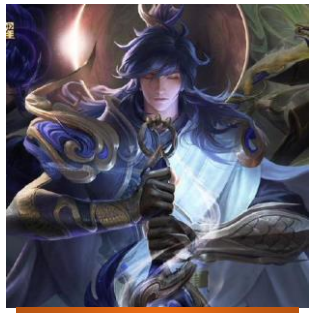
A Prover Network with Pricing by History

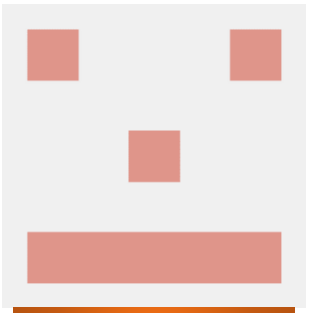
Bike Labs

Bike Labs Team

Rayer



Robert



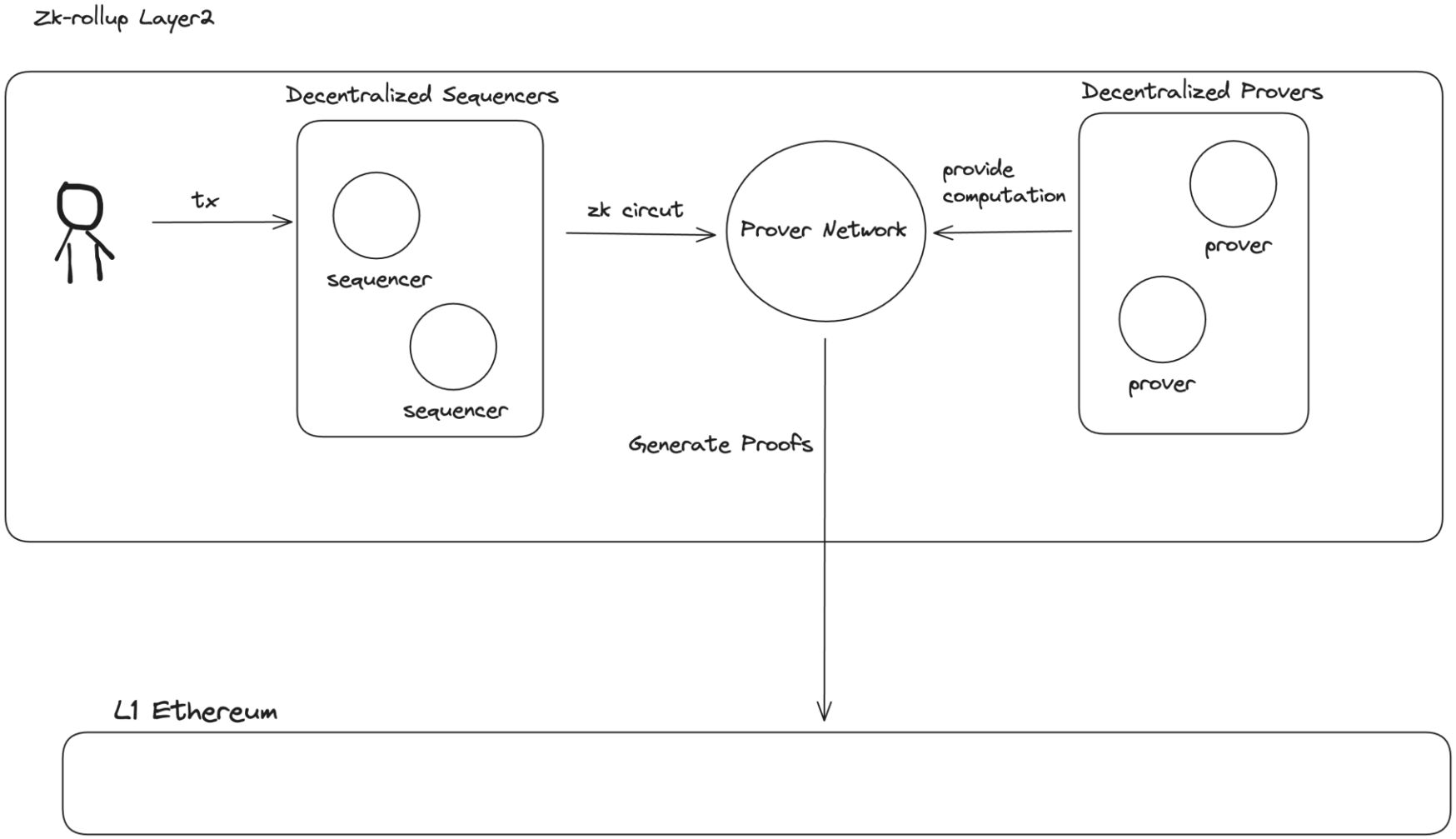
LJ Dragon

Ruo Yan

Stevending1st

Hakeen

Research Premise and Background



Issues with Existing Solutions

User Payment Methods

- First-price auction
- Order book
