SEC-205: Distributed Ledger and Blockchain

Lecture 4: Consensus Mechanisms

Instructed By:

Dr. Charnon Pattiyanon

Assistant Director of IT and Instructor
CMKL University

Artificial Intelligence and Computer Engineering (AICE) Program

Today’s Agenda

- In today’s lecture, we will explore and learn about:
    - Recapitulation of Byzantine Generals Problem.
    - Consensus Protocol for Byzantine Broadcast
    - Attacks for Consensus Protocols
    - Proof-of-Stake

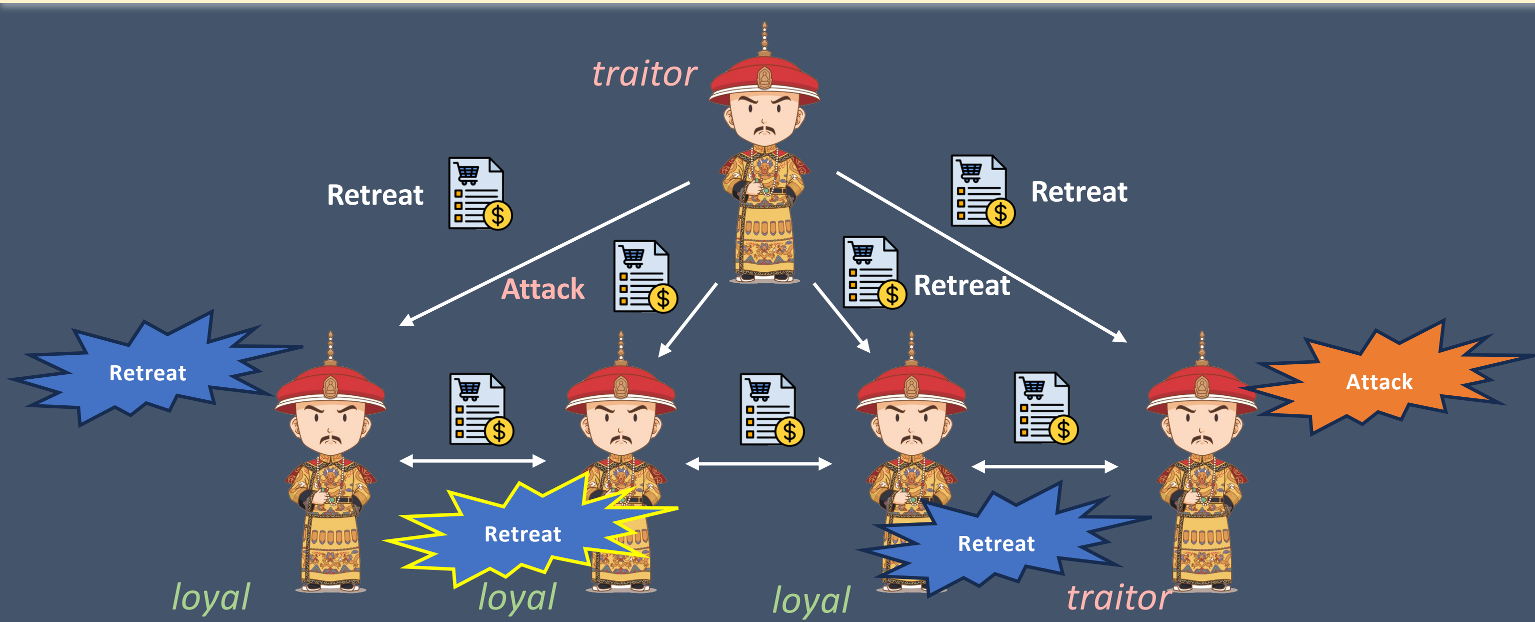
Recap of the Byzantine Generals Problem

- Byzantine Generals Problem was introduced by Lamport et al., 1982, to demonstrate the problem of reaching consensus between distributed entities.
- **Problem Statement:**

- There are n generals (where n is fixed), one of which is the ***commander***.
- Some generals are *loyal*, and some of them can be *traitors* (including the commander).
- The commander sends out an order that is either ***attack*** or ***retreat*** to each general.
- If the commander is *loyal*, it sends the same order to all generals.
- All generals take an action after some time.

Recap of the Byzantine Generals Problem

- **Goals:**
  - **Agreement:** ***No*** two *loyal* generals take ***different*** actions.
  - **Validity:** If the commander is *loyal*, then all *loyal* generals must take the action ***suggested by the commander***.
  - **Termination:** All *loyal* generals must eventually take some action.



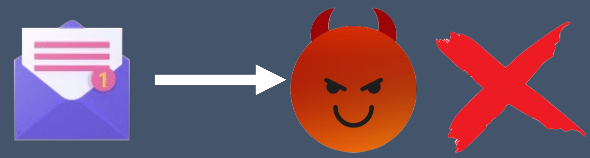
From Generals to Nodes

- Solutions to the Byzantine Generals Problem is a consensus protocol.
- In the network environment, when we modelling the consensus protocol:
  - Generals → Nodes
  - Commander → Leader
  - Loyalty → Honest, Traitor → Adversary
    - What can the adversary nodes do?

Adversary in Consensus Protocols

- The *adversary* can corrupt nodes, after which they are called adversarial.
  - **Crash faults** if the adversarial nodes do not send or receive any messages.

  - **Omission faults** if the adversarial nodes can selectively choose to drop or let through each messages sent or received.



  - **Byzantine faults (Byzantine Adversary)** if the adversarial nodes can deviate from the protocol arbitrarily.

