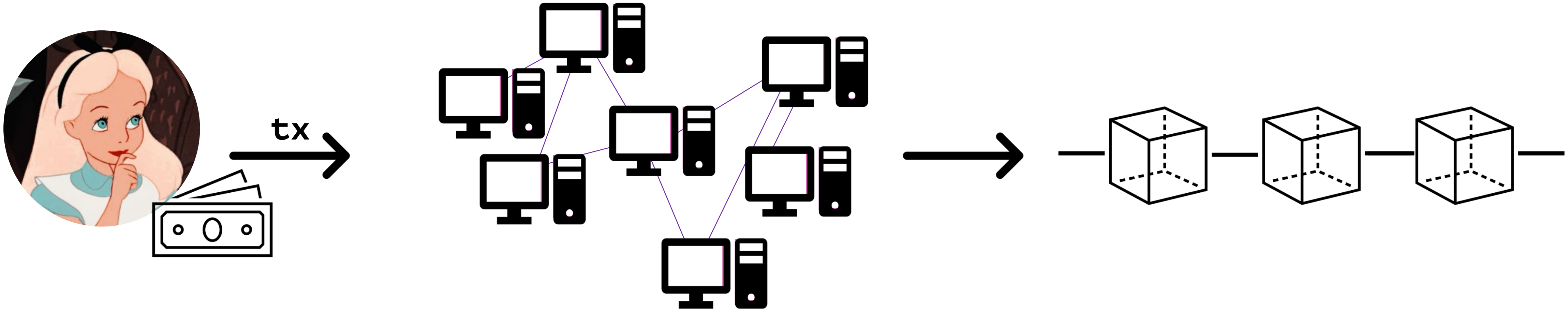
A FAIR & DECENTRALIZED CLOCK NETWORK FOR TRANSACTION ORDERING

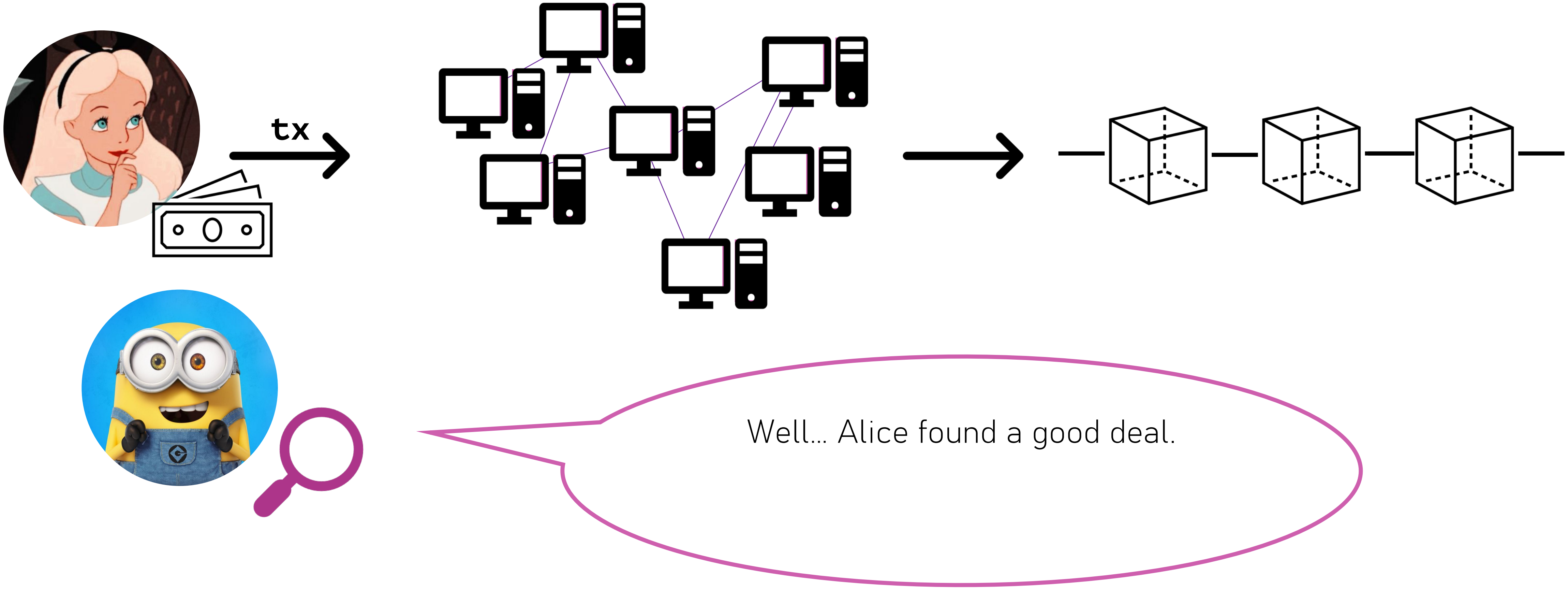
Andrei Constantinescu, Diana Ghinea, Lioba Heimbach, Zilin Wang, Roger Wattenhofer