- Estimated fees based on Layer 1 publication

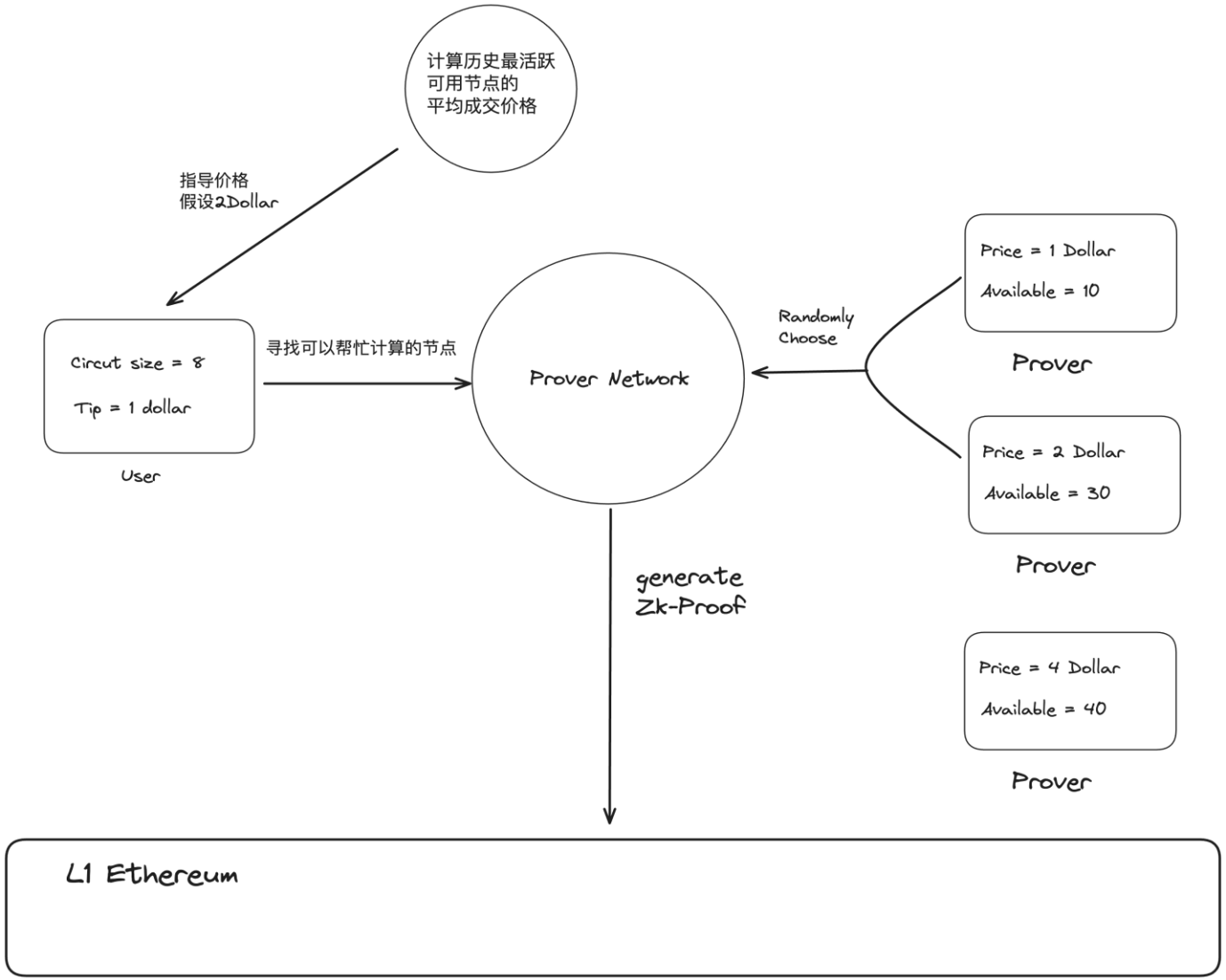
User Payment Alternatives

- ·Allocation by Sequencer
- Order book
- Proof-of-Stake (PoS) and weighted random selection
- Verifiable Random Function (VRF) for random selection

Design Philosophy

- User-Centricity
- Maximizing Prover Market Activity
- Minimizing User Entry Barriers

Overall Architecture

