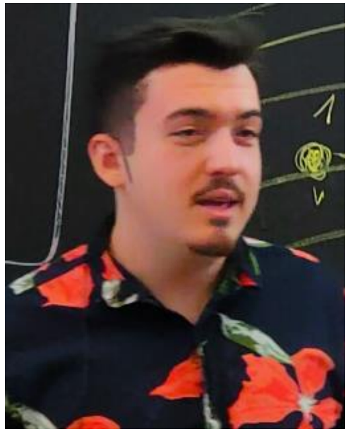
Transaction Fee Market Design for Parallel Execution

Bahar Acilan

Andrei Constantinescu

Lioba Heimbach

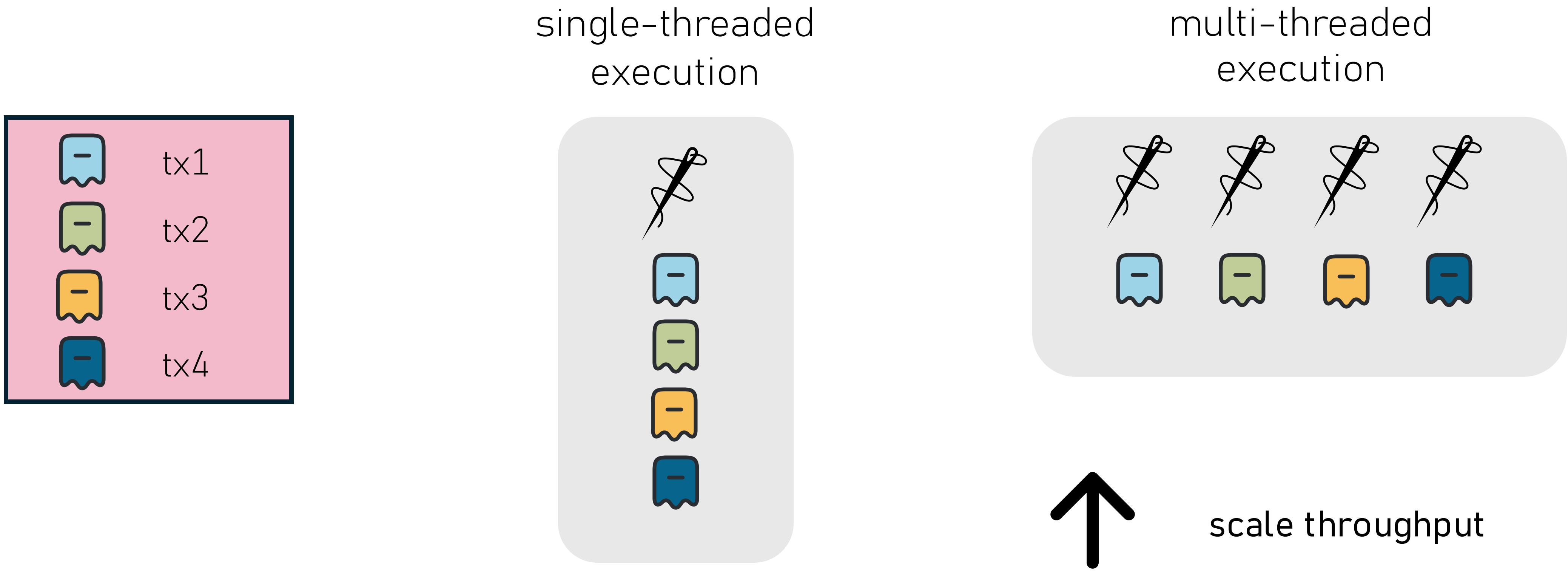
Roger Wattenhofer

ETH zürich

DFINITY

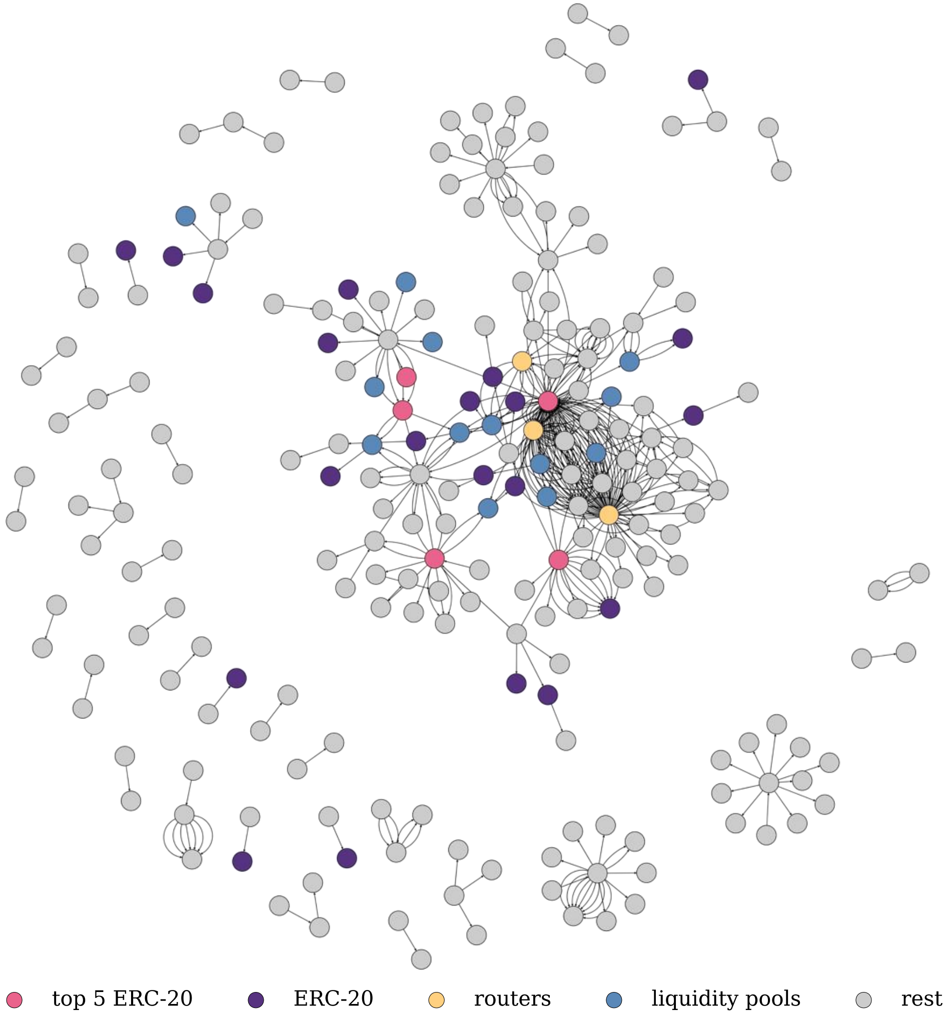
Category Labs

Blockchains are moving from single-threaded to multi-threaded execution.



What speedup is possible?