ETH Zürich

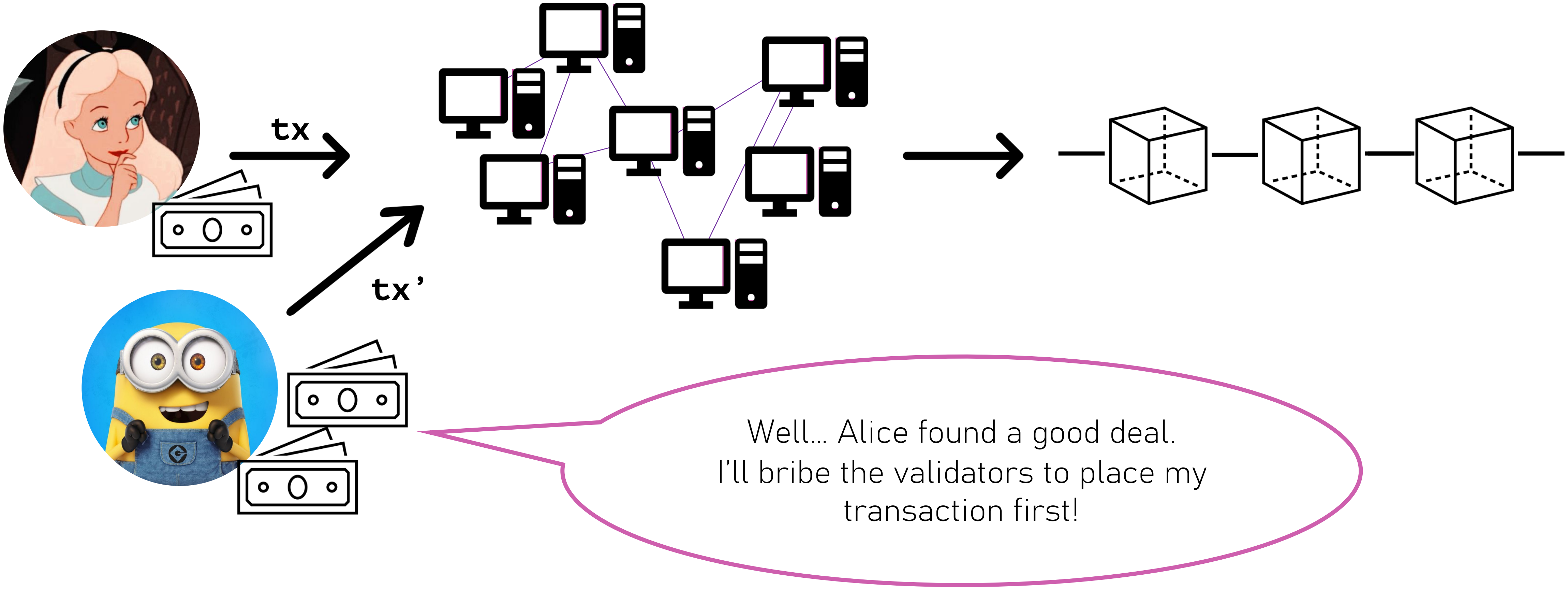
TRANSACTIONS & FRONT-RUNNING



TRANSACTIONS & FRONT-RUNNING



TRANSACTIONS & FRONT-RUNNING

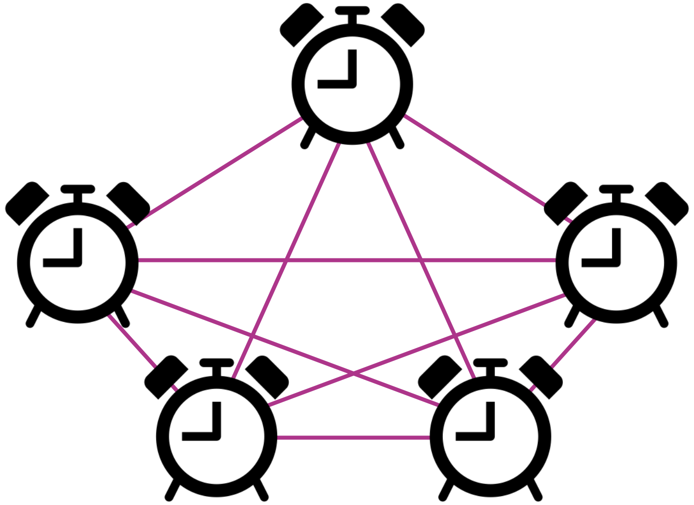


ADDITIONAL LAYER: CLOCKS

