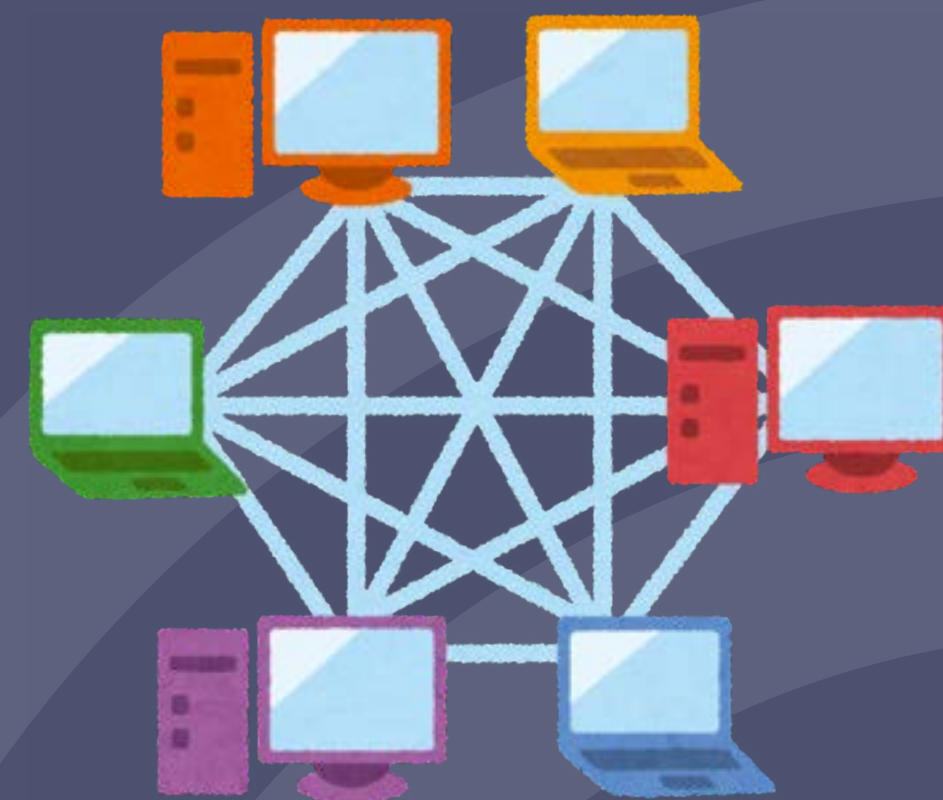


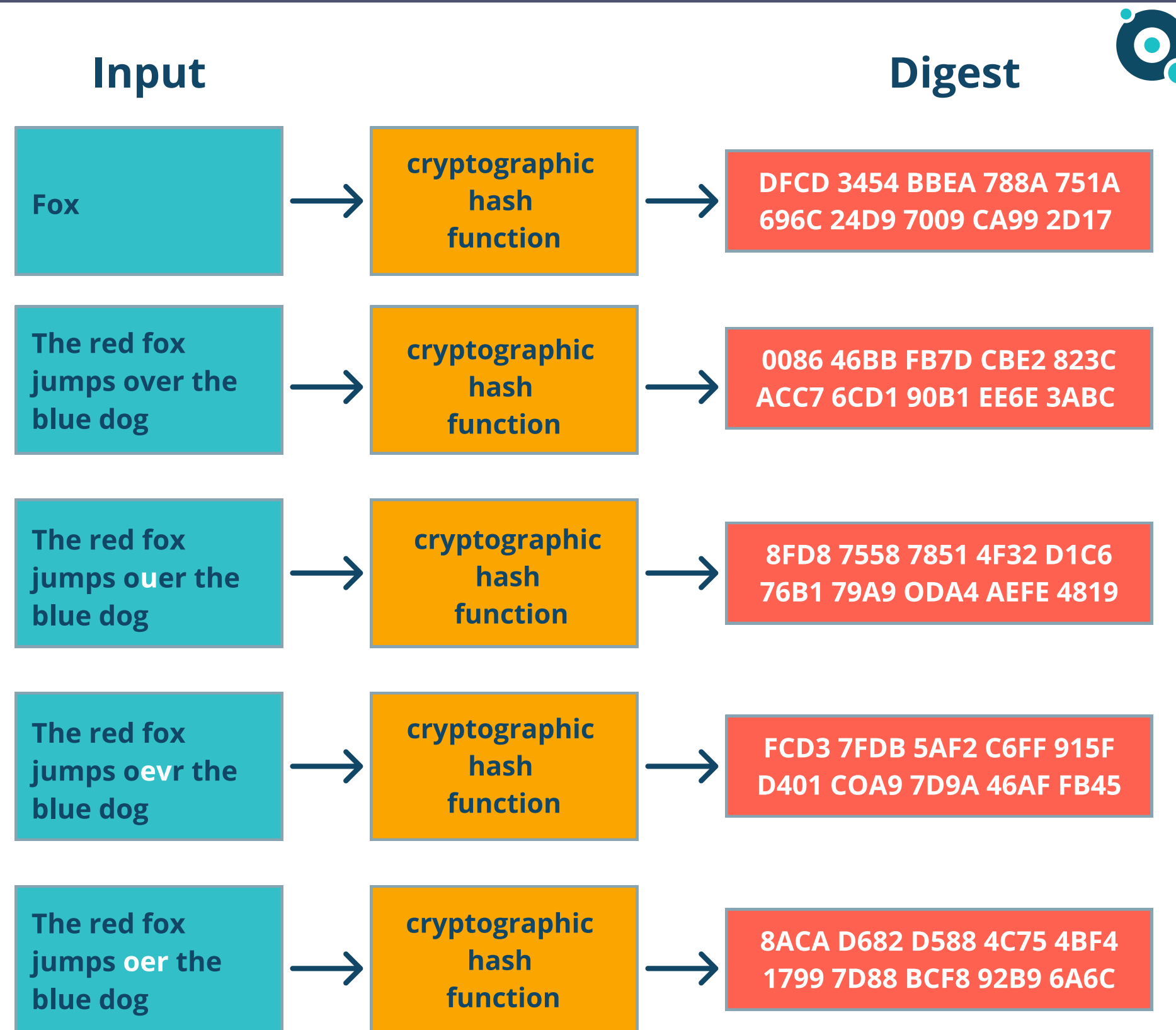
# SISTEMAS DISTRIBUÍDOS DESCENTRALIZADOS E DHTS



