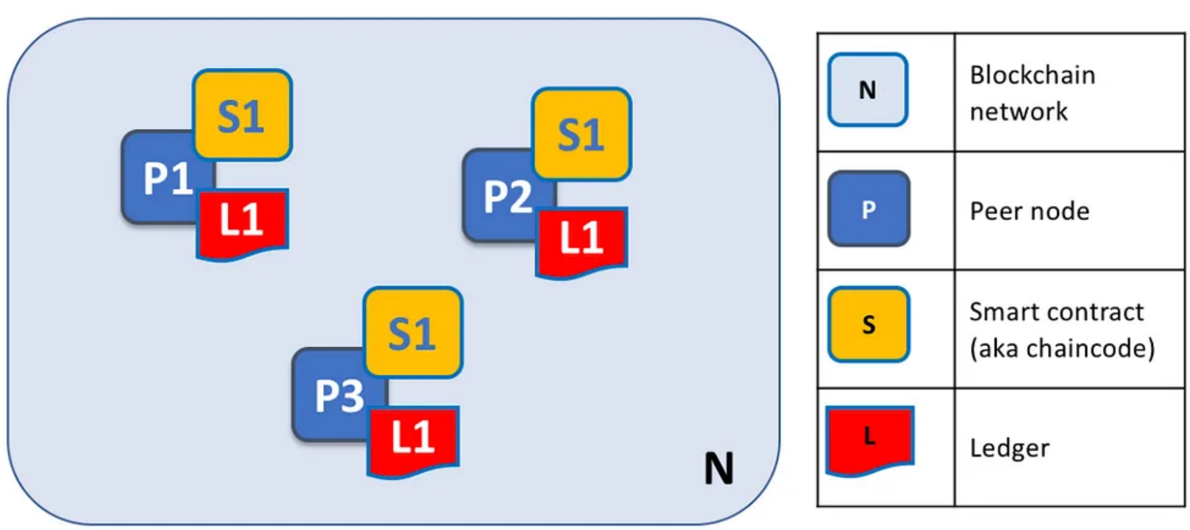
“What is Hyperledger Fabric Peer Nodes: A Beginner’s Guide”

Nova Novriansyah · Follow
Published in Novai-Hyperledger Fabric 101
8 min read · May 7, 2024

Hyperledger Fabric, a blockchain framework, relies on various components to function smoothly. Among these, peer nodes play a pivotal role. Let’s delve into what peer nodes are, how they function, and their significance within a Hyperledger Fabric network.



A Fabric blockchain network (above) is comprised of peers (non-ordering nodes), each of which stores and manages copies of ledgers and smart contracts. In this example, the Fabric network N consists of peers P1, P2 and P3, each of which maintains its own instance of the distributed ledger L1. P1, P2 and P3 each invoke the same chaincode, S1, to access their respective copies of the distributed ledger.

Understanding Peer Nodes

Imagine a peer node as a digital ledger-keeper within the Hyperledger Fabric ecosystem. These nodes manage ledgers and execute smart contracts, which are crucial for processing transactions securely.

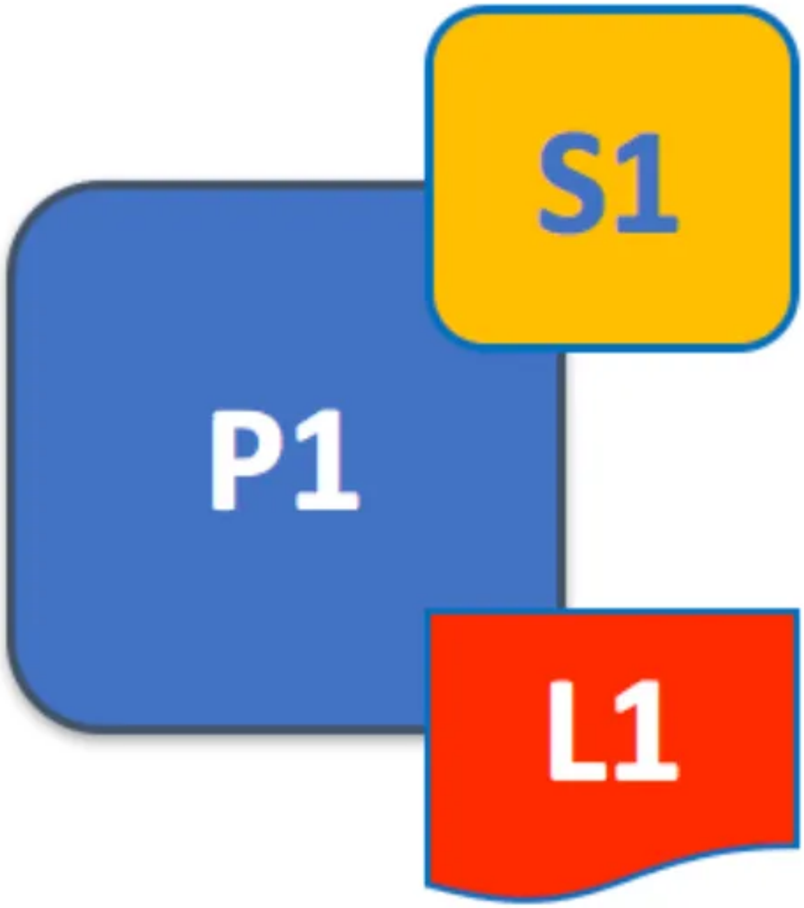
Peers are dynamic entities, capable of being created, started, stopped, and reconfigured as needed. They provide redundancy, ensuring there are no single points of failure within the network.