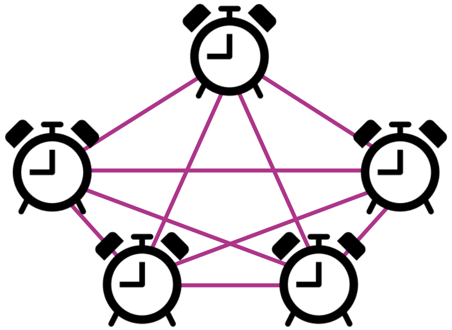
ADDITIONAL LAYER: CLOCKS

We consider $\mathbf{n}$ nodes equipped with clocks in an asynchronous network.

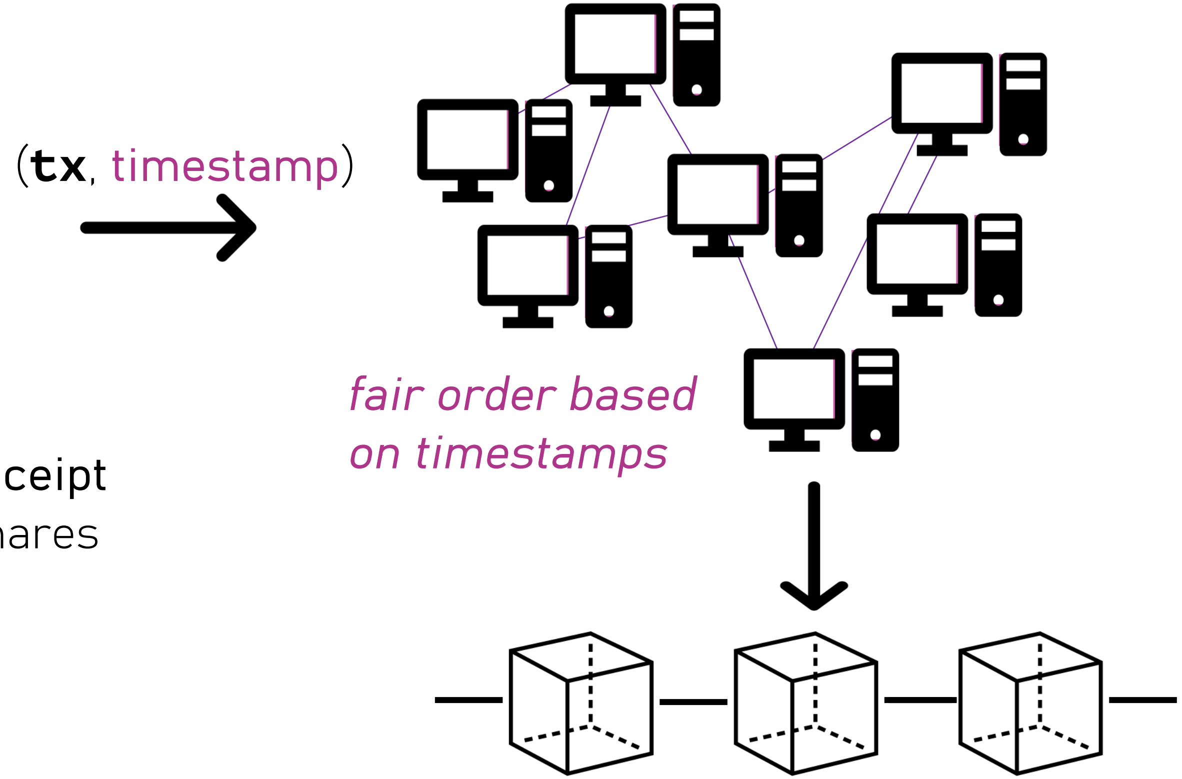
Out of these nodes, $\mathbf{f < n/3}$ may be byzantine.