Assumption on Adversarial Nodes

- We typically bound the adversary's power by assuming an upper bound (f) on the number of nodes (n) that can ever be adversarial.
  - E.g. $f < n$, $f < \frac{n}{2}$, $f < \frac{n}{3}$, …

Consensus Protocol for Byzantine Broadcast

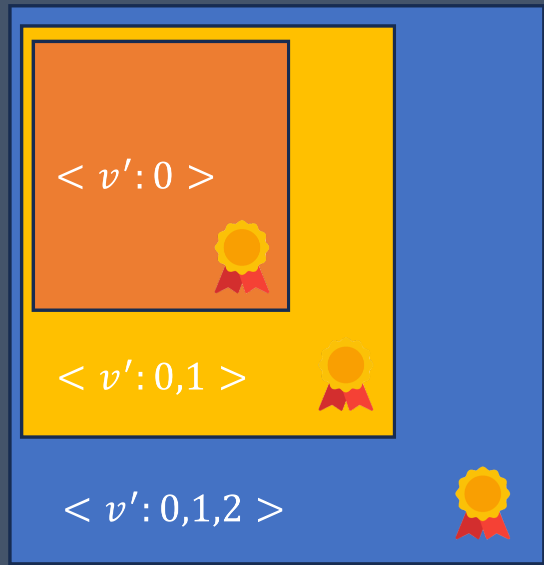
- There are n nodes (where n is fixed), one of which is the leader.
- For a public f, a subset of f nodes is adversarial, and all other nodes are honest.
- The leader has an input value 0 or 1.

Byzantine Broadcast Constraints:

- **Agreement: No two** honest nodes output different values. (Even when the leader is adversarial!)
- **Validity:** When the leader is honest => All the honest nodes output the value input from the leader.
- **Termination:** All honest nodes eventually output some value.

Consensus Protocol for Byzantine Broadcast

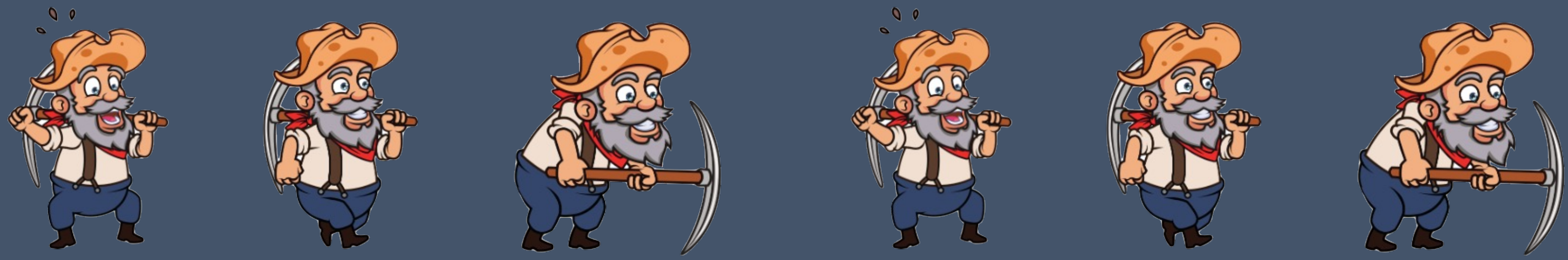
- Denote the nodes by the indices $i = 0,1,2, \ldots, n$
- Node 0 is the leader. Let v denotes its value.
- Let V_i denote the set of values received by node i.
- Time moves in *lock-step*.



- Let $< v': i >$ denotes the value v' signed by node i.
- Let $< v': i, j, k, \ldots, y, z >$ denotes a signature chain signed by nodes $i, j, k, \ldots, y, z$.
  - Recursive Definition: $< v': i, j, k, \ldots, y,\ z > = \ll v': i, j, k, \ldots, y >: z >$

Consensus in the Internet Setting

- How to select nodes to participate in the consensus protocol?

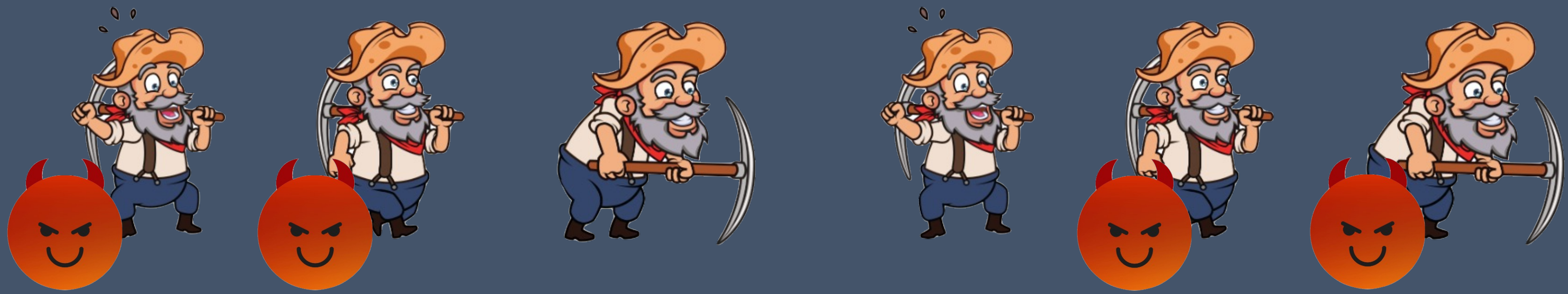


- **Two variants:**
  - Permissioned: There are *fixed set of nodes* that can participate in the consensus.
  - Permissionless: Anyone is *free to join* the protocol at any time.

Question: Can we accept any node that has a signing key to participate in consensus?

Consensus in the Internet Setting

- How to select nodes to participate in the consensus protocol?



- In a **Sybil Attack**, a single adversary impersonates many different nodes, outnumbering the honest nodes and potentially disrupting consensus.

Consensus Protocols in Sybil Resistance

- Consensus protocols with Sybil resistance are typically based on a bounded resource:

	Resource Dedicated to the Protocol	Some Examples Blockchain
Proof-of-Work	Total Computational Power	Bitcoin, PoW Ethereum, etc.
Proof-of-Stake	Total Number of Coins	Algorand, Cardano, Cosmos, PoS Ethereum, etc.
Proof-of-Space/Time	Total Storage Across Time	Chia, Filecoin, etc.