# DEFINIÇÃO DE DHT'S

- DHT (Distributed Hash Table): estrutura de dados distribuída.
- Funciona como uma tabela hash, mas espalhada em vários nós.
- Cada nó armazena parte das informações.
- Permite localizar dados de forma rápida, escalável e sem servidor central.

# DEFINIÇÃO DE DHT'S

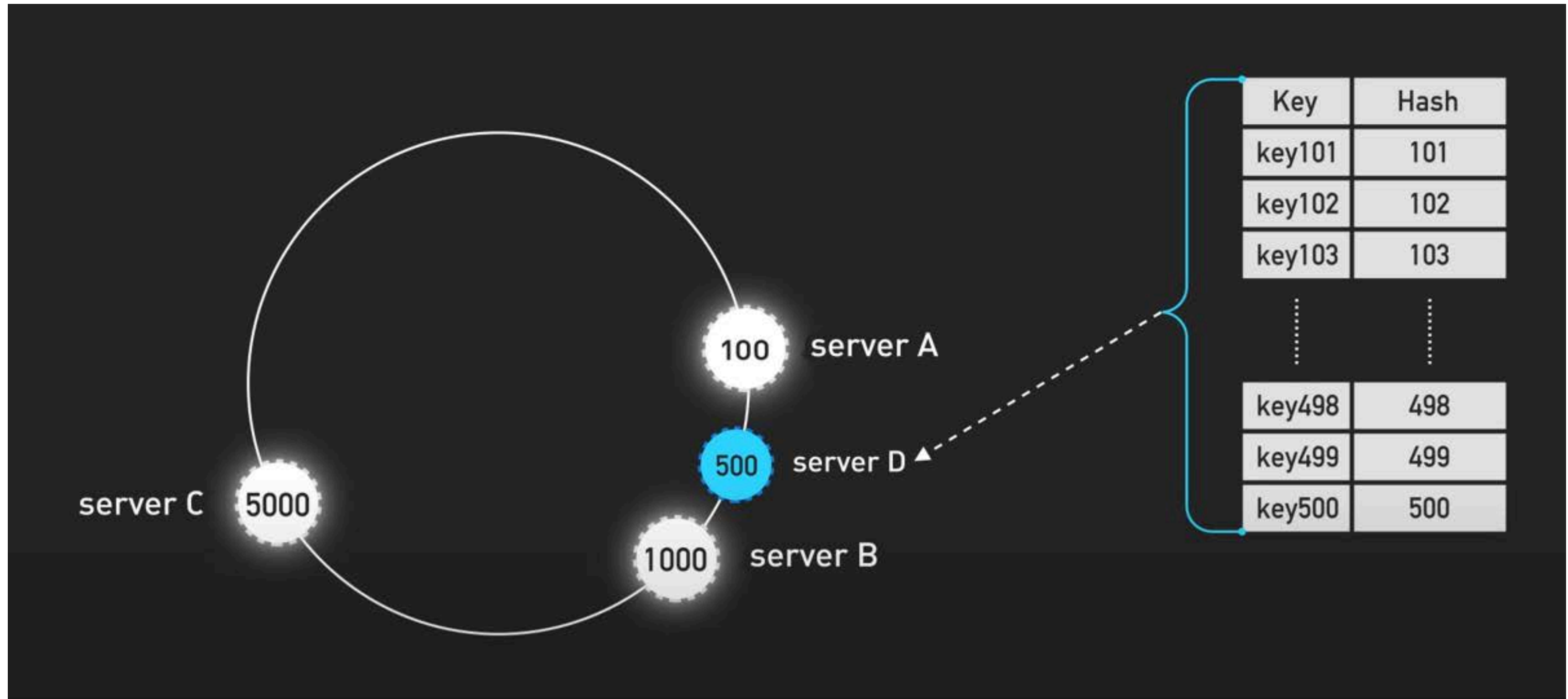


# ARQUITETURA CHORD

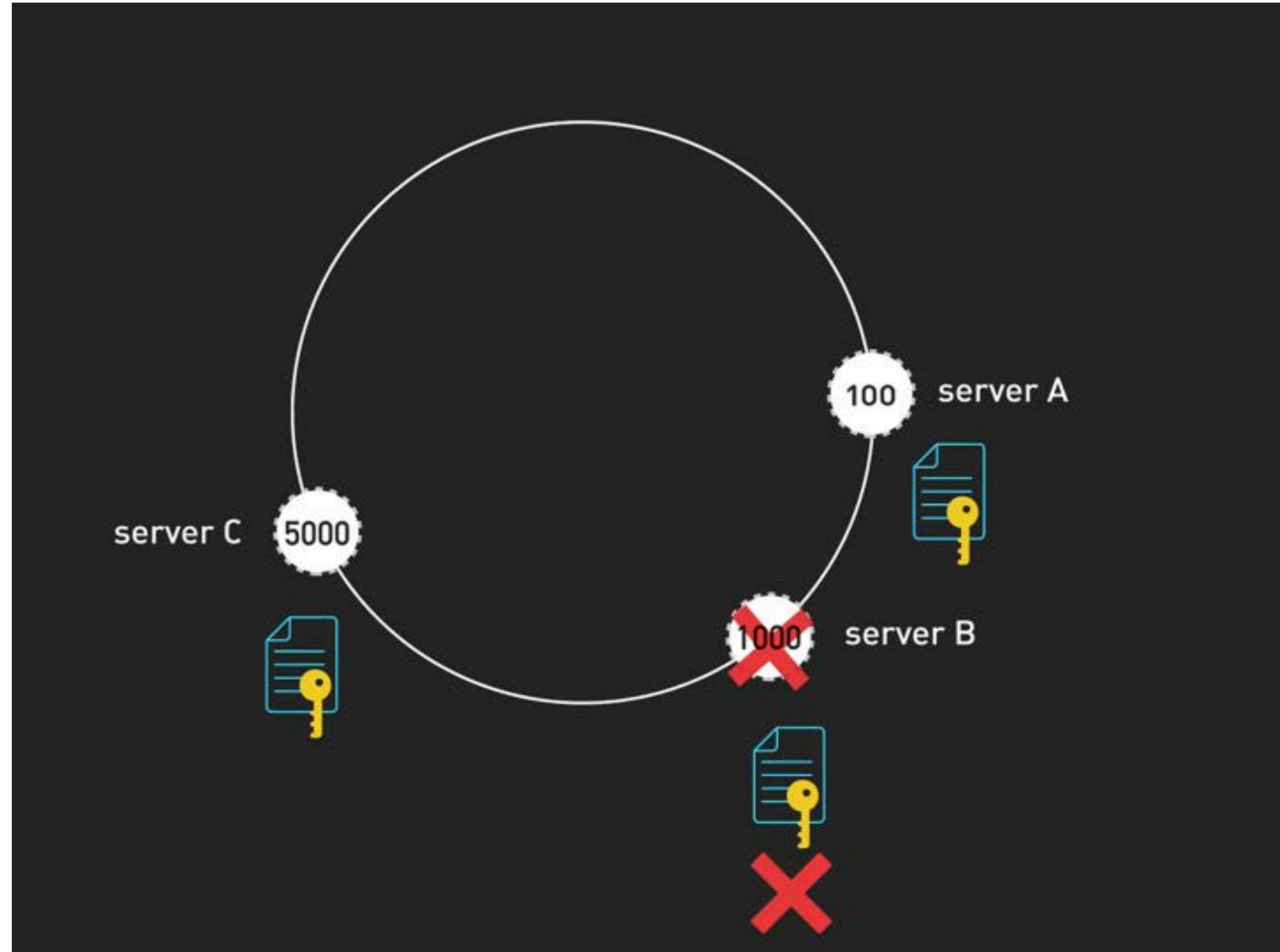
CHORD:

- Implementação de DHT baseada em anel lógico.
- Cada nó e cada chave recebem um ID via hash.
- As chaves são atribuídas ao nó sucessor no anel.