ADDITIONAL LAYER: CLOCKS



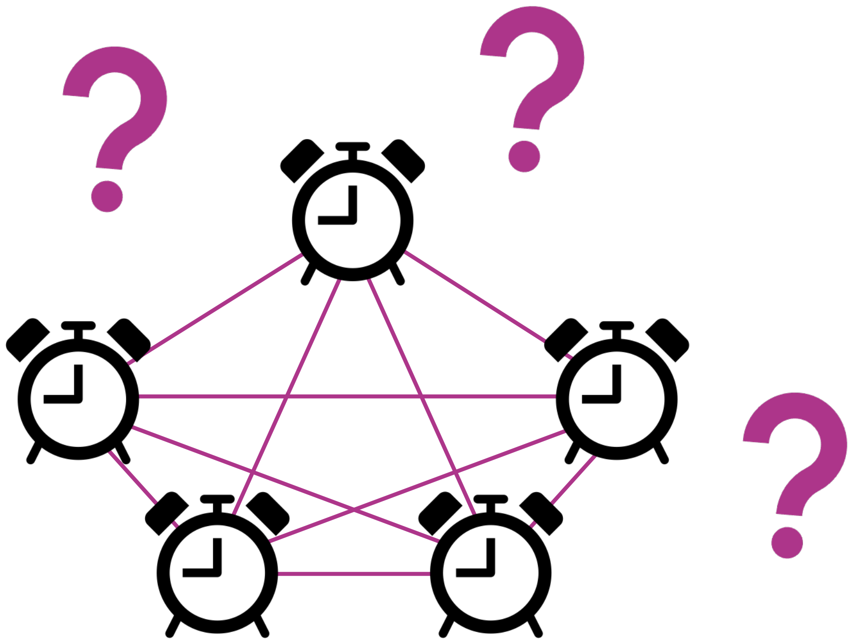
1. Agree on a timestamp of receipt
2. Reconstruct **tx** from the shares



WHAT IS A *FAIR* TIMESTAMP?

- If **tx1** is received before **tx2**,

  then **tx1** ‘s timestamp should not be later than **tx2**’s timestamp.



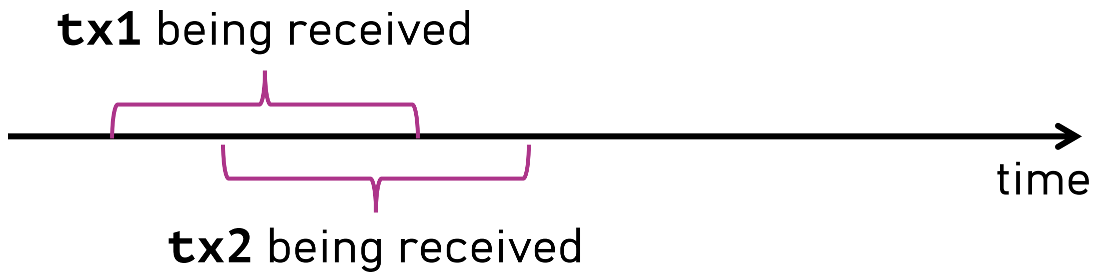
… but what does that mean for a decentralized network?

