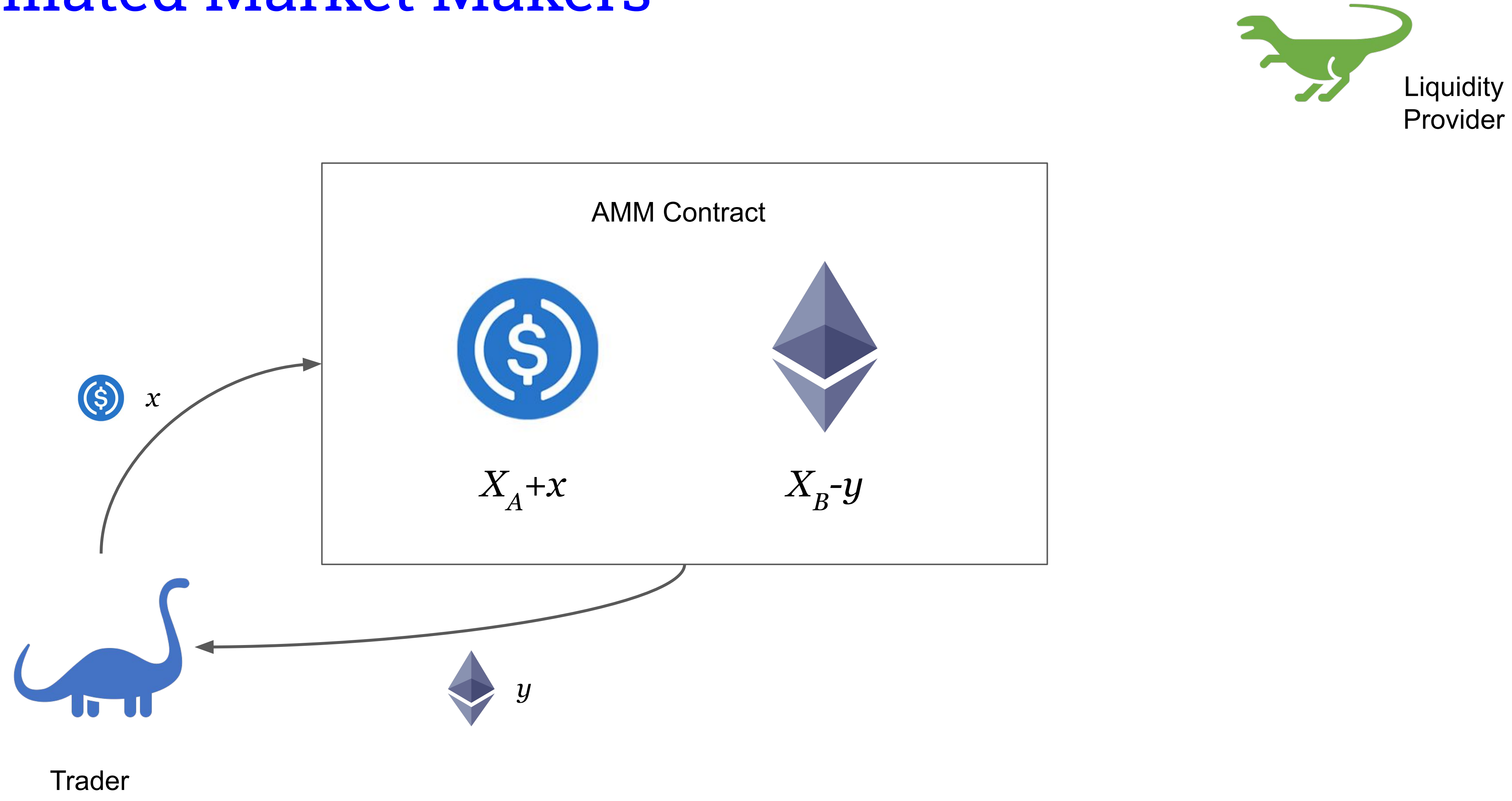
Professor Brett Hemenway Falk



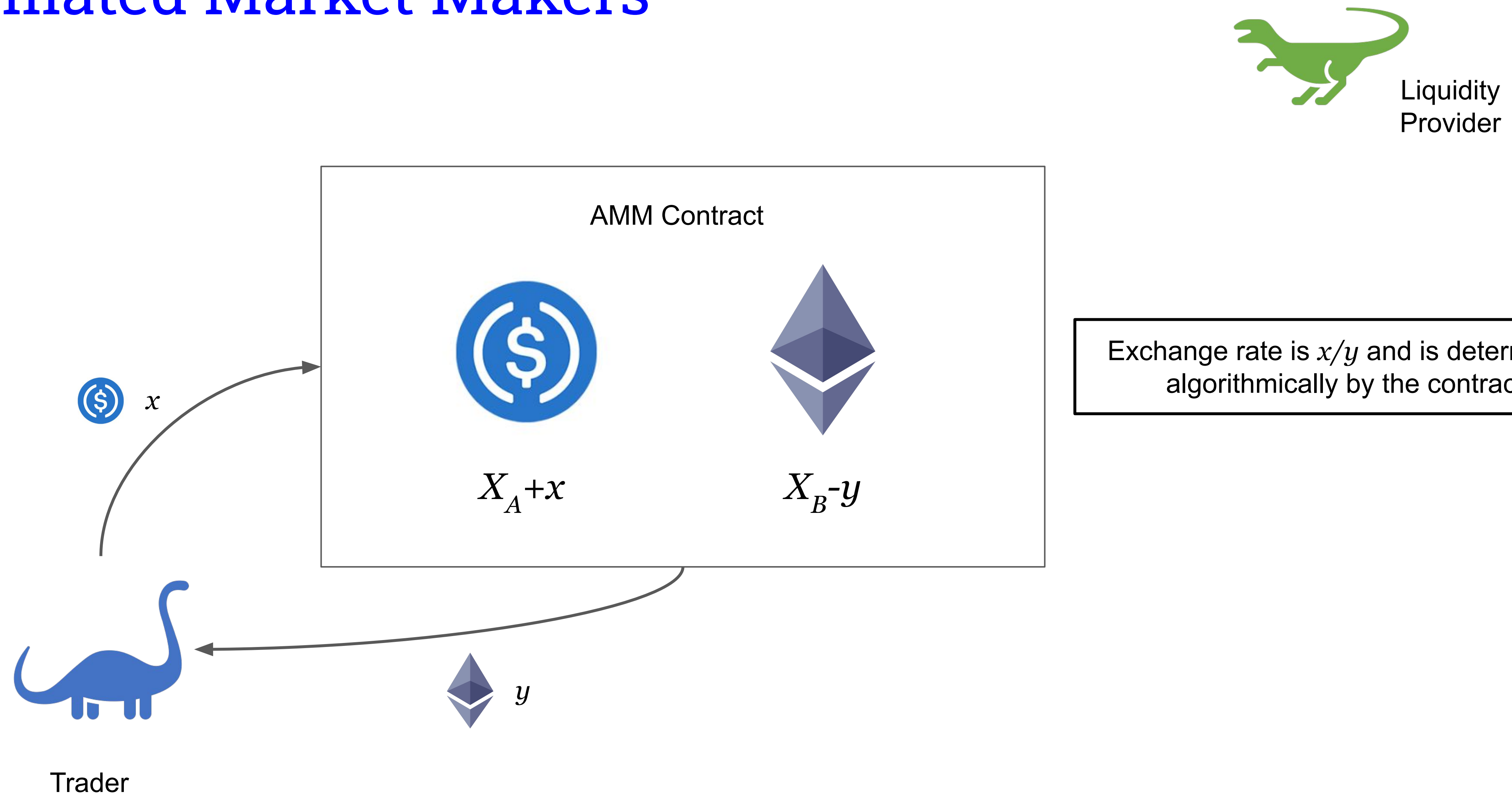
Automated Market Makers



Automated Market Makers

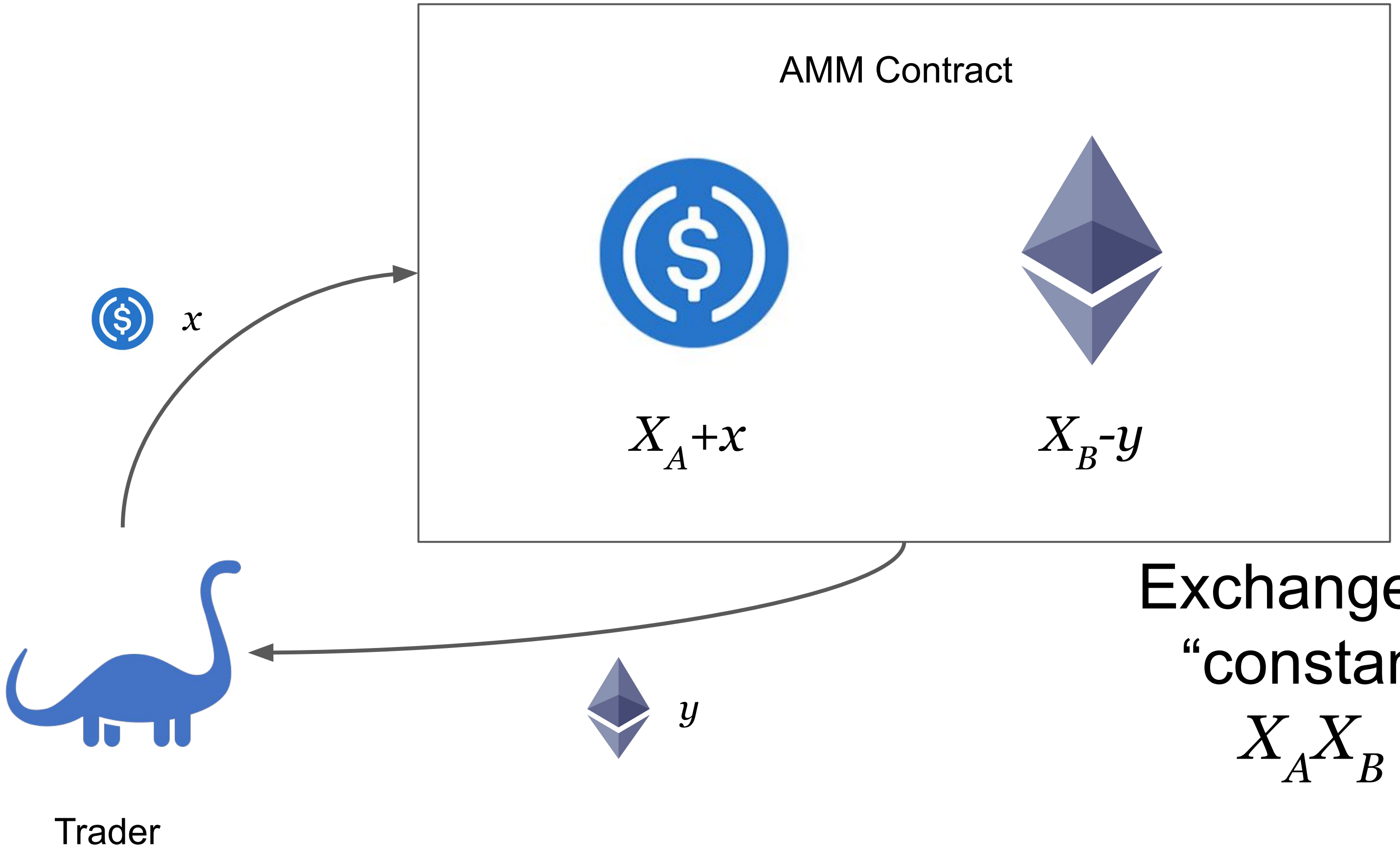