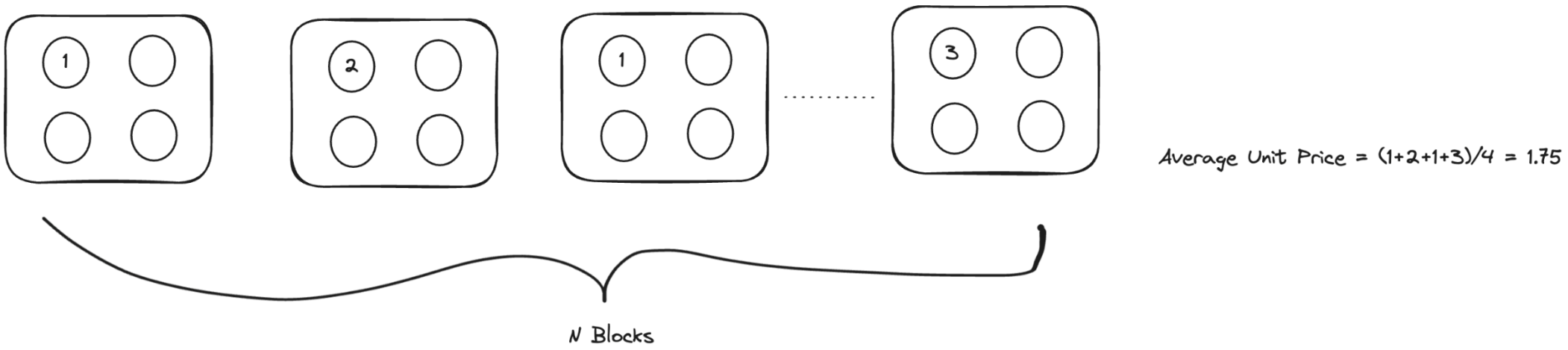


User Pricing Factors - User Factors

- **Circuit Size:** The more complex your circuit, the more you should pay.
- **Tips:** The more urgent your order, the more you should pay.

User Pricing Factors - Prover Factors

Average Unit Price: Based on the historical average transaction price of the **most active** m **available nodes** in the last n blocks.



