The Ripple Protocol Consensus Algorithm

- Authors: David Schwartz, Noah Youngs, and Arthur Britto

Presenter: Tianran Wang

ripple

Blockchain protocol, like Bitcoin and Ethereum

Provides fast, scalable, and stable payment services

XRP

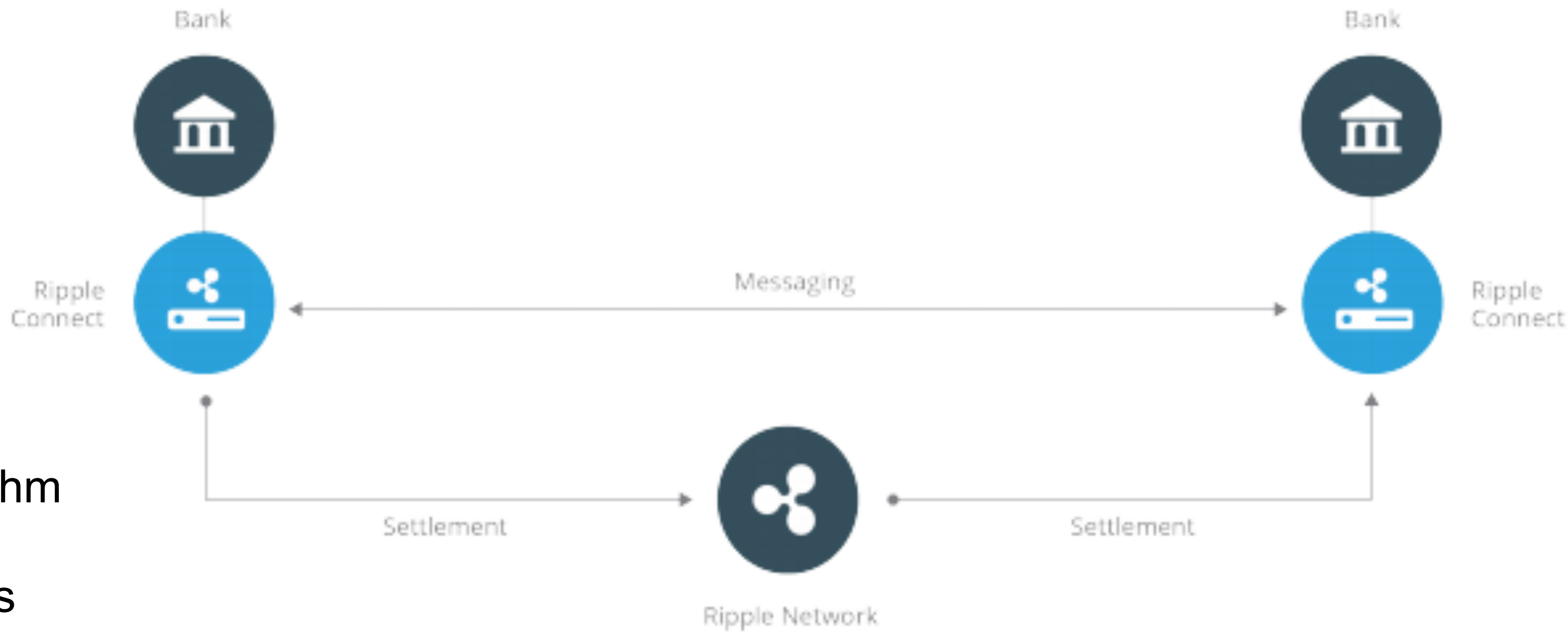
A native cryptocurrency

Acts as a central for money transferal

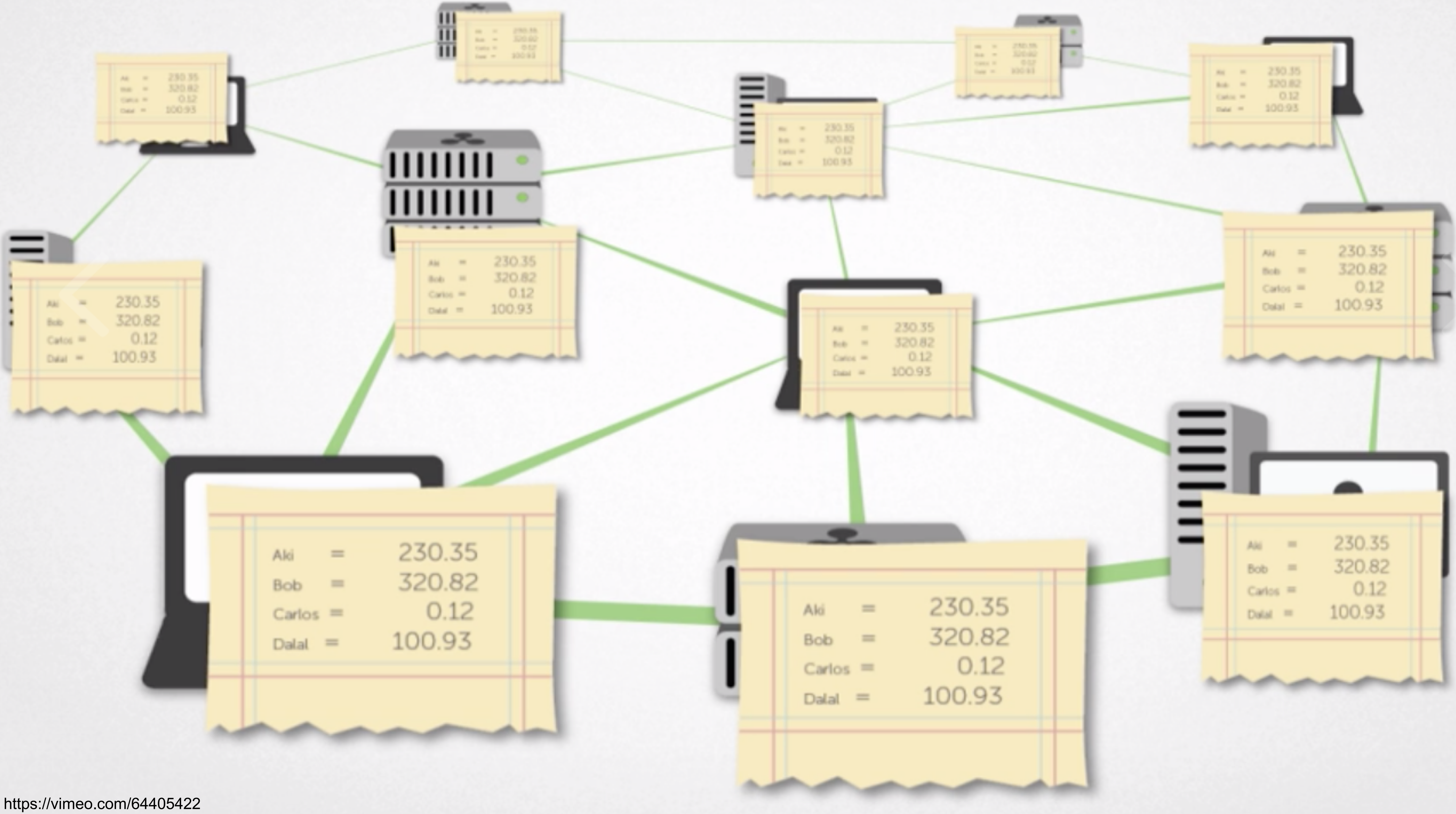