Automated Market Makers



Exchange rate is x/y and is determined algorithmically by the contract

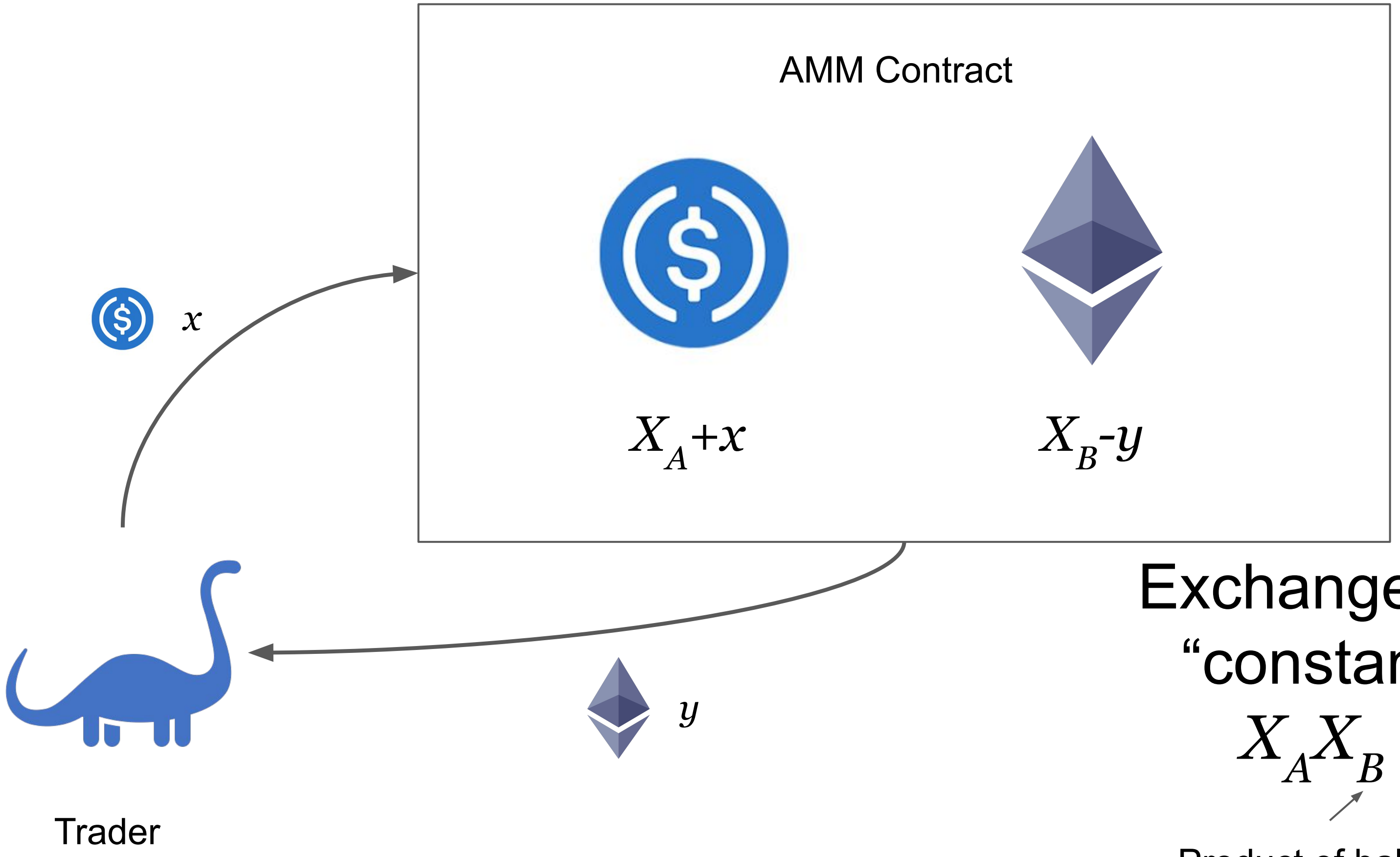
Automated Market Makers



Exchange rate determined by
“constant product formula”