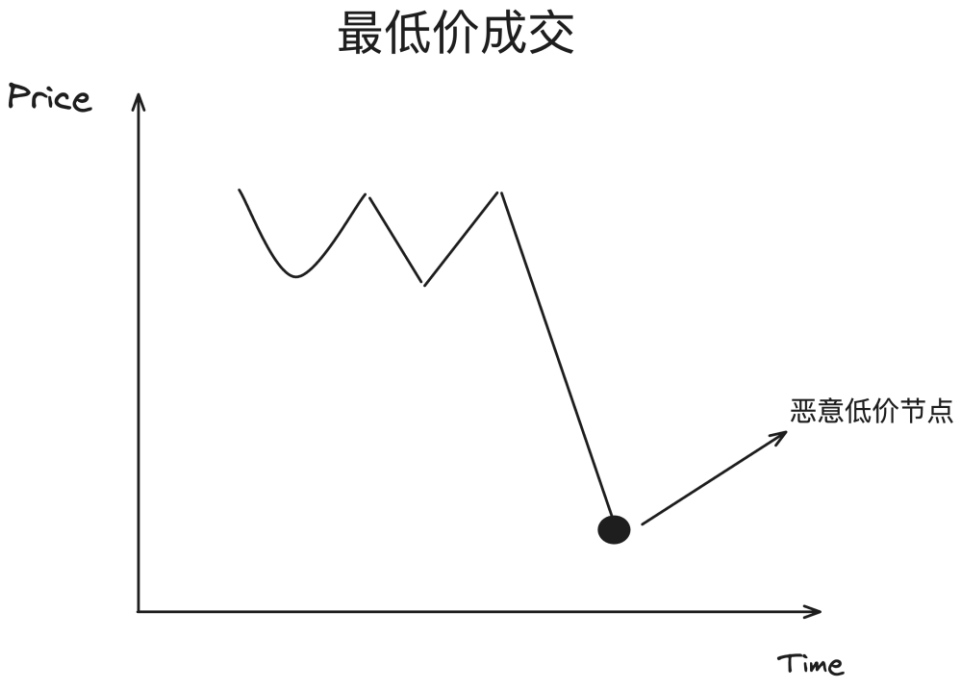
User Pricing Formula

- **Circuit Size:** The more complex your circuit, the more you should pay.
- **Tips:** The more urgent your order, the more you should pay.
- **Average Unit Price:** Based on the historical average transaction price of the most active m available nodes in the last n blocks.

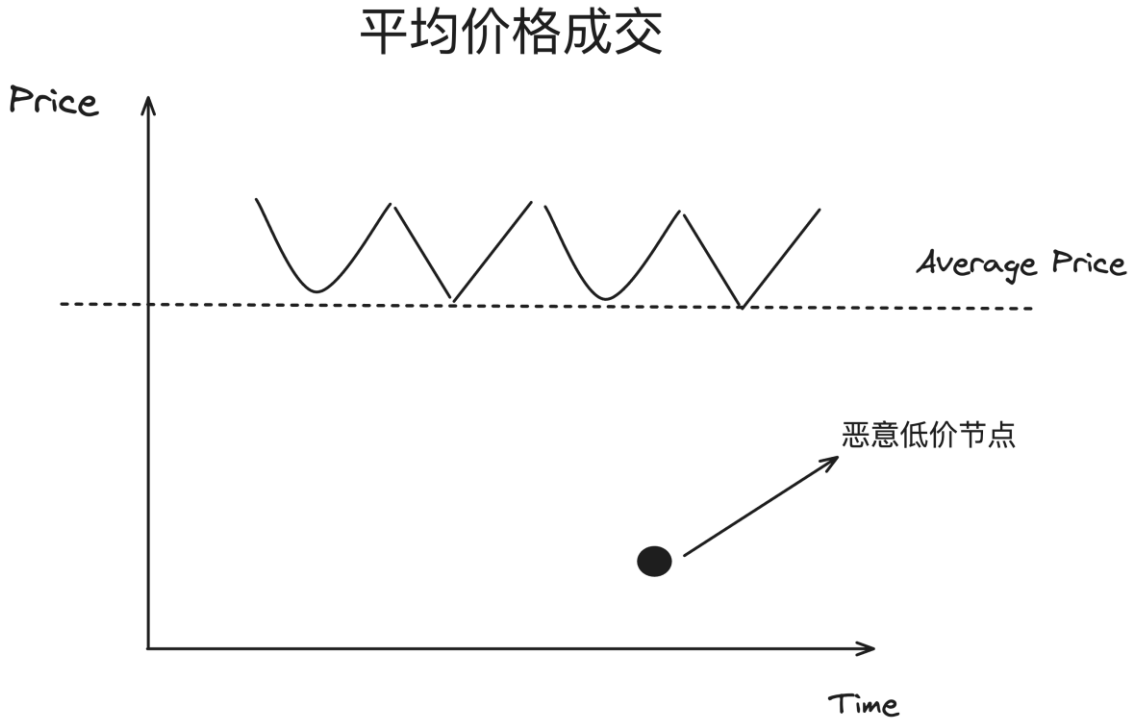
$$\text{TotalPrice} = (\text{averageUnitPrice} + \text{tips}) \times \text{circutSize}$$