Peer Nodes in Action

Let’s simplify this with an example. Consider a Hyperledger Fabric network called N, consisting of peers P1, P2, and P3. Each peer maintains its copy of the ledger L1. When transactions occur, these peers access the same chaincode, S1, to update their ledgers.

Hosting Ledgers and Chaincode

Peers host instances of ledgers and chaincodes. A ledger records transactions, while chaincodes execute the business logic. This redundancy ensures consistent data across all peers in a channel, enhancing network reliability.



A peer hosts instances of ledgers and instances of chaincodes (above). In this example, P1 hosts an instance of ledger L1 and an instance of chaincode S1. There can be many ledgers and chaincodes hosted on any individual peer.

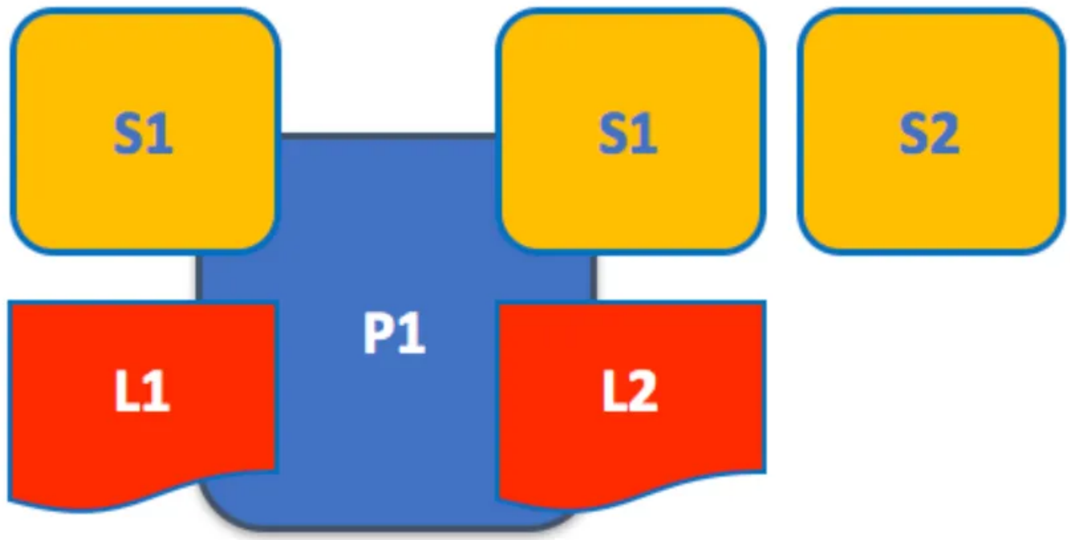
Because a peer is a *host* for ledgers, chaincodes and services, client applications and administrators must interact with a peer to access these resources.

Chaincodes installed on a peer can query or update (write to) a ledger. Note that peers also host special *system chaincodes* which relate to the overall configuration of the Fabric network.

Multi-Ledger and Multi-Chaincode Support

Peers can host multiple ledgers and chaincodes, allowing for flexible network configurations. For example, P1 can host ledgers L1 and L2, accessed by chaincodes

S1 and S2, respectively.



A peer hosting multiple ledgers (above). Peers host one or more ledgers and the chaincodes that access them. In this example, peer P1 hosts ledgers L1 and L2. Ledger L1 is accessed using chaincode S1; Ledger L2 can be accessed using either chaincode S1 or S2.

The Fabric Gateway Service

Starting in Hyperledger Fabric v2.4, peers manage transaction proposals and endorsements using the Fabric Gateway service. This service simplifies application development by handling transaction processing, making interactions smoother and more efficient.

Applications and Peers

Let's delve into how client applications interact with peers, specifically through the Fabric Gateway service, to access and update the ledger. When it comes to ledger operations, there's a distinction between querying and updating.

1. **Querying the Ledger:** When an application needs information from the ledger, it initiates a simple dialogue with a peer. The Fabric Gateway service acts as the intermediary, facilitating this interaction. Through a connection with the gateway, applications can easily access ledgers and chaincodes. With the

introduction of Gateway SDKs in Fabric v2.4, programmers find querying straightforward. The APIs provided by the SDKs empower applications to submit queries to chaincodes, receive responses, and handle events.

2. **Updating the Ledger:** Ledger updates, on the other hand, involve a more intricate process. When an application intends to modify the ledger, it initiates a multi-step workflow that engages peers, orderers, and the gateway service. This process ensures that every change to the ledger is meticulously validated and recorded, maintaining consistency and accuracy across the network.

3. Let’s dissect this ledger update process into three phases:

- *Phase 1: Transaction Proposal and Endorsement*. Here, the application submits a transaction proposal via the gateway service. Peers execute the proposed transaction, generate responses, and obtain endorsements. The gateway service orchestrates this process, ensuring that all necessary endorsements are secured.

- *Phase 2: Transaction Submission and Ordering.* Once endorsed, the transaction is forwarded to the ordering service, where it undergoes validation and is organized into blocks. These blocks are then distributed to peers for further processing.

- *Phase 3: Transaction Validation and Commitment.* Peers validate the transactions within the received blocks, ensuring consistency and adherence to network rules. Valid transactions are then committed to the ledger, ensuring that all peers maintain synchronized and up-to-date records.