RPCA

A fast and low-cost consensus algorithm

Can tolerate (n-1)/5 Byzantine failures



https://hackernoon.com/whitepaper-in-four-minutes-ripple-a27103e4d265



https://vimeo.com/64405422

RPCA Components

Unique Node List(UNL):

Lists other servers queried by this server
A subset of the network trusted by this server

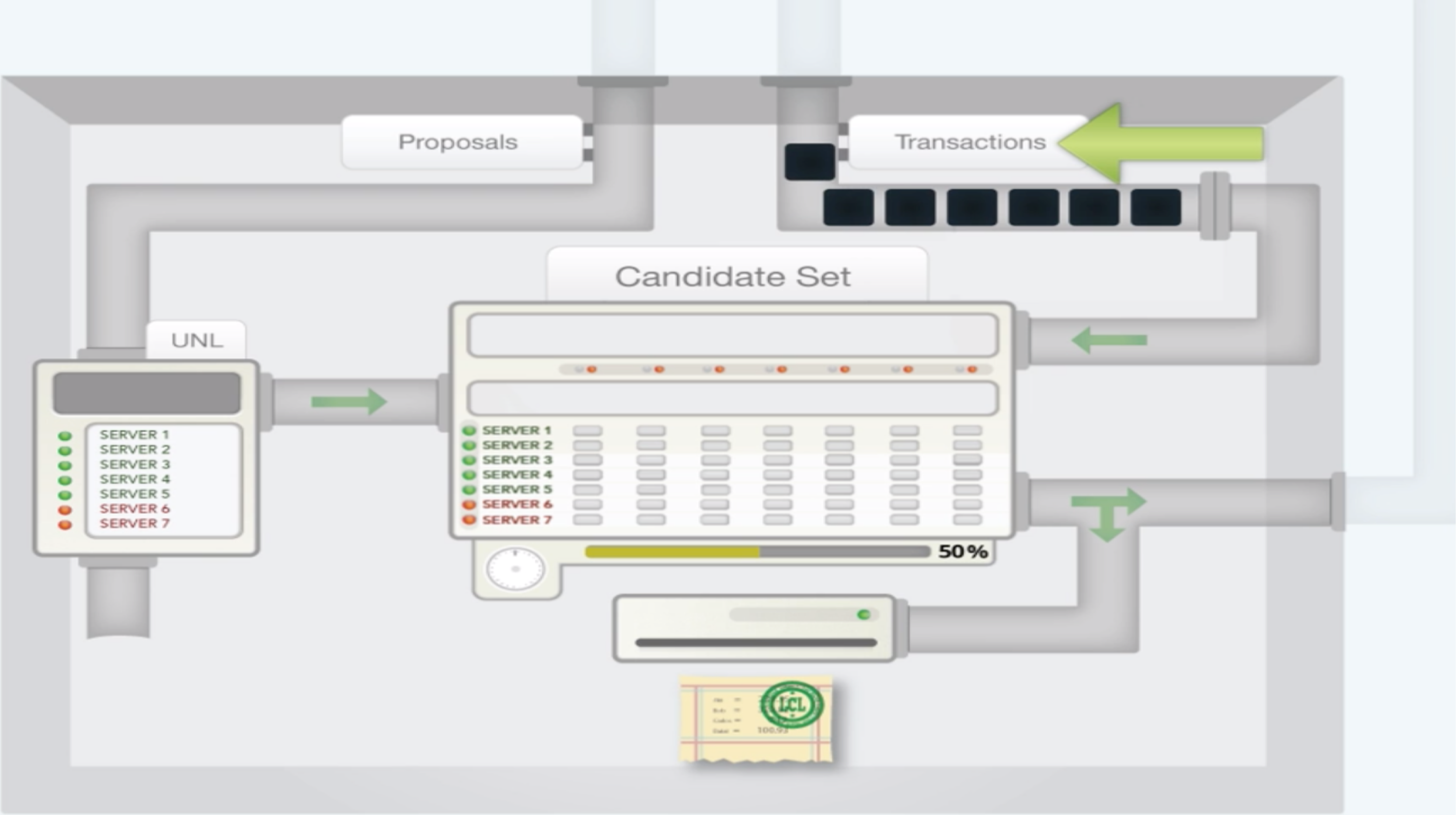
Last-Closed Ledger:

Represents the most recent consensus among all servers
Should be identical

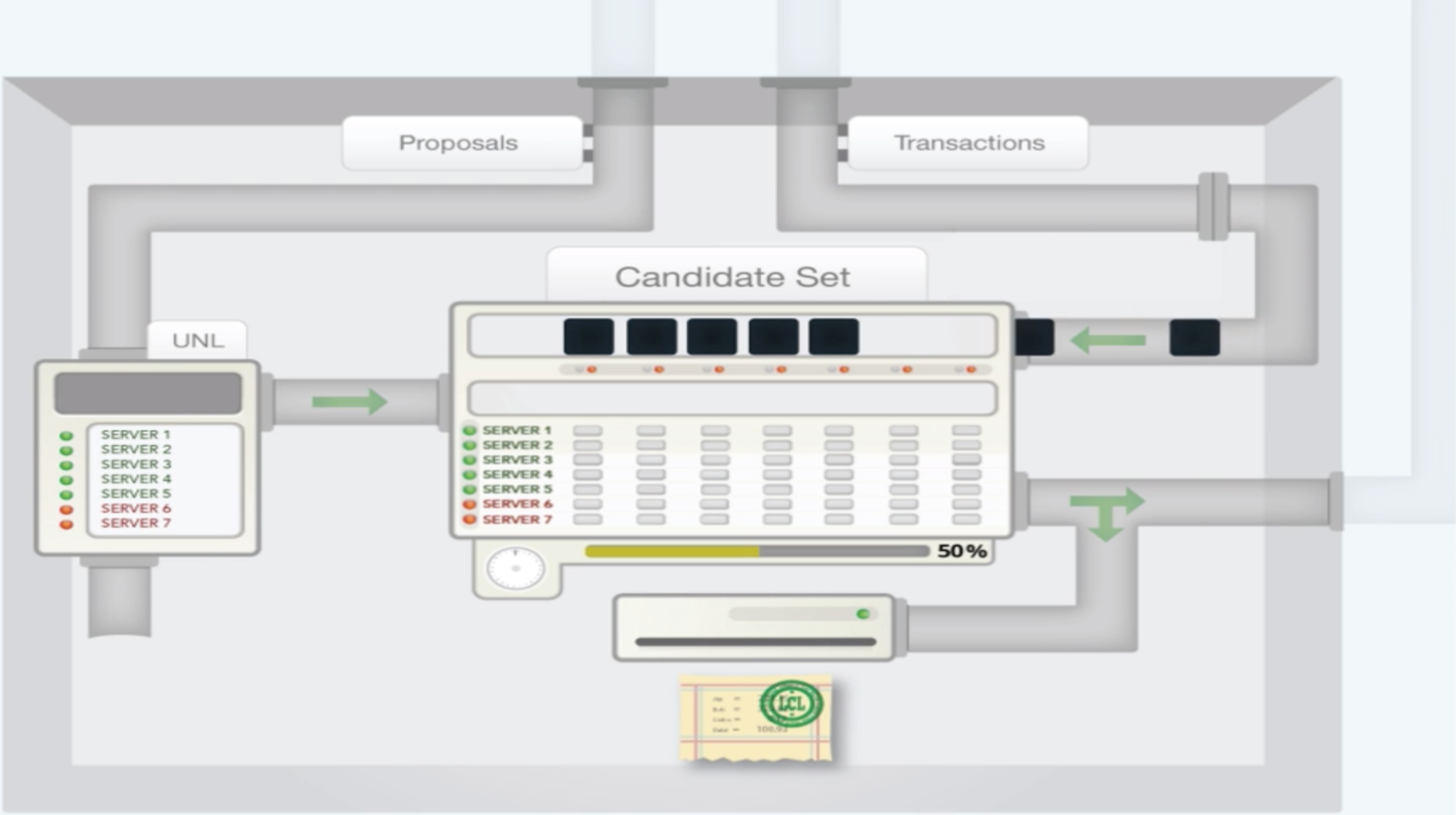
Open Ledger:

Represents current status of this server
Different among servers

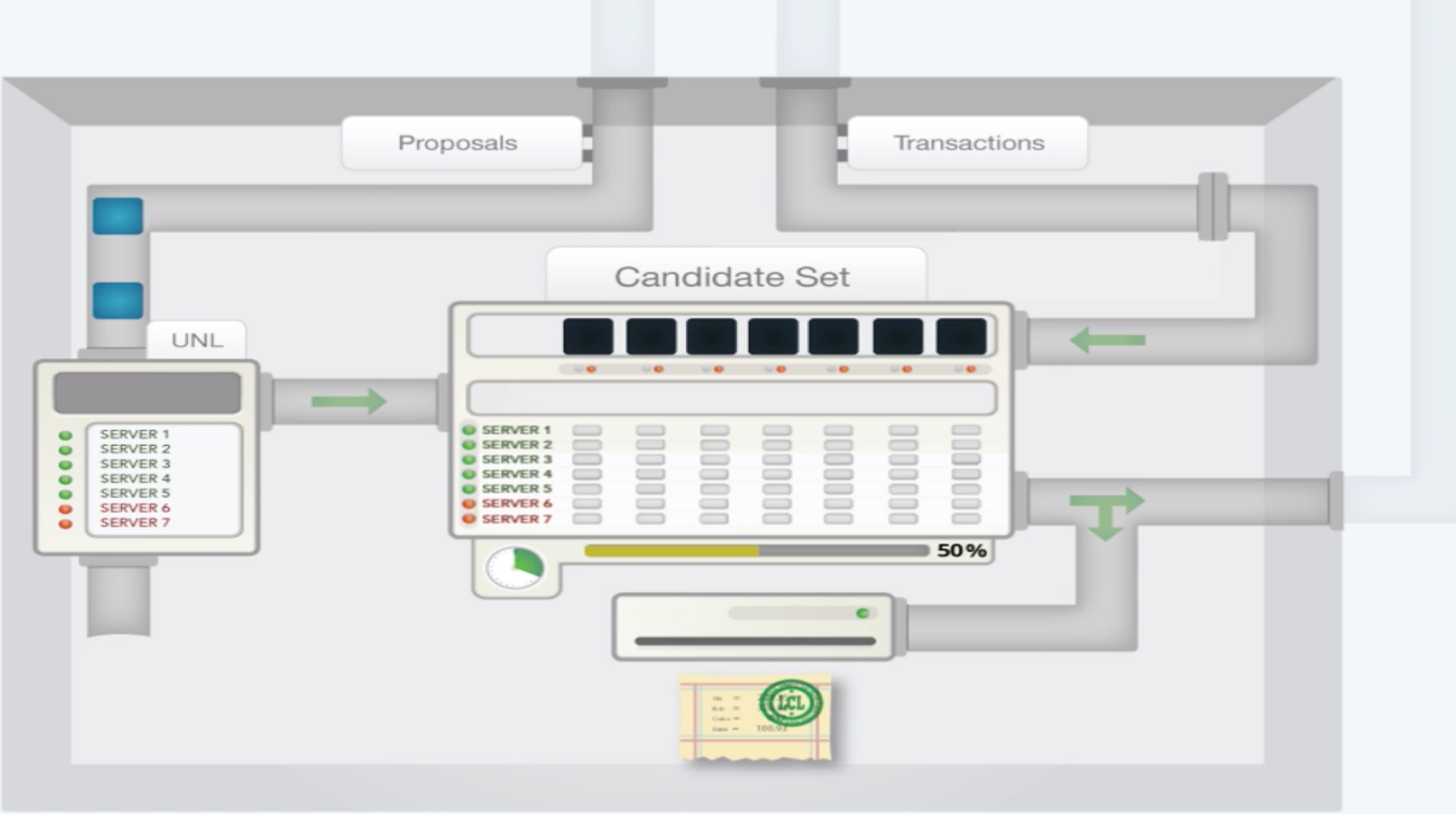
Once consensus is reached, a set of transactions will be applied on the open ledger.
Then it becomes the last-closed ledger.



