# Consensus Protocols

## Practical Byzantine Fault Tolerance
## - pBFT -

Creat de Buhai Darius, Grupa 334

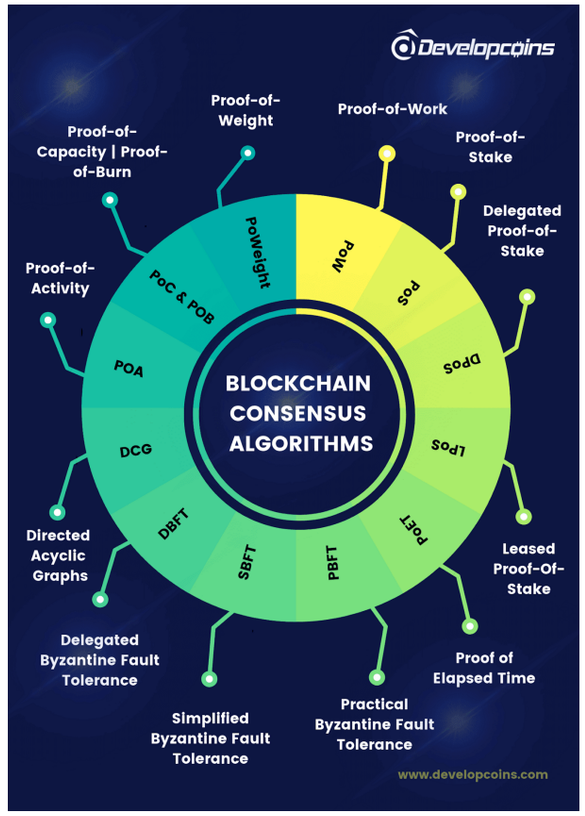
# Motivatie

## Consensus Protocols

- Crypto monedele si alte aplicatii ale blockchain-ului (**descentralizate**) au la baza Consensus Protocols.


- Retelele de tip blockchain au implementate diferite metode de Consensus Protocols, dintre care **Proof of Work** si **Proof of Stake** fiind cele mai utilizate.

- Principala problema intampinata la **Proof of Work** o reprezinta puterea de procesare necesara generarii numerelor de validare.



# Motivatie

## Proof of Stake

- Pentru a se evita folosirea resurselor in mod excesiv (pentru generarea numerelor de validare - Proof of Work), putem folosi **Proof of Stake**, ce se bazeaza pe alegerea unui validator care aproba sau nu blocul ce urmeaza a fi introdus in blockchain.

# Proof of Stake

## Practical Byzantine Fault Tolerance

- Practical Byzantine Fault Tolerance a fost introdus la sfarsitul anilor 90 de catre Barbara Liskov si Miguel Castro.
- Acest protocol ajuta retelele distribuite sa ajunga la un consens.
- Obiectivul principal al pBFT este de a proteja reteaua impotriva erorilor de sistem

# Practical Byzantine Fault Tolerance

## Avantaje

- Eficienta Consumului de energie
- Finalitatea Tranzactiilor
- Reward-uri mici

# Practical Byzantine Fault Tolerance

## Mod de functionare

- Sistemele **pBFT** pot functiona doar in cazul in care numarul maxim de noduri malitioase nu este mai mare sau egal decat **1/3** din toate nodurile din sistem.

- Cu cat numarul de noduri din sistem creste, cu atat securitatea acestuia devine mai buna.

- Nodurile din pBFT sunt impartite in noduri **primare** si noduri de **backup**.

# Practical Byzantine Fault Tolerance

## Mod de functionare

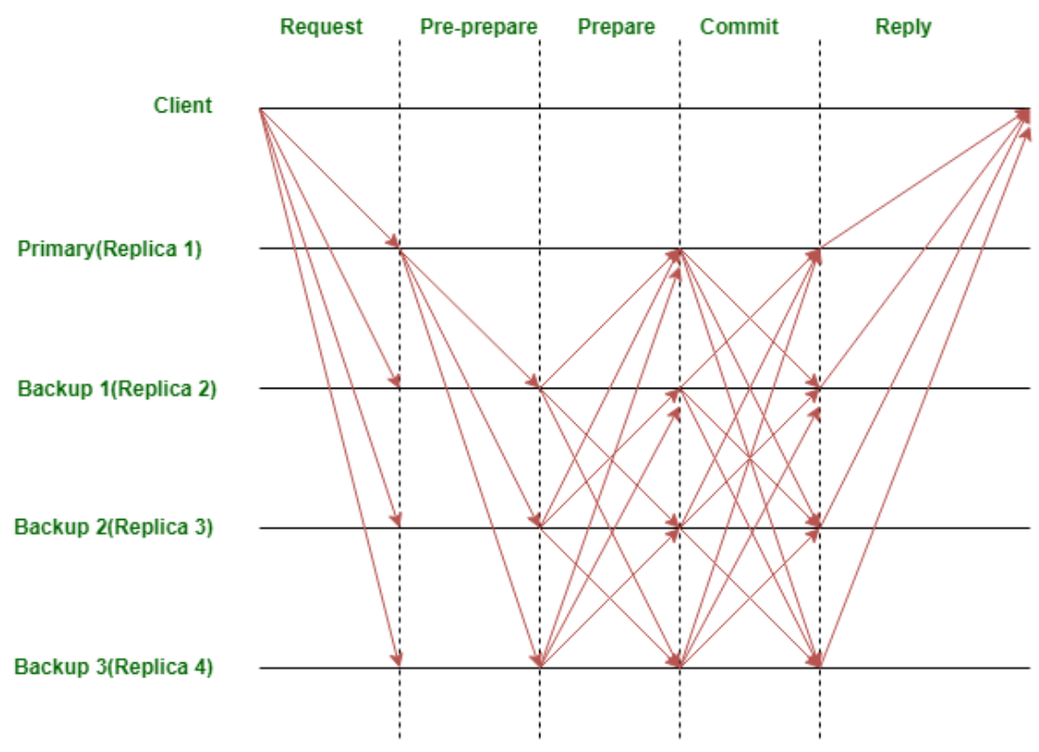
- **pBFT** este impartit in 4 faze de functionare:

- 1. Clientul trimite un request catre nodul primar (selectat aleator).

- 2. Nodul primar trimite request-ul catre toate nodurile de back-up.

- 3. Nodurile primare si de back-up ruleaza serviciul cerut si trimit un raspuns catre client.

- 4. Request-ul clientului este validat atunci cand acesta primeste m+1 raspunsuri de la noduri diferite din retea (cu acelasi rezultat), unde m = numarul maxim de noduri gresite permise.



# Practical Byzantine Fault Tolerance

## Alegerea nodului primar

- Nodul primar este schimbat la fiecare alegere (runda de consens pBFT) si poate fi substituit printr-un protocol de tip: **view change protocol**, daca un nod predefinit nu actioneaza in timp util.
- Daca este necesar, o majoritate de noduri 'oneste' pot vota inlocuirea nodului primar cu urmatorul nod.

# Practical Byzantine Fault Tolerance

## Probleme si Limitari

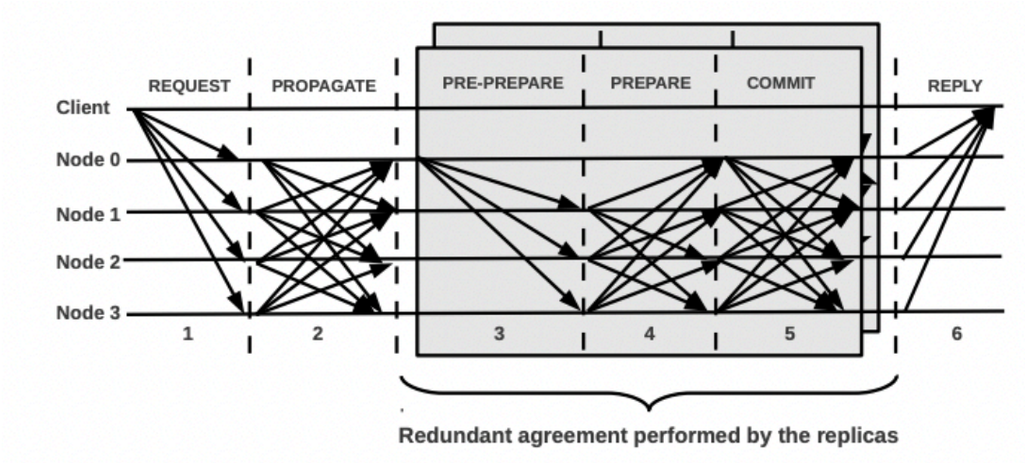
- **Scalabilitate**: atunci cand reteaua de noduri devine prea mare, acest protocol poate deveni ineficient.

- **Atacuri de tip Sybil**: Retelele de tip pBFT sunt susceptibile atacurilor Sybil, unde un singur nod creeaza sau manipuleaza majoritatea nodurilor din retea (sau cel putin > 1/3) si compromite securitatea sistemului.

# Practical Byzantine Fault Tolerance

## Variatii

- RBFT - Reduntant BFT




- ABsTRACTs
- Q/U
- HQ - Hybrid Quorum Protocol for BFT
- Adapt

# Practical Byzantine Fault Tolerance

## In practica

- **Zilliqa** foloseste o versiune optimizate de pBFT pentru a ajunge la consens legat de blocurile din blockchain.

- **Hyperledger Fabric** este un environment open-source pentru blockchain, hostat de Linux Foundation si foloseste o versiune permissioned de pBFT pentru platforma.

zilliqa

# Bibliografie


- https://www.section.io/engineering-education/blockchain-consensus-protocols/
- https://www.geeksforgeeks.org/practical-byzantine-fault-tolerancepbft
- https://crushcrypto.com/what-is-practical-byzantine-fault-tolerance/
- https://pakupaku.me/plaublin/rbft/5000a297.pdf (RBFT)