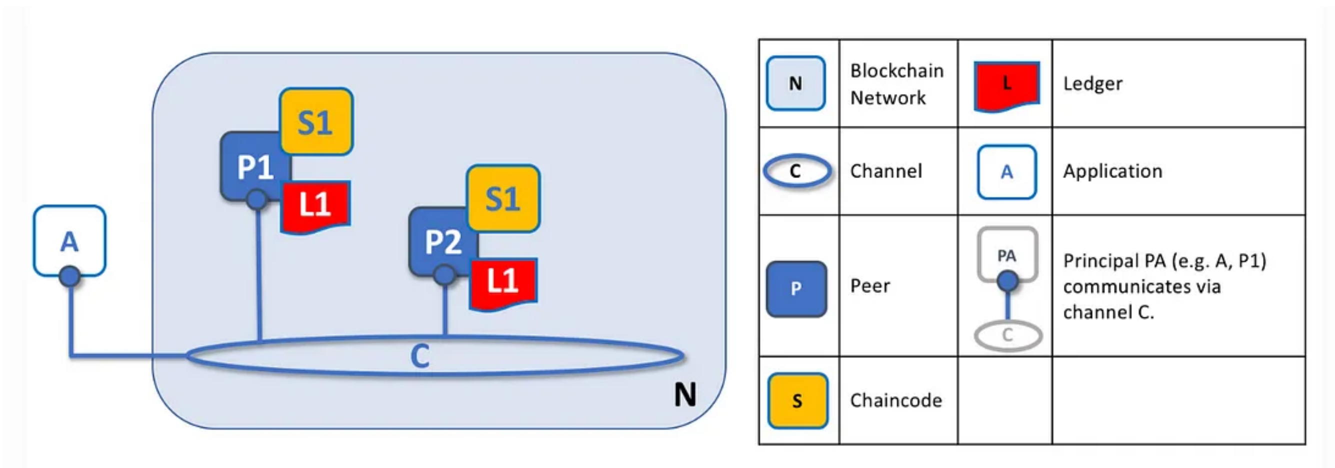
It’s important to note that while these phases offer insights into the inner workings of ledger updates, developers can seamlessly implement them using Fabric Gateway SDKs. These SDKs abstract the complexity, allowing developers to focus on building robust applications without getting bogged down in the intricacies of the underlying transaction management process.

Peer Interaction on Channels

Channels serve as private communication pathways within a Fabric network. Peers interact with each other and applications on specific channels, ensuring secure and private transactions. Just like groups of friends, channels maintain distinct rules and memberships.

Channel components include peer nodes, orderer nodes and applications, and by joining a channel, they agree to collaborate to collectively manage and share identical copies of the ledger.

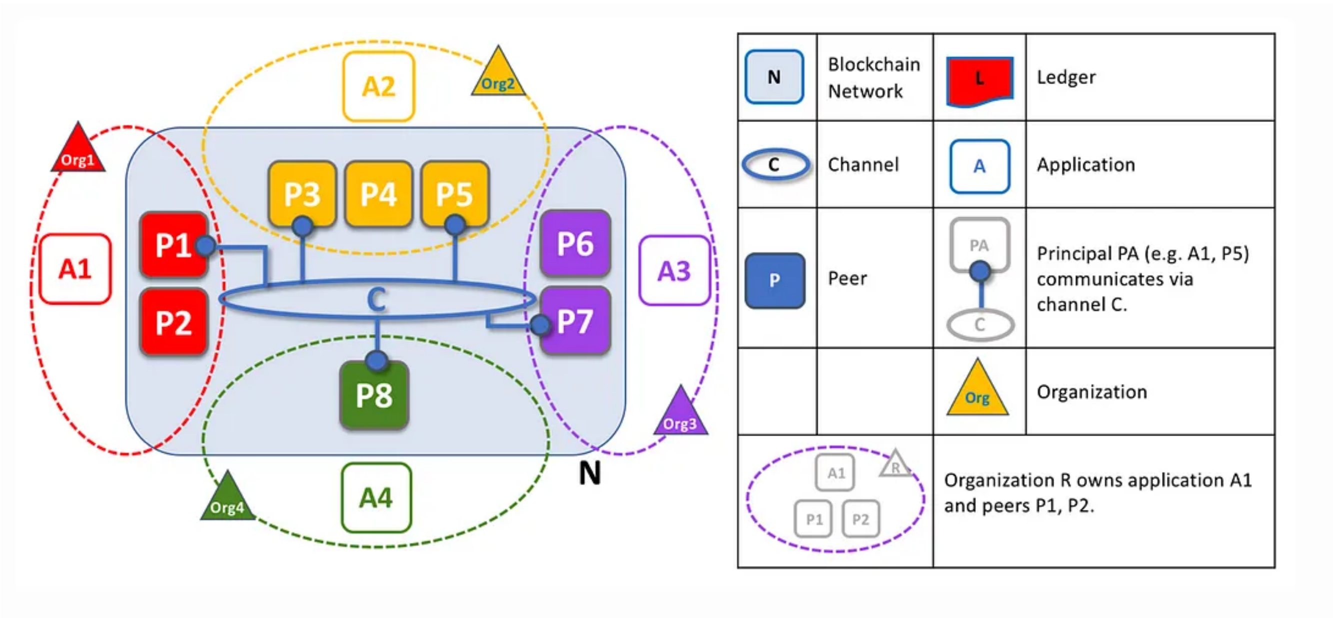
Channel membership works the same way as in other groups; any one peer may belong to several channels and maintain a ledger and chaincodes specific to each channel. Or a peer may belong to only a single channel, and therefore have only one set of rules to follow.



Channels allow a specific set of applications and peers (and organizations) to communicate with each other on a Fabric blockchain network. In the example above, application A communicates with peers P1 and P2, through the gateway service, on channel C. The channel is a pathway for communications between specific applications and peers.