Impact of Pricing Mechanism

- **Smooths Price Fluctuations**

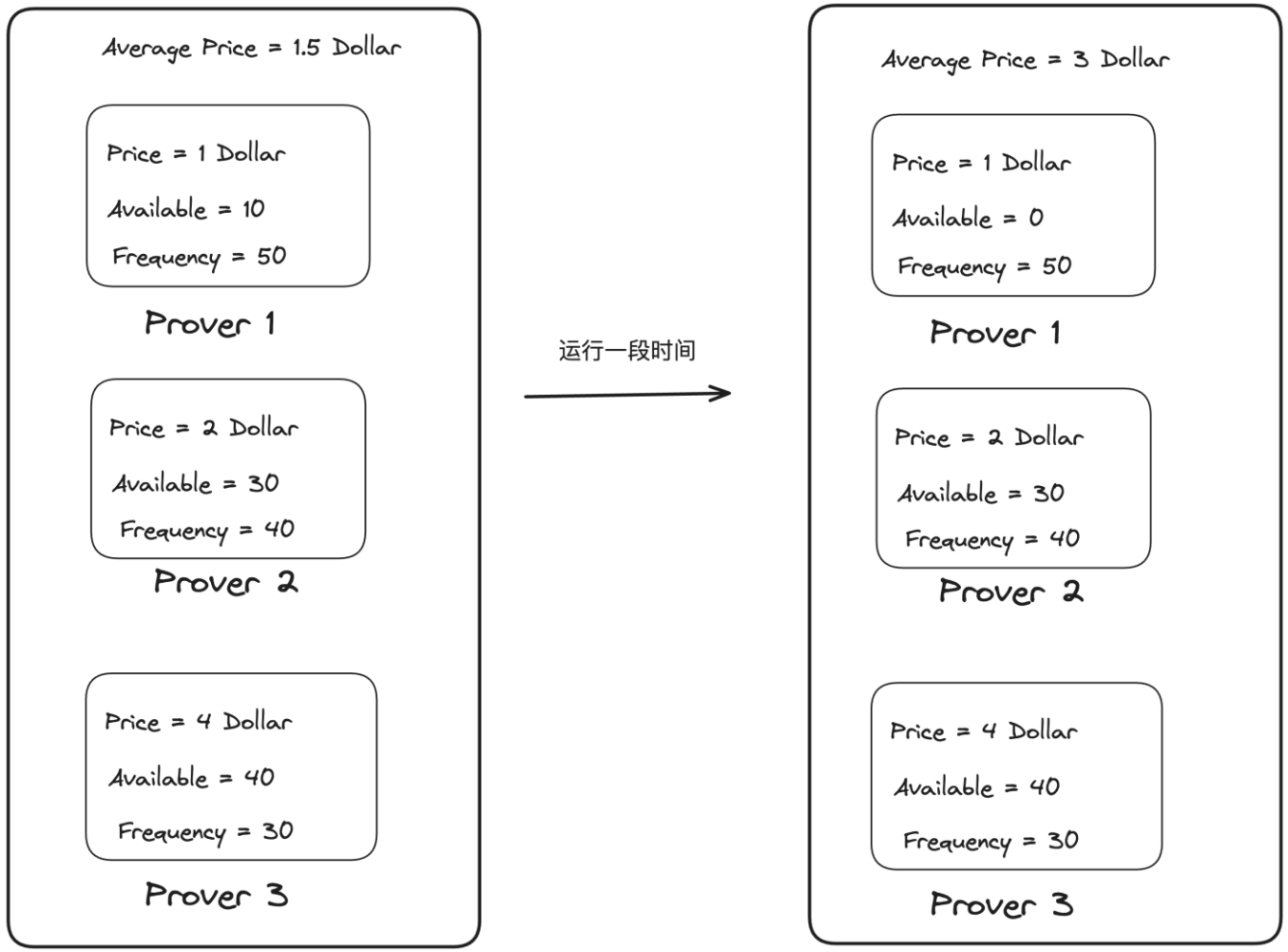


- **Average Unit Price:** Based on **the historical average transaction price** of the **most active** m available nodes in the last n blocks.