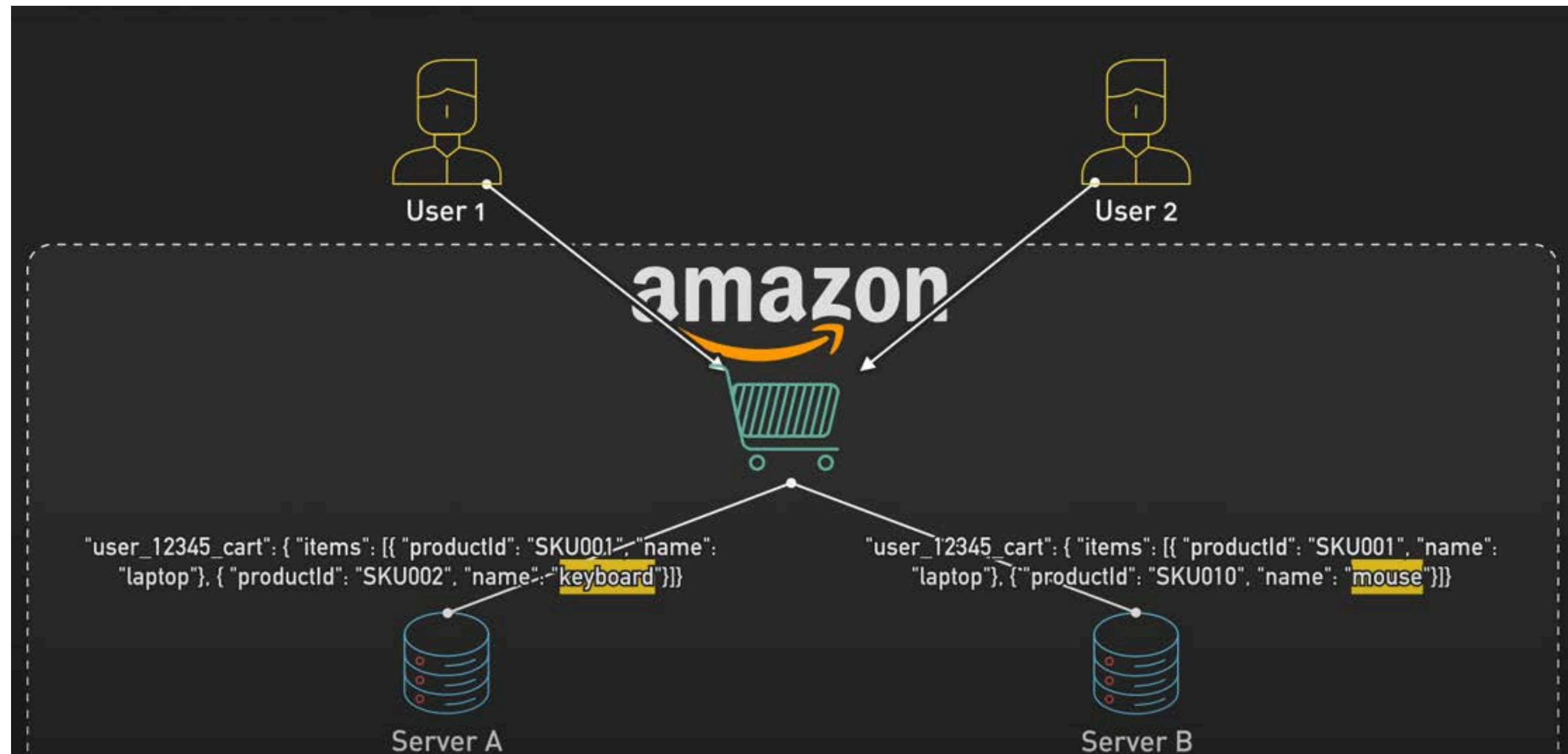
# ARQUITETURA CHORD



# ARQUITETURA CHORD



# ARQUITETURA CHORD





# VANTAGENS E LIMITAÇÕES

- O Teorema CAP entra diretamente na parte de vantagens e limitações de DHTs e de qualquer sistema distribuído descentralizado.





# TEOREMA CAP

O TEOREMA DIZ QUE EM UM SISTEMA DISTRIBUÍDO, NÃO PODE TER OS 3 ATRIBUTOS AO MESMO TEMPO. VOCÊ SÓ PODE GARANTIR 2 DE 3 (CONSISTENCY, AVAILABILITY, PARTITION TOLERANCE).