$$X_A X_B = (X_A + x)(X_B - y)$$

Automated Market Makers

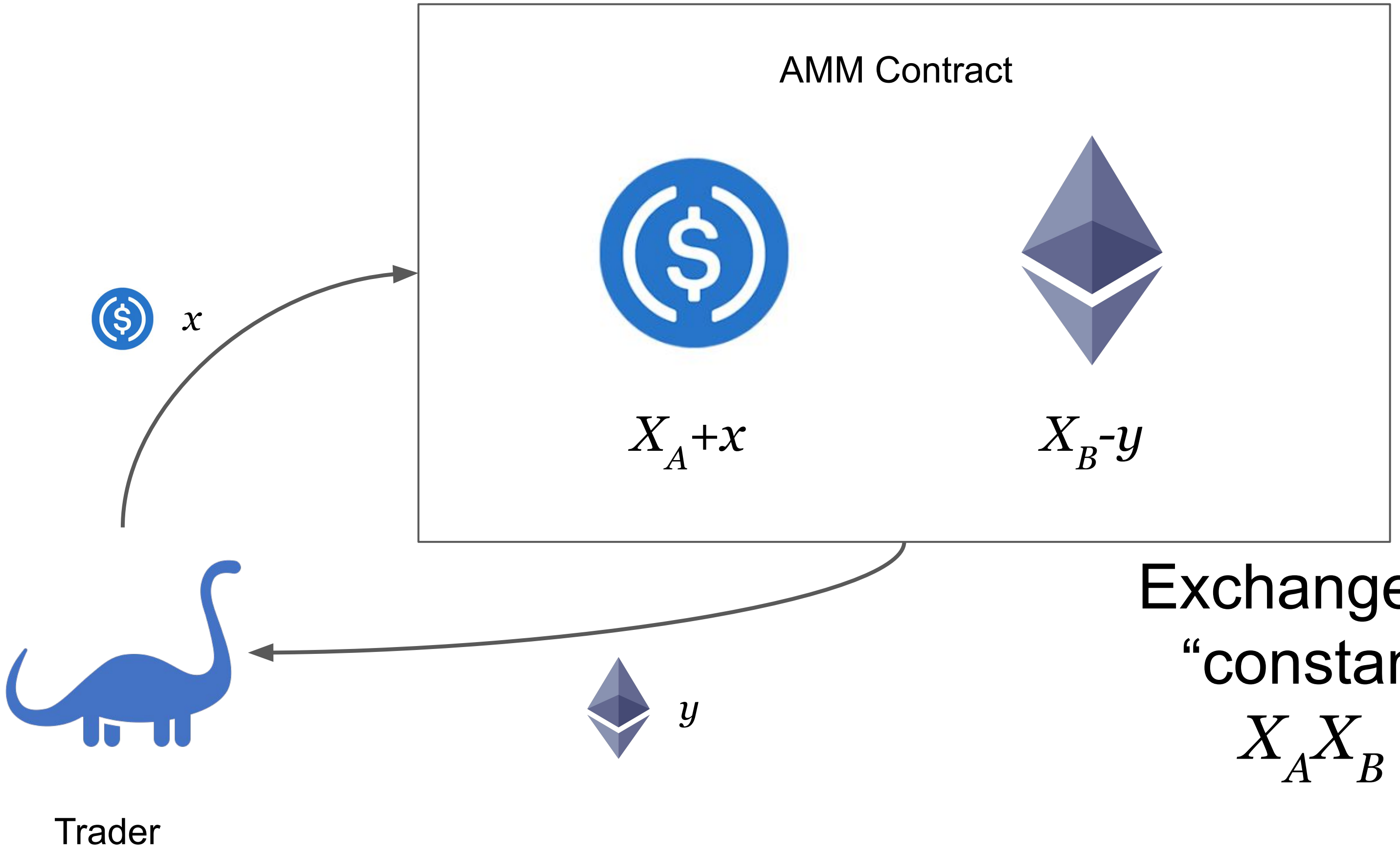


Exchange rate determined by "constant product formula"

$$X_A X_B = (X_A + x)(X_B - y)$$

Product of balance **before** the trade

Automated Market Makers



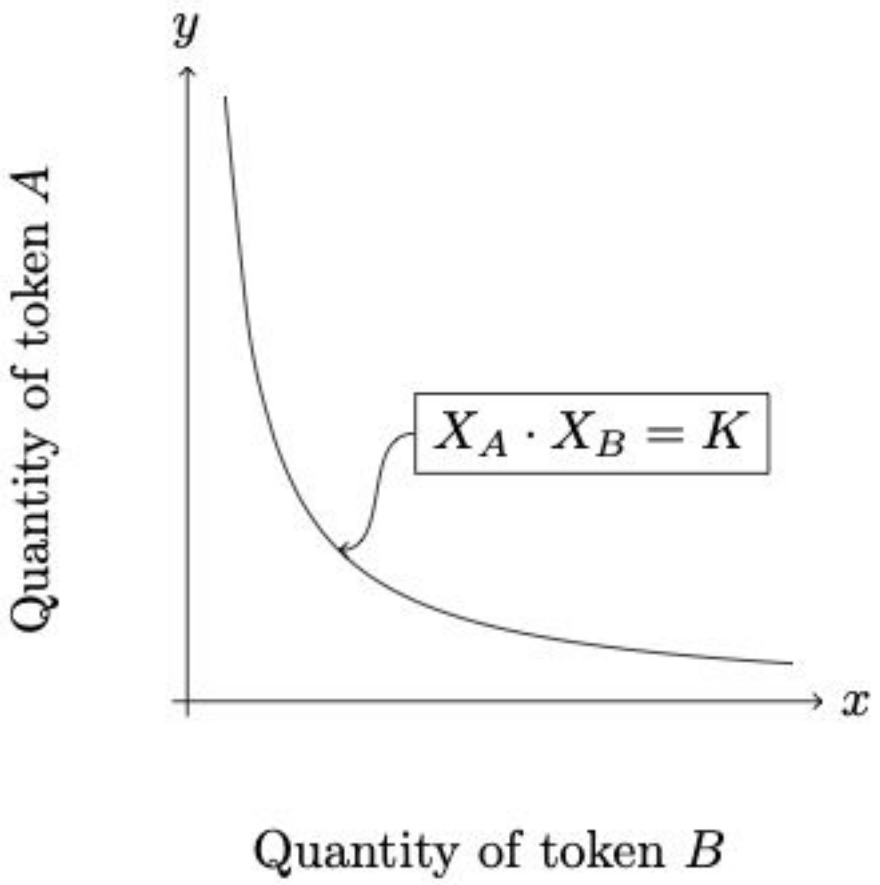
Exchange rate determined by
"constant product formula"

