**ETHEREUM:** <https://vn.zastrin.com/courses/vn-ethereum-primer/lessons/1-1>

\*Ý tưởng:

Ngay sau khi Bitcoin ra đời, chủ yếu áp dụng cơ chế đồng thuận phân tán trong giao dịch tiền ảo. Vậy tại sao không thể mở rộng nó ra ngoài lĩnh vực tài chính, khi có thể trao đổi các dữ liệu khác theo cơ chế này. Nhu cầu áp dụng cơ chế này vào việc có thể hiện thực các ứng dụng mang nhiều ý nghĩa thực tế khác trên mạng Blockchain là hoàn toàn khả thi nhưng mạng Bitcoin không đáp ứng. Do đó Ethereum ra đời cho các ứng dụng DApp và có cả việc giao dịch đồng ETH.

\*Khái niệm:

Ethereum là một nền tảng điện toán phân tán public dựa trên blockchain. Có thể coi đó là một máy tính lớn cấu thành bởi các máy tính nhỏ trên toàn thế giới. Có thể viết các ứng dụng và chạy chúng trên chiếc máy tính toàn cầu này. Nền tảng này đảm bảo rằng ứng dụng của bạn sẽ chạy mà không gặp phải downtime, kiểm duyệt, lừa đảo hay sự can thiệp từ bên thứ ba. Bên cạnh việc chạy các ứng dụng, Ethereum blockchain còn có thể chuyển tiền giữa hai bên mà không cần một đơn vị trung gian.

Tất cả các máy tính này (hay còn gọi là node) được kết nối tới nhau và có một bản copy đầy đủ các mã và dữ liệu. Khi deploy mã trên Ethereum blockchain, mã này được nhân bản trên tất cả các node trên mạng lưới. Khi ứng dụng của bạn lưu trữ các dữ liệu bất kì, cả các dữ liệu đó cũng được nhân bản trên tất cả node. Có hàng ngàn node trên mạng lưới và điều đó khiến việc dừng tất cả các node là không thể với bất kì ai. Điều này đảm bảo rằng ứng dụng của chúng ta sẽ luôn có thể truy cập được.

\*Cấu trúc mạng Ethereum:

Mỗi client browser giao tiếp với chính instance của nó trên ứng dụng. Không có một server trung tâm nào để các client kết nối tới. Điều này có nghĩa là, trong thế giới phi tập trung lí tưởng, mỗi người mà muốn tương tác với dapp (Ứng dụng Phi tập trung – Decentralized Application) sẽ cần một bản copy đầy đủ của blockchain chạy trên máy tính/điện thoại của họ. Nghĩa là, trước khi có thể sử dụng một ứng dụng, bạn cần phải download toàn bộ blockchain và sau đó bắt đầu sử dụng ứng dụng. Chúng ta không sống trong một thế giới lí tưởng và sẽ không hợp lí khi kì vọng tất cả mọi người phải chạy một blockchain server để sử dụng các ứng dụng này. Nhưng ý tưởng đằng sau phi tập trung là để không phụ thuộc vào một server riêng lẻ/tập trung. Vậy, cộng đồng này đã tìm ra các giải pháp (hosted blockchain server, metamask …) mà ở đó bạn sẽ không phải tốn một phần lớn ổ cứng và RAM để download và chạy một bản copy đầy đủ của blockchain nhưng cũng không phải thỏa hiệp về khía cạnh phi tập trung. Bây giờ, chính xác thì Ethereum blockchain là gì? Blockchain có 2 thành phần chính:

1. Database: Mỗi giao dịch trong network được lưu trên blockchain. Khi bạn deploy ứng dụng của mình, nó sẽ được coi là một giao dịch. Nếu bạn có, giả sử một ứng dụng bầu cử mà cho phép bất cứ ai bầu cho các ứng viên, một vote cho một ứng viên có thể được coi là một giao dịch. Tất cả các giao dịch này được public và bất cứ ai cũng có thể xem giao dịch này và có chính bản dữ liệu đó và để đảm bảo không có dữ liệu không hợp lệ nào được ghi vào database này, Ethereum sử dụng một thuật toán gọi là Proof of Work để bảo vệ mạng lưới. (<http://ethereum.stackexchange.com/questions/14/what-proof-of-work-function-does-ethereum-use>)

