**Hyperledger Fabric & Ethereum**

**Hyperledger Fabric**

An open source plug-and-play development architecture maintained by Linux Foundation.

It is focused on blockchain related use cases that’ll work under a variety of industrial sectors. It serves as a backbone for creating blockchain-based goods, services and applications that are intended for usage by private businesses.

Unlike other blockchain architectures, hyperledger fabric supports private transactions and confidential contracts. Hyperledger Fabric is a consortium blockchains (permission blockchains) and its most noticeable change is that the participating peers are now decoupled into two separate runtimes with three distinct rolls (Endorser, Comitter and Consenter).

**Hyperledger Fabric Transaction Flow**

Client application sends a transaction proposal starting the transaction flow only after each organization peers approve, peers then simulate the proposed transaction confirming the identity and legitimacy of the submitting client then sends back an endorsement signature to the client only if what was anticipated matched.

Once the transaction proposed has the required number of endorsements as specified in the endorsement policy the client application gathers recommendations from peers and transmits the transaction to the ordering service.

Lastly, in order to comply with the endorsement policy the ordering service has to verify that the transaction has the required number of endorsements, the accepted transactions are then chronologically ordered and bundled into blocks and sent to peers within each organization.

The ordering service sends new blocks of transaction to peer nodes who subsequently do a final validation on the transaction in that block and as soon as that finishes the ledger state is updated and a new block is added.

Business processes to be automated are documented in a smart contract with self-executing clauses between parties written in lines of code. Keep in mind that these transactions are retractable and irreversible.

**Hyperledger Fabric Use Cases**

1. Supply Chain Operations, by boosting transparency and transaction traceability within the network.
2. Financial and Trading Consortiums, by easily creating a blockchain network for all parties to transact and process trade-related paperwork electronically.
3. Insurance Companies, by identifying duplicate or fabricated claims by referring to transaction data kept on the ledger.

**Ethereum**

Ethereum is an open source distributed blockchain network written in **Solidarity Language** and maintained by Ethereum Developer Community.

It is a platform for decentralized apps **(DAPPS)** that improvised Bitcoin’s protocols to support applications beyond currency issuance. Its major breakthrough is the ability to easily write and deploy smart contracts.

Ethereum comes with a built-in cryptocurrency **Ether** that is used to pay for transaction fees and computational services.

Anyone across the globe can connect with Ethereum blockchain and can maintain the current state of the network.

**Ethereum Structure**

A fundamental concept in Ethereum architecture are **Accounts**. There are two kinds of Accounts, **EOA** and **Contract Accounts.**

Externally Owned Accounts (EOA) are controlled by private key and can be used by anyone having the private key. Contract Accounts has an associated code that executes when it receives a transaction from an EOA.

A contract account cannot initiate the transaction on its own as they must always originate from an EOA. However, contracts can call other contracts as part of their execution.

Once a smart contract is deployed onto the Ethereum network, it cannot be edited or corrected nor stopped even by its original author. It's immutable.

The only way to change it would be by convincing the entire Ethereum network that a change would be made and that is virtually impossible.

**Ethereum Transaction Flow**

Transactions require a fee to be paid in order to be executed, this is because it does require the nodes in the network to do some kind of

Transaction in the Ethereum network consists of multiple fields.

**Recipient field**, to whom the transaction is being sent to and this could be either another EOA which would be a transfer value transaction or to a contract to trigger a code to run or execute.

**Value field**, holds the amount of Ether to be transacted.

**Nonce field**, a nonce is a unique number that is required for every single transaction. For EOA’s this number is the amount of transactions sent from the account and for contract accounts it is the number of contracts created by the account. The nonce value is critical because a different value is needed for every transaction to essentially prevent the network from processing the same transaction over and over again as an attempt to gain more fees.

**Signature field**, this contains a digital signature generated from the private key that allows the account owner to authorize the transaction.

**Data field**, an optional field used to include any arbitrary data that is used primarily whenever you’re calling smart contracts to indicate which functions to run, parameters to use and so on.

**gasLimit field**, gas is the fee required to process a transaction and it’s based on how much computational effort is required by the network. A gas limit indicates the maximum amount to be spent on the transaction.

**Ethereum Use Cases**

1. Decentralized Finance **(DeFi)**, preventing any financial institution or bank from executing payments.
2. Decentralized Apps **(DAPPS)**, digital applications that run on a blockchain network of computers instead of relying on a single computer.
3. Non-Fungible Tokens **(NFTs)**, tokens that are used to show ownership of unique possessions.

**Hyperledger Fabric VS Ethereum**

**Network**

Ethereum is a **public/private blockchain** without any permissions, anyone with an internet connection and a computer can participate in the network.

Hyperledger Fabric is a **consortium blockchain**, only added organizations can participate in the network.

**Confidentiality**

Ethereum does not support confidential transactions.

Hyperledger Fabric supports confidential transactions.

**Cryptocurrency**

Ethereum’s currency is **Ether**, used to pay for transactions and buy gas which is in turn used to pay for computations committed on the Ethereum Virtual Machine **(EVM)**.

Hyperledger Fabric has no specific currency, a freedom of implementing any blockchain technology kind is granted for the developers.

**Consensus Mechanism**

Ethereum uses **Proof-Of-Work** mechanism for its consensus. New created Ether is rewarded for participants that publicly verify the transaction on behalf of the entire network.

Hyperledger Fabric allows developers to choose whether they require a consensus mechanism at all. If a consensus mechanism is needed, an algorithm called **Practical Byzantine Fault Tolerance (PB FD)** is used.

**Smart Contracts & Languages**

Ethereum uses a combination of GoLang and Python and their smart contracts are written in specific scripting languages (Solidarity). Hyperledger Fabric uses a combination of GoLang and Java and their smart contracts are written in chain codes.