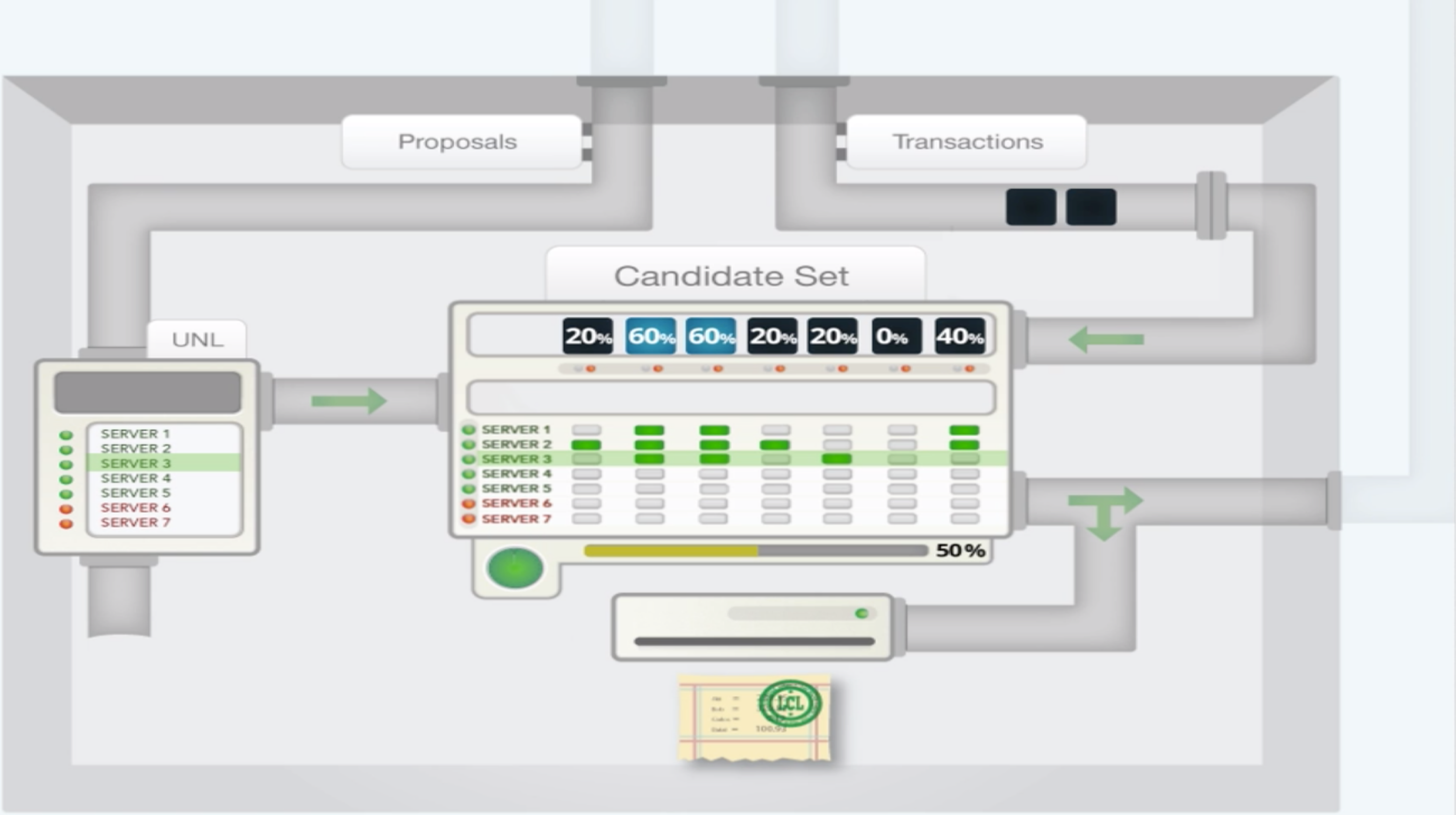
https://vimeo.com/64405422



https://vimeo.com/64405422

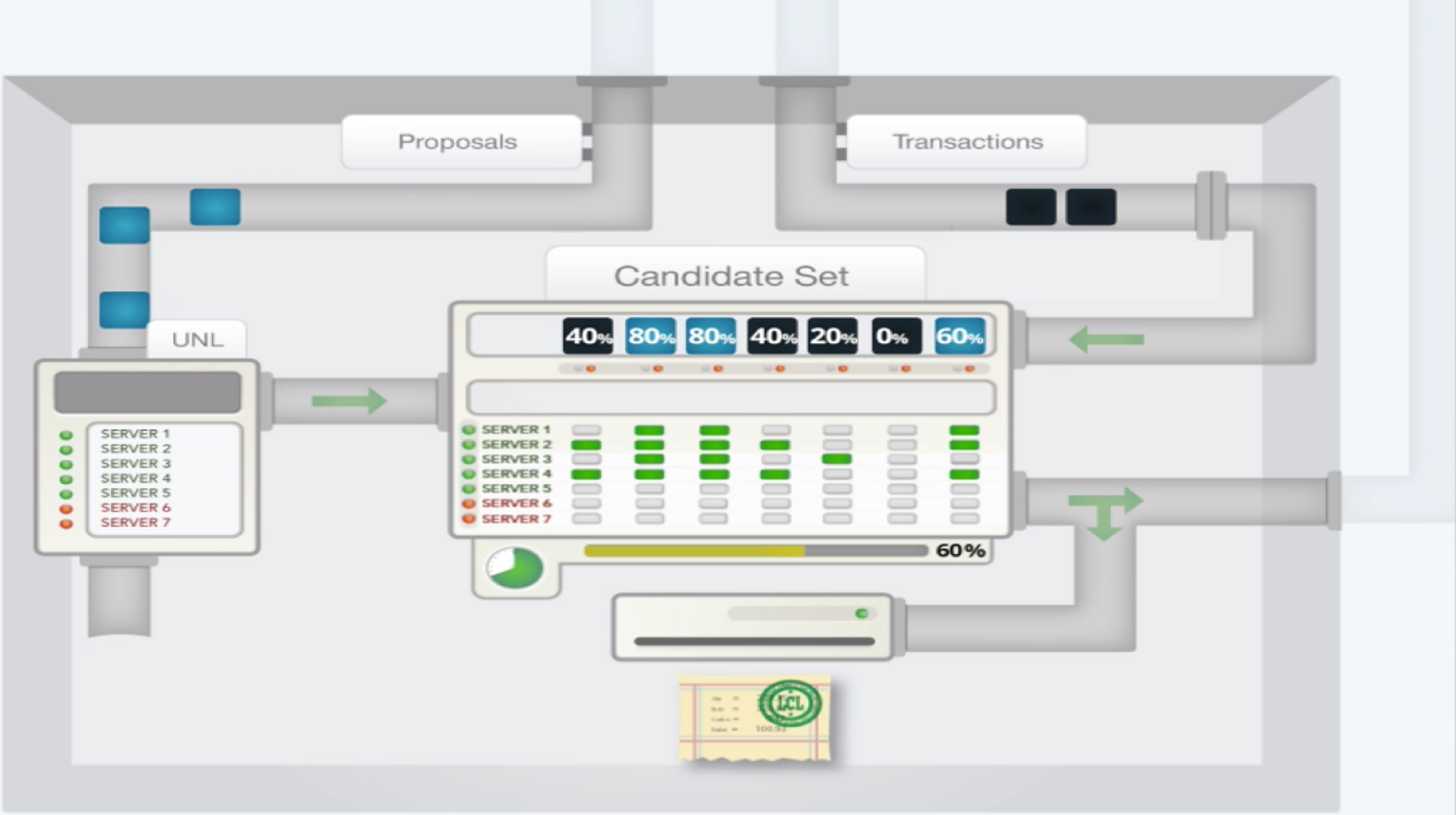


https://vimeo.com/64405422



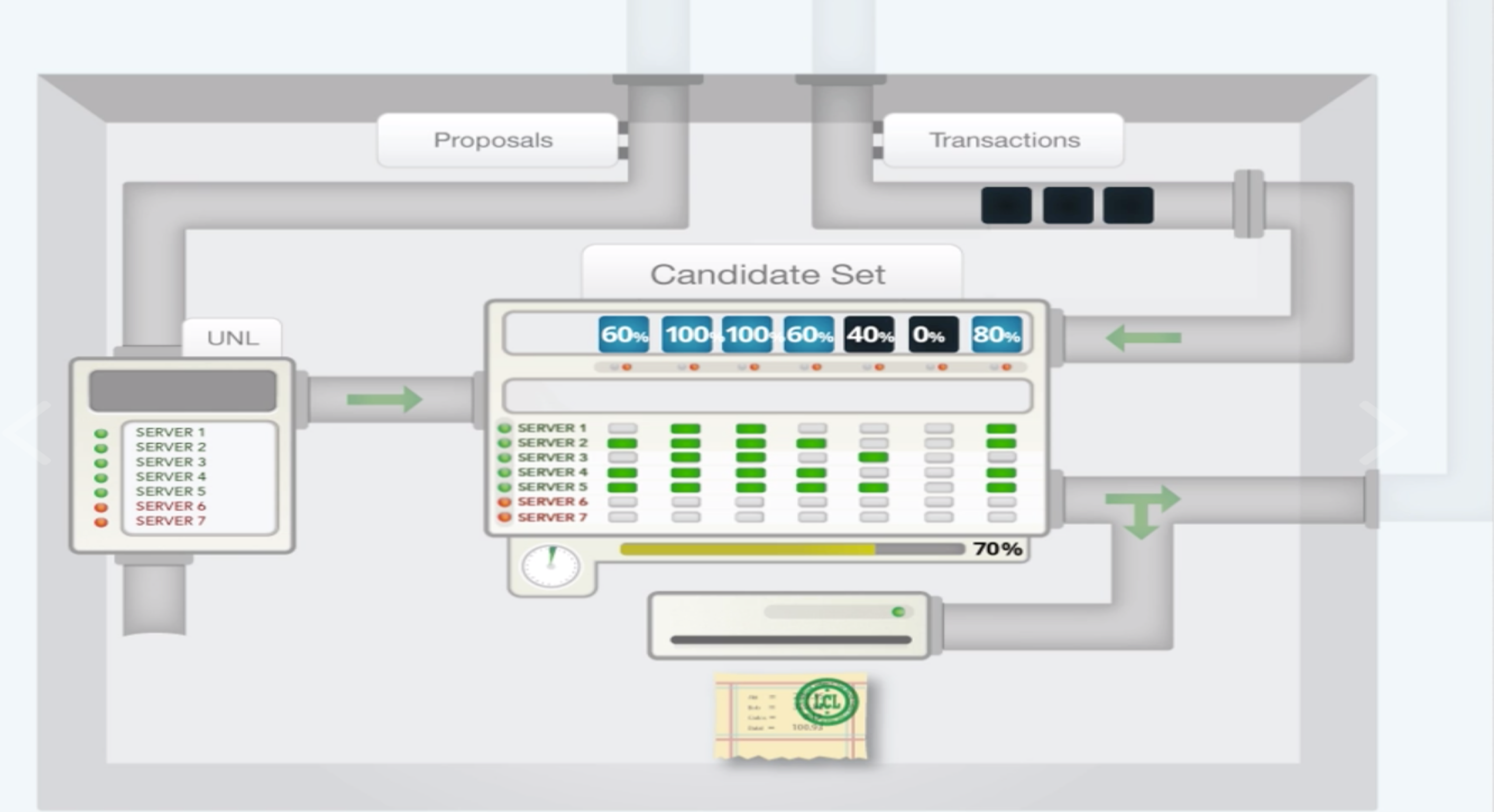
	20%	60%	60%	20%	20%	0%	40%
SERVER 1	□	🟩	🟩	□	□	□	🟩
SERVER 2	🟩	🟩	🟩	🟩	□	□	🟩
SERVER 3	□	🟩	🟩	□	🟩	□	□
SERVER 4	□	□	□	□	□	□	□
SERVER 5	□	□	□	□	□	□	□
SERVER 6	□	□	□	□	□	□	□
SERVER 7	□	□	□	□	□	□	□