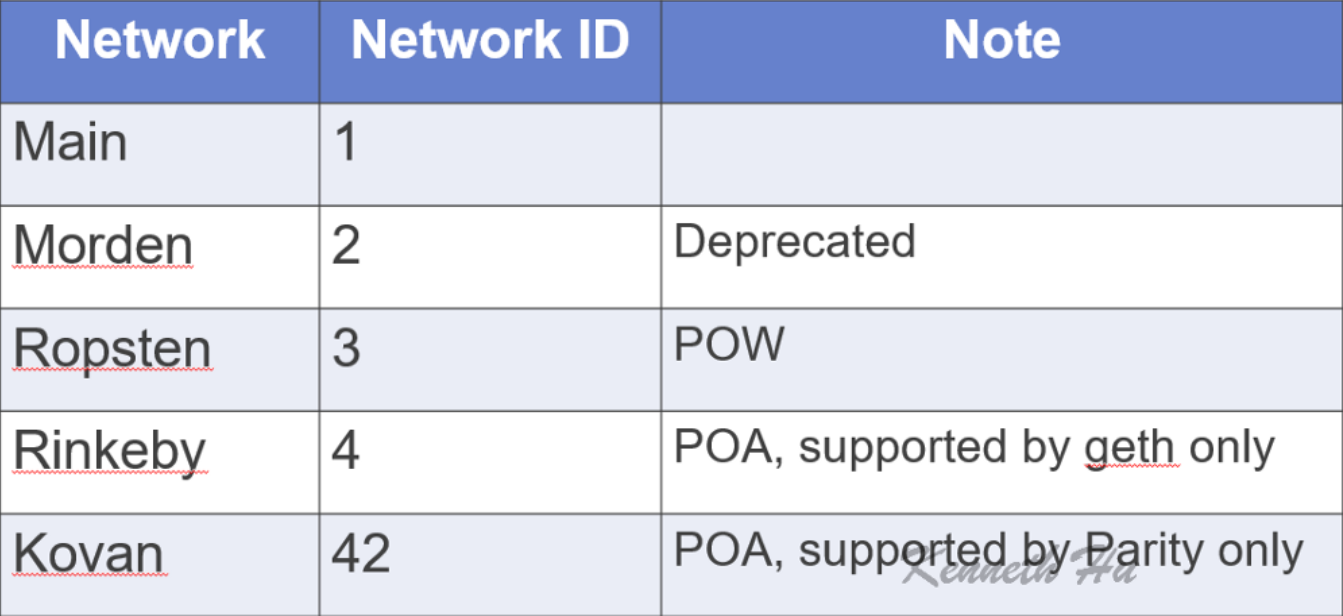
2. Code: Database của blockchain chỉ lưu trữ các giao dịch. Vậy còn các logic để vote cho ứng viên, gọi tổng số lượt vote… ở đây? Trong thế giới Ethereum, khi viết các code ứng dụng/logic (gọi là contract) trong một ngôn ngữ gọi là Solidity. Sau đó bạn sử dụng solidity compiler để compile nó thành Ethereum Byte Code và sau đó deploy byte code đó tới blockchain (Có một vài ngôn ngữ khác có thể dùng để viết contract nhưng tới nay solidity là ngôn ngữ phổ biến nhất và tương đối dễ dùng). Vậy, Ethereum blockchain không chỉ lưu trữ giao dịch, nó còn lưu trữ và thực hiện các contract code. Vậy về cơ bản, blockchain lưu trữ dữ liệu, lưu trữ code và đồng thời chạy code trên EVM (Ethereum Virtual Machine). Để xây dựng các Dapps trên web, Ethereum cung cấp một thư viện javascript hữu dụng gọi là web3.js, nó kết nối tới blockchain node. Vậy chỉ cần bao gồm thư viện này trong js framework nổi tiếng như reactjs, angularjs… và bắt đầu xây dựng.

