Resumo: Blockchain GoLedger Módulo 4

Gabriel dos Santos Schmitz

Novembro 2024

This article explores key concepts of Hyperledger Fabric, focusing on transaction flow, private data handling, and network characteristics. It begins with the transaction process, where a client initiates a proposal, which is validated by endorsing peers before being organized and added to the ledger by the Ordering Service. This flow is divided into three stages: proposal and execution, submission and ordering, and validation and commitment. The article also covers specific scenarios like managing Private Data Collections (PDC) for sensitive data, highlighting how PDC adds complexity to transaction flow while maintaining confidentiality. Additionally, network scalability is discussed, addressing the adaptation to organizational changes and the role of Peer Snapshots in data efficiency. Techniques for chaincode interoperability, transaction tuning, and monitoring for malicious activities underscore the flexibility and robustness of Hyperledger Fabric. The piece concludes with a mention of future improvements, particularly the planned Byzantine Fault Tolerant (BFT) consensus in Fabric 3.0, enhancing reliability and security in decentralized environments.

Gabriel dos Santos Schmitz: UTFPR. Eu agradeço colegas e orientador por comentários e discussões úteis. Este trabalho foi apoiado por Grupo PQC UTFPR

1. Conceitos de Hyperledger Fabric II

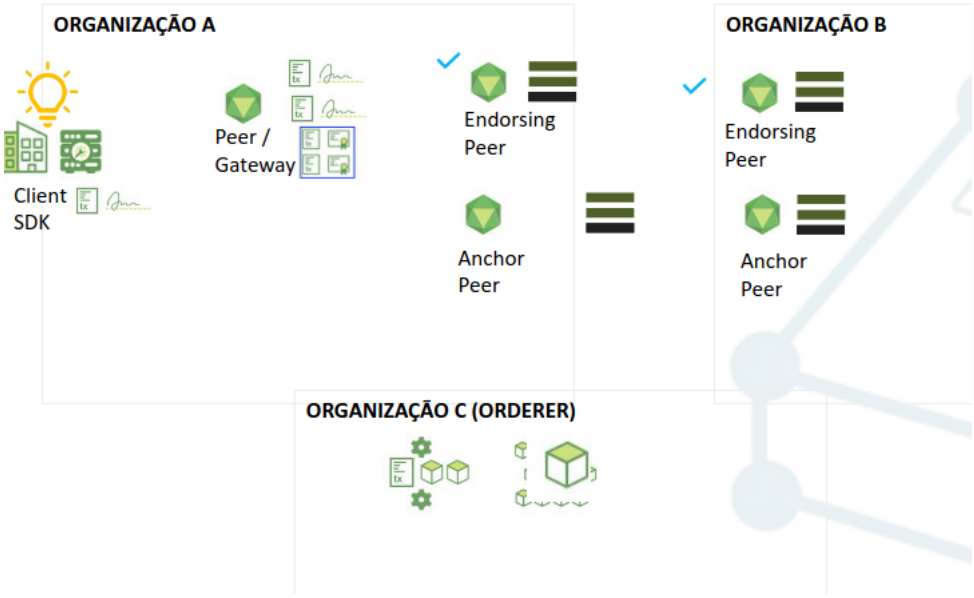

FIGURE 1. Transaction flow

1. Client Creates and Sends a Transaction: The client creates a **PROPOSE MESSAGE** and sends it to a peer with the Gateway Service, which forwards it to the necessary endorsing peers according to the chaincode's endorsing policy. The **PROPOSE MESSAGE** includes:

- clientID
- chaincodeID
- txPayload
- Timestamp
- clientSig

2. Endorsing Peers Simulate and Sign the Transaction: The endorsing peers execute the chaincode using the information from the **PROPOSE MESSAGE**. The result is digitally signed, generating a signed transaction, which is sent back to the peer with the Gateway Service.

3. Client Sends Signed Transactions to the Ordering Service: The client receives the signed transactions from the endorsing peers via the Gateway Service and creates an **ENDORSEMENT** package. This package is then sent to the Ordering Service.

4. Ordering Service Delivers the Block to Peers: The Ordering Service validates the transactions based on the chaincode's endorsing policy. If all required endorsing peers have signed the transaction, it is delivered as a block to all peers, updating their state.

5. Peers Validate Transactions: Each peer validates the consistency of the read/write sets of the transactions. Inconsistent transactions are marked as invalid, while valid transactions update the state.