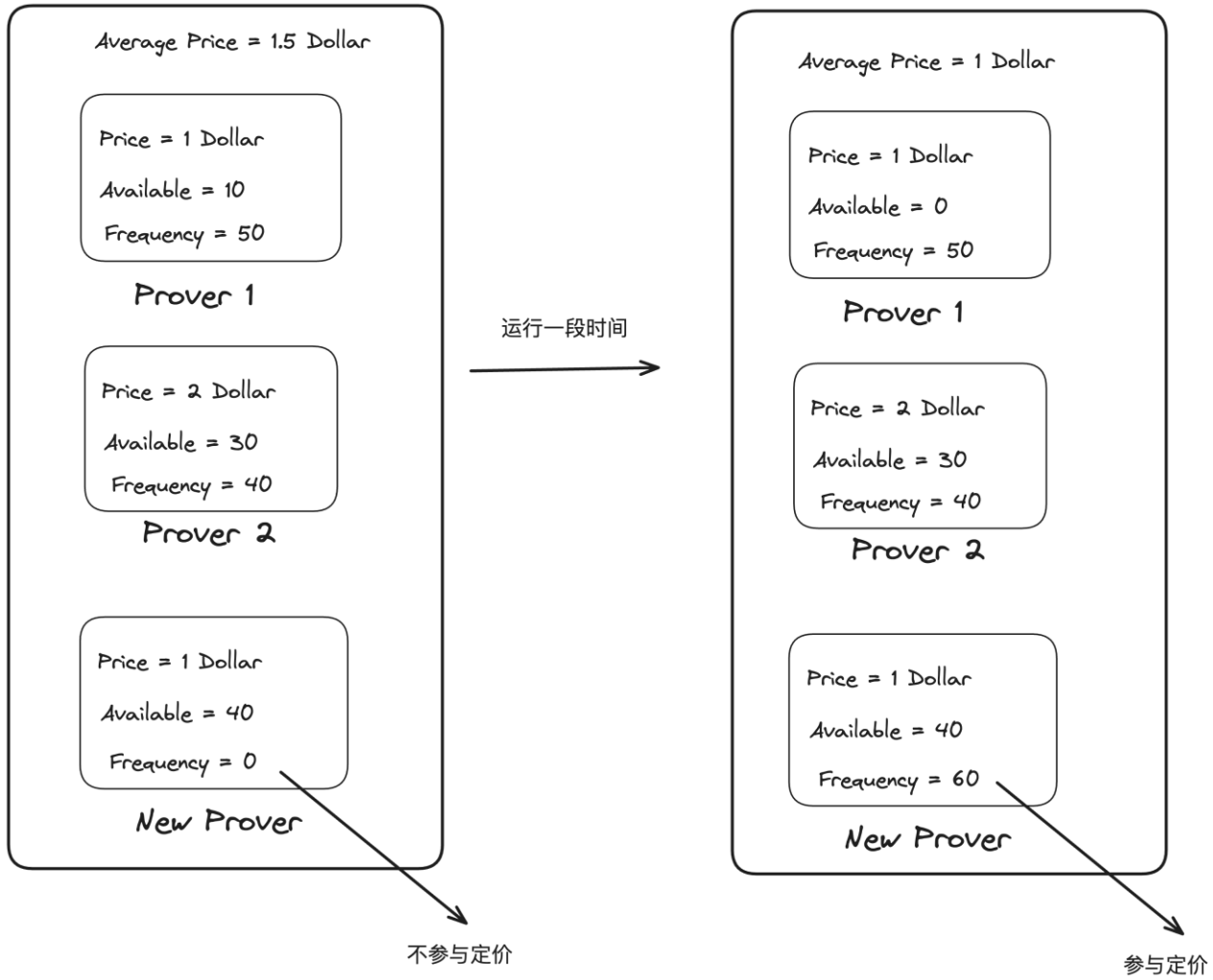
Matching Mechanism

- **Ensuring the liquidity of the computing power provided by Provers.**
- **Average Unit Price:** Based on the historical average transaction price of the most active m available nodes in the last n blocks.