How does Proof-of-Work prevent Sybil attacks?

We assume that the adversary controls a small fraction of the scarce resource!
Resource gives the power to influence the protocol.
Adversary has less influence than honest nodes.

Quick Introduction to Proof-of-Stake

In a Proof-of-Stake protocol, nodes lock up (i.e., stake) their coins in the protocol to become eligible to participate in consensus.

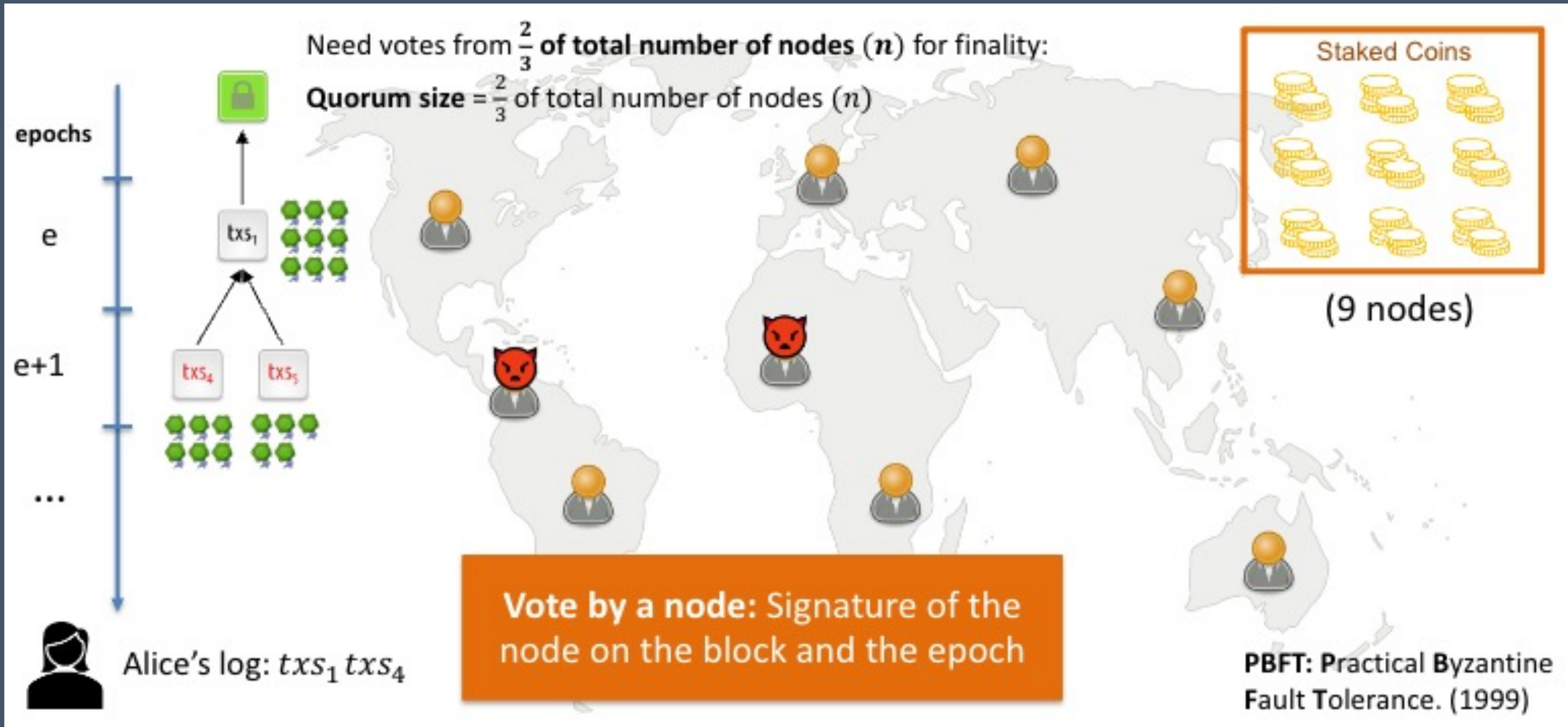
The more coins staked by a node...

- **Higher** the probability that the node is elected as a leader.
- **Larger** the weight of that node's actions.