WHAT IS A *FAIR* TIMESTAMP?

- If the median time when honest nodes receive **`tx1`**

  is before the median time when honest nodes receive **`tx2`**,

  then **`tx1`** ‘s timestamp should not be later than **`tx2`**’s timestamp.



…Byzantine Agreement with Median Validity?

BYZANTINE AGREEMENT

Assume $\mathbf{n}$ nodes hold input values.

Even when $\mathbf{f}$ out of these nodes are byzantine:

(**Termination**) All honest nodes output some value.

(**Agreement**) All honest nodes output the same value.

(All-Same Validity) **If all honest nodes hold the same input value, that's the value they output.**

BYZANTINE AGREEMENT + MEDIAN VALIDITY

Assume $\mathbf{n}$ nodes hold input values.

Even when $\mathbf{f}$ out of these nodes are byzantine:

(Termination) All honest nodes output some value.

(Agreement) All honest nodes output the same value.

(Median Validity) Honest nodes' outputs are *close* to the honest inputs' median.

δ-MEDIAN VALIDITY

$$\boldsymbol{\tau_1 \leq \tau_2 \leq \cdots \leq \tau_{median-\delta} \leq \cdots \leq \tau_{median} \leq \cdots \leq \tau_{median+\delta} \leq \cdots \leq \tau_{n-f}}$$

Any value in $[\boldsymbol{\tau_{median-\delta}}, \boldsymbol{\tau_{median+\delta}}]$ is valid.

δ-MEDIAN VALIDITY

Synchronous model: $\mathbf{\delta \geq f/2}$ sufficient and necessary.

[OPODIS:StoWat15, SRDS:MelWat18]

Asynchronous model: ? ? ?

δ-MEDIAN VALIDITY

Synchronous model: $\boldsymbol{\delta \geq f/2}$ sufficient and necessary.

[OPODIS:StoWat15, SRDS:MelWat18]

Asynchronous model: $\boldsymbol{\delta \geq f}$ necessary!!!

This is quite weak when $\boldsymbol{f < n/3}$...

...Compromise?

TIMESTAMP AGREEMENT

Assume $\mathbf{n}$ nodes hold **(integer) timestamps** as input values.

Even when $\mathbf{f}$ out of these nodes are byzantine:

(**Termination**) All honest nodes output some value.

(**Agreement**) All honest nodes output the same value.

(**$\boldsymbol{\delta}$-Median Validity**) Honest outputs are in $[\tau_{\mathbf{median}-\delta}, \tau_{\mathbf{median}+\delta}]$.

For $\boldsymbol{\delta} = \mathbf{f/2}$ if the network is synchronous and $\boldsymbol{\delta} = \mathbf{f}$ otherwise.

ACHIEVING TIMESTAMP AGREEMENT

1. Every node sends its initial timestamp to everyone.

   When sufficient time to allow for **one synchronous round has passed**

   and $\mathbf{n} - \mathbf{f} + k$ values were received (with $\mathbf{0} \leq \mathbf{k} \leq \mathbf{f}$):

   $\boldsymbol{\tau}_{\mathbf{median}}$:= the $\lceil (\mathrm{n} - \mathrm{f})/2 \rceil + \lfloor \mathrm{k}/2 \rfloor$-th lowest value received.

ACHIEVING TIMESTAMP AGREEMENT

1. Every node sends its initial timestamp to everyone.

   When sufficient time to allow for **one synchronous round has passed**

   and $\mathbf{n - f} + k$ values were received (with $\mathbf{0 \leq k \leq f}$):

   $\boldsymbol{\tau}_{\mathbf{median}}$:= the $\lceil (\mathrm{n - f})/2 \rceil + \lfloor \mathrm{k}/2 \rfloor$-th lowest value received.

=> $\boldsymbol{\tau}_{\mathbf{median}}$ satisfies δ–Median Validity for:

synchronous network => $\delta = \mathrm{f}/2$.

asynchronous network => $\delta = \mathrm{f}$.