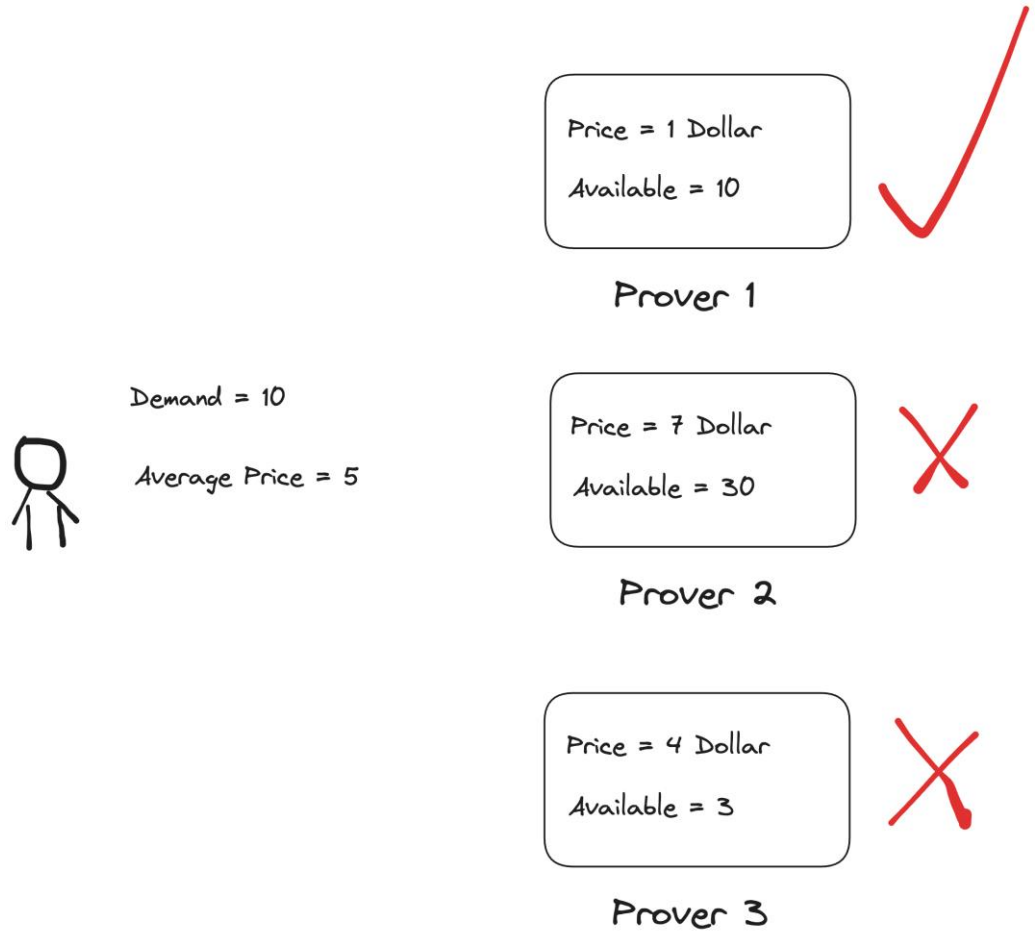
Impact of Pricing Mechanism

- **Facilitates New Provers Joining**
- **Average Unit Price:** Based on the historical average transaction price of the most active m available nodes in the last n blocks.