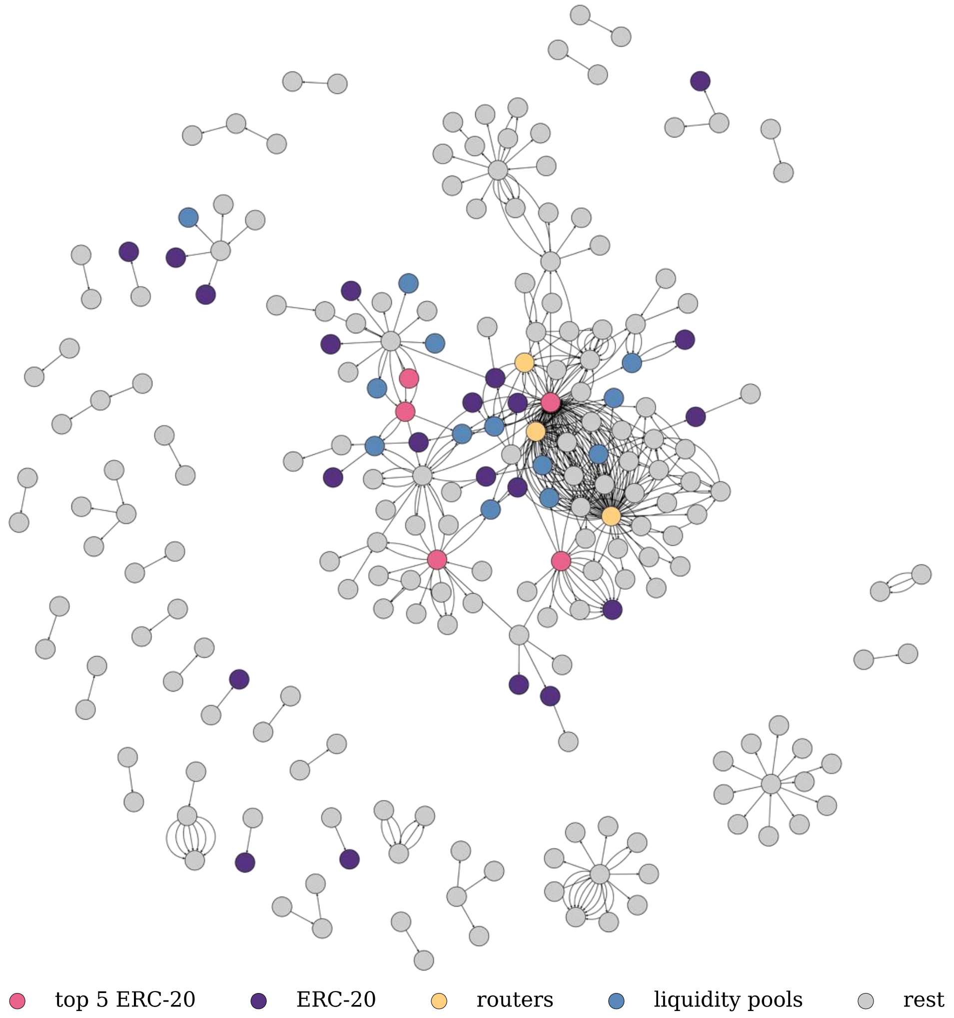
Many transactions in Ethereum interact with the same resources, but …



vertices: addresses (contracts or wallets)

edges: transaction calls

Many transactions in Ethereum interact with the same resources, but a fourfold speedup is realistic.



a fourfold speedup is realistic

vertices: addresses (contracts or wallets)

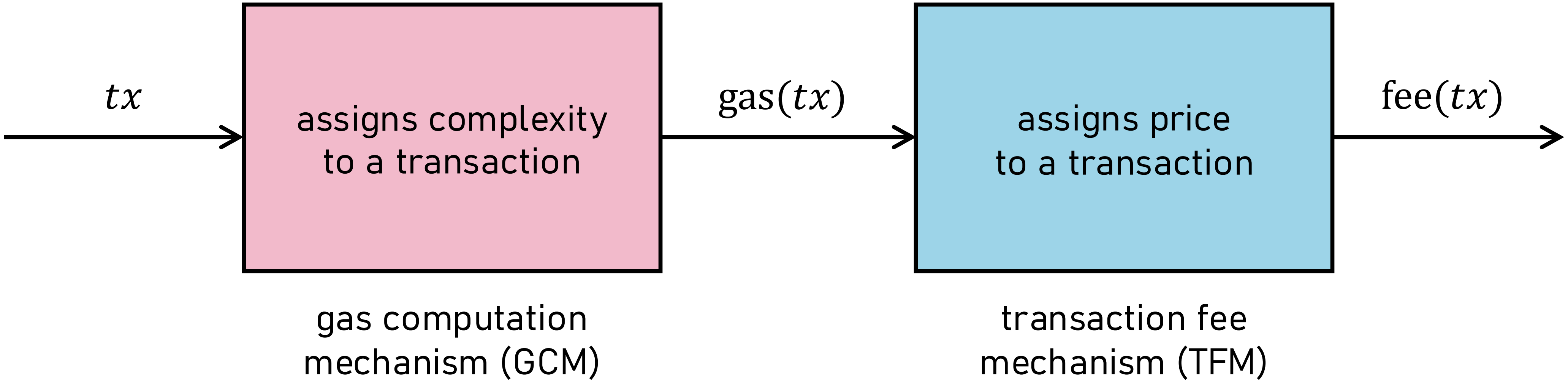
edges: transaction calls

Are there other considerations when moving to parallel execution?

↪ a suitable fee market

What is a fee market?

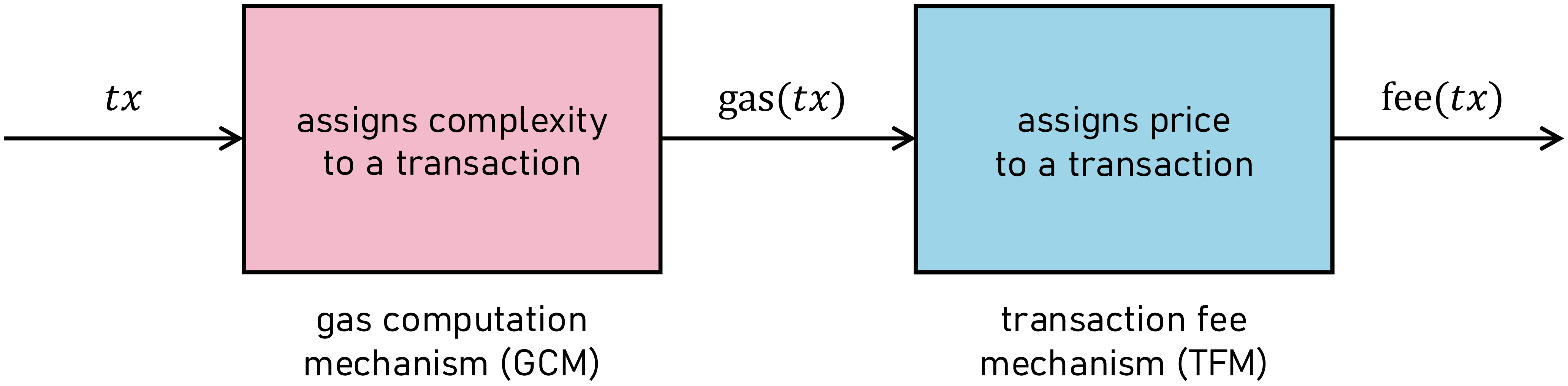
1. fees should be higher for complex transactions
2. fees should be higher when demand is high



