

Complete Guide to Blockchain Development (with Hyperledger Fabric)

What is a Blockchain?

A **blockchain** is a distributed and immutable digital ledger that records data in a secure, transparent, and tamper-proof manner. It works by linking blocks of data using cryptographic hashes.

Key Characteristics:

1. **Immutability** – Data can't be changed once recorded.
 2. **Decentralization** – No central authority.
 3. **Transparency** – All nodes can verify transactions.
 4. **Consensus** – Agreed rules ensure only valid transactions are recorded.
 5. **Security** – Public-key cryptography and hashing secure the data.
-

Core Components of a Blockchain

Component	Description
Block	Contains data, timestamp, hash, previous hash
Chain	A linked list of blocks
Node	A participant in the network
Consensus	Method to agree on valid transactions
Smart Contract	Code that runs on-chain to enforce logic

Types of Blockchains

Type	Description	Use Cases
Public	Open to everyone, decentralized	Bitcoin, Ethereum, NFTs
Private	Controlled by a single entity or group	Enterprise solutions, supply chains
Consortium	Multiple organizations manage access	Inter-bank systems
Hybrid	Combination of public and private features	Identity verification, government

When to Use Hyperledger Fabric

Hyperledger Fabric is a permissioned blockchain platform ideal for:

- Enterprise-grade applications
- Private and secure networks
- Scalable and modular deployments
- No need for native tokens or mining

Benefits:

- Pluggable consensus
- Private channels
- Chaincode in Go, Java, or Node.js
- Membership Service Providers (MSP) for access control

Concepts to Know Before Building a Blockchain

1. **Hashing** – SHA-256, Keccak
2. **Public/Private Key Cryptography** – Digital signatures
3. **Merkle Trees** – Efficient verification of data
4. **Consensus Algorithms** – PoW, PoS, PBFT
5. **Peer-to-Peer Networks** – Decentralized communication
6. **Chaincode** – Fabric's term for smart contracts
7. **Event Handling** – Triggering external services from blockchain

Setting Up Hyperledger Fabric

1. Install Required Tools

- Docker & Docker Compose
- Go 1.20+
- Node.js (v18 LTS)
- Git
- cURL
- jq

2. Download Fabric Binaries

```
curl -sSL https://bit.ly/2ysbiFn | bash -s -- 2.5.0
```

3. Start Test Network

```
cd fabric-samples/test-network  
./network.sh up createChannel -ca
```

4. Deploy Chaincode

```
./network.sh deployCC -ccn basic -ccp ../asset-transfer-basic/chaincode-go -ccl  
go
```



Project Use Case: Income Traceability System

A system to:

- Record wage payments immutably
- Use AI to classify BPL/APL status
- Provide real-time dashboard for policymakers
- Integrate UPI/PAN/Aadhaar data

Input:

- PAN/Aadhaar (anonymized)
- Job type, wage amount, date
- Govt-defined thresholds

Output:

- Immutable wage history
- Estimated annual income (via AI)
- Real-time BPL/APL flags
- Eligibility dashboard



Architecture Overview

- Blockchain: Hyperledger Fabric (private)
 - Backend: Node.js + Fabric SDK
 - AI Engine: Python
 - Frontend: React/Dash/Tailwind
 - Smart Contracts: Go (chaincode)
-

Privacy & Access

- Fabric CA for identity
 - Private data collections
 - Attribute-based access control (ABAC)
-

Integration Strategy

- Record payment txns on Fabric
 - Trigger AI service off-chain for classification
 - Store eligibility flags back on blockchain
 - Update dashboards in real-time
-

Final Notes

Hyperledger Fabric is optimal when:

- Participants are known
- Privacy and trust are critical
- Fast, reliable consensus is needed

Avoid public blockchains unless:

- You need token economy
 - Transparency must be absolute
-

Next Steps:

1. Write custom smart contract (chaincode)
 2. Connect backend with Fabric
 3. Build real-time dashboard
 4. Integrate AI classifier pipeline
-

Prepared for: Innovative Blockchain-based Welfare Distribution Project (Group 6 TRACIENT) **Compiled by:** Niranj | August 2025