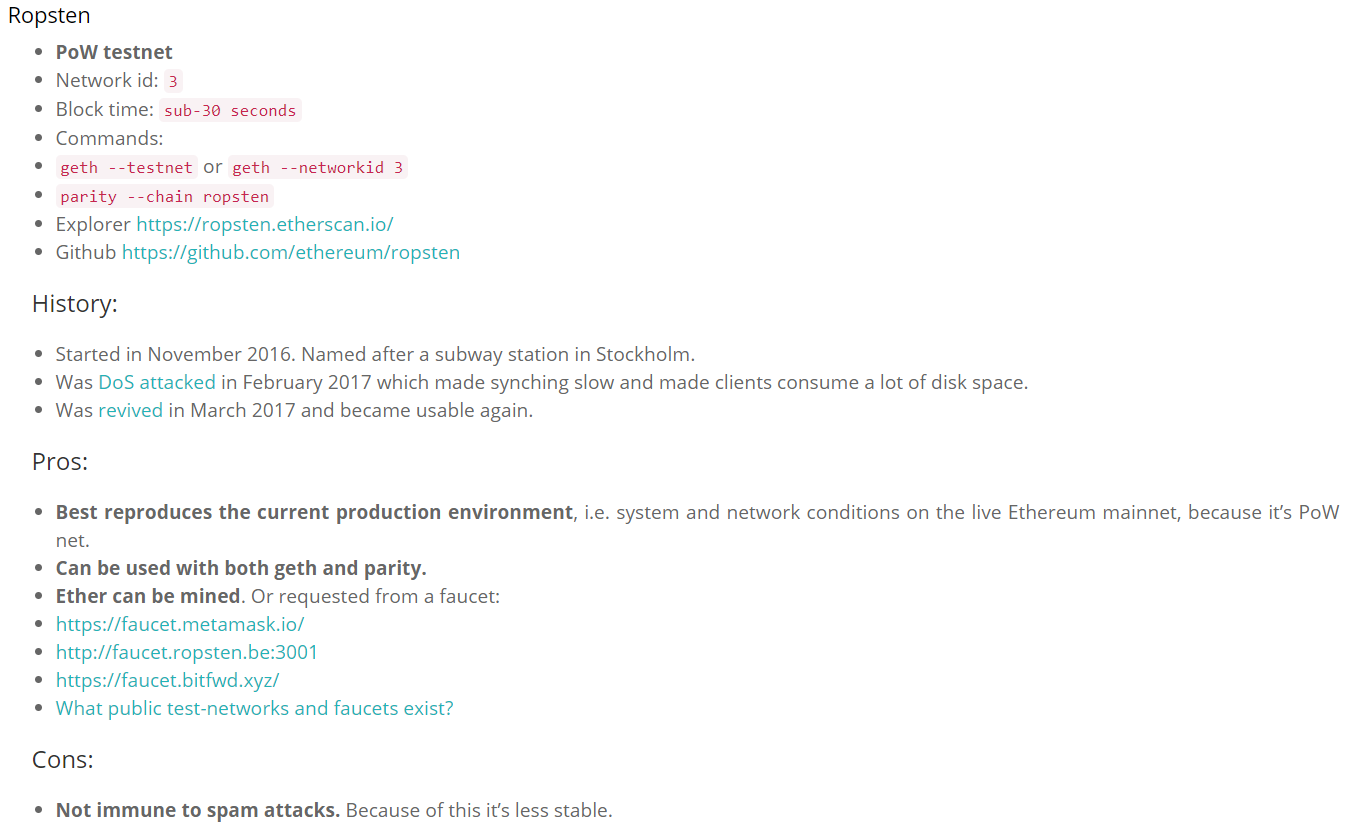
**CÁC MẠNG ETHEREUM:**

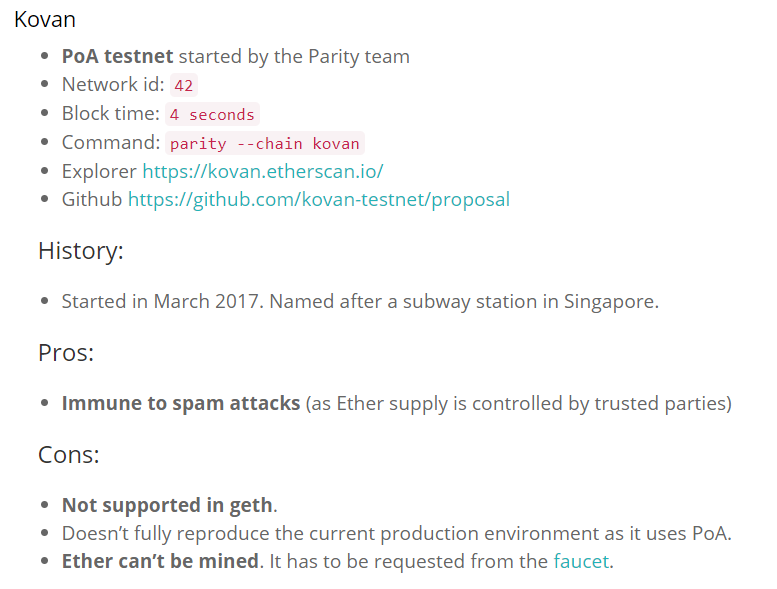
\*Mainnet:

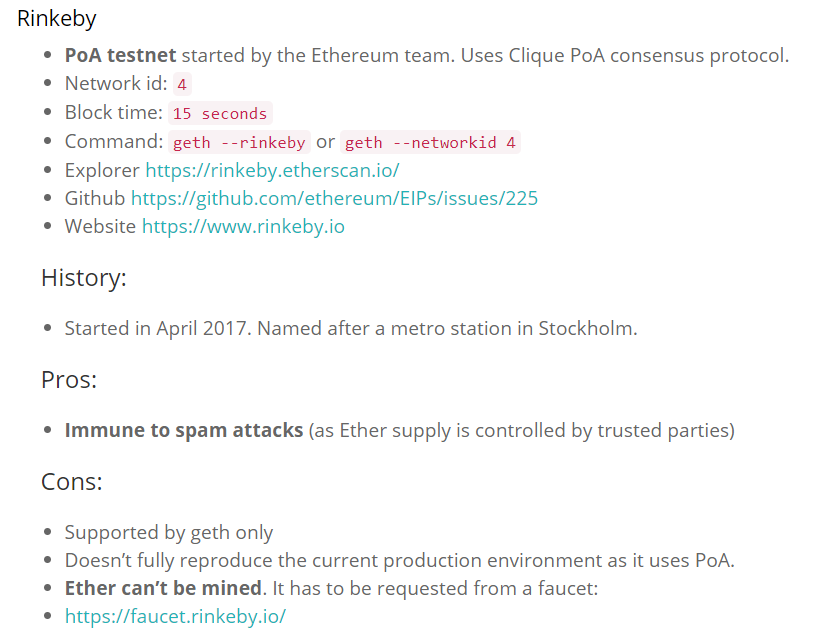
Mạng chính của Ethereum để thực hiện các giao dịch chính thức với đồng ETH.

\*Testnet: Mạng test để thực hiện việc test, kiểm thử các giao dịch và xin Ethereum miễn phí một cách dễ dàng.









\*Một số khái niệm cần phân biệt:

- Các cơ chế đồng thuận:

+ Proof of Work (POW):

Còn gọi là bằng chứng công việc. Nền tảng đồng Bitcoin, giải bài toán đồng thuận để tạo block mới và nhận thưởng. Xem them chi tiết:

<https://blogtienao.com/proof-of-work-pow-la-gi/>

+ Proof of Authority (POA):

Thuật toán bằng chứng ủy quyền, đề cao danh tính hơn giá trị tài sản. Có hiệu suất cao hơn tuy nhiên phải đánh đổi việc đánh mất cơ chế phi tập trung. Xem them chi tiết:

<https://bitcoinvietnamnews.com/proof-of-authority-la-gi>

- Các nền tảng client:

+ Geth:

Geth là client software chính thức cung cấp bởi Ethereum Foundation. Nó được viết trên ngôn ngữ lập trình Go. Khi bạn khởi động Geth, nó kết nối với các client Ethereum khác (còn gọi là node) trên network và tải một bản copy của blockchain. Nó sẽ liên tục giao tiếp với các node khác để giữ các bản copy của blockchain được cập nhật. Nó cũng có khả năng đào block và thêm các giao dịch vào blockchain, xác thực các giao dịch trong block và cũng thực thi các giao dịch. Nó cũng đóng vai trò như một server bằng cách đưa ra các API bạn có thể tương tác với thông qua RPC. Nó đi kèm với javascript client (geth console) có thể dùng để kết nối với blockchain.

+ Parity:

Parity là một phần mềm tốt khác của giao thức Ethereum và được viết trên ngôn ngữ lập trình Rust. Nó được duy trì bởi một công ty tên là Parity Inc. Bất cứ ai cũng có thể thực hiện client software và tham gia Ethereum network. Có các client viết trên C++ và Python và các ngôn ngữ khác. Nếu muốn bạn có thể thực hiện client của chính bạn theo các chi tiết tại yellow paper! Một minh họa đơn giản của Ethereum network với các client kết nối với nhau.

**BlockExplorer (Etherscan):**

Etherscan là một ứng dụng frontend hữu ích (trang web) cung cấp một dịch vụ miễn phí bởi một công ty gọi là Etherscan để khám phá Ethereum blockchain. Nếu bạn tương tác với Ethereum blockchain public, tất cả các địa chỉ, các giao dịch, chi tiết khối v.v. có thể được xem trên website đó. Bạn có thể nhìn thấy bản minh họa của Ethereum network và cách Etherscan hòa nhập vào network đó. Bạn có thể thấy các đường dẫn tới Etherscan frontend cho các network public khác nhau dưới đây:

Mainnet: <https://etherscan.io/>

Ropsten: <https://ropsten.etherscan.io/>

Rinkeby: <https://rinkeby.etherscan.io/>

Kovan: <https://kovan.etherscan.io/>

**Block**

Tốc độ tạo block mới Ethereum khoảng 14s (trong khi đó Bitcoin 10mins)

Không gian block nhỏ hơn (~1MB) với Gas Limit ~ 1 500 000 gas. Mỗi transaction khoảng 21 000 gas. Do đó mỗi Block sẽ có khoảng 70 transaction để làm đầy Block (1 500 000/21 000). Với bitcoin con số này khoảng 1500 – 2000 transaction mỗi block.