What is the current fee market in Ethereum?

total fee = gas units · gas price

execution component of gas currently ≈ time

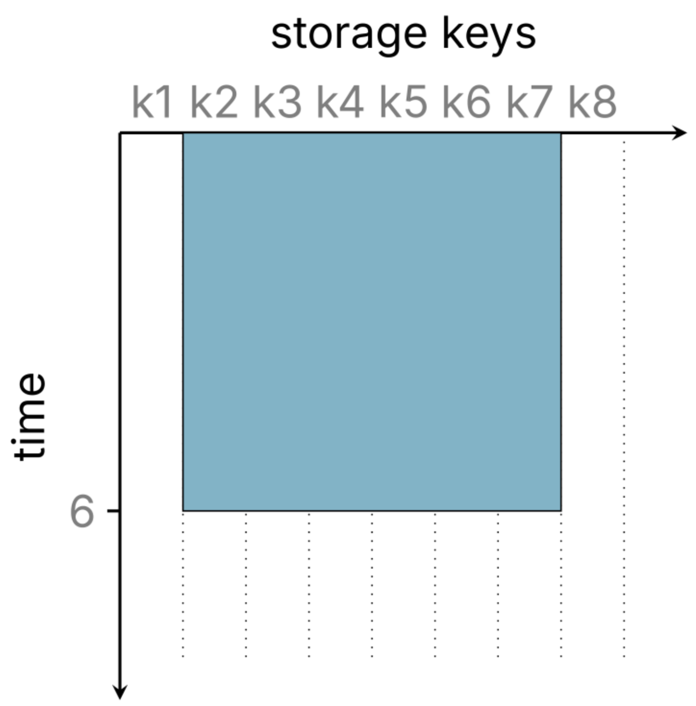


Why does the current fee market not suffice in light of parallel execution?

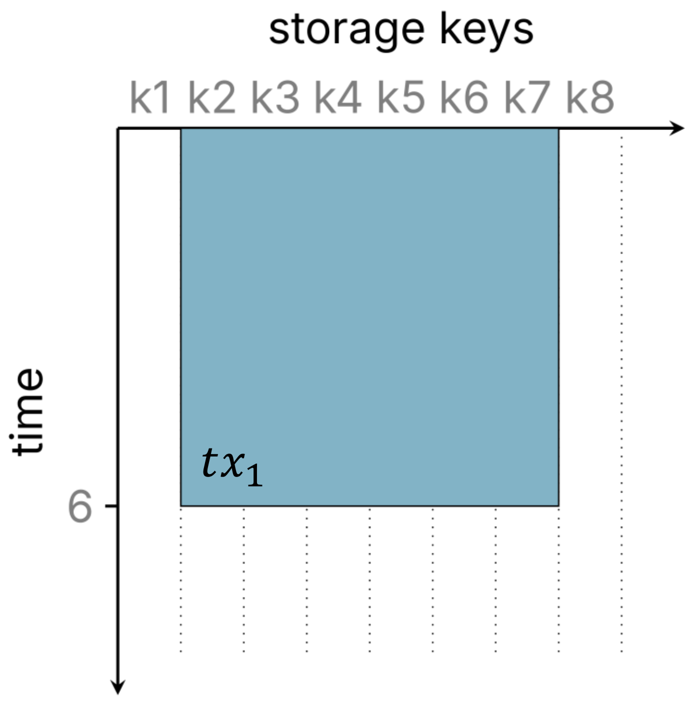
$$tx_1 = (6, \{k2, k3, k4, k5, k6, k7\})$$

time

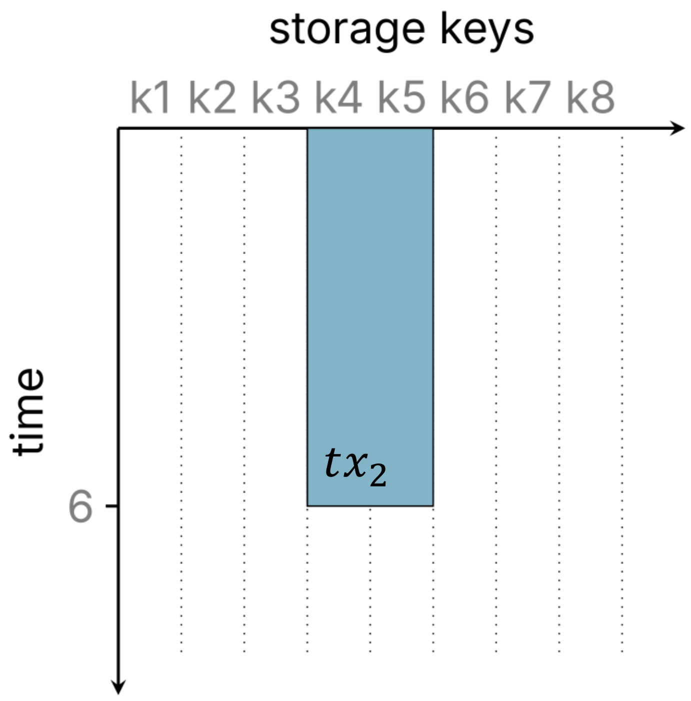
storage key set
assumed to be known



These transactions use the same amount of gas, but one uses three times the number of storage keys.



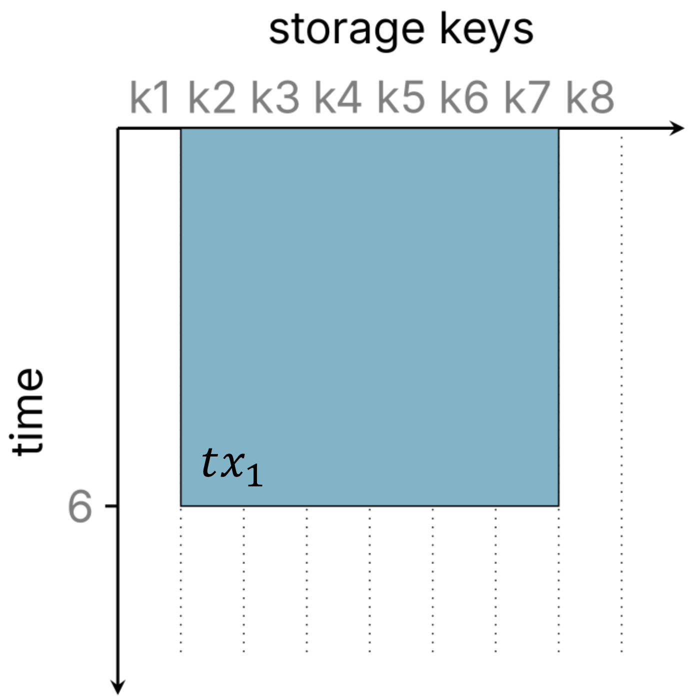
$$\text{gas}(tx_1) = \text{time}(tx_1) = 6$$