$\boldsymbol{\delta}$–Median Validity ✓

Agreement ✗

ACHIEVING TIMESTAMP AGREEMENT

2. Join ApproximateAgreement with $\boldsymbol{\tau}_{\mathbf{median}}$ as input. Obtain output $\boldsymbol{\tau}_{\mathbf{AA}}$.

=> honest nodes obtain $\boldsymbol{\varepsilon}$-close outputs $\boldsymbol{\tau}_{\mathbf{AA}}$ $(0 < \boldsymbol{\varepsilon} < 0.5)$

within the range of their inputs.

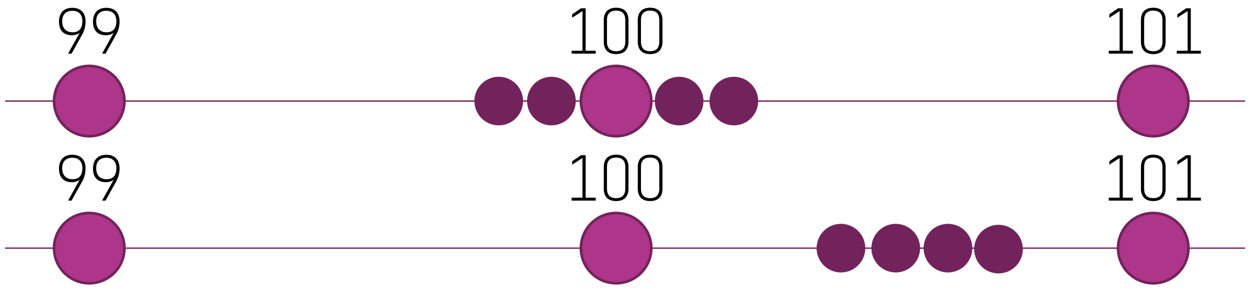
$\boldsymbol{\delta}$-Median Validity ✓

Agreement Up to an error < 0.5

ACHIEVING TIMESTAMP AGREEMENT

3. Nodes need to decide whether to round their values $\boldsymbol{\tau_{AA}}$ up or down,

...such that they end up with the same value.



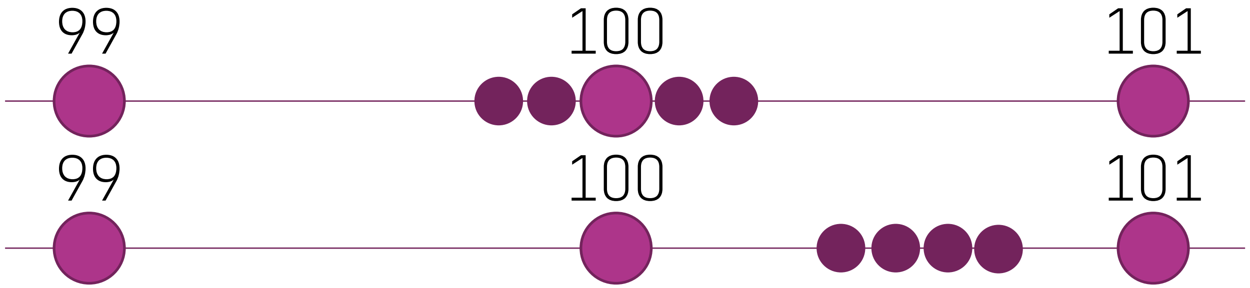
$\boldsymbol{\delta}$-Median Validity ✓

Agreement ✓

ACHIEVING TIMESTAMP AGREEMENT

3. Nodes need to decide whether to round their values $\boldsymbol{\tau_{AA}}$ up or down,

…such that they end up with the same value.



Run **ByzantineAgreement with input $\mathbf{b = 0}$ if the closest integer is even**

and $\mathbf{b = 1}$ if the closest integer is odd.

Output the closest even integer if **output bit = 0**, otherwise the closest odd integer.

THANK YOU & SUMMARY

- Transactions are *fairly* ordered based on the ~**median** timestamp of receipt.
- The timestap of receipt is decided by a network of nodes equiped with clocks.
- **Timestamp Agreement** protocol:
  - Asynchronous Byzantine Agreement with Median Validity.
  - Optimal resilience guarantees.
  - Optimal Median Validity guarantees for the actual network conditions.