Peers and Organizations

Fabric networks are administered by multiple organizations, each contributing peers and resources. Peers serve as connection points for these organizations, forming the backbone of the decentralized network.



*The example above shows organizations and their peers in a Fabric blockchain network. We see four organizations contributing a total of eight peers to form a network. Channel C connects five of these peers in the network N — P1, P3, P5, P7 and P8. The other peers owned by these organizations have not joined channel C, but typically join at least one other channel. Applications developed by an organization connect to peers, via the Fabric Gateway service, in the same organization as well as peers in other organizations on a channel.

In the formation of a Fabric blockchain network, it’s crucial to understand the collaborative effort of multiple organizations. These organizations not only contribute peers but also provide vital resources like chaincode and ordering service nodes. This collaborative principle underscores the network’s existence and resilience.

The network’s vitality hinges on organizations pooling their resources. Peers serve as the backbone, but it’s the collective contribution of all resources that breathes life into the network. Unlike centralized systems, where reliance on a single entity is common, Fabric networks thrive on shared responsibility.

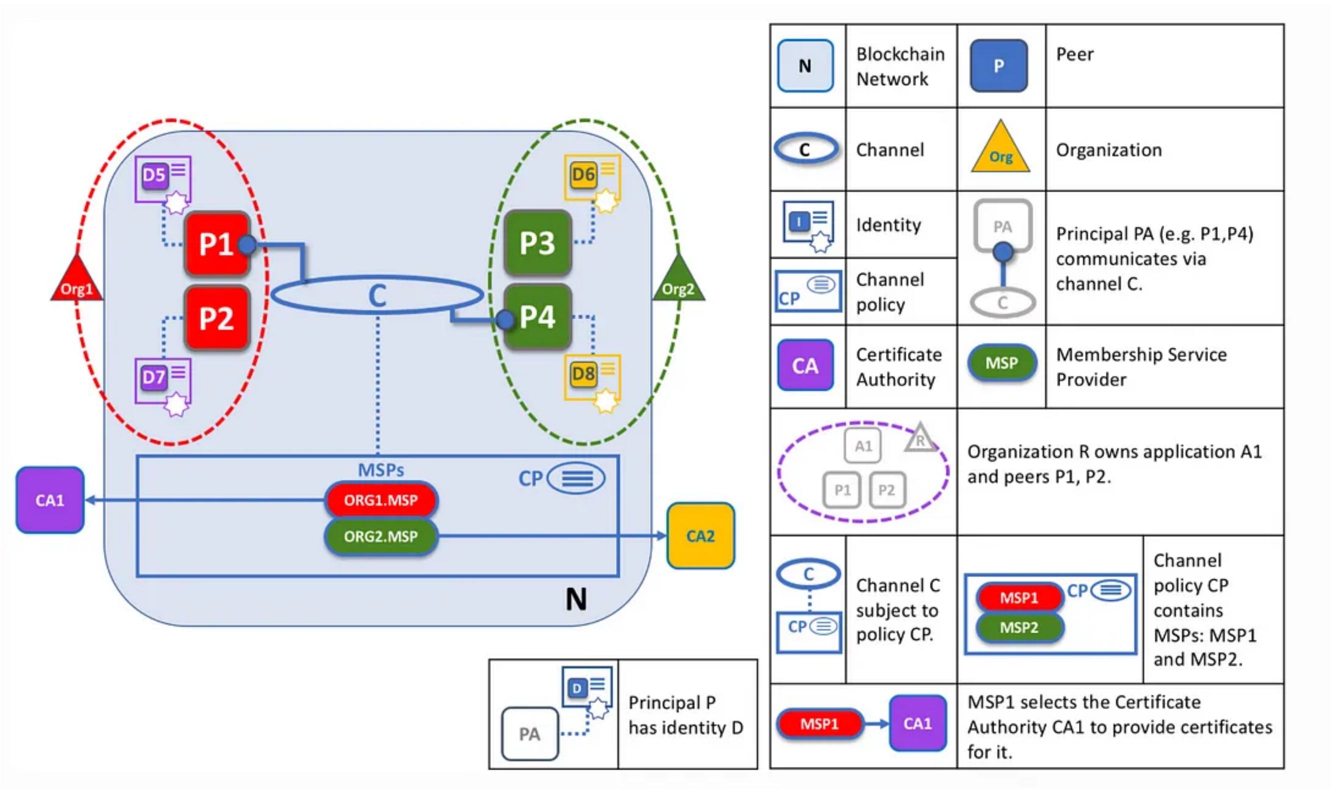
In essence, the network’s strength lies in its decentralized nature. It’s a network where no single organization holds dominance, and contributions from all members are valued equally. Even if organizations exit, the network persists as long as its membership meets the defined criteria, showcasing its inherent resilience.

Furthermore, the diversity among organizations extends to their use of applications.

Each organization has the freedom to tailor their applications and presentation logic according to their unique needs. Despite these variations, all peers maintain an identical ledger, ensuring consistency and integrity across the network. This flexibility underscores the adaptability of Fabric networks, accommodating diverse organizational requirements while upholding data integrity and security.

Identity and Ownership

Peers are assigned digital certificates by their owning organizations, linking them to specific roles within the network. These certificates determine access rights and ensure secure interactions among network participants.



When a peer connects to a channel, its digital certificate identifies its owning organization via a channel MSP. In the example above, P1 and P2 have identities issued by a certificate authority (CA1). Channel C determines, from a policy in its channel configuration, that identities from CA1 should be associated with Org1 using ORG1.MSP. Similarly, P3 and P4 are identified by ORG2.MSP as part of Org2.