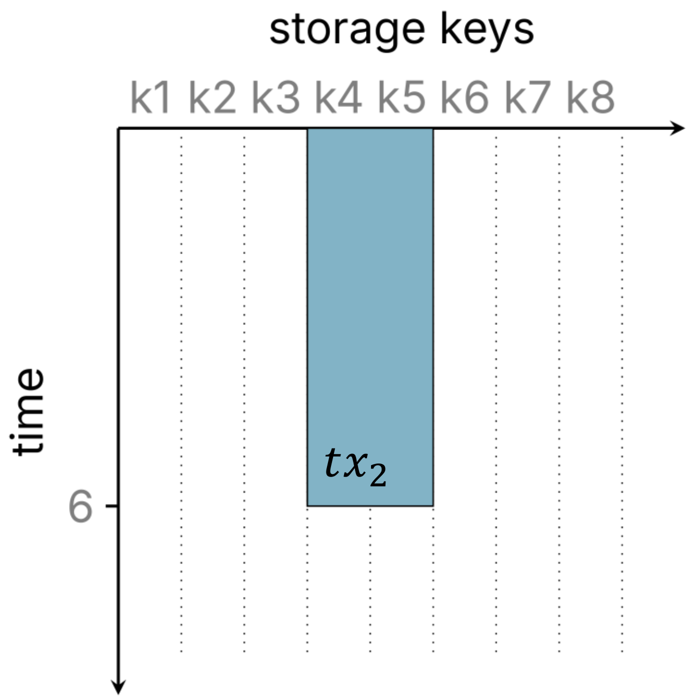


$$\text{gas}(tx_2) = \text{time}(tx_2) = 6$$

What properties do we want in a fee market that supports parallel execution?

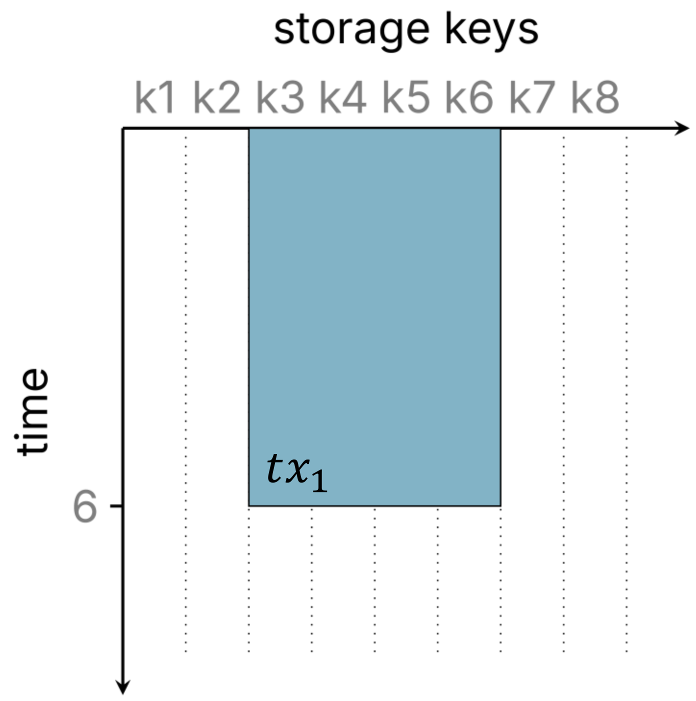
Storage key monotonicity.

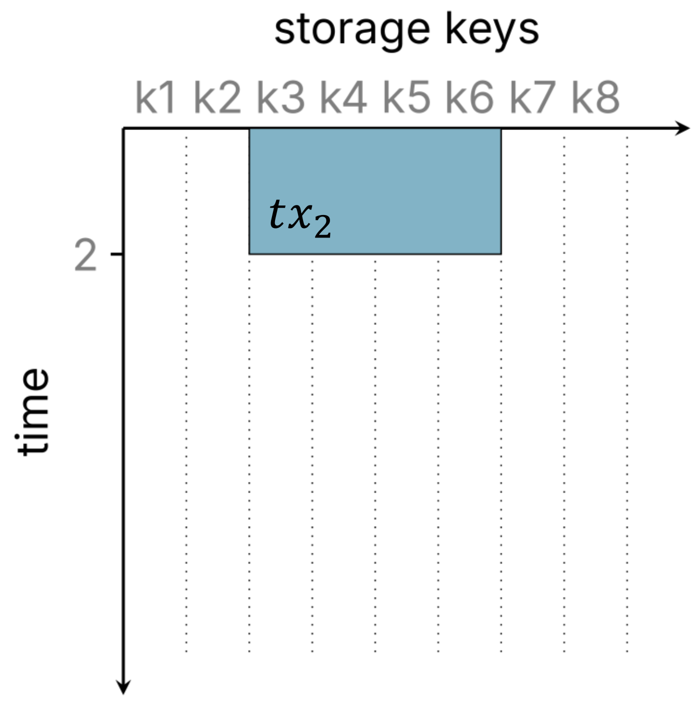




$$\text{gas}(tx_1) > \text{gas}(tx_2)$$

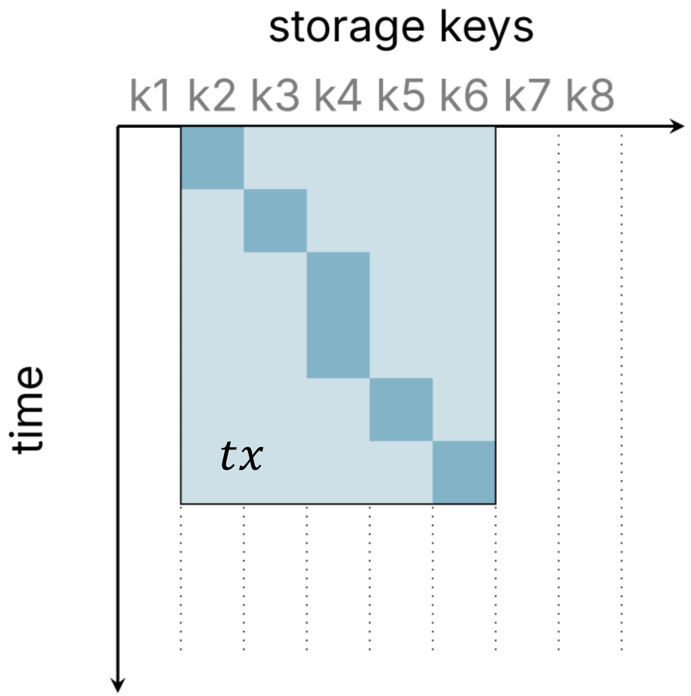
Time monotonicity.