Mỗi gas có giá gas price khoảng 0.1 – 100+ Gwei

**Transaction**

<https://topdev.vn/blog/transaction-trong-ethereum-xay-ra-nhu-the-nao/>

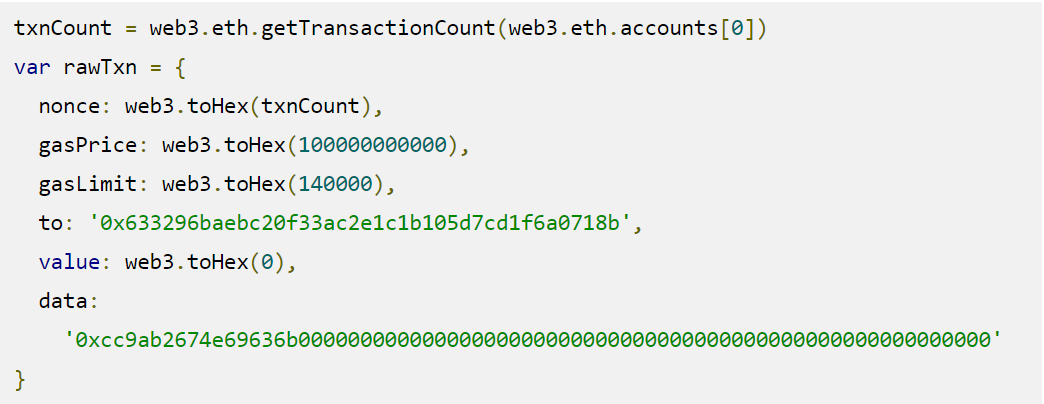
Transaction là một hành động giao dịch dữ liệu

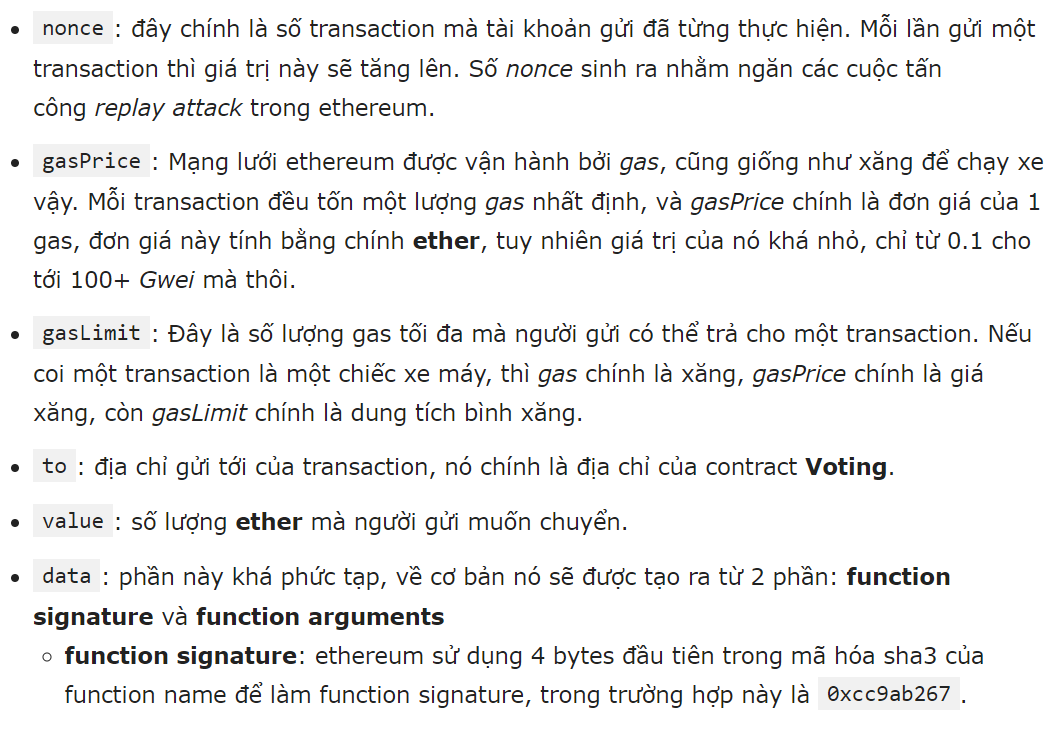
Hiểu một cách đơn giản, toàn bộ các hoạt động chuyển tiền, nhận tiền, sinh contract, tạo token…. trên mạng ethereum đều là transaction. Những transaction này sẽ được ghi lại vào trong các block và tồn tại mãi mãi không ai thay đổi được.