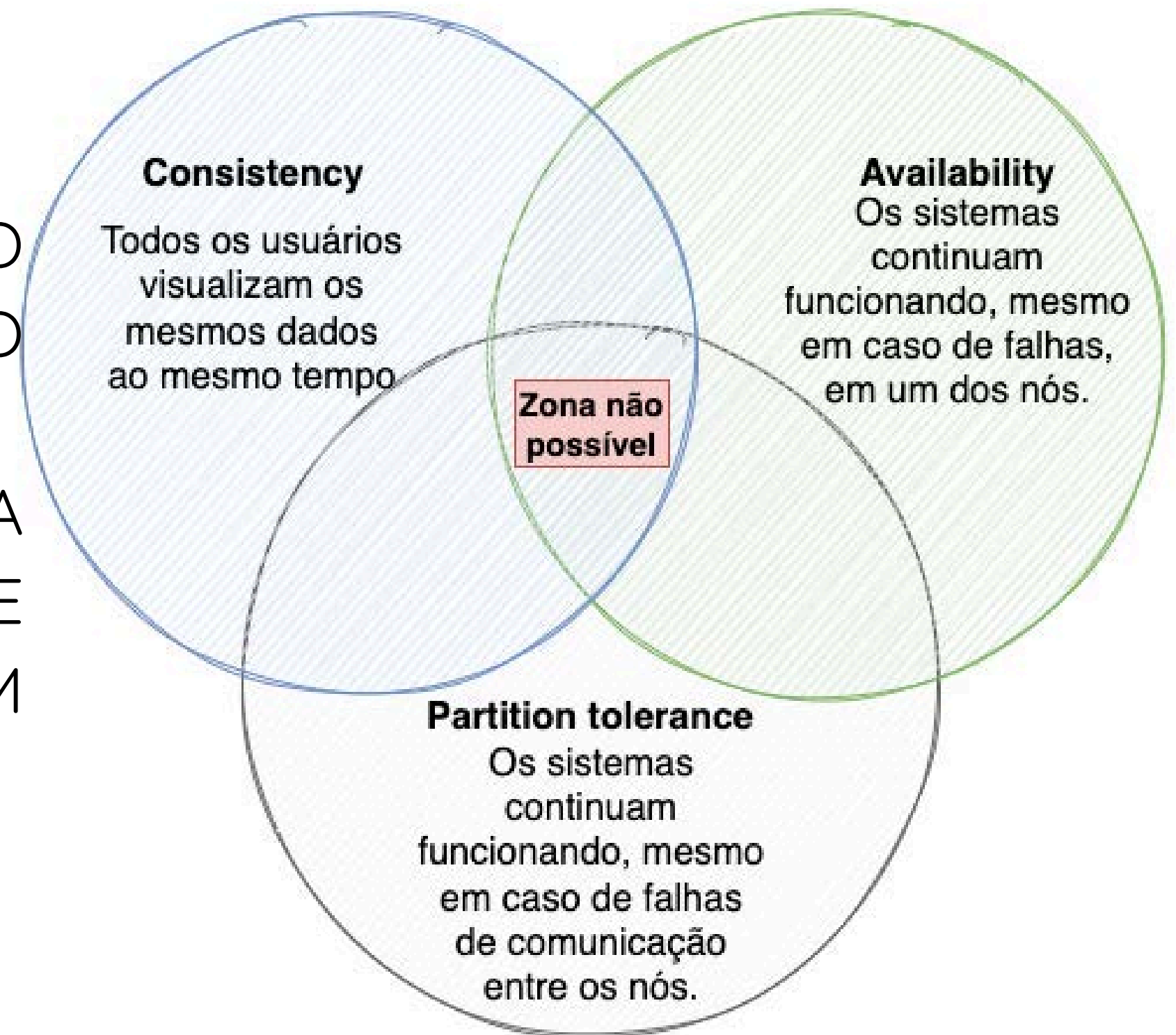
- CA (CONSISTÊNCIA + DISPONIBILIDADE) → SÓ FUNCIONA BEM SE NÃO HOUVER FALHA DE REDE.
- AP (DISPONIBILIDADE + PARTIÇÃO) → A REDE SEMPRE RESPONDE, MAS PODE RESPONDER COM DADOS DESATUALIZADOS.
- CP (CONSISTÊNCIA + PARTIÇÃO) → TODOS VEEM OS MESMOS DADOS, MAS ÀS VEZES VOCÊ NÃO CONSEGUE ACESSAR (SEM DISPONIBILIDADE).