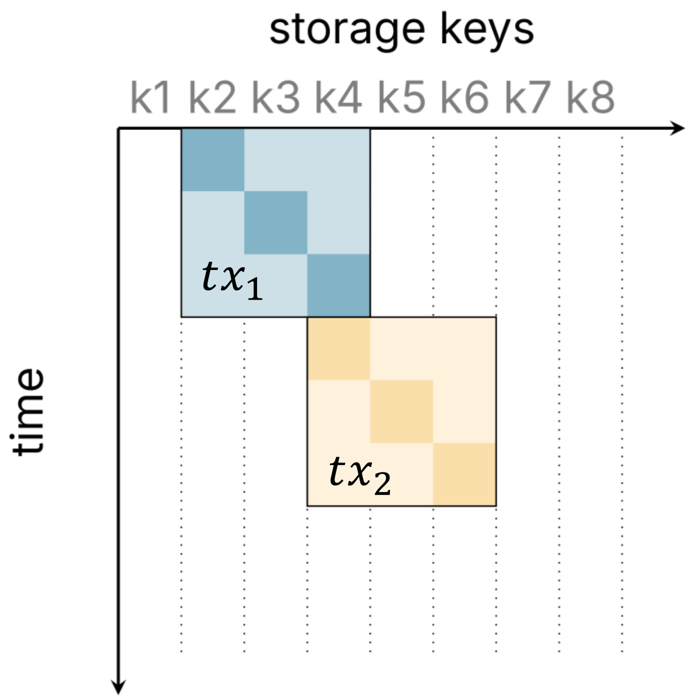




$$\text{gas}(tx_1) > \text{gas}(tx_2)$$

Transaction bundling.





$$\text{gas}(tx) > \text{gas}(tx_1) + \text{gas}(tx_2)$$

Further properties.

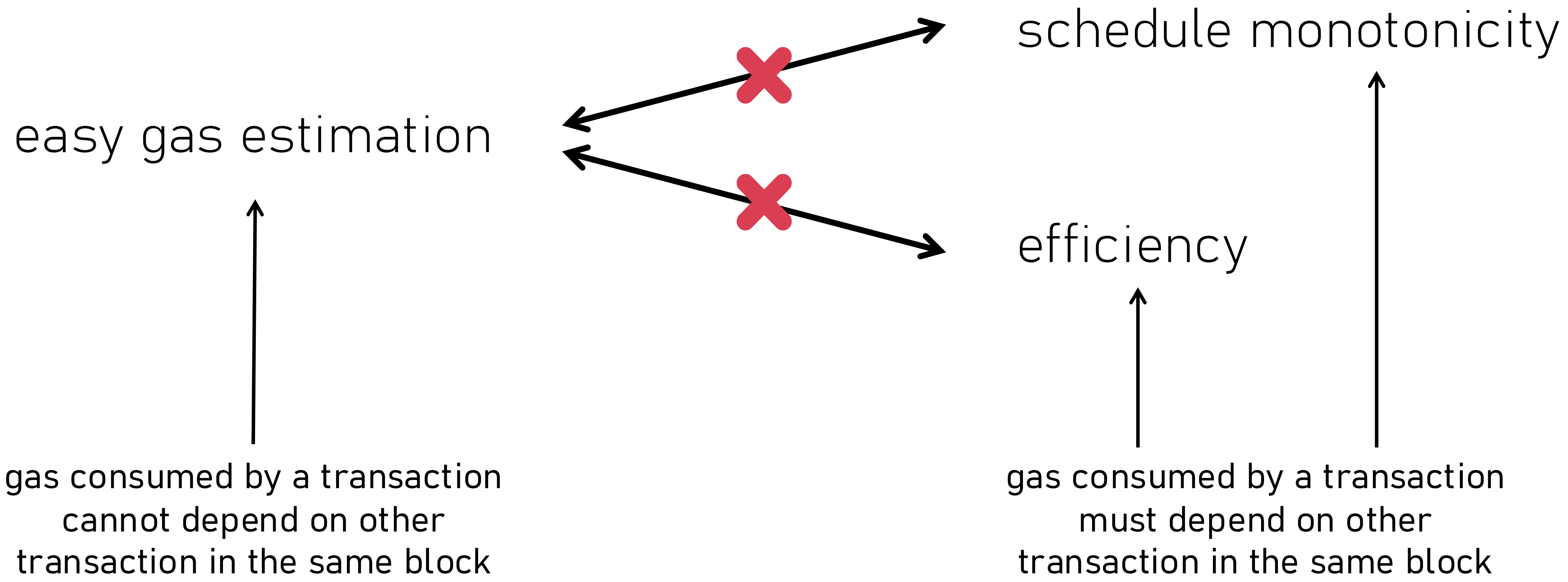
Schedule monotonicity: Transactions with higher marginal contributions to the execution time should consume more gas.

Efficiency: The gas consumption of all transactions in a block should collectively account for the total time needed to execute the block.

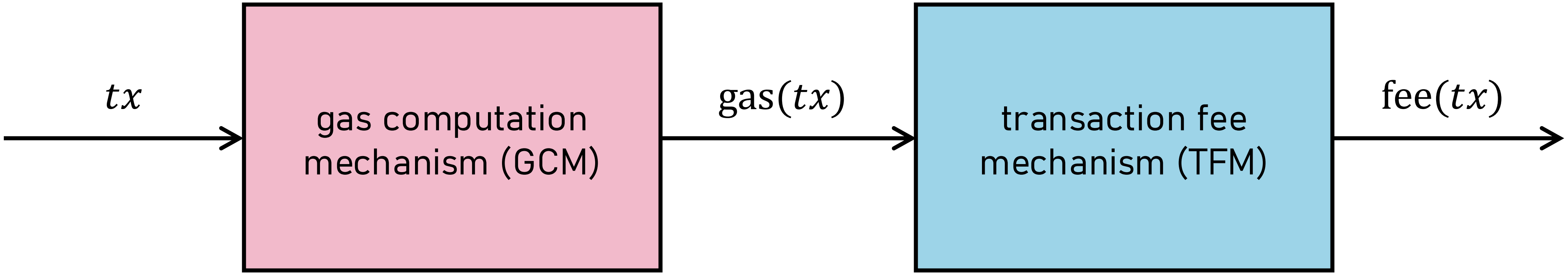
Easy gas estimation: Transaction submitters should be able to estimate a transaction's gas consumption in advance.

Can we achieve all properties with one mechanism?

Impossible to achieve all properties simultaneously.



Why is easy gas estimation important?



Easy gas estimation is essential for seamless composition with existing TFMs, i.e., retaining their properties.