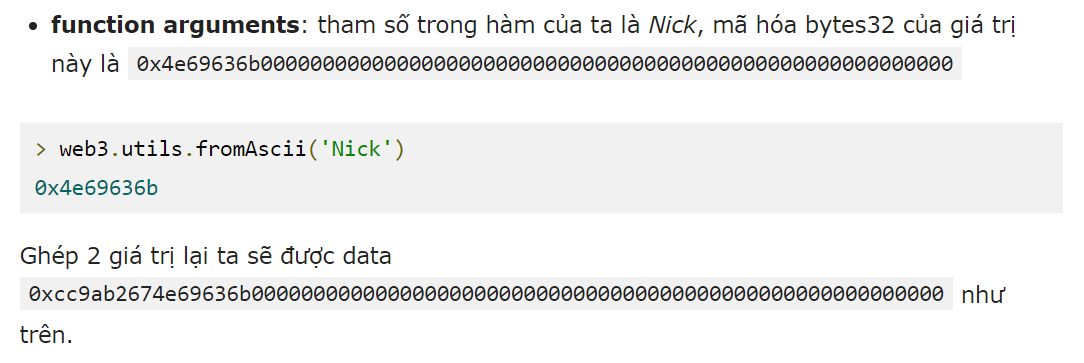
**Hoạt động gửi và xác nhận transaction**

Khi có một smart contract. Ta gọi hàm deploy lên mạng Ethereum để tạo thành một transaction. Transaction đó sẽ được broad cast đi toàn bộ các node trong mạng lưới Ethereum.

Thư viện web3js chuyển transaction sang dạng raw transaction object gồm các nội dung:







Đoạn mã trên là Txn Hash (4 bytes đầu mã sha3 của tên hàm function name + mã hóa bytes32 của tham số function arguments) là id đặc trưng cho mỗi transaction để tìm kiếm và kiểm tra thông tin giao dịch.

Sau đó, để xác thực chủ nhân thực hiện transaction thông qua việc ký transaction bằng private key

Bước tiếp theo, là xác nhận lại local node để đảm bảo mọi thứ hợp lệ trước khi gửi lên toàn mạng Ethereum

Sau đó, Broadcast transaction đi toàn bộ network và chúng ta có thể theo dõi quá trình thông qua etherscan.io.

Toàn bộ các constraction được đưa vào một nơi gọi mining pool với sự sắp xếp chi phí xác nhận cho các miner từ cao đến thấp.

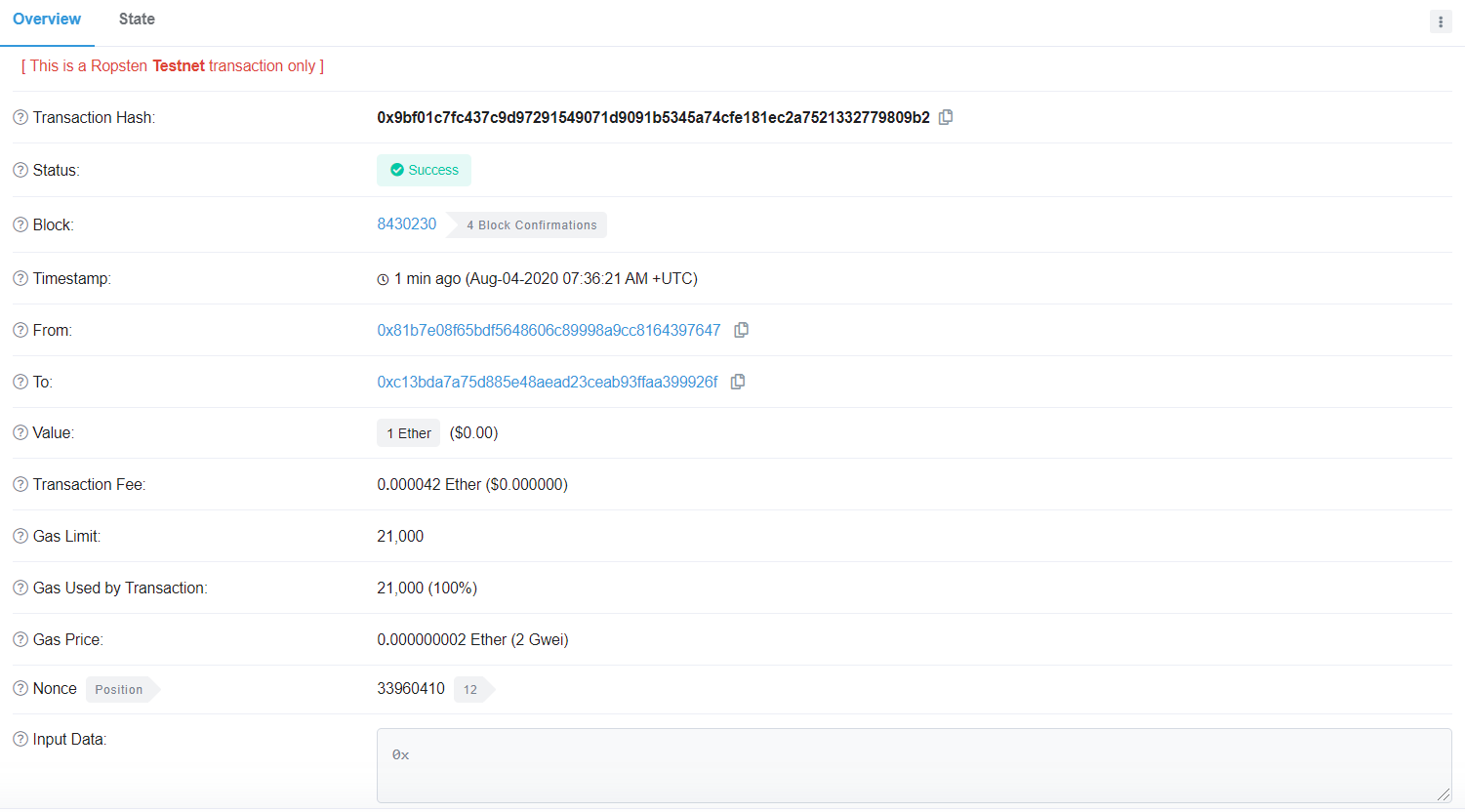
Khi đó các miner node sẽ xem xét mức giá giao dịch và chấp nhận giao dịch của chúng ta từ đó tính toán gom các constraction khác đủ để hình thành một block hợp lệ.