$$X_A X_B = (X_A + x)(X_B - y)$$

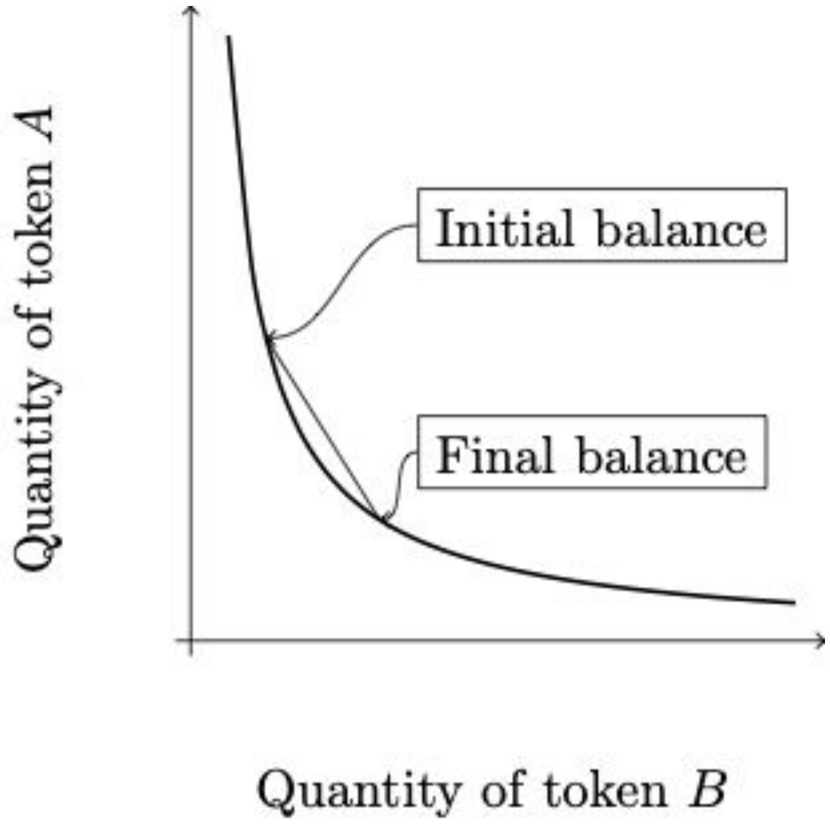
Product of balance
after the trade

Constant-Function Market Makers

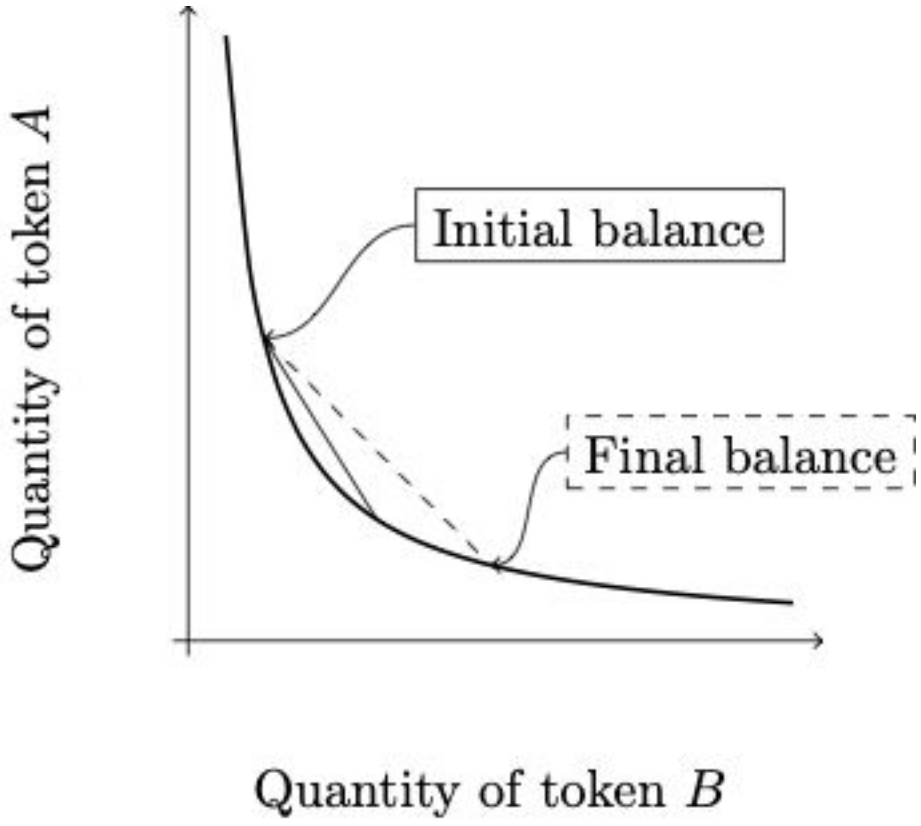
- Contract balance lies on hyperbola
- Contract Maintains invariant $X_A \cdot X_B = K$
- Client wants to trade x units of A for B
  - Receives y units of B
  - $(X_A + x) \cdot (X_B - y) = K$
  - $y = X_B - K/(X_A + x)$



Exchange rate depends on quantity



Small trade



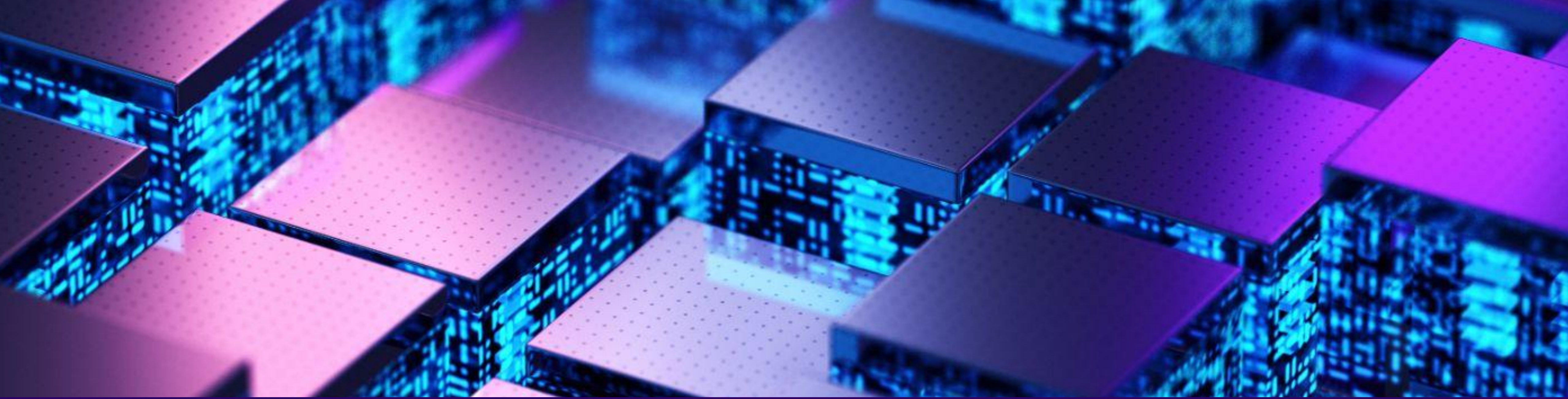
Large trade

Alternative AMMs

- There is a more general class of Constant-Function AMMs
- XY remains constant
- Arbitrary function: f(X,Y) remains constant
    - f(X,Y) = XY
    - f(X,Y) = X^aY^b (Balancer)
    - Some weighted average (Curve)

Analysis

- Liquidity Provider Returns in Geometric Mean Market Makers
- SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols
- The Adoption of Blockchain-based Decentralized Exchanges
- Optimal Fees for Geometric Mean Market Makers
- A Note on Privacy in Constant Function Market Makers
- Improved Price Oracles: Constant Function Market Makers
- An analysis of Uniswap markets
- When does the tail wag the dog? Curvature and market making
- Learning from DeFi: Would Automated Market Makers Improve Equity Trading?
- Automated market making and Loss-versus-rebalancing



Why AMMs?

Problems with limit order books

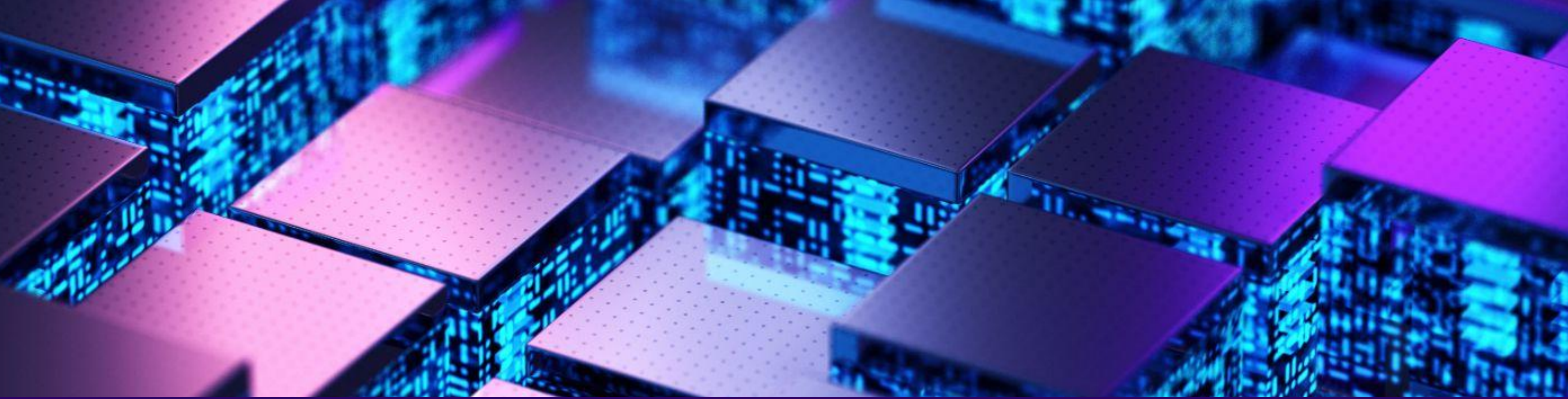
- Most *centralized* platforms use Central Limit Order Books
    - Requires placing / canceling orders
        - You can write smart contract limit orders
            - You must pay to place / cancel an order
            - Hard to prioritize order cancellation over fulfilment
    - Requires matching orders
        - Ethereum is too slow / expensive to run matching logic
        - Solana is fast enough -- (Serum)
    - Centralized exchanges are not composable -- you cannot trade on a centralized exchange and use the outputs of that trade *in a single transaction*

Benefits of AMMs

- Users can provide liquidity (and earn trading fees) with minimal market knowledge
- Trades are composable
  - multi-hop trades can be part of the same transaction
  - Flash loans
  - Riskless arbitrage
- Efficient price discovery
- Easier to reset after liveness failures