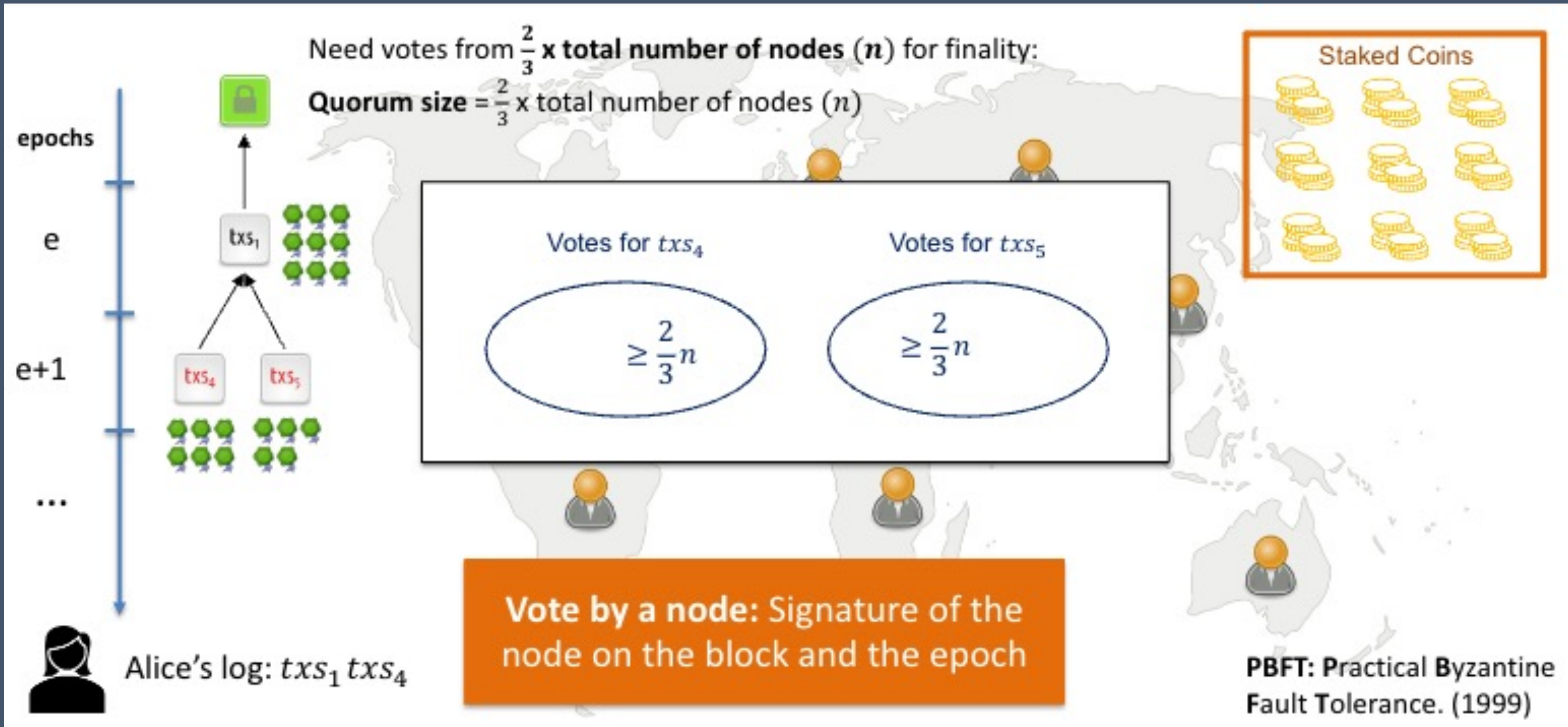
If a node is caught doing an adversarial action (e.g., sending two values), it can be punished by burning its locked coins (stake)!
This is called *slashing*.

Thus, in a Proof-of-Stake protocol, nodes can be held *accountable* for their actions (unlike in Bitcoin, where nodes do not lock up coins).

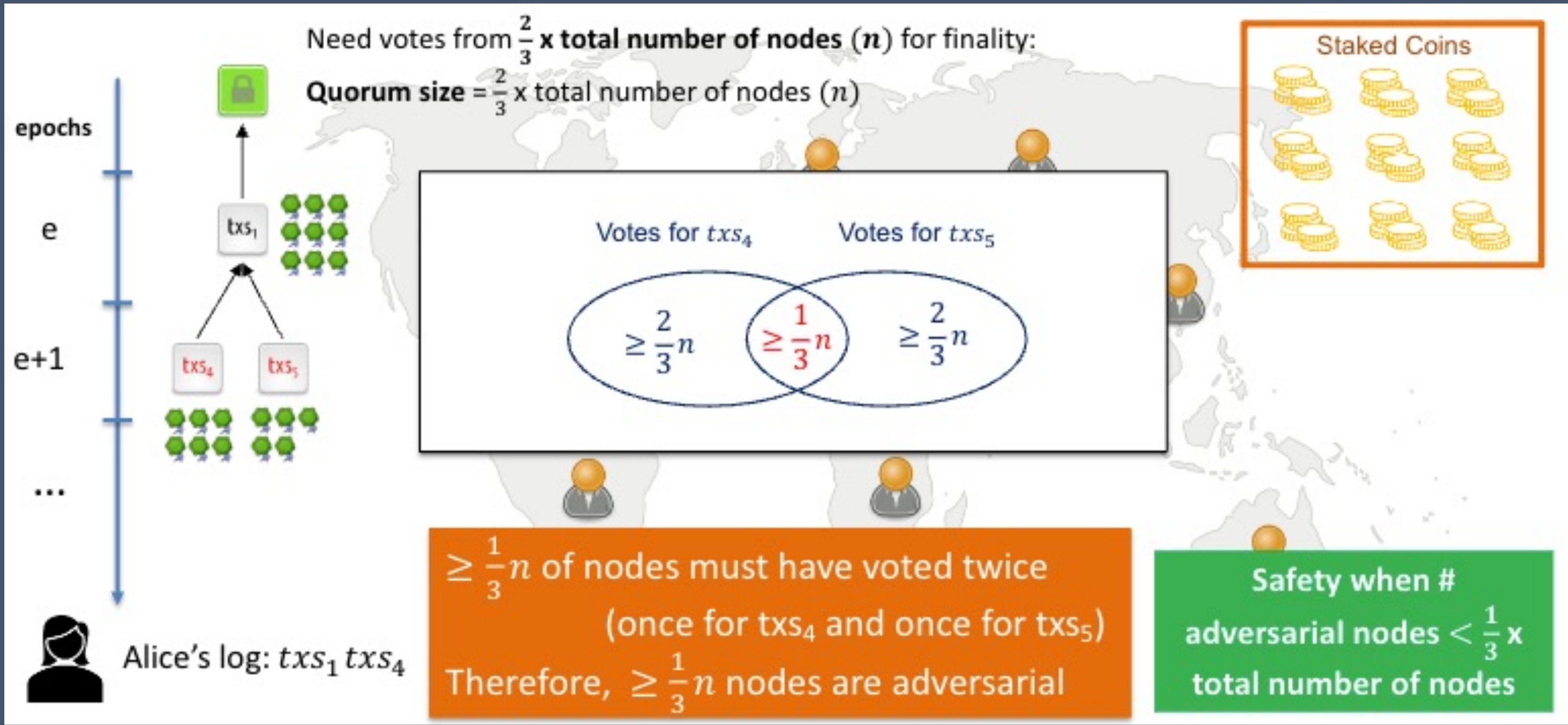
A Simple (PBFT-Style) Proof-of-Stake Protocol