# TEOREMA CAP

## AVAILABILITY (DISPONIBILIDADE)

SEMPRE QUE VOCÊ FAZ UM PEDIDO, O SISTEMA RESPONDE, MESMO QUE NÃO SEJA A RESPOSTA PERFEITA.

ISSO SIGNIFICA: MESMO QUE PARTE DA REDE CAIA, VOCÊ AINDA CONSEGUE ARMAZENAR E BUSCAR DADOS EM ALGUM NÓ.

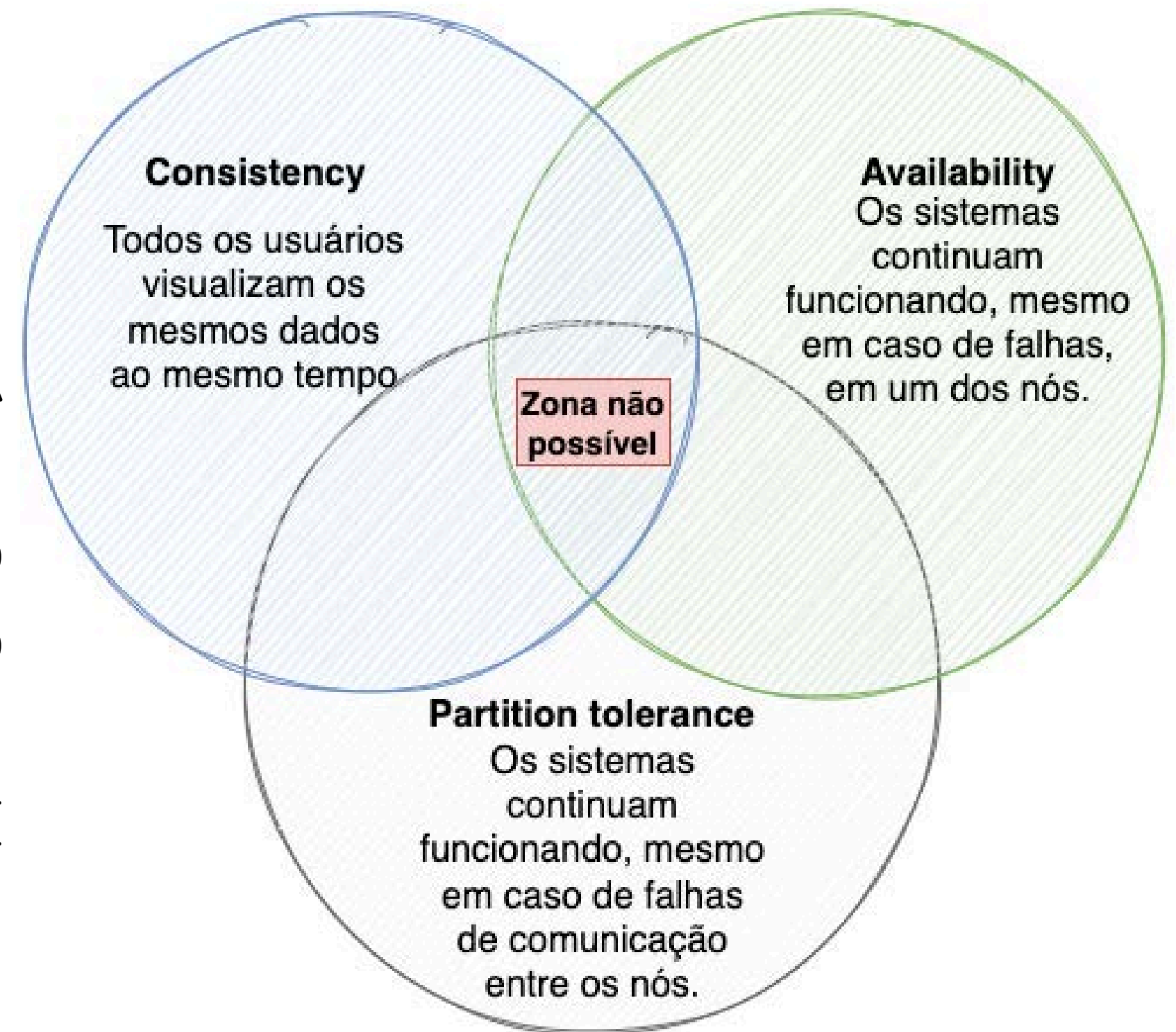


# TEOREMA CAP

## PARTITION TOLERANCE (TOLERÂNCIA A PARTIÇÕES)

A REDE PODE “QUEBRAR” EM PEDAÇOS E, MESMO ASSIM, O SISTEMA CONTINUA FUNCIONANDO.