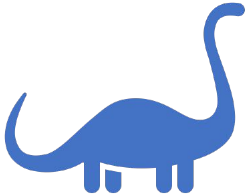
Limitations of AMMs

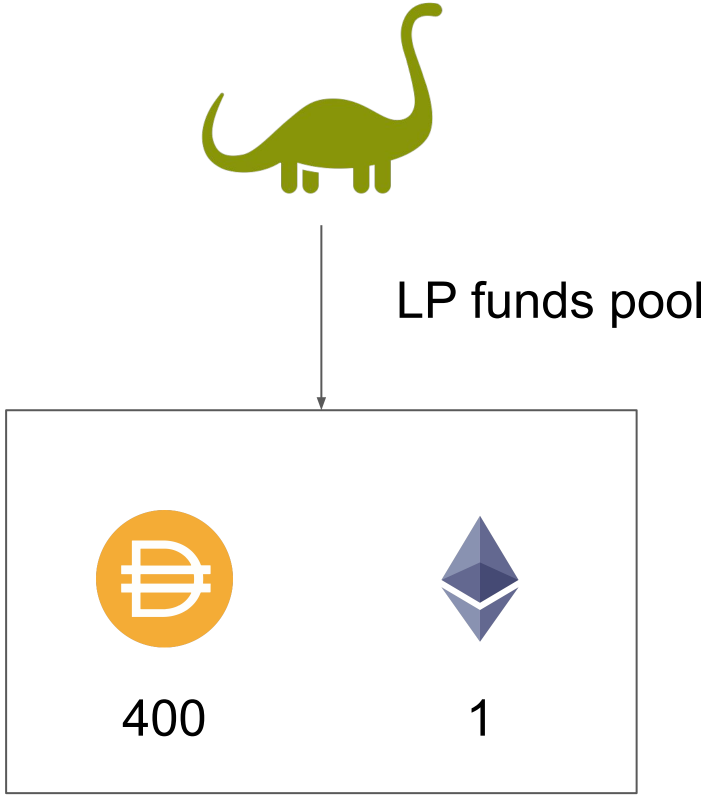
- Impermanent loss
  - Loss-versus rebalancing
- Hacks
- Slippage
- Front-running, back-running and Sandwich attacks



Impermanent Loss

Impermanent Loss

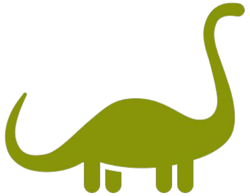


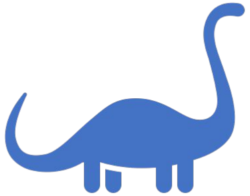


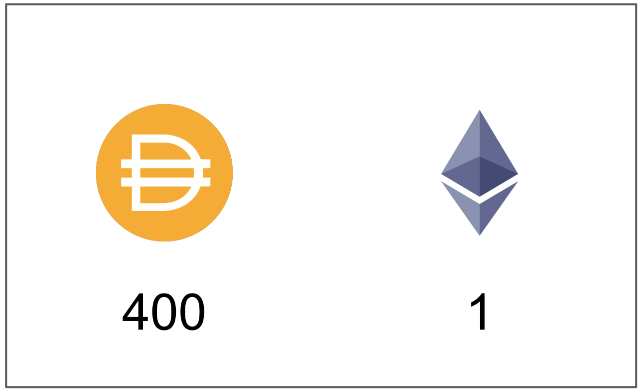
External price of ETH is $400

Initial deposit $800

Impermanent Loss

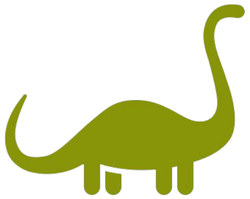


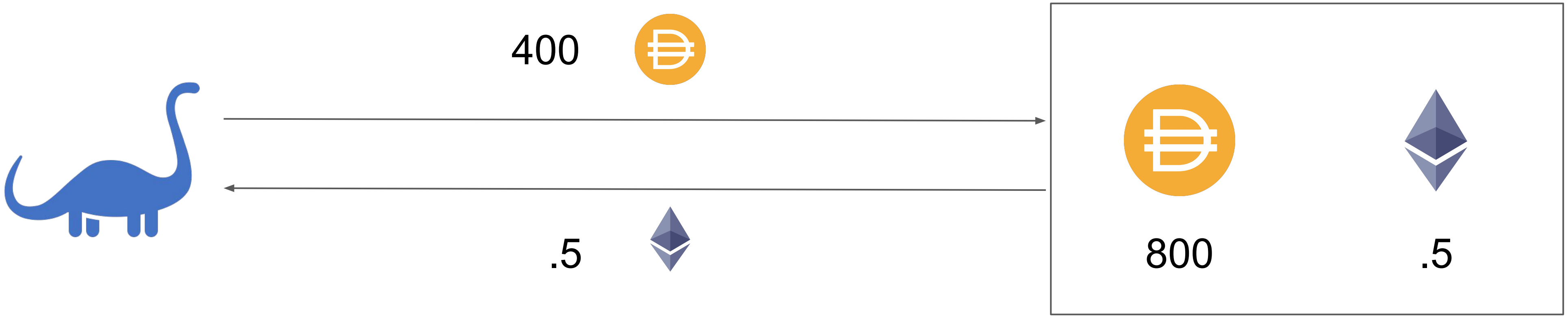




External price of ETH is $1600

Impermanent Loss

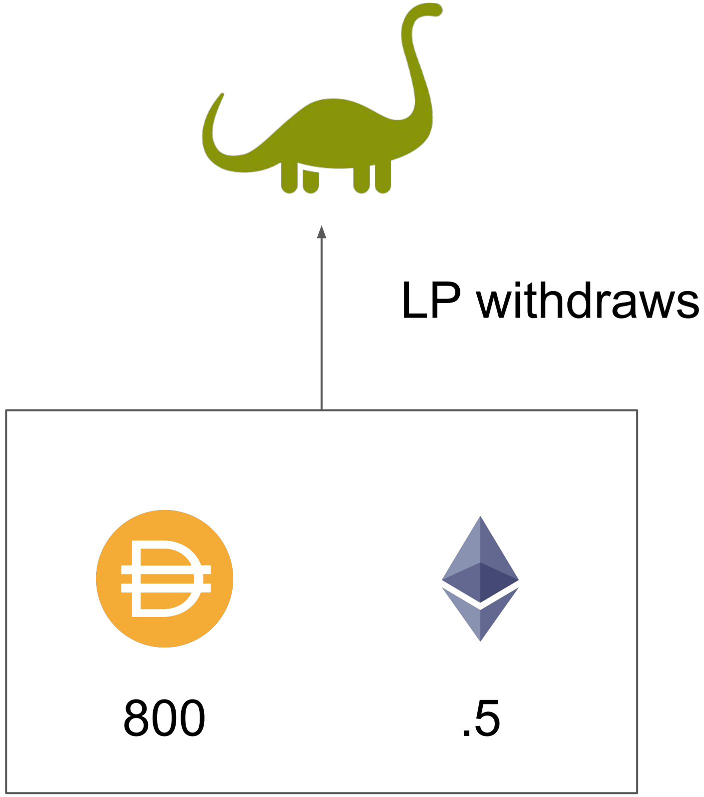


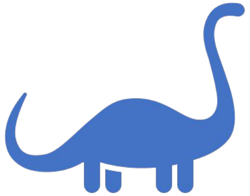


Pool invariant is maintained:
$400 \cdot 1 = 800 \cdot .5$

External price of ETH is \$1600

Impermanent Loss

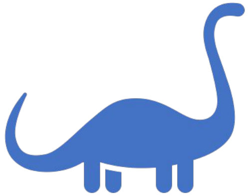




LP obtains $800 + .5*$1600 = $1600
(plus fees)

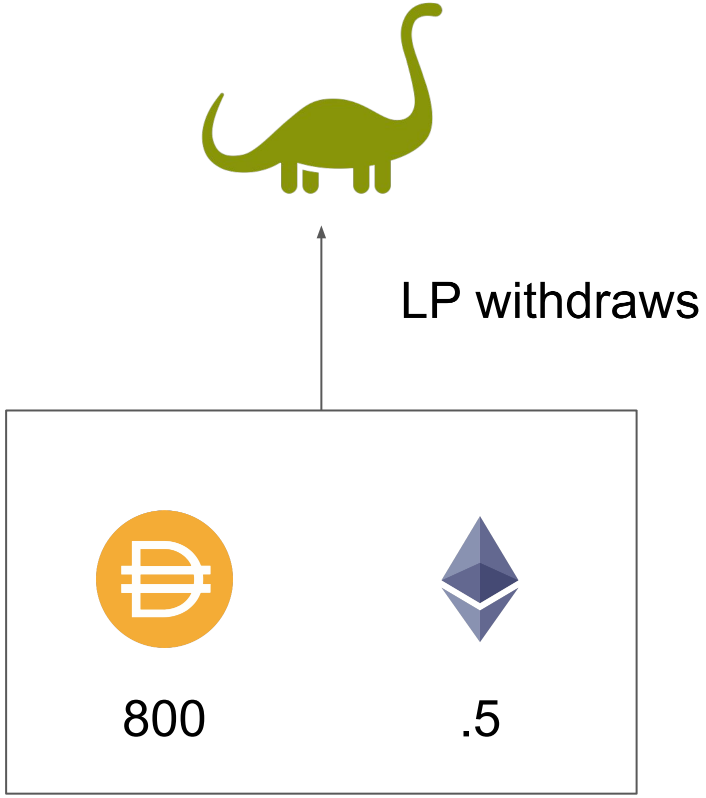
External price of ETH is $1600

Impermanent Loss



LP obtains \$800 + .5*\$1600 = \$1600
(plus fees)