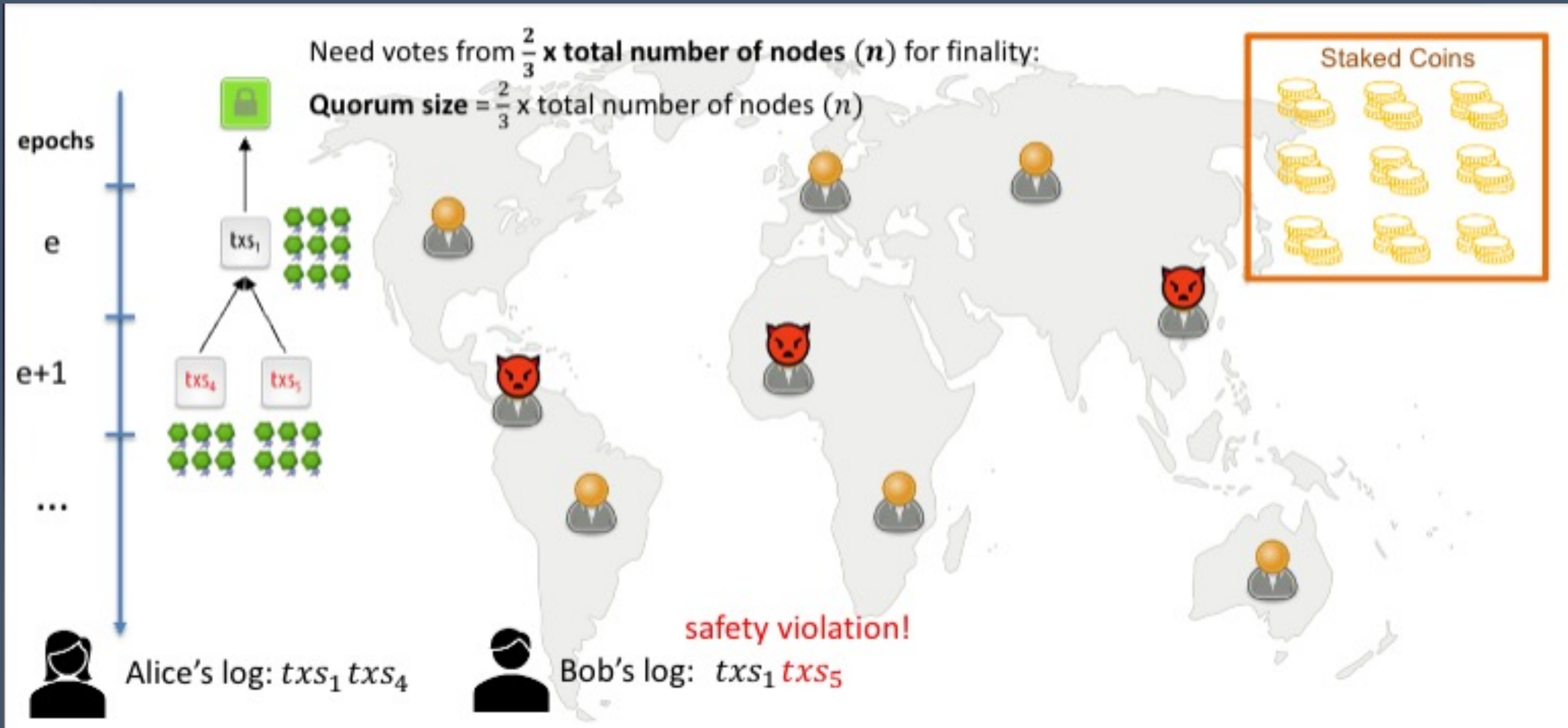
A Simple (PBFT-Style) Proof-of-Stake Protocol