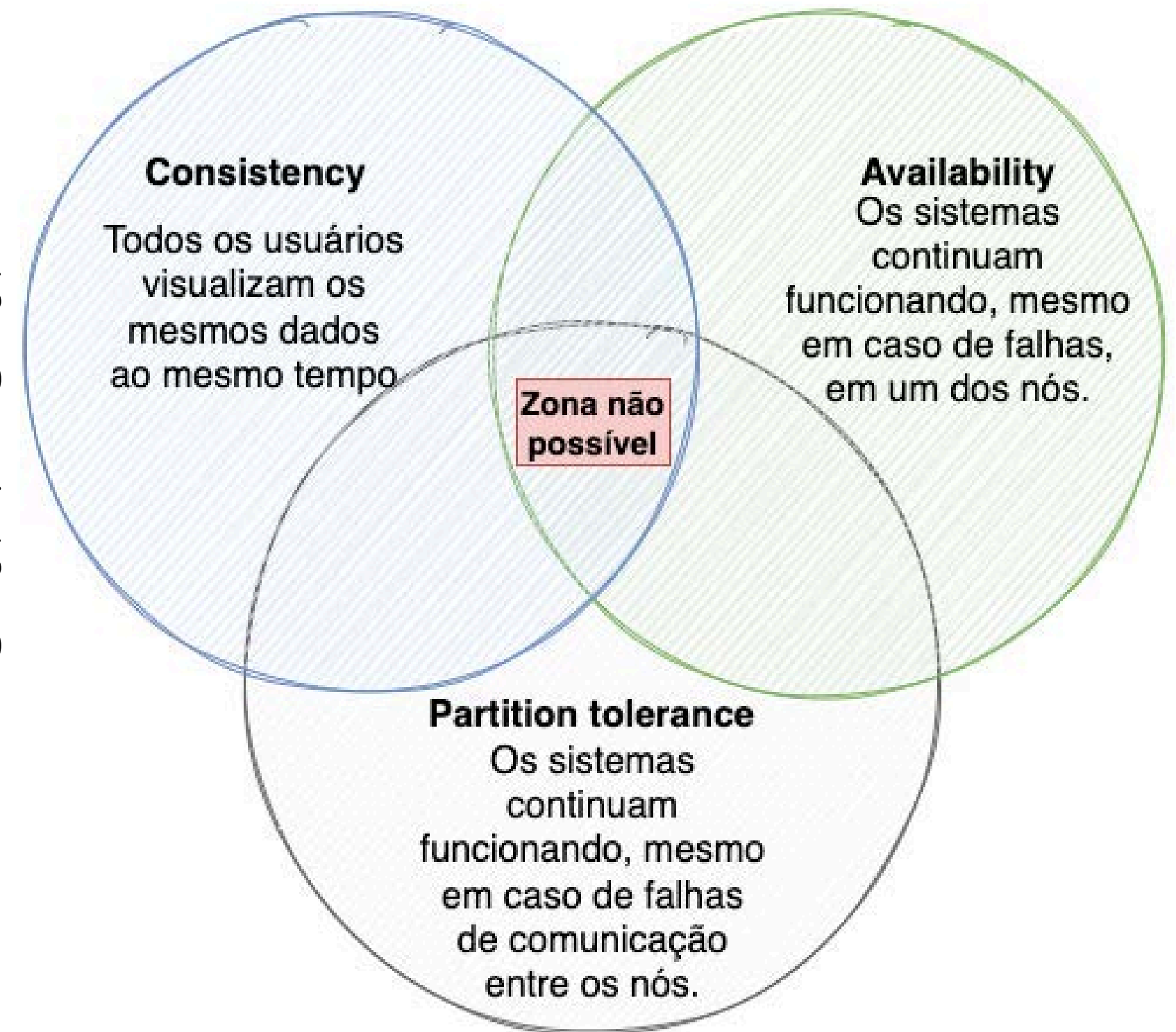
EM SISTEMAS DISTRIBUÍDOS: MESMO QUE HAJA FALHAS DE COMUNICAÇÃO ENTRE NÓS, OS NÓS QUE RESTAM CONTINUAM FUNCIONANDO E RESPONDENDO.



# TEOREMA CAP

## CONSISTENCY (CONSISTÊNCIA)

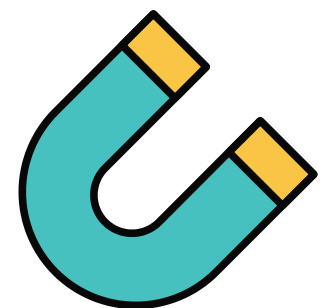
TODOS OS NÓS VEEM OS MESMOS DADOS AO MESMO TEMPO. QUANDO VOCÊ INSERE OU ATUALIZA UM VALOR EM UM NÓ, TODOS OS OUTROS NÓS REFLETEM ESSA ATUALIZAÇÃO IMEDIATAMENTE.