Let's simplify how peers are connected to organizations in a blockchain network:

1. *Digital Certificates:* Peers are like members of a club, and to prove their membership, they carry a special ID card called a digital certificate. This *certificate is issued by a certificate authority* and contains important details about

the peer.

2. *Membership Service Provider (MSP):* When a peer joins a Fabric network channel, its rights and privileges are determined by its identity, which is managed by a Membership Service Provider (MSP). Think of the *MSP as the gatekeeper that assigns roles to peers within organizations*. Each peer belongs to one organization and is linked to a specific MSP, much like a person belongs to one club and follows its rules.

3. *Organizational Identity:* Peers, along with other network participants like applications and administrators, obtain their organizational identity from their digital certificate and the associated MSP. This identity is crucial for accessing network resources. It’s important to note that a peer’s identity is independent of its physical location. For example, even if a peer is hosted in Org1’s data center, if its digital certificate is issued by CA2, it’s considered owned by Org2.

In essence, peers, applications, and other network entities need to have an identity assigned by an MSP to interact with the blockchain network. This ensures that everyone follows the rules of their respective organizations while accessing network resources.

Collaboration of Peers with Orderers

Peers are like the backbone of a Fabric blockchain network. They store important data (ledgers) and smart contracts, which can be accessed and updated by applications connected to them. But how do these updates happen? Enter the orderers, special nodes that play a crucial role in the process.

Peers are like the building blocks of a Fabric blockchain network. They store ledgers and smart contracts, which are essentially the rules of the network. These ledgers can be checked or changed by applications connected to the peers. But there’s a special role for orderers. They ensure that when changes happen to the ledger, everyone agrees on what those changes are.

Three Phases of Updating the Ledger

1. Phase 1: Proposing the Change

- An application wants to make a change to the ledger, so it asks the Fabric Gateway service for permission.
- The application selects a peer to carry out the change.
- The selected peer tries out the change (but doesn’t actually apply it to the ledger) and reports back.
- Meanwhile, the change proposal is sent to other peers for their approval.
- If all goes well and everyone (every orgnatinsations) agrees, the proposal moves to the next phase.

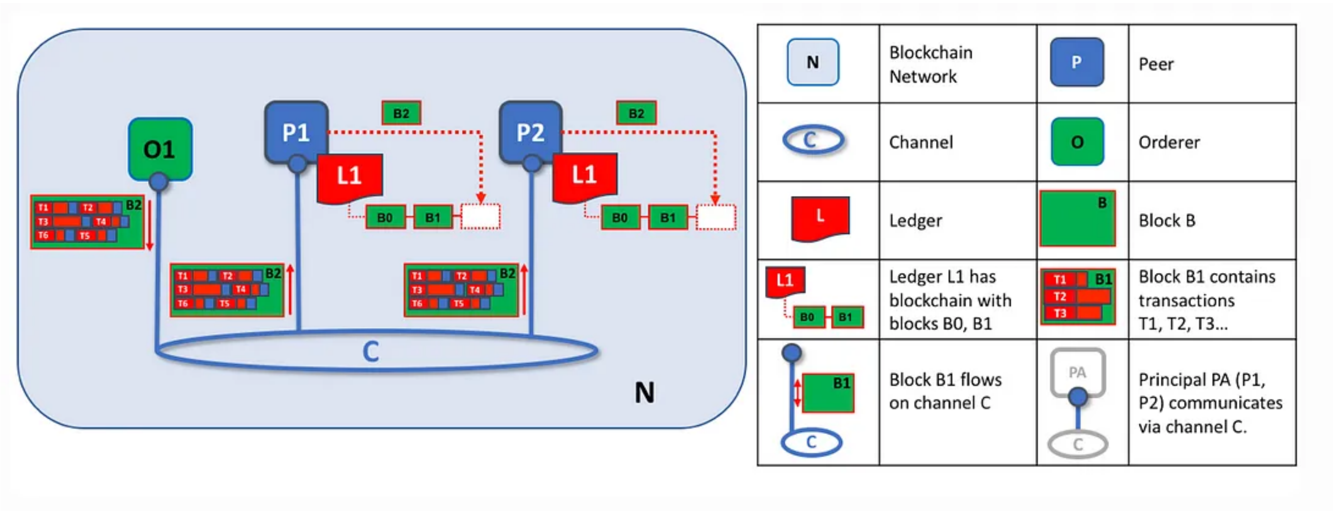
2. Phase 2: Ordering the Transactions

- Now, the approved changes (by each organisations), along with any associated updates to the rules (chaincode), are sent to the orderer nodes.
- These orderer nodes organize the changes into blocks, like putting puzzle pieces together.
- These blocks are then arranged in a specific order, forming the ledger.

3. Phase 3: Confirming and Applying Changes

- Each peer receives these blocks and makes sure they’re valid and in the right order.
- If everything has been endorsed correctly, the peer applies the changes to its copy of the ledger.
- This ensures that all peers have the same ledger, maintaining consistency.
- During this phase, the chaincode isn’t run again; it’s just confirming that the proposed changes fit the rules set by the chaincode.
- Additionally, successful transactions trigger chaincode execution, leading to further updates on the ledger.

- Note that phase 3 does not require running chaincode — this is done only during phase 1, and that’s important. It means that chaincodes only have to be available on endorsing nodes, rather than throughout the blockchain network.
- This keeps the logic of the chaincode confidential to endorsing organizations only. This is in contrast to the output of the chaincodes (transaction proposal responses), which are shared with every peer in the channel, whether or not they endorsed the transaction.



