Scalable Byzantine Fault Tolerance Protocol

Implemented by Aadarsh Venugopal, Bharath Kinnal, Yathesh Lekkalapudi and Sadaf Arshad
Implementation of **SBFT** protocol

- **SBFT** has two main protocols
  - Agreement
  - View change

- **SBFT** uses collectors to ensure linear communication in agreement protocol

- Agreement protocol has three phases - linear **PBFT**

- View change protocol has two phases
  - Leader election: Executed during view change trigger
  - View change: Assigning new leader as primary
What we have implemented so far

Agreement Protocol

- It is implemented as a state machine with the below states
  - Pre-prepare
  - Commit
  - Execute

- Modules in agreement protocol
  - Replica State Machine
  - Blockchain

Fig. 1. Schematic message flow for $n=4, f=1, c=0$. 
What we have implemented so far

Types of Replicas in the network

- **Primary**
  - This node is responsible for initiating the client transaction

- **C-Collector**
  - This replica stores a message queue containing the sign share from all replicas
  - If the C-Collector receives $2f+c+1$ messages, it will send a commit proof to all replicas

- **E-Collector**
  - This replica stores a message queue containing the sign share messages from the replicas
  - When the replica receives $f+1$ messages, it creates a full execute proof and sends it to other replicas and the client for acknowledgement

- **Replica**
What we have implemented so far

View Change

- View change triggered by a replica when primary is found to be faulty
  - Timeout is set for making requests and when primary doesn’t respond view change is triggered
- Each replica votes for new leader; votes tallied to decide new leader
  - Leader chosen among non faulty replicas only
- Each replica sends request to make new leader as primary
  - Primary updated in network configuration file