Sau khi miner tính toán Proof of work đủ điều kiện để tìm ra một block mới hợp lệ tối đa hóa lợi nhuận thì Broadcast nó đi toàn hệ thống và hình thành gắn vào chuỗi (chain) có chứa hash của block trước cuối cùng trên mạng Ethereum. Khi đó, local node của chúng ta sẽ nhận về block chứa transaction của mình.

Sử dụng metamask (wallet) để thay thế cho local node nhằm tiết kiệm không gian dữ liệu máy tinh cho việc copy toàn bộ dữ liệu mạng BC Ethereum và tạo chữ ký trực tiếp trên browser.

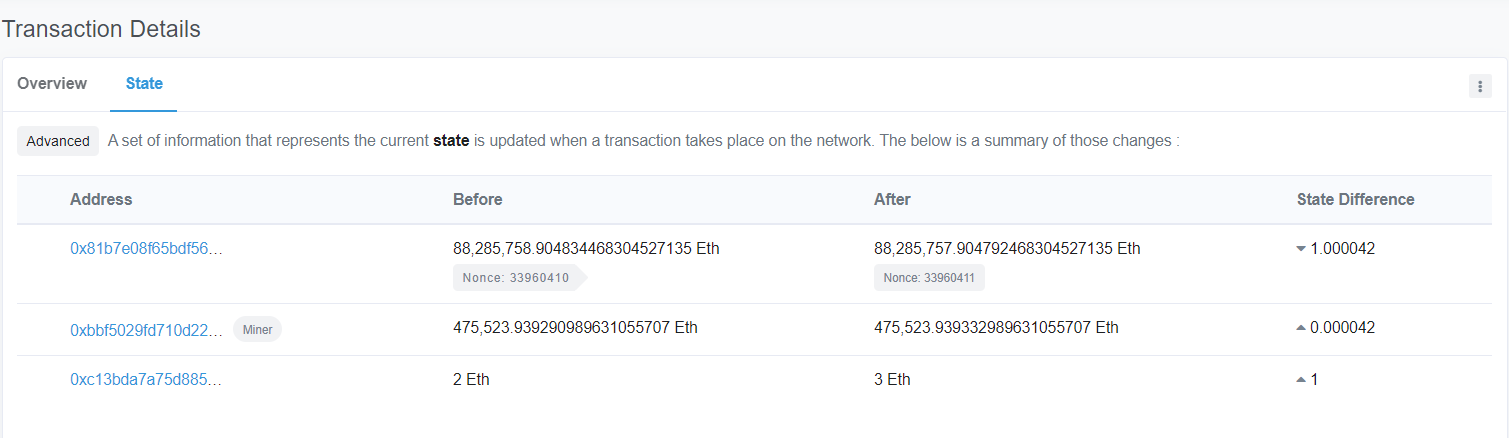
**Receipt Transaction**

Thể hiện các thông tin biên lai cơ bản của một giao dịch transaction. Thể hiện như hình dưới gồm các thông số cần biết:



**Log Transaction**

Ghi lại nhật ký, trạng thái hoạt động về quy trình đường đi của một transaction để người dùng theo dõi. Bao gồm: Chênh lệch số dư của địa chỉ gửi, của địa chỉ miner và của địa chỉ nhận. Từ đó biết được chính xác transaction của chúng ta thực sự đang nằm ở đâu và đã được xử lý thành công hay chưa.