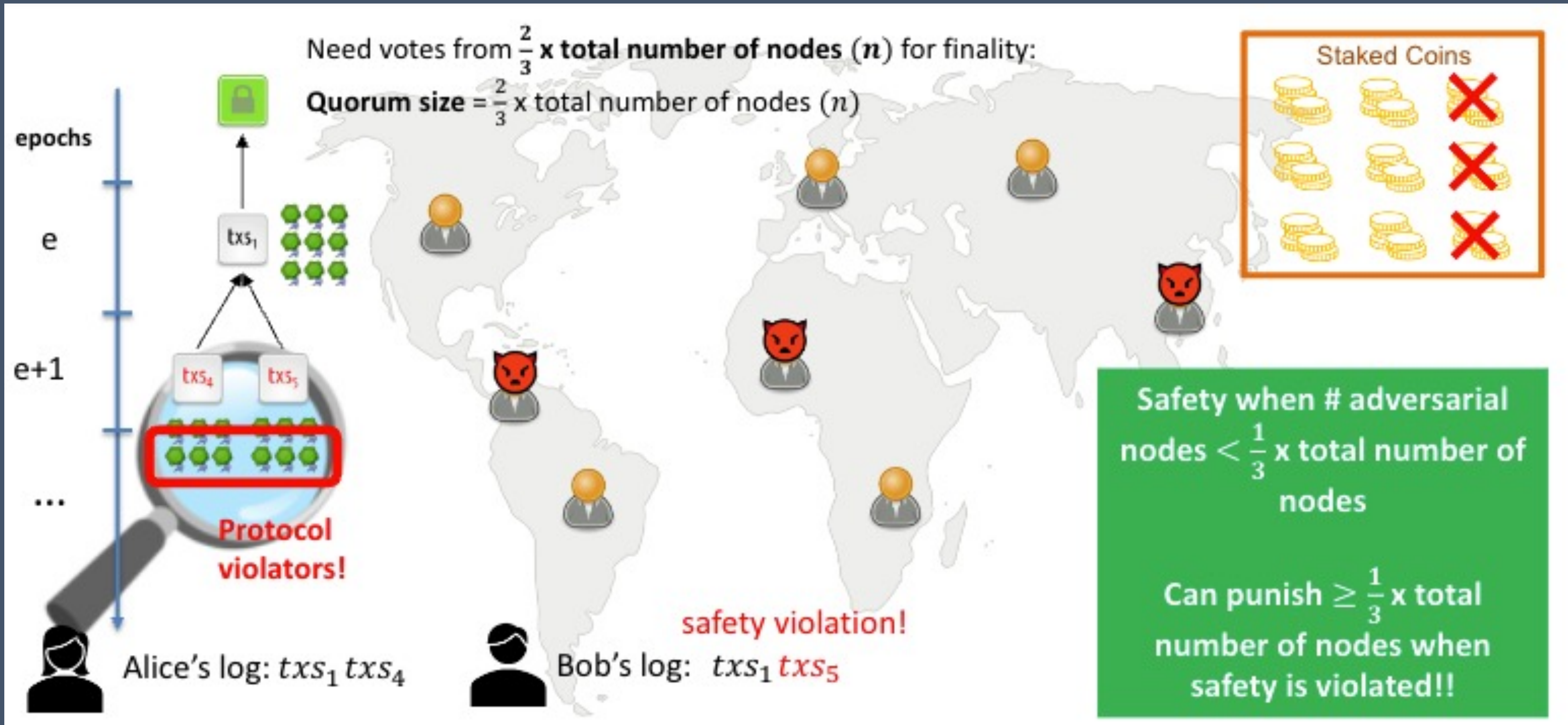
A Simple (PBFT-Style) Proof-of-Stake Protocol



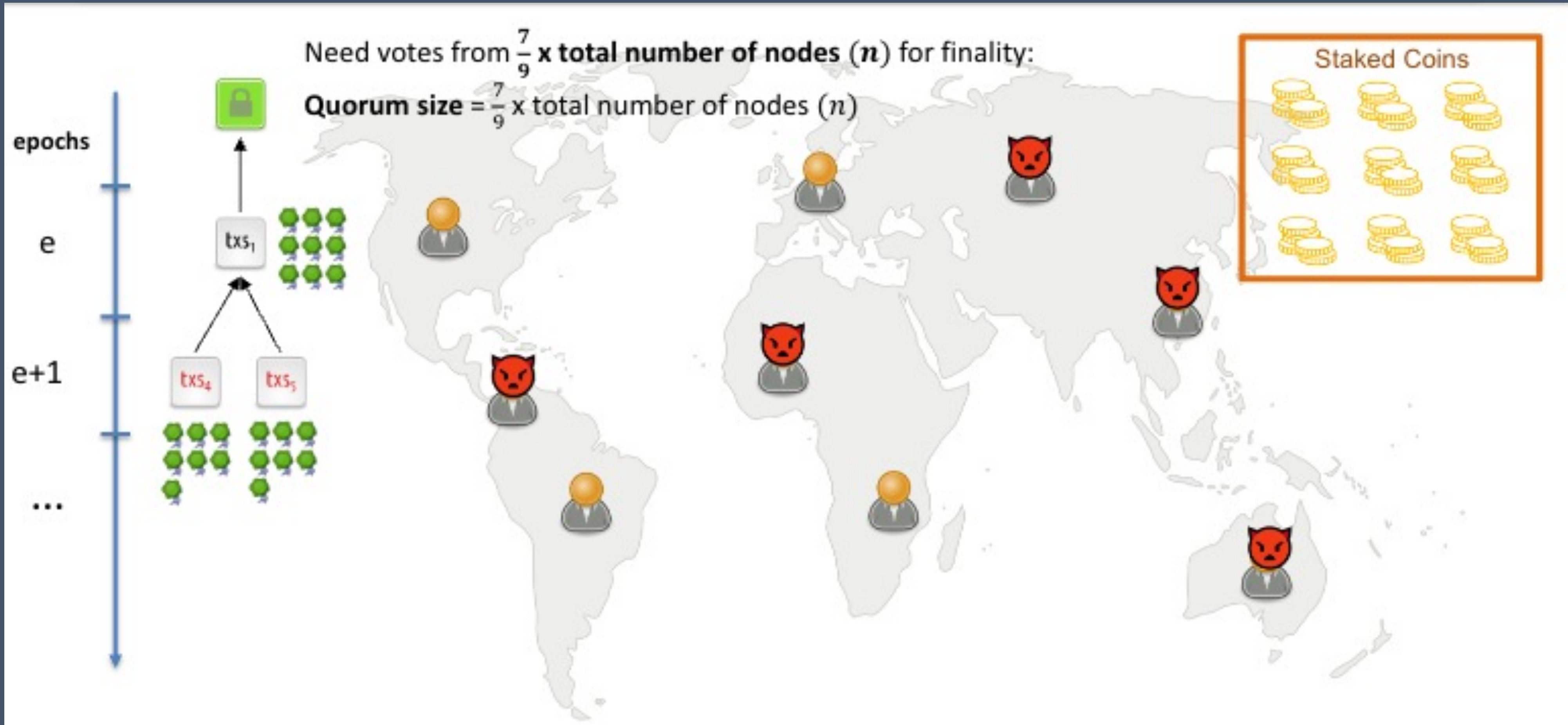
A Simple (PBFT-Style) Proof-of-Stake Protocol