LP *would have* had 400 + 1*1600 = \$2000



External price of ETH is \$1600

Impermanent loss

- Initial price ratio is p_o
- User deposits
  - x_o ETH
  - $x_o\, p_o$ USDC
  - Pool value is $2x_o p_o$
- Price changes to p_1
  - Traders trade until exchange rate is p_1

$$\left.\begin{array}{r} \frac{y}{x} = p_1 \\ xy = x_0^2 p_0 \end{array}\right\} \Rightarrow y^2 = x_0^2 p_0 p_1 \Rightarrow \left\{\begin{array}{l} y = x_0\sqrt{p_0 p_1} \\ x = x_0\sqrt{\frac{p_0}{p_1}} \end{array}\right.$$

  - Pool value is $2x_0\sqrt{p_0 p_1}$

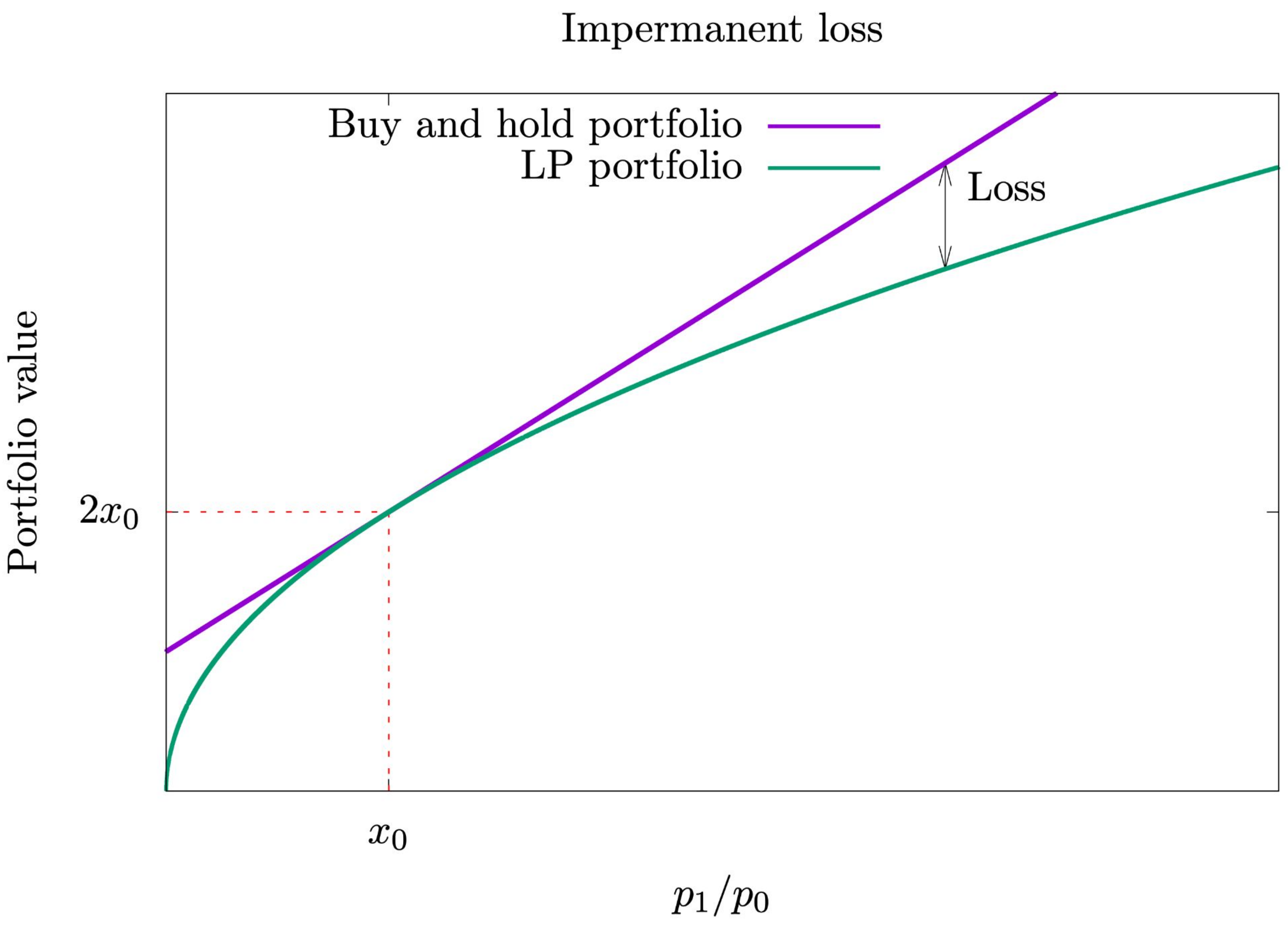
Price of ETH is p_0 USDC

Impermanent loss

- Initial value
  - x_o ETH at $x_o p_o$
  - $x_o p_o$ USDC
  - Total initial value $2x_o p_o$
- Final value (ETH valued at p_1)
  - $x_0 \sqrt{\frac{p_0}{p_1}}$ ETH
  - $x_0 \sqrt{p_0 p_1}$ USDC
  - Total final value $2x_0 \sqrt{p_0 p_1}$

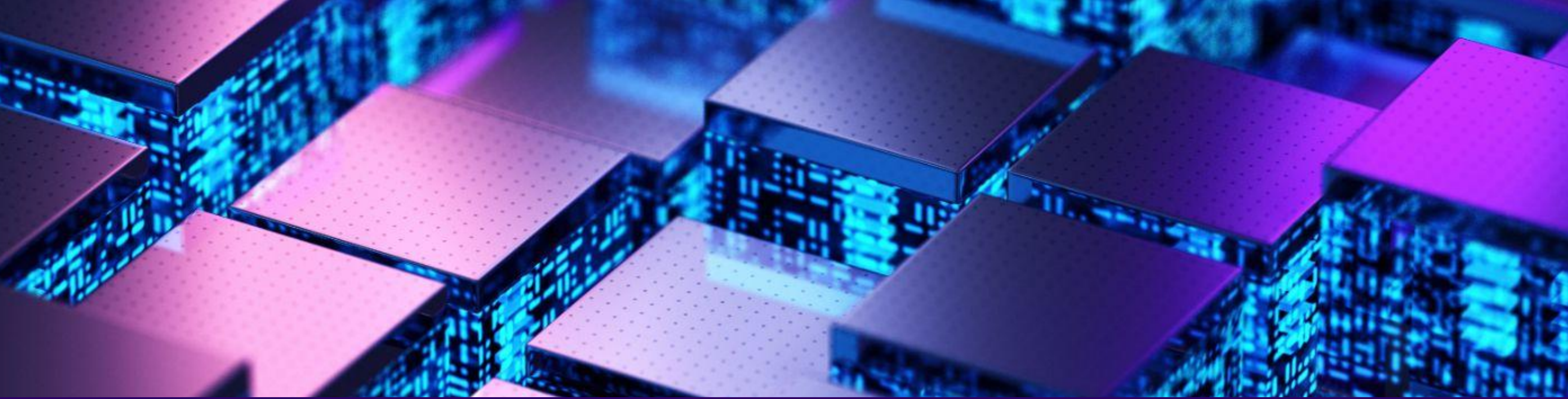
- Final value (of buy and hold strategy)
  - x_o ETH at $x_o p_1$
  - $x_o p_o$ USDC
  - Total initial value $x_o(p_o + p_1)$