An orderer node distributes ordered blocks to peers for validation and commitment. In this example above, orderer O1 distributes block B2 to peers P1 and P2. Peer P1 processes block B2, resulting in a new block being added to ledger L1 on P1. In parallel, peer P2 processes block B2, resulting in a new block being added to ledger L1 on P2. Once complete, the ledger L1 has been consistently updated on peers P1 and P2, and the gateway service informs the relevant applications that the transaction has been committed.

Orderers and Consensus

The entire process of updating the ledger is called consensus because all peers need to agree on the order and content of transactions, with help from orderers. It’s a multi-step process, and ledger updates only happen when everyone agrees. Orderers act like traffic controllers, organizing and distributing proposed ledger updates to peers for their approval before adding them to the ledger.

Conclusion

In summary, peer nodes are the backbone of a Hyperledger Fabric network, facilitating secure transactions and data management. Their flexibility, redundancy, and collaborative nature make them essential components in building decentralized blockchain solutions. Understanding peer nodes is key to unlocking the full potential of Hyperledger Fabric in various enterprise applications.

Hyperledger Fabric

Follow

Published in Novai-Hyperledger Fabric 101

7 Followers · Last published Jun 5, 2024

Welcome to our Hyperledger Fabric channel, your portal to understanding the intricacies of enterprise-grade blockchain solutions. Explore the robust features and practical applications of Hyperledger Fabric, a leading framework for building permissioned blockchain networks.

Follow

Written by Nova Novriansyah

109 Followers · 34 Following

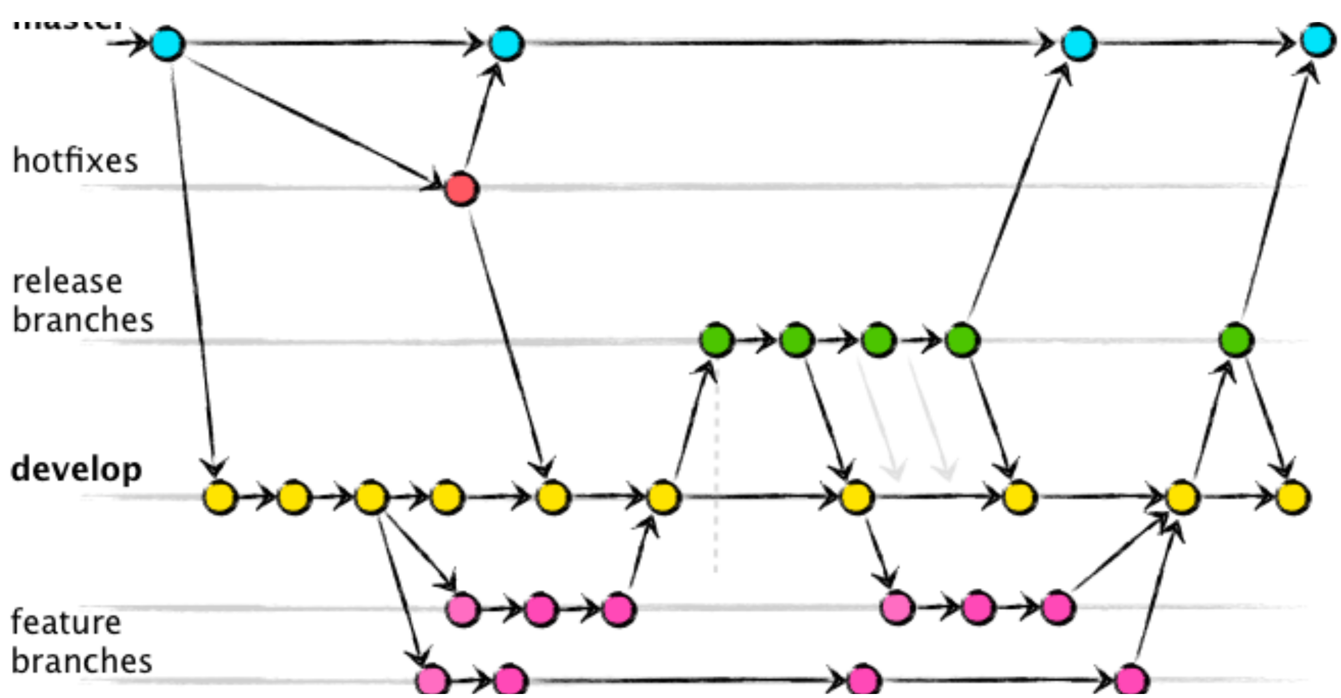
C|CISO, CEH, CC, CVA,CertBlockchainPractitioner, Google Machine Learning , Tensorflow, Unity Cert, Arduino Cert, AWS Arch Cert. CTO, IT leaders. Platform owners

No responses yet

What are your thoughts?