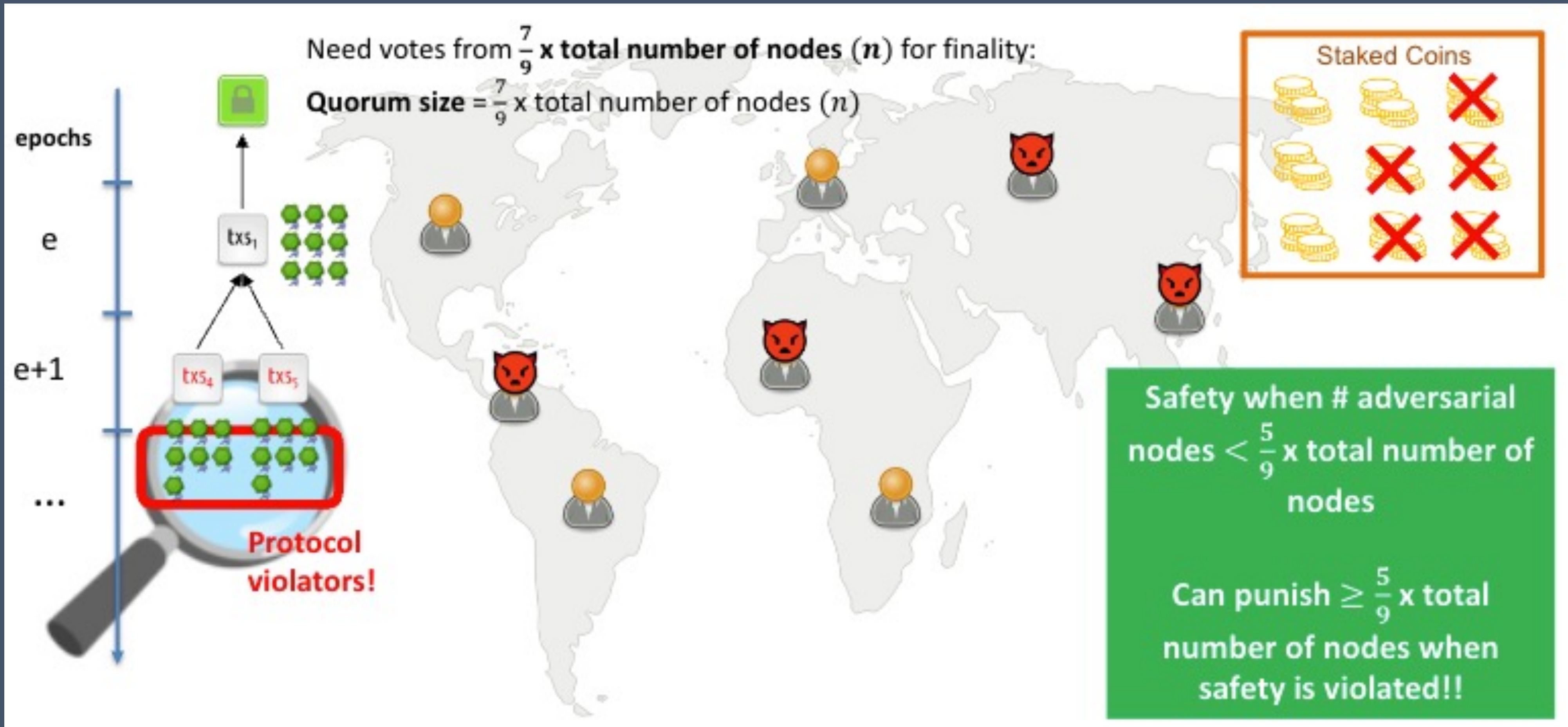
A Simple (PBFT-Style) Proof-of-Stake Protocol



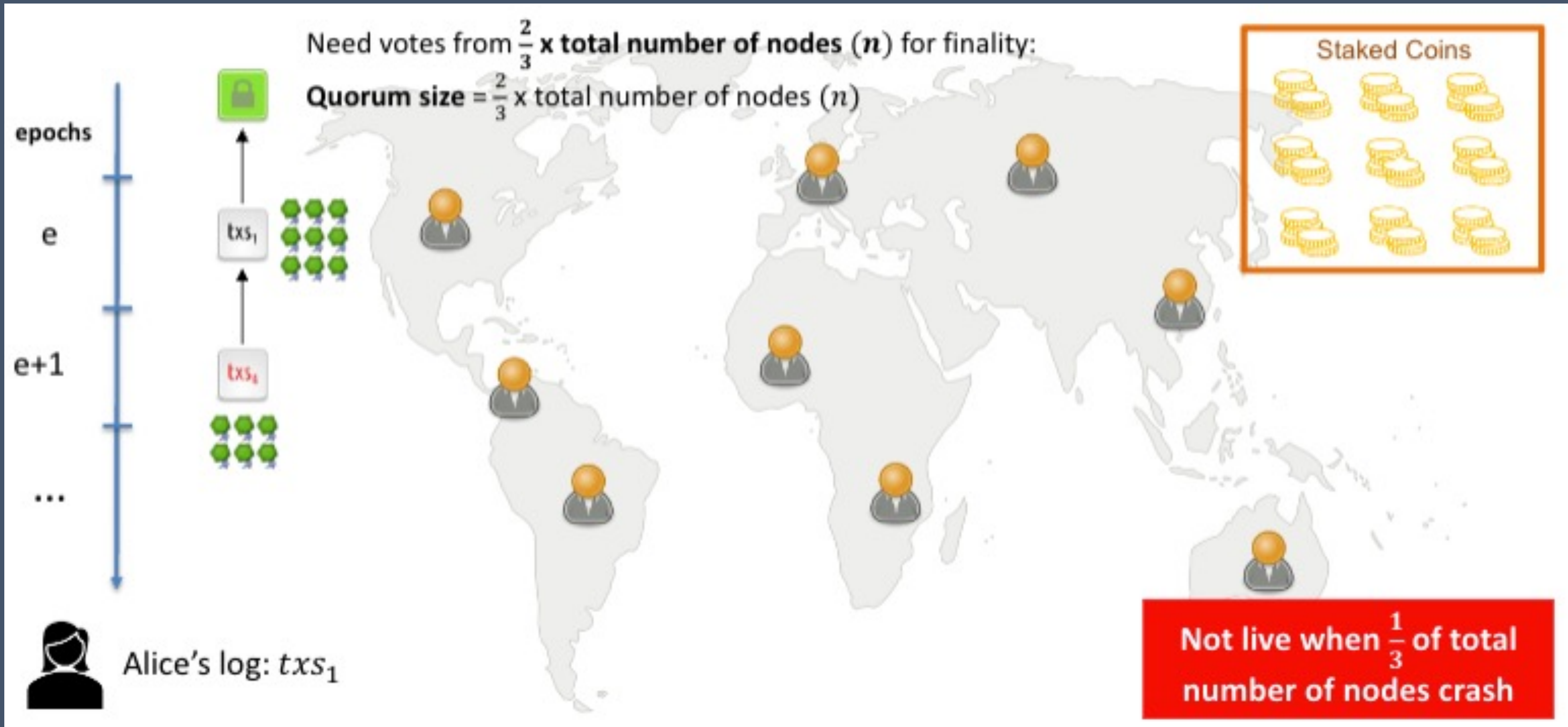
A Simple (PBFT-Style) Proof-of-Stake Protocol