What transaction fee mechanism (TFM) properties would we like to hold?

incentive compatible for users

incentive compatible for block producers

good welfare

off-chain agreement proof

} standard TFM properties

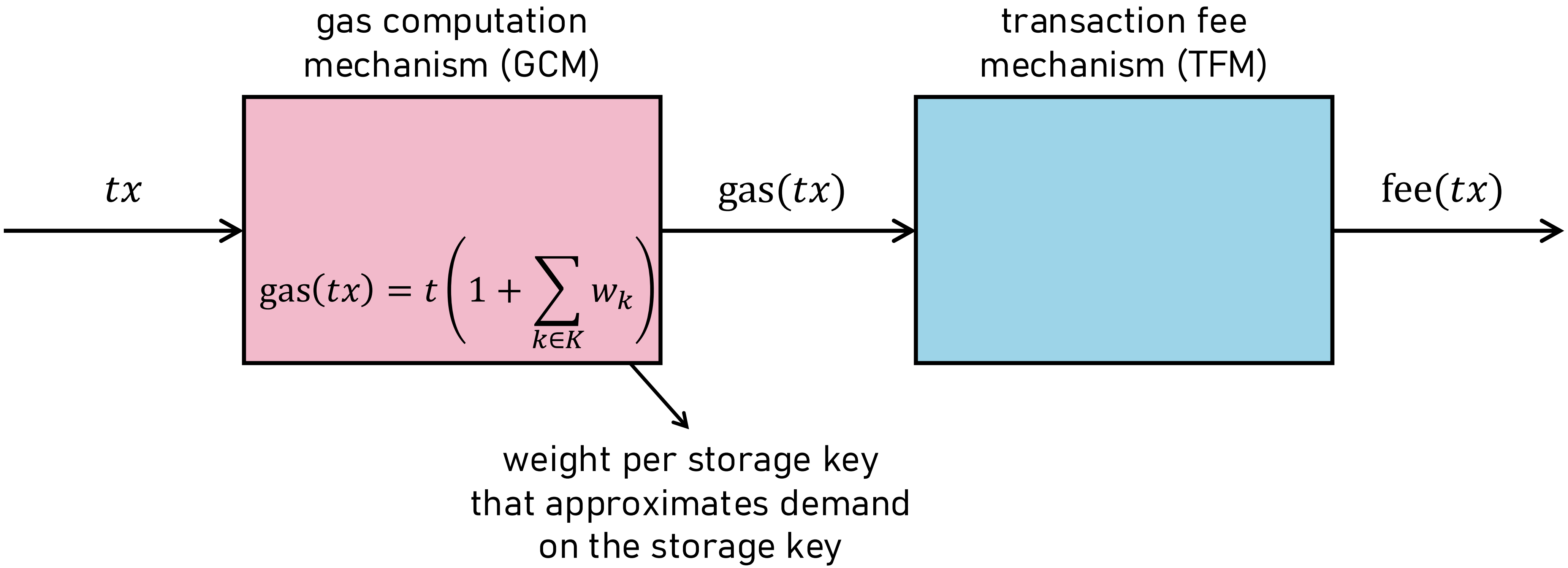
We explore two worlds of mechanisms,…

…those with easy gas estimation. …those without easy gas estimation.

Property	Current	W. Area	Shapley	Banzhaf	TPM	ESM	XSM
Storage Key Monotonicity	$=$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Time Monotonicity	$<$	$<$	$<$	$<$	$<$	$\leq$	$\leq$
Resource-Time Monotonicity	$\leq$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Set Inclusion	$<$	$<$	✗	✗	$\leq$	$\leq$	✗
Transaction Bundling	$=$	$\leq$	✗	$\leq$	$\leq$	✗	$<$
Scheduling Monotonicity	✗	✗	✗	✗	✗	$<$	$<$
Efficiency	✗	✗	✓	✗	✓	✓	✗
Easy Gas Estimation	✓	✓	✗	✗	✗	✗	✗
Poly-time Computable	✓	✓	$S(v)$	$B(v)$	v	v	v

What is a suitable fee market for our wish list with easy gas estimation?

total fee = gas units · gas price

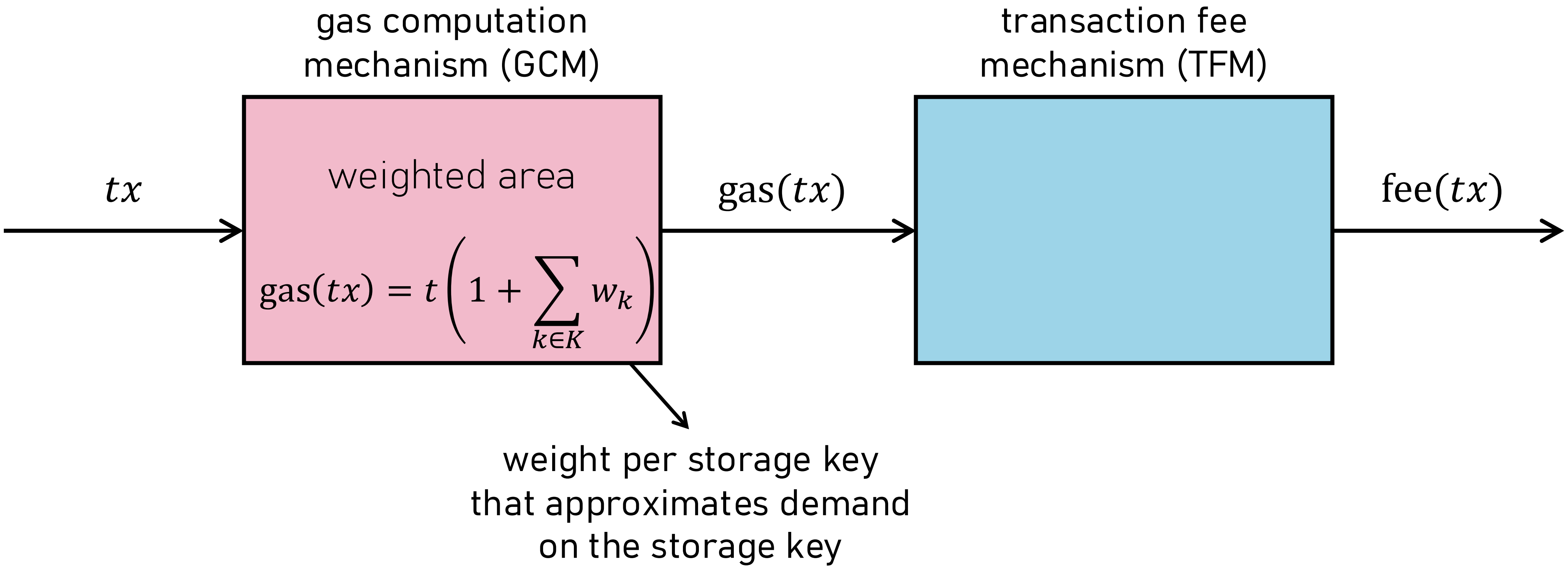


We explore two worlds of mechanisms,...

...those with easy gas estimation. ...those without easy gas estimation.