Respond

More from Nova Novriansyah and Novai-Hyperledger Fabric 101



In NovAI- Agile & DevOPS 101 by Nova Novriansyah

Top 4 Branching Strategies and Their Comparison: A Guide with Recommendations

Branching strategies are critical in version control, helping teams manage and organize code changes efficiently. Choosing the right…

Aug 15 14

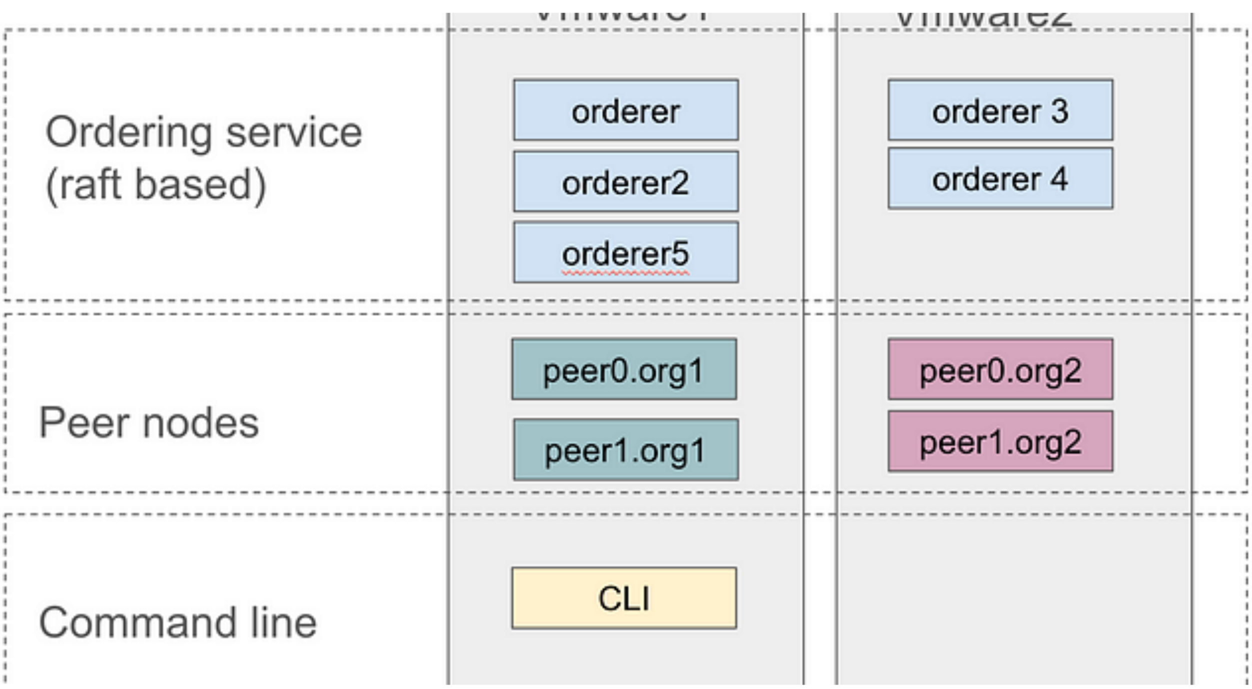
```
│
├── keystore            (Directory for storing private keys)
│   ├── admin_sk        (Admin user's private key)
│   ├── peer1_sk        (Peer's private key)
│   └── orderer1_sk     (Orderer's private key)
│
├── signcerts           (Directory for storing X.509 certificates)
│   ├── admin_cert.pem  (Admin user's X.509 certificate)
│   ├── peer1_cert.pem  (Peer's X.509 certificate)
│   └── orderer1_cert.pem (Orderer's X.509 certificate)
│
├── cacerts             (Directory for storing CA certificates)
│   └── ca-cert.pem     (Root CA certificate)
│
└── tlscacerts          (Directory for storing TLS CA certificates)
```

In Novai-Hyperledger Fabric 101 by Nova Novriansyah

Using Fabric CA on Hyperledger Fabric 2.4: A Guide

Introduction to Fabric CA

Jun 5



In Novai-Hyperledger Fabric 101 by Nova Novriansyah

Multi host Hyperledger Fabric 2.x deployment (Docker Swarms)

Hyperledger Fabric 2.4 is a leading open-source blockchain platform designed for building enterprise-grade distributed ledger applications...

In NovAI Cloud Computing — GCP by Nova Novriansyah

How to Install Google Cloud CLI (Command-Line Interface) on Mac, Windows, and Linux

Google Cloud CLI, known as gcloud, is an essential tool for managing Google Cloud Platform (GCP) resources from the command line...