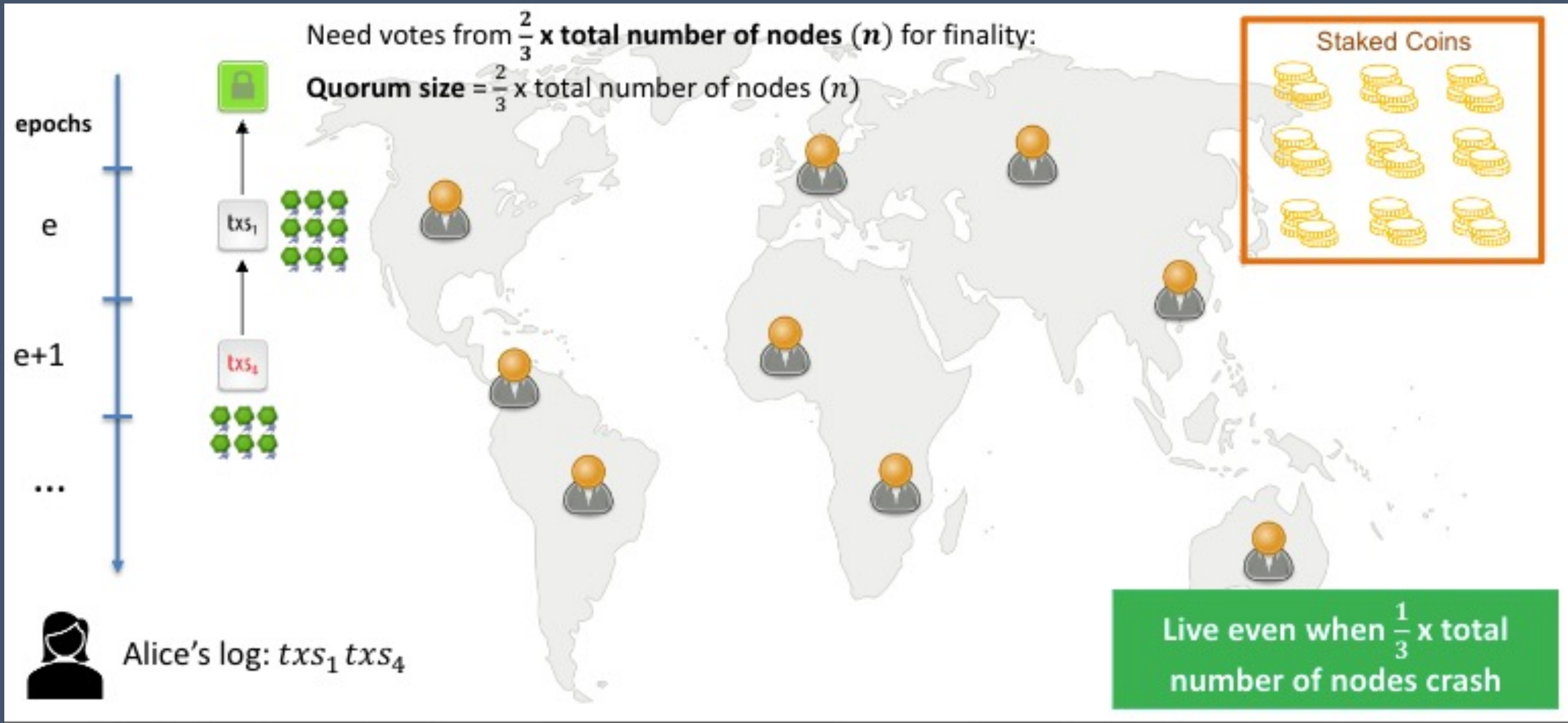
A Simple (PBFT-Style) Proof-of-Stake Protocol



A Simple (PBFT-Style) Proof-of-Stake Protocol



A Simple (PBFT-Style) Proof-of-Stake Protocol



Today's Agenda

- Upon successful of today's lecture, we have learned about:
  - Recapitulation of Byzantine Generals Problem, including the problem statement, the network node analogy.
  - Consensus Protocol for Byzantine Broadcast, including the common constraints, and the procedure to ensure consensus.
  - Attacks for Consensus Protocols, such as Sybil attacks.
  - Proof-of-Stake as a simple PBFT-style PoS protocol.

End of the lecture! 🥳

Please feel free to ask any questions.

If you need further discussion, please contact me:

- Email me at charnon@cmkl.ac.th
- Appoint me for 1-on-1 discussion during the office hours.