https://vimeo.com/64405422



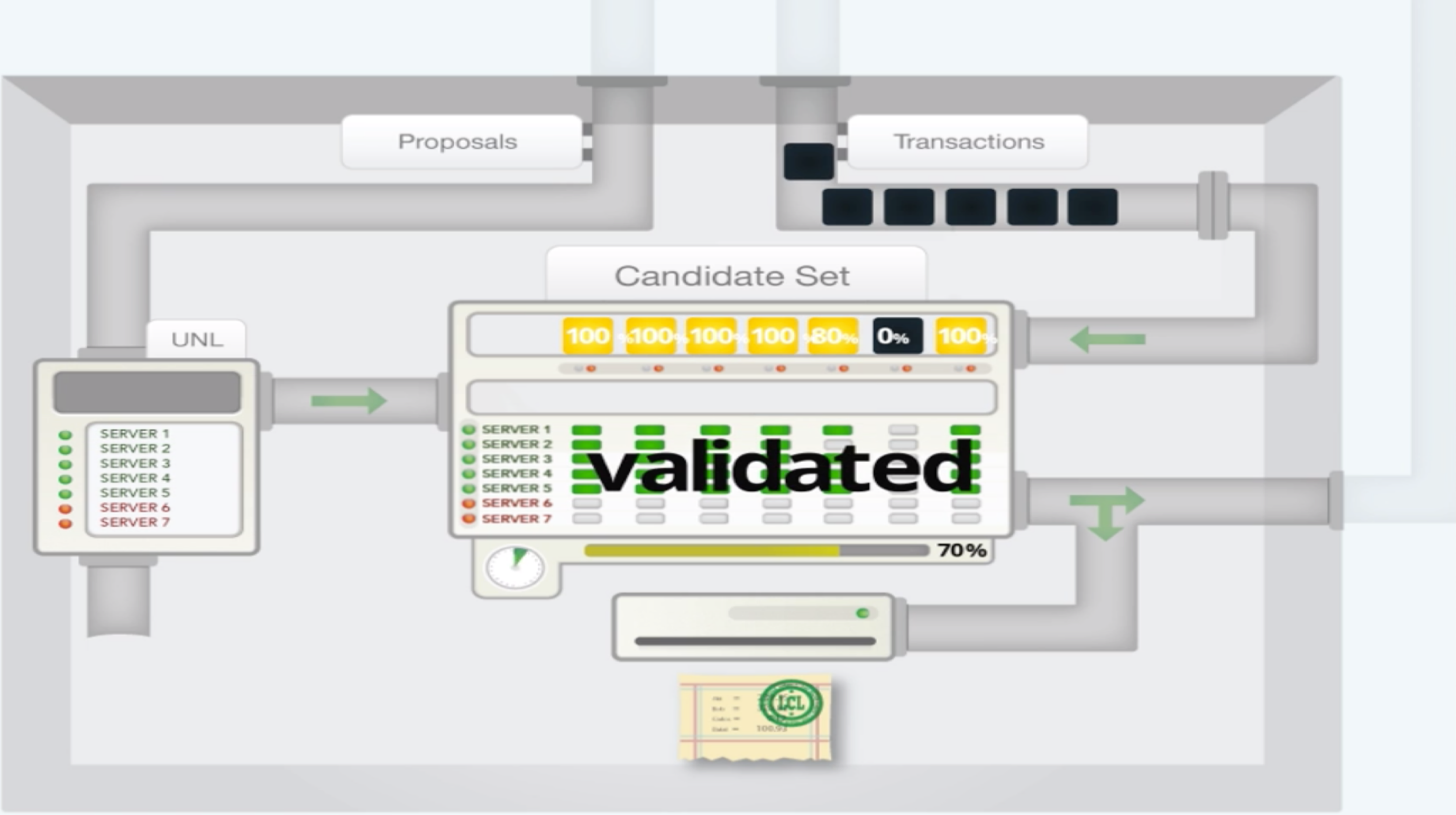
	40%	80%	80%	40%	20%	0%	60%
SERVER 1	⬜	🟩	🟩	⬜	⬜	⬜	🟩
SERVER 2	🟩	🟩	🟩	🟩	⬜	⬜	🟩
SERVER 3	⬜	🟩	🟩	⬜	🟩	⬜	⬜
SERVER 4	🟩	🟩	🟩	🟩	⬜	⬜	🟩
SERVER 5	⬜	⬜	⬜	⬜	⬜	⬜	⬜
SERVER 6	⬜	⬜	⬜	⬜	⬜	⬜	⬜
SERVER 7	⬜	⬜	⬜	⬜	⬜	⬜	⬜