Jun 21 1 1

See all from Nova Novriansyah

See all from Novai-Hyperledger Fabric 101

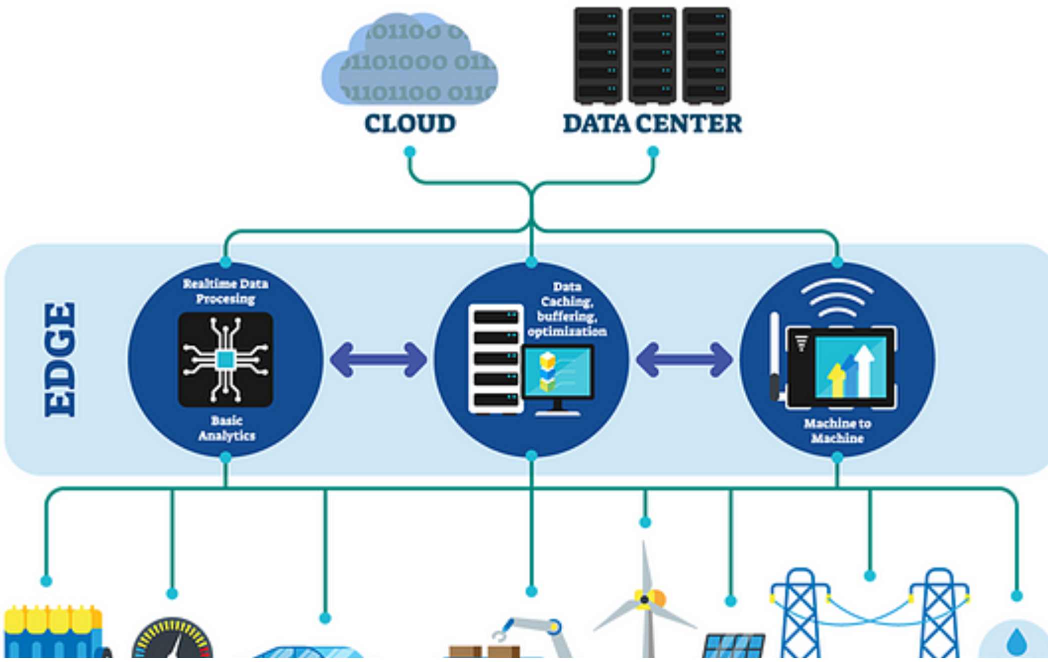
Recommended from Medium

In Personal Finance & Investing by Dr. Ghulam Mohey-ud-din

Understanding Blockchain Technology: Overview and Practical Applications

“The magic of blockchain lies in its consensus mechanism. For a block to be added to the chain, the majority of the network must agree that…

Jul 4 4



In Modernizing Maintenance by Stylianos Chiotis

MQTT or OPC UA | My Roadmap to IIoT Communication Protocols

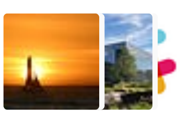
Through the eyes of a Mechanical Maintenance Engineer

Aug 19 1

Lists

Staff picks

778 stories · 1485 saves

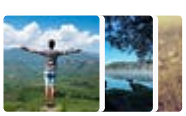


Stories to Help You Level-Up at Work

19 stories · 887 saves

Self-Improvement 101

20 stories · 3113 saves



Productivity 101

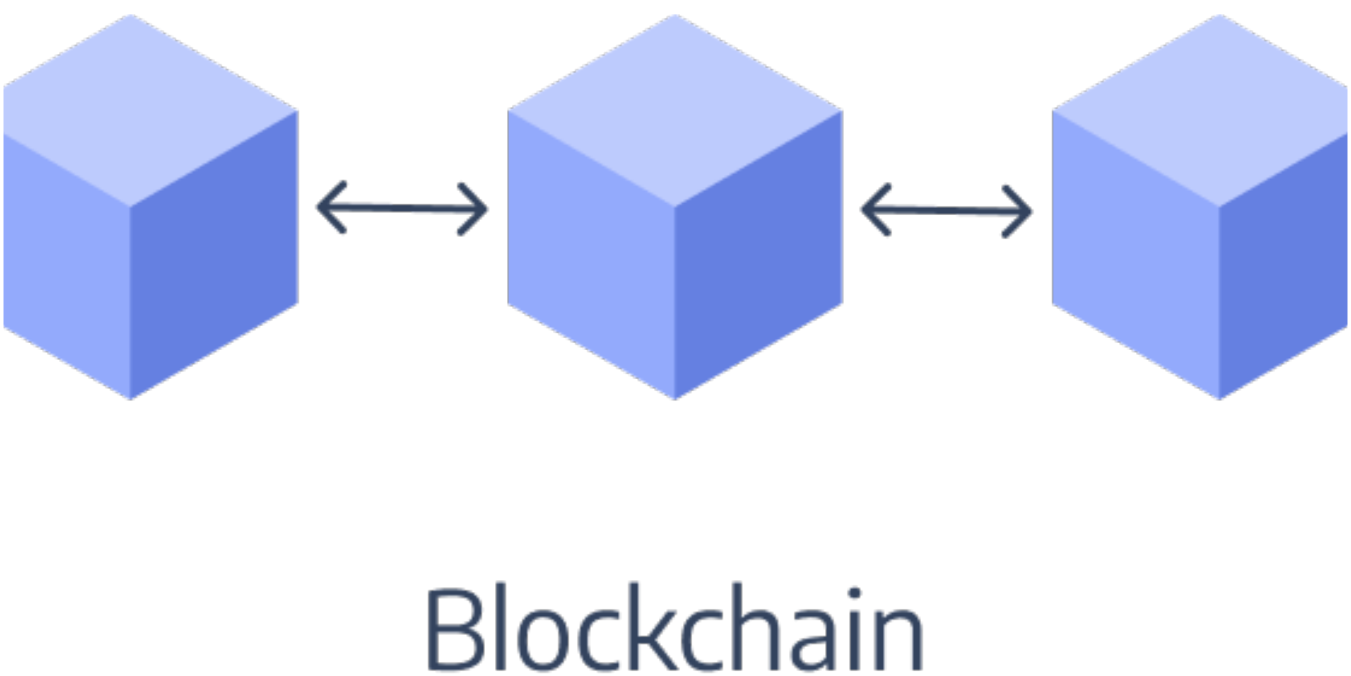
20 stories · 2617 saves

NRT0401

Roadmap for Learning New Technologies in 2025 – Blockchain Technology

Introduction Blockchain technology has emerged as a groundbreaking innovation, offering solutions beyond cryptocurrencies. As we approach…

Oct 17 5



Sithara Wanigasooriya

Blockchain in 2024: An Expert’s Guide to its Core Components and Evolution

Blockchain is now recognized as a decentralized, secure, and transparent way to store and manage data across a network of computers without…

Oct 6

In Novai-Hyperledger Fabric 101 by Nova Novriansyah

Getting Started with Smart Contracts in Hyperledger Fabric using Go

Introduction: Smart contracts play a crucial role in blockchain technology, facilitating automated execution of predefined logic upon…

Jun 3

In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

Nov 19 3.6K 111

See more recommendations