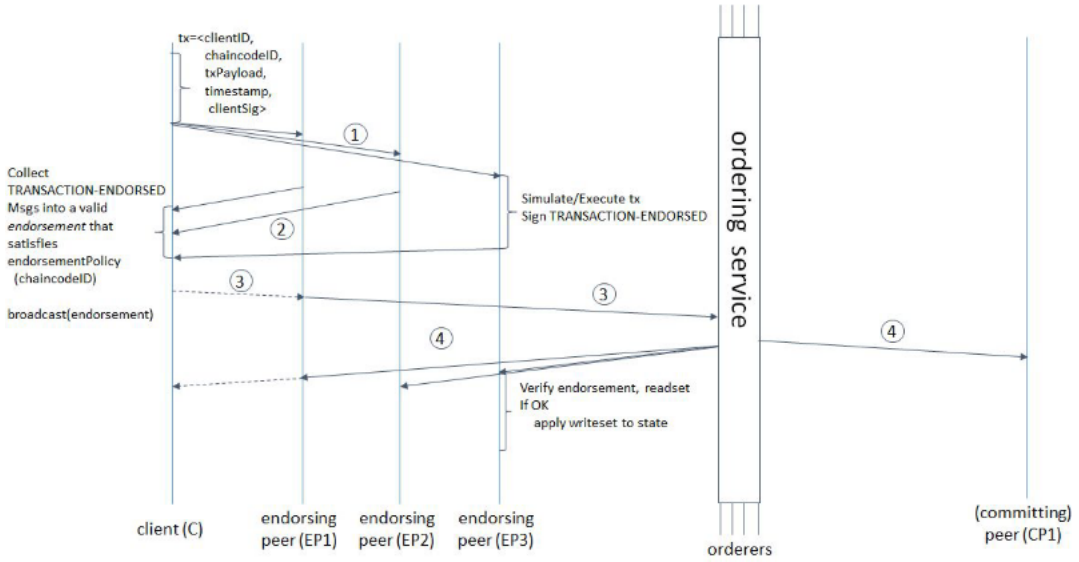

FIGURE 2. Fluxo de uma transação Hyperledger Fabric

1.1. Exemplo de Fluxo

. A Organização A compra e vende legumes da Organização B, e o consenso do chaincode estabelece que ambas as partes devem assinar as transações. Tanto A quanto B possuem um peer para enviar pedidos de compra, aceite ou rejeição. O processo começa quando o cliente da Org A inicia a transação usando um SDK para enviar uma proposta de transação aos endorsing peers de A e B.

. Os endorsing peers recebem a proposta de transação, verificam sua autenticidade (incluindo formato, não duplicação, e autorização do cliente), e executam o chaincode. Se tudo estiver correto, o resultado é enviado de volta ao Gateway. O cliente coleta essas respostas dos endorsing peers e aguarda o próximo passo.

. O Gateway de A monta a transação de endorsement e envia para o Ordering Service, que valida as transações de acordo com a política de endosso e organiza-as cronologicamente em blocos. Depois disso, os blocos são adicionados ao canal.

. Os peers verificam as transações no bloco e atualizam o ledger. Um evento é então enviado ao cliente A, informando que a transação foi registrada com sucesso. Esse processo atualiza o estado global (World State) do ledger.

. O fluxo é dividido em três fases:

Proposta e Execução: A proposta é feita, o chaincode é executado, e o conjunto de leitura/escrita (R/W Set) é assinado pelos peers.

Envio e Ordenação: A transação é enviada ao Orderer para verificação e ordenação.

Validação e Comprometimento: As transações são validadas quanto à consistência, o estado é atualizado, e o cliente é notificado.

. Se o chaincode usa Private Data Collections (PDC), o fluxo inclui uma etapa adicional para gerar uma resposta de proposta de transação. Dados privados são armazenados separadamente no CouchDB, e replicados apenas entre os peers autorizados pelas políticas de PDC. O uso de dados transitórios permite que argumentos sensíveis não sejam gravados nos blocos.