https://vimeo.com/64405422

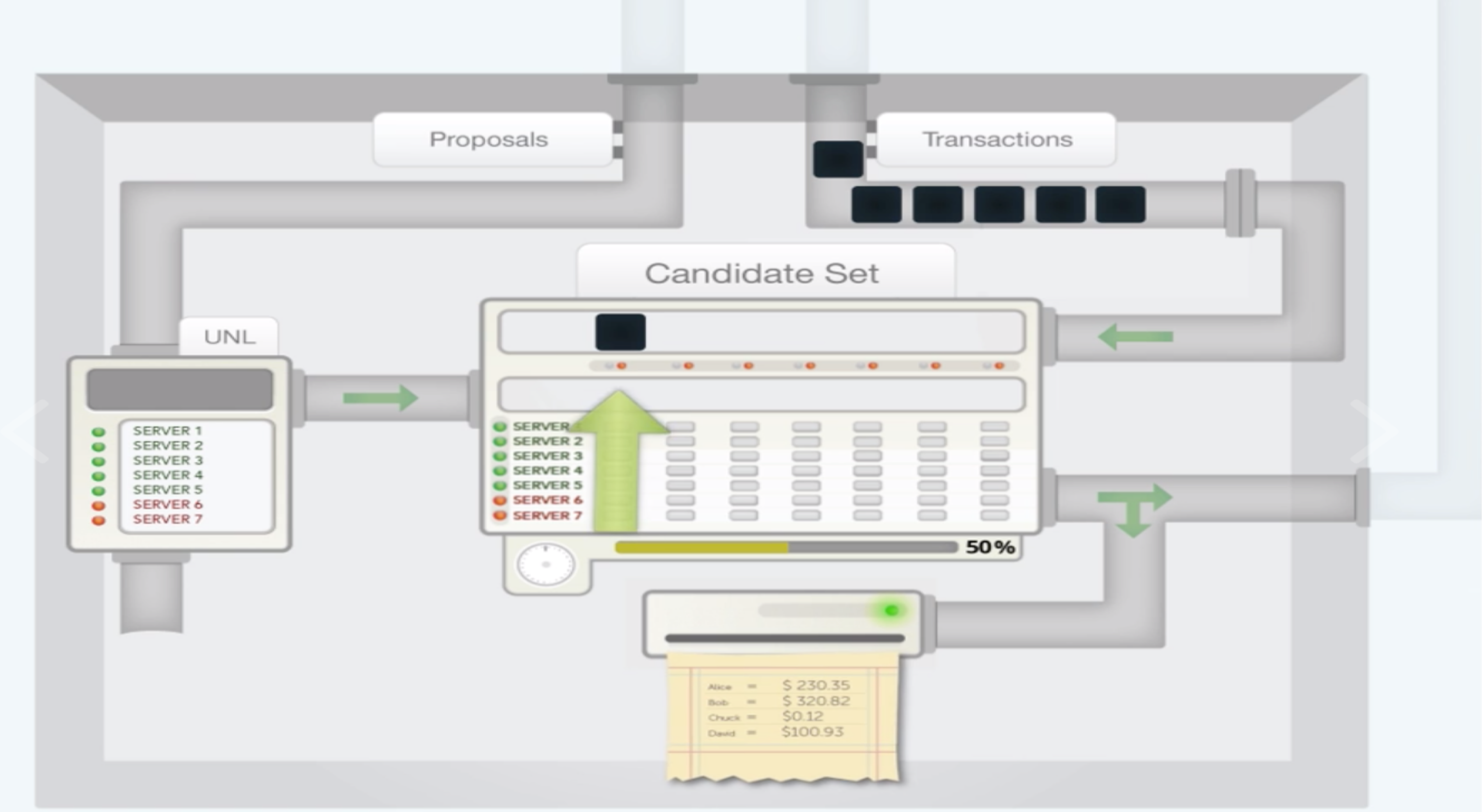


	60%	100%	100%	60%	40%	0%	80%
SERVER 1	⬜	🟩	🟩	⬜	⬜	⬜	🟩
SERVER 2	🟩	🟩	🟩	🟩	⬜	⬜	🟩
SERVER 3	⬜	🟩	🟩	⬜	🟩	⬜	⬜
SERVER 4	🟩	🟩	🟩	🟩	⬜	⬜	🟩
SERVER 5	🟩	🟩	🟩	🟩	🟩	⬜	🟩
SERVER 6	⬜	⬜	⬜	⬜	⬜	⬜	⬜
SERVER 7	⬜	⬜	⬜	⬜	⬜	⬜	⬜

https://vimeo.com/64405422



https://vimeo.com/64405422



Alice	=	$ 230.35
Bob	=	$ 320.82
Chuck	=	$0.12
David	=	$100.93

https://vimeo.com/64405422

Correctness

A transaction is only approved if 80% of the UNL of a server agrees with it.