# APLICAÇÕES DISTRIBUÍDAS QUE UTILIZAM DHTS

## BITTORRENT

- USADO PARA COMPARTILHAR ARQUIVOS GRANDES (FILMES, JOGOS, MÚSICAS).
- O DHT SERVE COMO UMA “AGENDA TELEFÔNICA” → EM VEZ DE DEPENDER DE UM SERVIDOR CENTRAL (TRACKER), OS PEERS USAM O DHT PARA DESCOBRIR QUEM TEM O ARQUIVO.
- VANTAGEM: FUNCIONA MESMO SE O SERVIDOR CENTRAL CAIR.



# APLICAÇÕES DISTRIBUÍDAS QUE UTILIZAM DHTS

SISTEMAS DE CLOUD STORAGE DESCENTRALIZADOS (STORJ, SIA, FILECOIN)

- GUARDAM ARQUIVOS ESPALHADOS EM VÁRIOS COMPUTADORES DO MUNDO.
- O DHT É USADO PARA LOCALIZAR ONDE CADA PEDAÇO DE ARQUIVO ESTÁ ARMAZENADO.
- PERMITE UMA “NUVEM SEM DONO”, DIFERENTE DO GOOGLE DRIVE/AWS.

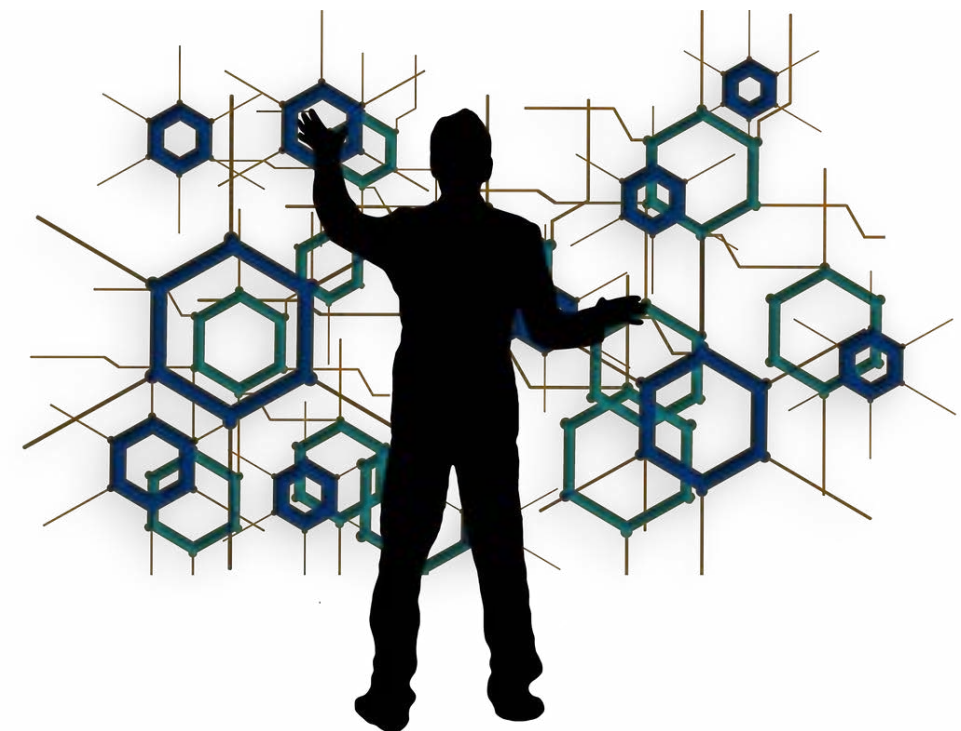




# APLICAÇÕES DISTRIBUÍDAS QUE UTILIZAM DHTS

## BLOCKCHAIN / CRIPTOMOEDAS

- ALGUMAS BLOCKCHAINS USAM DHTS PARA DESCOBRIR NÓS E ARMAZENAR DADOS AUXILIARES.
- EXEMPLO: ETHEREUM USA UMA DHT (KADEMLIA) PARA ORGANIZAR SUA REDE P2P.



# APLICAÇÕES DISTRIBUÍDAS QUE UTILIZAM DHTS

## AMAZON DYNAMODB

- BANCO DE DADOS NOSQL USADO PELA AMAZON (EX.: CARRINHO DE COMPRAS).
- BASEADO EM DHT + CONSISTENT HASHING → OS DADOS SÃO ESPALHADOS EM VÁRIOS SERVIDORES.
- SE UM SERVIDOR CAIR, OUTRO ASSUME, GARANTINDO ALTA DISPONIBILIDADE.