Impermanent loss



Impermanent Loss and LVR

- IL considers pool balance against “buy and hold” portfolio
- A better metric would be to compare pool balance against “rebalancing” portfolio
    - Investor constantly rebalances assets so the balance is 50/50 at current market prices
- Called Loss-Versus Rebalancing (LVR)
- Key Questions:
    - **Can trading fees offset these losses?**

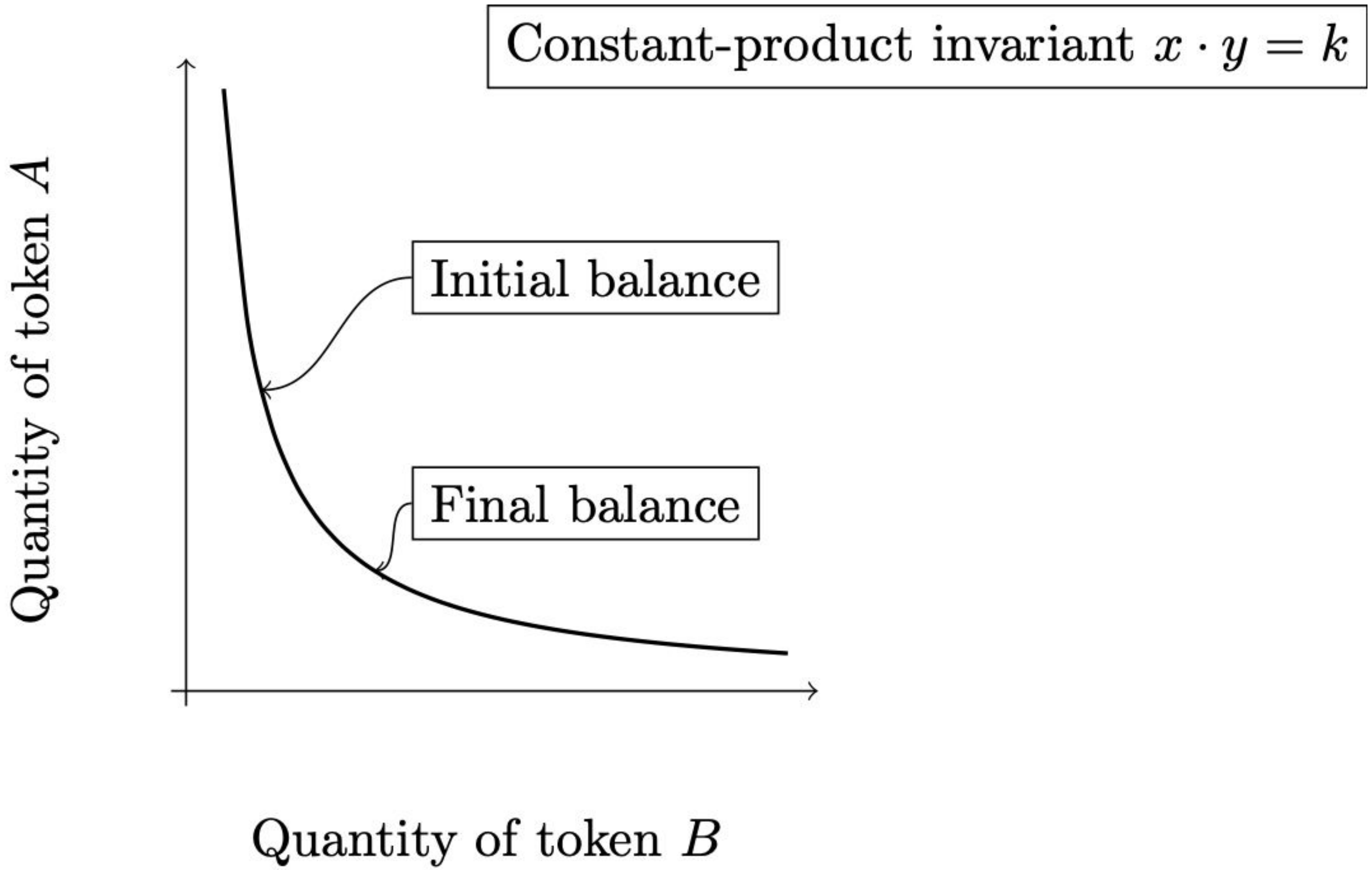


Slippage on Constant-Function AMMs

Constant-Function Market Makers

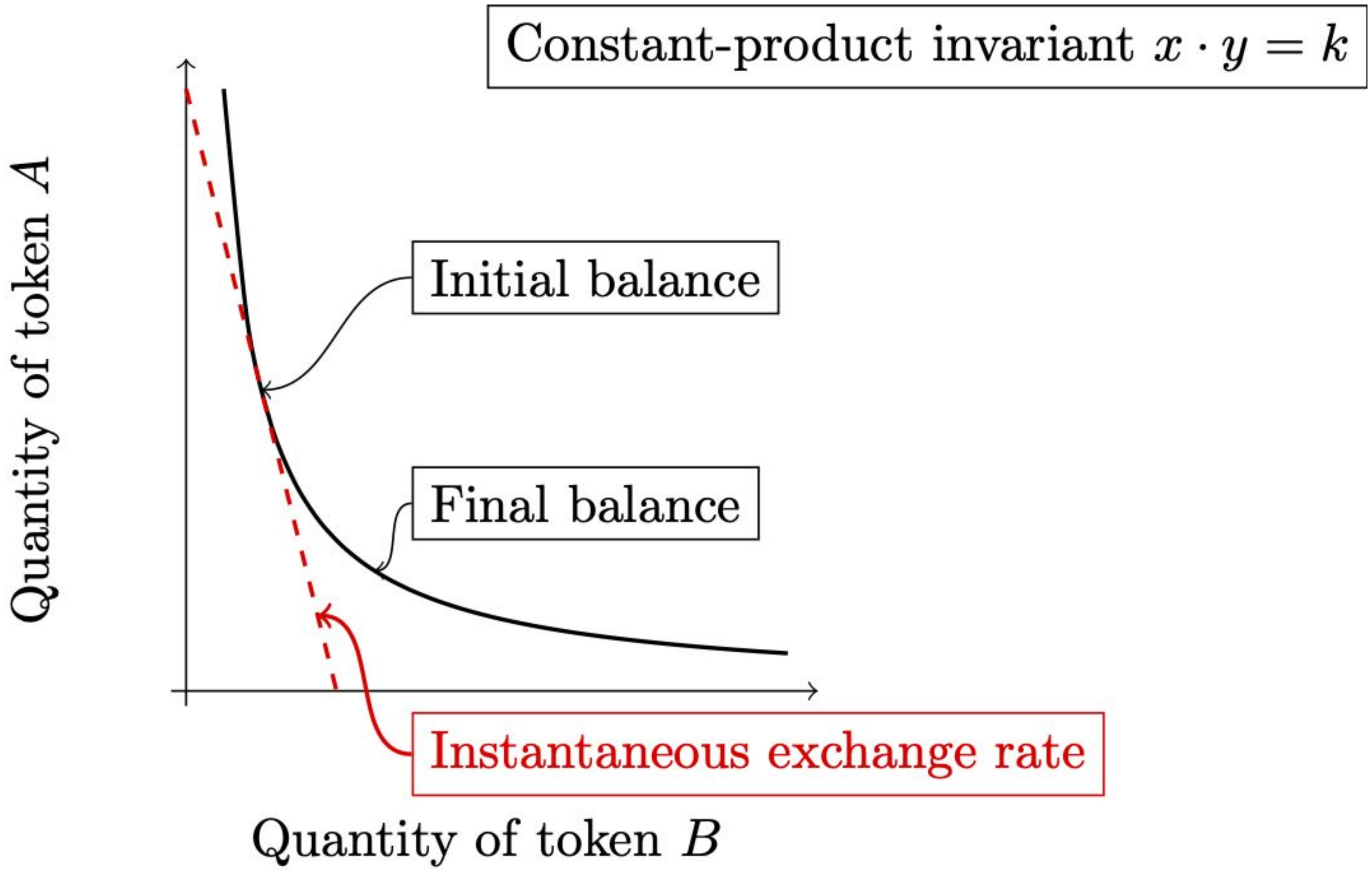
- Contract holds
  - X units of token A
  - Y units of token B
- Users can swap A for B and vice versa
- Maintain invariant $XY = K$
- Contract can always execute trade
  - (Exchange-rate may be bad)