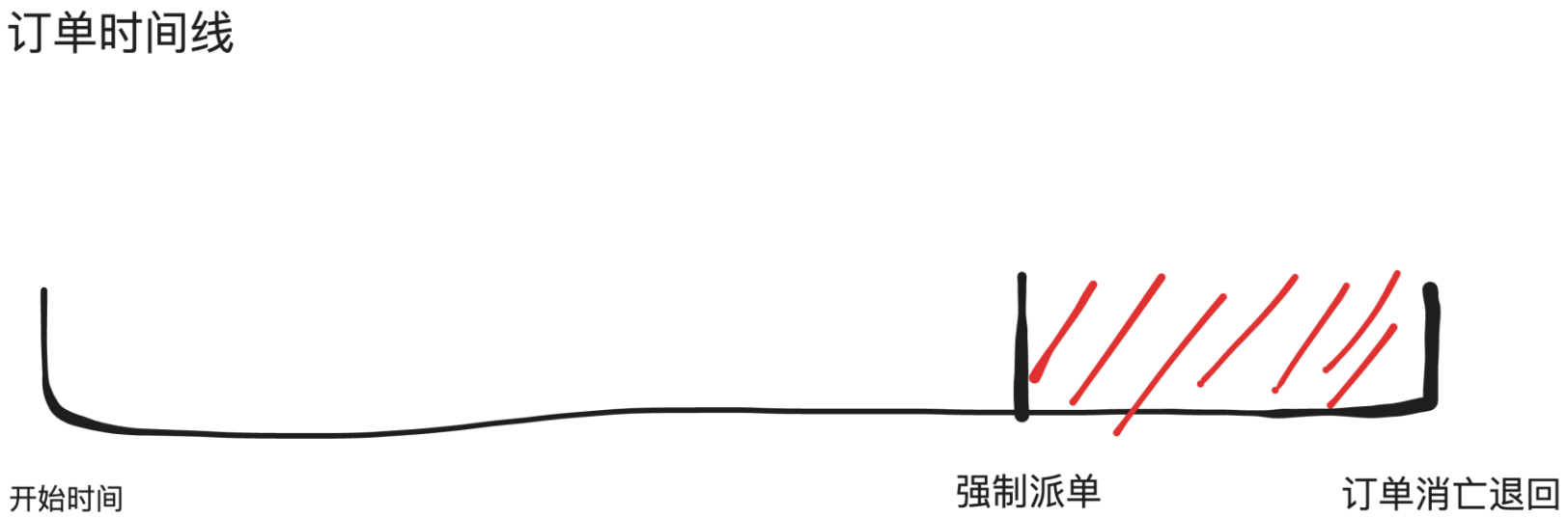
Matching Mechanism

Any node meeting price and computing power requirements is randomly matched.



Low-Price Orders & Security Deposit Mechanism

- Firstly, before a Prover can go online, they must place a security deposit to ensure compliance with rules.
- If a low-priced order exists, it remains unexecuted because others have been adding to the price.
- At a certain point, forced assignment triggers, requiring the Prover to execute the order; otherwise, part of the security deposit is deducted as compensation to the user.



Security Deposit & Penalty Mechanism

- The amount of the security deposit a Prover places must be proportional to their computing power:
  - Because the more computing power you possess, the greater your impact on the network and potential earnings, you should place a higher security deposit.Because the more computing power you possess, the greater your impact on the network and potential earnings, you should place a higher security deposit.。

$$\text{stakedPrice} = k \cdot \text{Capability}$$

- Provers who fail to complete tasks on time lose a percentage of their security deposit as compensation to the user:
  - Why is it a percentage of the security deposit rather than compensating the user's bid?
  - Because a user's bid may be insignificant compared to a large Prover's earnings. We design the system to increase the cost of malicious behavior for Provers.

Conclusion

- We designed a Prover network achieving the following goals:
  - **User-Friendly:** Users only need to pay the market average price for services, and no additional knowledge is required.
  - **Prover-Incentive Friendly:** Your Prover will receive orders with values greater than your costs, ensuring profitability.
  - **Maximized Transaction Volume:** By being friendly to both parties and eliminating the need for competitive bidding, transaction efficiency is greatly increased.