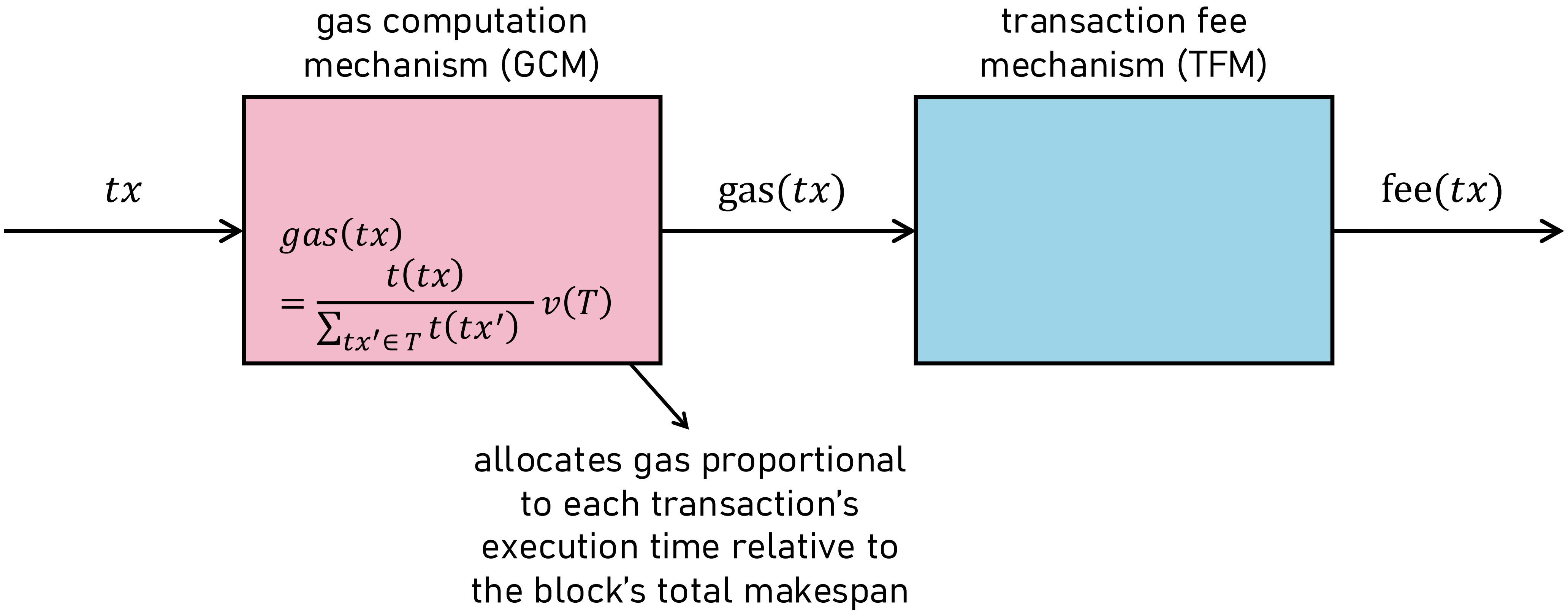
Property	Current	W. Area	Shapley	Banzhaf	TPM	ESM	XSM
Storage Key Monotonicity	$=$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Time Monotonicity	$<$	$<$	$<$	$<$	$<$	$\leq$	$\leq$
Resource-Time Monotonicity	$\leq$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Set Inclusion	$<$	$<$	$\times$	$\times$	$\leq$	$\leq$	$\times$
Transaction Bundling	$=$	$\leq$	$\times$	$\leq$	$\leq$	$\times$	$<$
Scheduling Monotonicity	$\times$	$\times$	$\times$	$\times$	$\times$	$<$	$<$
Efficiency	$\times$	$\times$	$\checkmark$	$\times$	$\checkmark$	$\checkmark$	$\times$
Easy Gas Estimation	$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$
Poly-time Computable	$\checkmark$	$\checkmark$	$S(v)$	$B(v)$	v	v	v

total fee = gas units · gas price



What is a suitable fee market for our wish list without easy gas estimation?

total fee = gas units · gas price

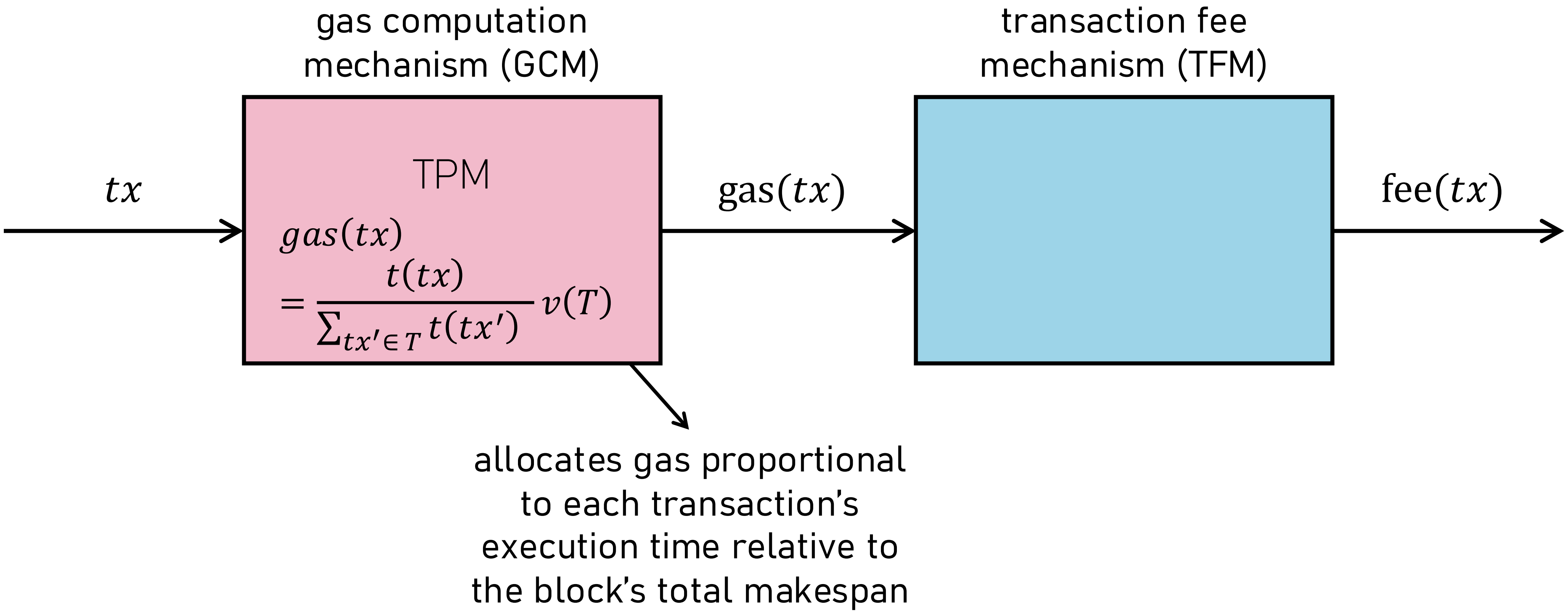


We explore two worlds of mechanisms,...