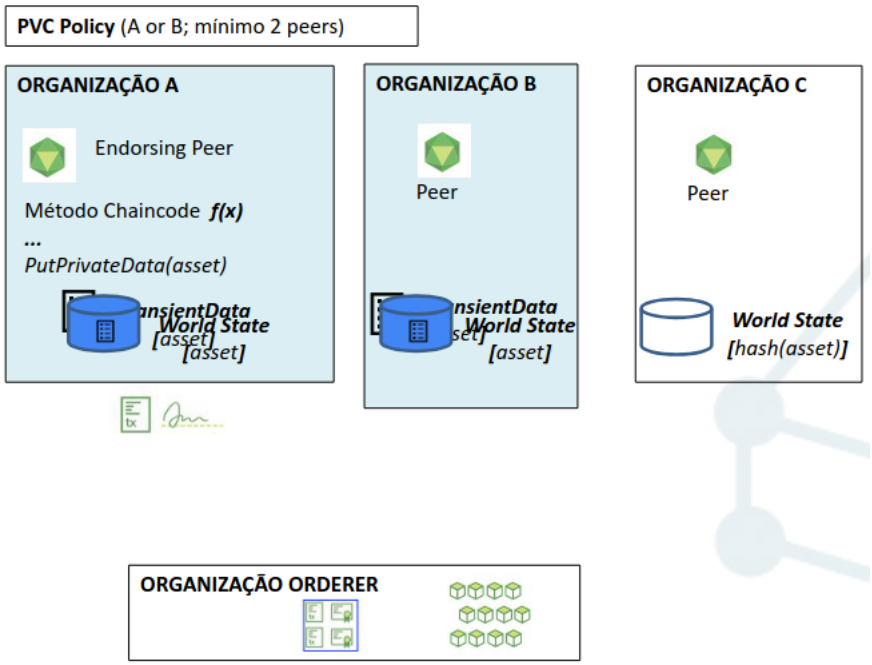

FIGURE 3. Private Data Collections

. Para iniciar uma transação, o SDK do Hyperledger precisa de informações sobre a rede, como políticas de endosso, peers e orderers de um canal. O Service Discovery fornece essas informações de forma dinâmica, complementando o perfil de conexão da rede.

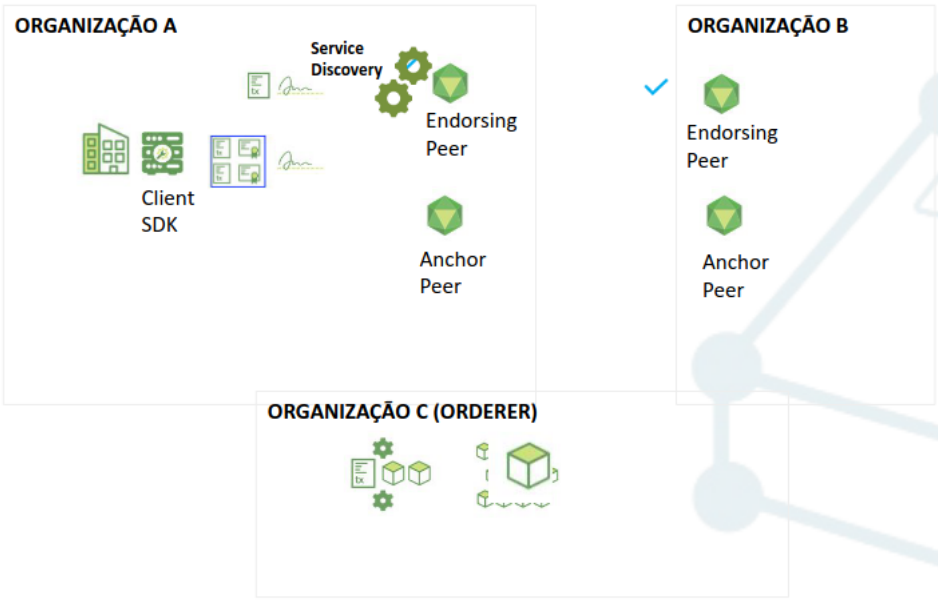

FIGURE 4. Service Discovery

1.2. Características Redes Fabric

. Para manter a rede escalável, alguns eventos podem exigir mudanças na configuração, como a adição de novas organizações ou nós, revogação de certificados (CRL) para excluir entidades, mudanças na hierarquia de consenso e atualizações de contratos inteligentes. Essas adaptações garantem que a rede possa crescer e se ajustar a novas necessidades sem comprometer a segurança ou funcionalidade.

. O recurso de Peer Snapshot permite reduzir a quantidade de dados armazenados no ledger de um peer, economizando espaço e diminuindo o tempo de sincronização de novos peers. Por exemplo, um peer pode começar a partir do bloco #1000, ignorando dados anteriores. No entanto, isso limita as consultas ao histórico e não suporta dados privados (private-data).

. Um chaincode pode interagir com outro utilizando o método invokeChaincode. Quando a chamada ocorre no mesmo canal, o read/write set do chaincode chamado é integrado à resposta principal. Se a chamada é feita entre canais diferentes, apenas a resposta é retornada, sem alterar a proposta de transação original.

. A otimização da velocidade de registro depende de ajustes em várias áreas, como a política de endosso, métodos específicos do chaincode, parâmetros do cliente e uso de Private Data Collections (PDC). Essas configurações permitem balancear segurança e desempenho conforme as necessidades do sistema.

. Embora o consenso do Hyperledger Fabric não seja Byzantine Fault Tolerant (BFT), existem mecanismos para detectar comportamentos suspeitos. Clientes podem comparar as respostas de propostas de transação, e peers que enviam respostas inconsistentes podem ser tratados de forma diferenciada. Em cenários de consenso democrático, peers anômalos podem ser sinalizados para administração, que decide sobre permissões e atualizações. O suporte a BFT está previsto para a versão 3.0 do Fabric.