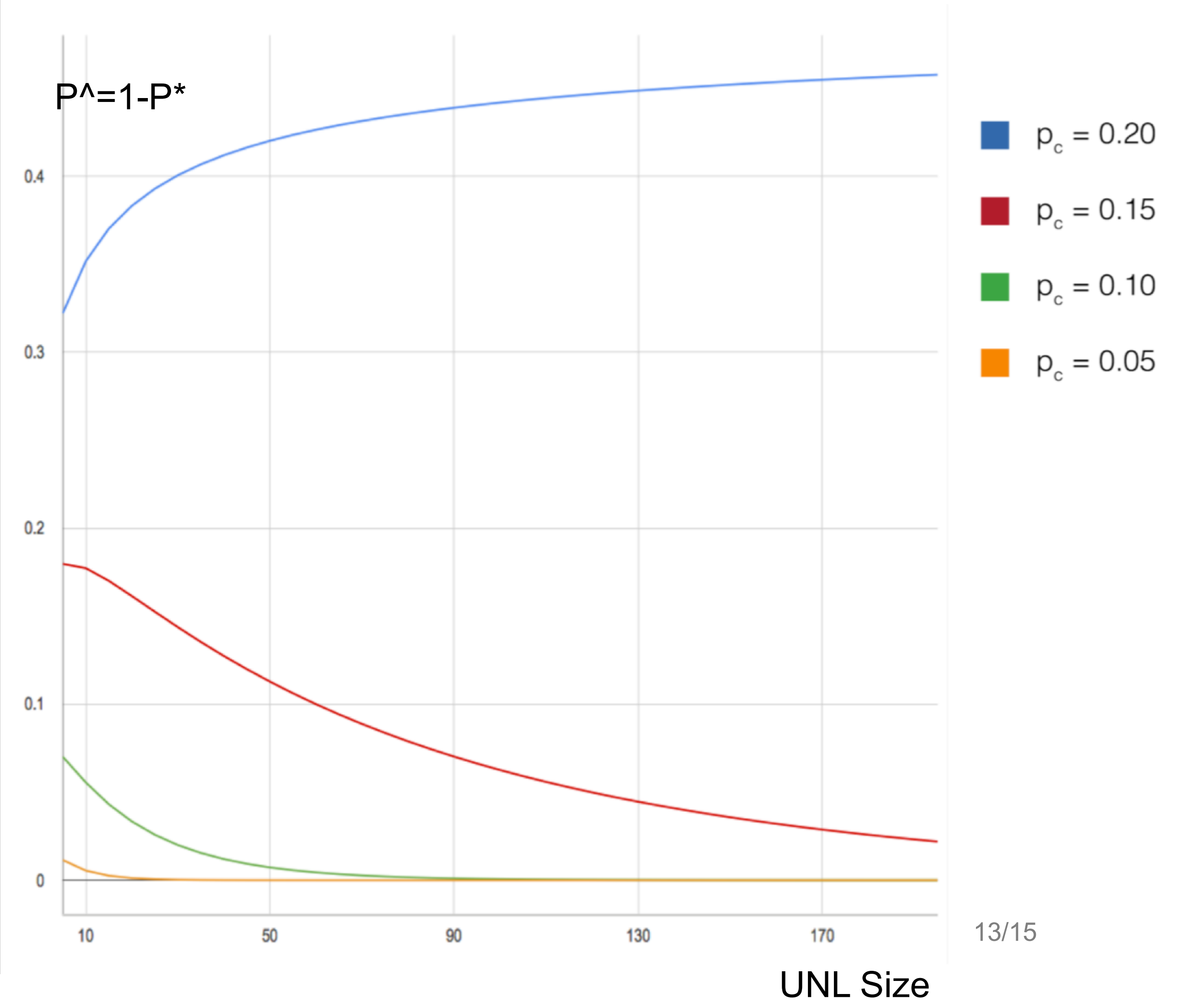
The protocol will maintain correctness if **f<=(n-1)/5**.

Pc: the probability that servers in the UNL will be fraudulent

P*: the probability of correctness

$$p^* = \sum_{i=0}^{\lceil(\frac{n-1}{5})\rceil} \binom{n}{i} p_c^i (1-p_c)^{n-i}$$

To achieve correctness:
Make sure Pc is smaller than 20%
Use a large UNL



Agreement

Correctness cannot guarantee agreement.

Correctness: no malicious transactions
Agreement: maintain a single global truth set of txns

The Requirement on the UNL Size:
Size(UNL) > 0.2*N

The Requirement on the connectivity:

$$|UNL_i \cap UNL_j| \geq \frac{1}{5}\max(|UNL_i|, |UNL_j|) \forall i, j$$

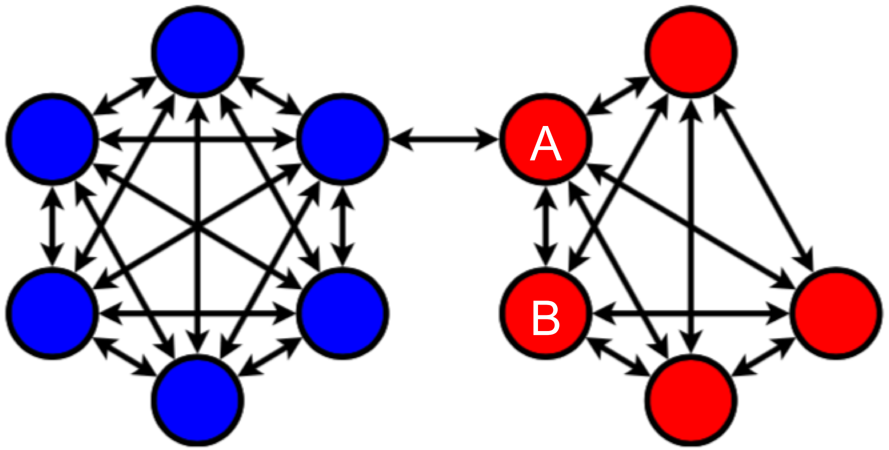


Figure 4: Disjoint UNL

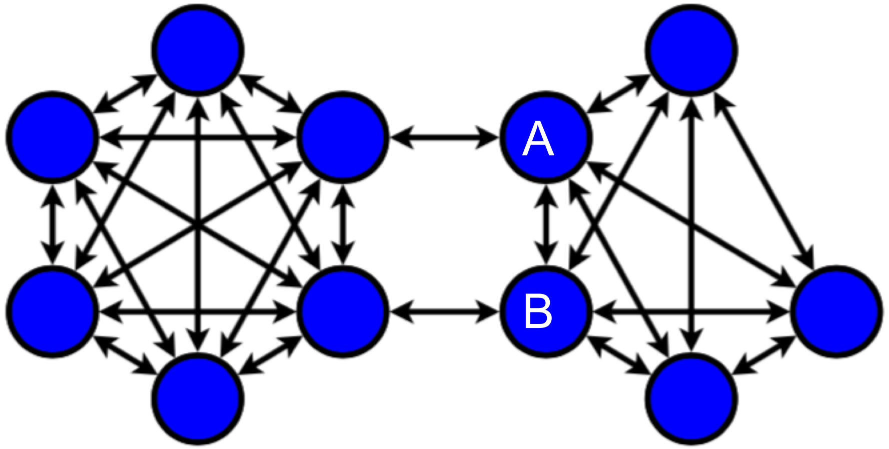


Figure 6: UNL cliques in consensus

https://arxiv.org/pdf/1802.07242.pdf

Utility & Conclusion

To make sure consensus is reached in finite time.
High latency nodes will be removed from all UNLs.

A default UNL is provided to minimize Pc.

A network split function algorithm is employed to avoid a fork in the network.

Can tolerate only (n-1)/5 Byzantine failures.

Utilizes collectively-trusted subnetworks within the whole network.

A fast and low-cost distributed payment consensus algorithm.