Slippage



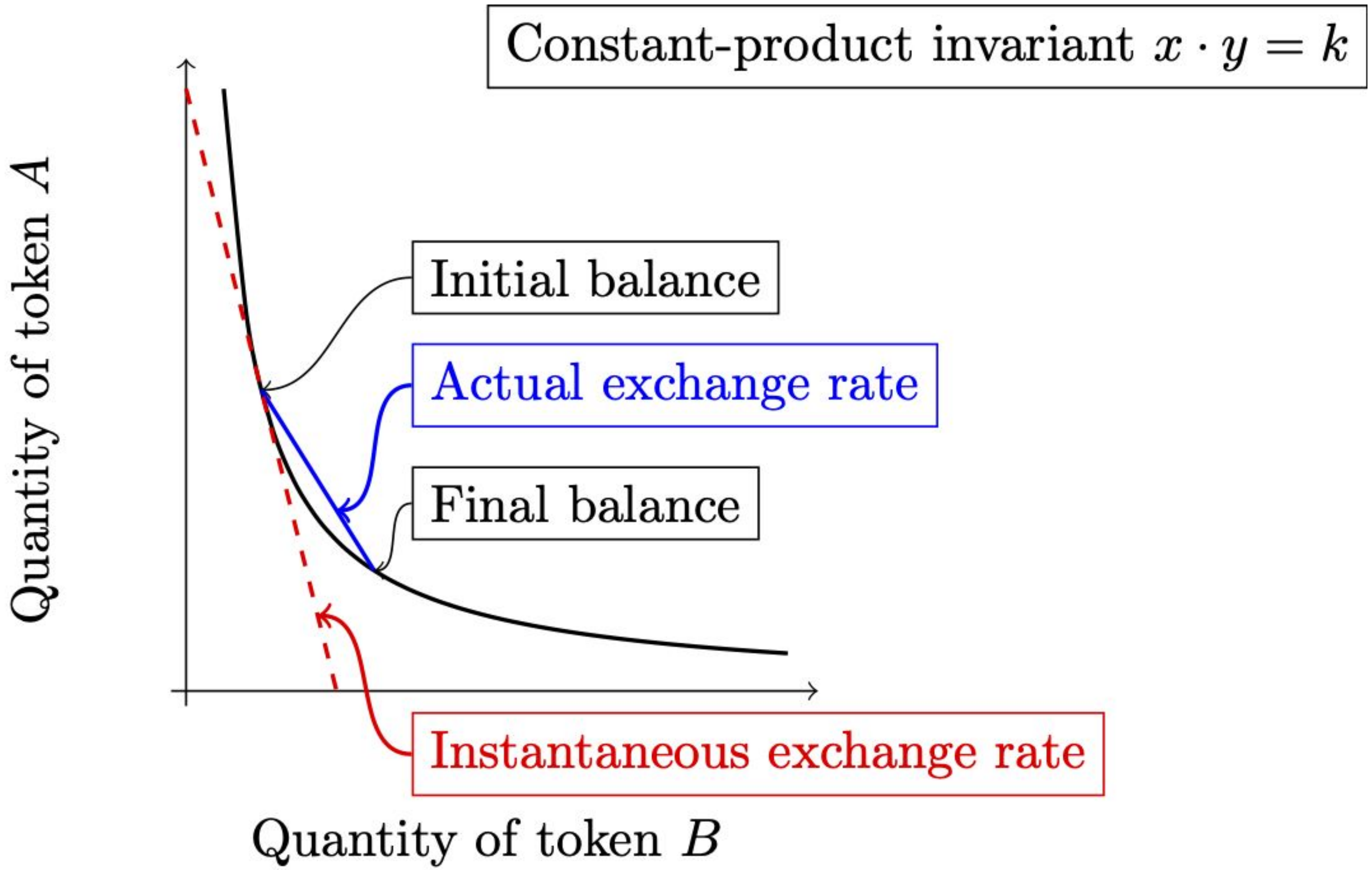
Constant-product invariant $x \cdot y = k$
Quantity of token A
Initial balance
Final balance
Quantity of token B

Slippage



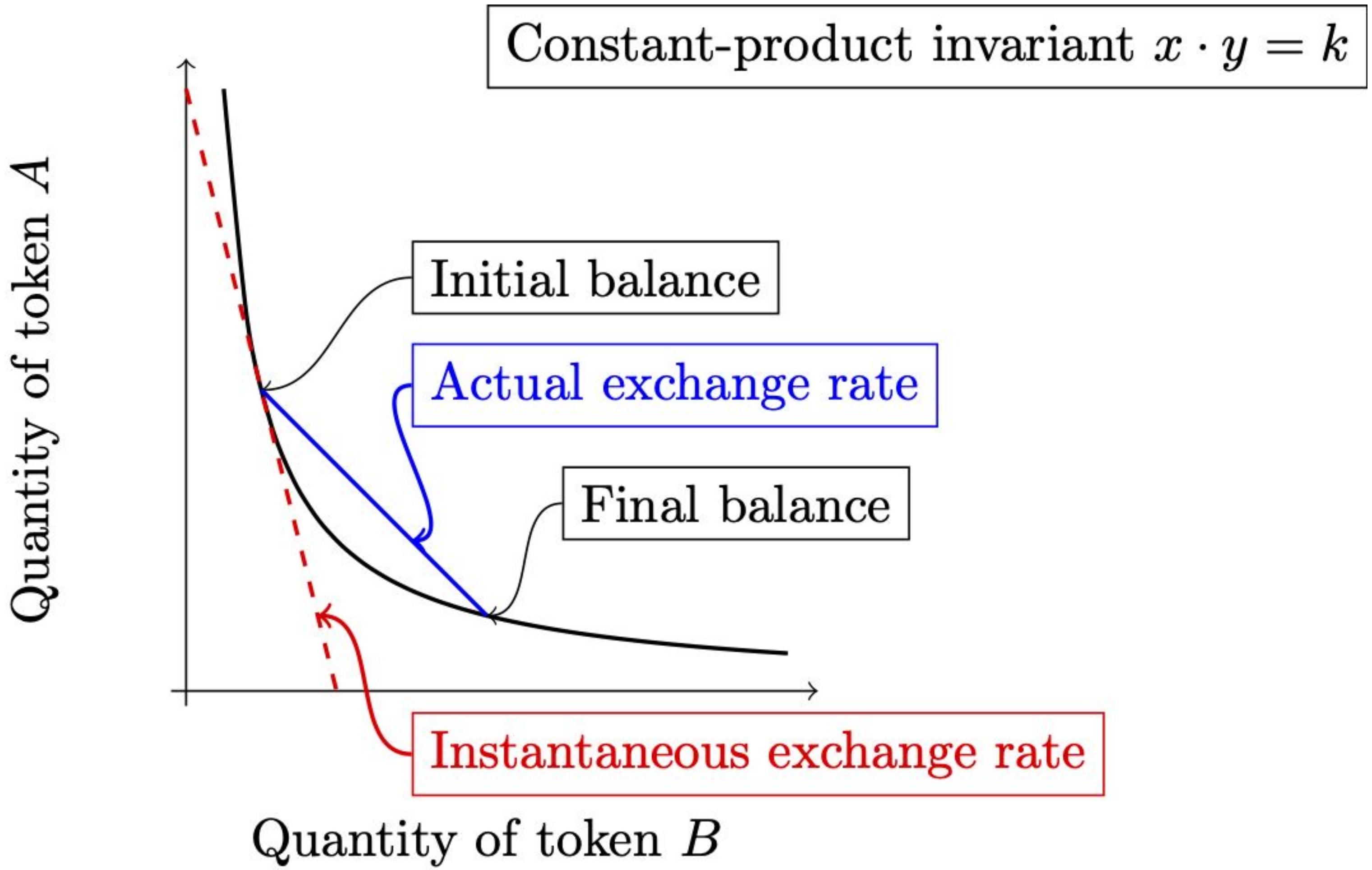
Constant-product invariant $x \cdot y = k$
Quantity of token A
Initial balance
Final balance
Instantaneous exchange rate
Quantity of token B

Slippage



Constant-product invariant $x \cdot y = k$
Quantity of token A
Initial balance
Actual exchange rate
Final balance
Instantaneous exchange rate
Quantity of token B

Slippage



Constant-product invariant $x \cdot y = k$
Quantity of token A
Initial balance
Actual exchange rate
Final balance
Instantaneous exchange rate
Quantity of token B

Slippage

- Low slippage when
  - Liquidity is large
  - Trades are small
- Lots of slippage for large trades