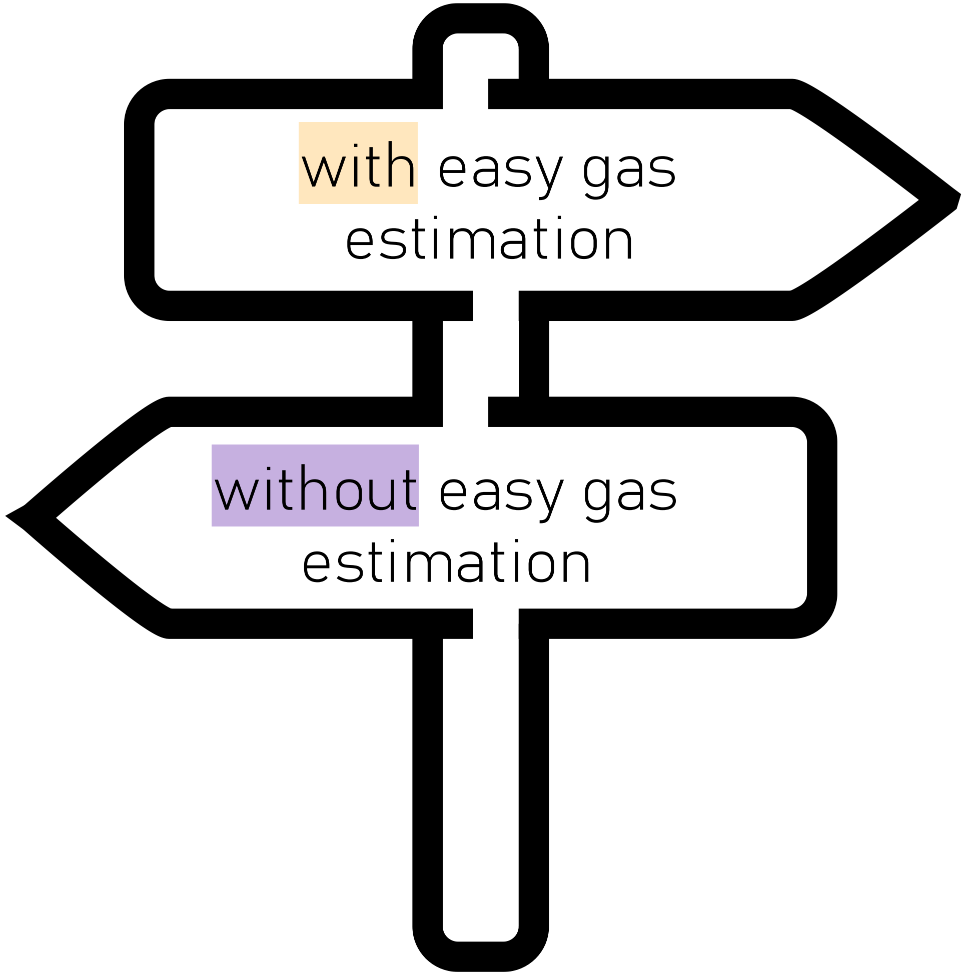
...those with easy gas estimation. ...those without easy gas estimation.

Property	Current	W. Area	Shapley	Banzhaf	TPM	ESM	XSM
Storage Key Monotonicity	$=$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Time Monotonicity	$<$	$<$	$<$	$<$	$<$	$\leq$	$\leq$
Resource-Time Monotonicity	$\leq$	$<$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$
Set Inclusion	$<$	$<$	✗	✗	$\leq$	$\leq$	✗
Transaction Bundling	$=$	$\leq$	✗	$\leq$	$\leq$	✗	$<$
Scheduling Monotonicity	✗	✗	✗	✗	✗	$<$	$<$
Efficiency	✗	✗	✓	✗	✓	✓	✗
Easy Gas Estimation	✓	✓	✗	✗	✗	✗	✗
Poly-time Computable	✓	✓	$S(v)$	$B(v)$	v	v	v

total fee = gas units · gas price



Which way should protocols go?



Our work ….

- distills a wish list of properties for fee markets in light of parallel execution.
- explores the design space of GCMs and identifies a promising candidates.
- paves the way for practical implementations of GCMs for parallel execution needed to unlock full execution layer scaling potential.

Transaction Fee Market Design for Parallel Execution

@liobaheimbach

@liobaheimbach

lheimbach@category.xyz

Weighted area GCM.

Definition 4.1 (Weighted Area GCM). Given a set of transactions T and a transaction $tx \in T$ with $tx \simeq (t, R)$, the *Weighted Area* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^{WA}(tx) := t \cdot \left(1 + \sum_{r \in R} w_r\right) \tag{1}$$

Mechanisms with easy gas estimation.

Definition 4.1 (Weighted Area GCM). Given a set of transactions T and a transaction $tx \in T$ with $tx \simeq (t, R)$, the *Weighted Area* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^{WA}(tx) := t \cdot \left(1 + \sum_{r \in R} w_r\right) \tag{1}$$

Mechanisms without easy gas estimation.

Definition 4.2 (Shapley GCM). Given a set of transactions T consisting of $|T| = n$ transactions and a transaction $tx \in T$, the *Shapley* GCM computes the amount of gas used by tx as follows:

$$\begin{aligned} \text{gas}_T^S(tx) :=& \frac{1}{n!} \sum_{\sigma} \left[v(P_{tx}^{\sigma} \cup \{tx\}) - v(P_{tx}^{\sigma}) \right] & (2) \\ =& \sum_{S \subseteq T \setminus \{tx\}} \frac{|S|! \cdot (n - |S| - 1)!}{n!} \left[v(S \cup \{tx\}) - v(S) \right]. & (3) \end{aligned}$$

Here, σ ranges over the $n!$ possible ways to order the transactions in T and P_{tx}^{σ} denotes the set of transactions that precede tx in the order σ. The equality between Eqs. (2) and (3) follows by counting the number of orders σ such that $P_{tx}^{\sigma} = S$, which there are $|S|! \cdot (n - |S| - 1)!$ of.

Definition 4.4 (Banzhaf GCM). Given a set of transactions T consisting of $|T| = n$ transactions and a transaction $tx \in T$, the *Banzhaf* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^B(tx) := \frac{1}{2^{n-1}} \sum_{S \subseteq T \setminus \{tx\}} \left[v(S \cup \{tx\}) - v(S) \right].$$

Mechanisms without easy gas estimation.

Definition 4.6 (Time-Proportional Makespan GCM). Given a set of transactions T and a transaction $tx \in T$, the *Time-Proportional Makespan (TPM)* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^{TPM}(tx) := \frac{t(tx)}{\sum_{tx' \in T} t(tx')} \cdot v(T).$$

Definition 4.7 (Equally-Split Makespan GCM). Given a set of transactions T and a transaction $tx \in T$, the *Equally-Split Makespan (ESM)* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^{ESM}(tx) := \frac{v(T)}{|T|}.$$

Definition 4.8 (Exponentially-Split Makespan GCM). Given a set of transactions T and a transaction $tx \in T$, the *Exponentially-Split Makespan (XSM)* GCM computes the amount of gas used by tx as follows:

$$\text{gas}_T^{XSM}(tx) := \frac{v(T)}{3^{|T|}}.$$