**Token**

<https://coin98.net/ethereum-la-gi/#2_Ethereum_Token_Standard>

Token là **tài sản số (các loại đồng tiền)** gồm 2 loại:

- Native Token (đồng tiền ảo gốc chính thức đang phát hành có giá trị giao dịch trao đổi vd ETH, BTC)

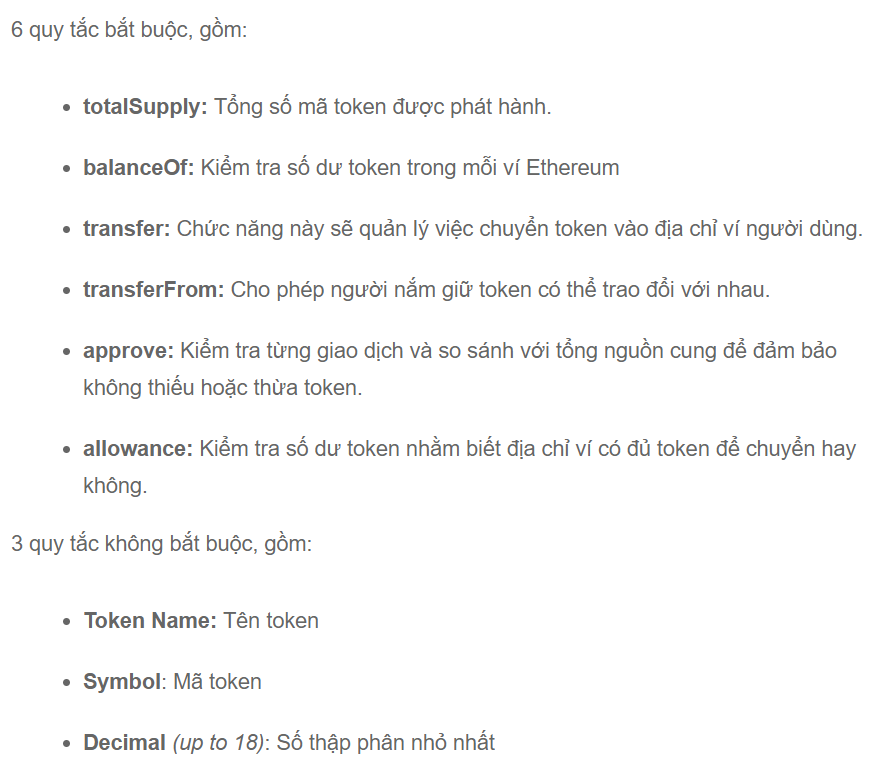
Nền tảng Blockchain Ethereum tổng cộng 72 triệu ETH. Team phát triển giữ lại 12 triệu ETH. SỐ còn lại phát hành ra thị trường thông qua ICO (phát hành đồng tiền token để gây quỹ cho dự án công ty).

- Smart Contract Token (tài sản hợp đồng thông minh). Có các file dạng \*.sol

Là các loại đồng tiền (token) được các nhà triển khai dự án tự phát hành đồng tiền của riêng mình. Khi tạo token phải niêm yết ngay từ đầu sẵn tổng giá trị của token đó và số lượng đồng tiền sẽ phát hành ra thị trường. Nhà đầu tư sẽ mua loại token này bằng ETH, BTC hay USD như một hình thức gọi vốn. Khi dự án phát triển và có lãi thì giá trị đồng token của riêng họ sẽ tăng lên và thu về lợi nhuận cho nhà đầu tư khi giao dịch chúng.

Có các bộ quy tắc tạo token như:

+ Linh hoạt Fungile: Bộ qui tắc ERC20 (Ethereum Request for Comments 20). Bùng nổ với hiện tượng đầu tư ICO với token có thể trao đổi với nhau.



+ Không linh hoạt Non-fungible: <https://viblo.asia/p/nhung-dieu-nen-biet-ve-erc721-OeVKBqEE5kW>

ERC721 xác thực tài sản số không thể trao đổi giá trị (vd tài sản số đại diện cho iteam A không thể trao đổi ngang hang với tài sản số đại diện cho item B)

Non-Fungible Token (NFT) được hiểu là một loại token đại diện cho một thứ gì đó độc nhất. Vì vậy, một NFT này không thể hoán đổi, thay thế bởi một NFT khác. Ví dụ: Một token đại diện cho vé xem phim Spiderman không thể hoán đổi với một token đại diện cho vé xem phim Avenger được. Nhờ có tiêu chuẩn ERC721, các nhà phát triển trên Ethereum đã mở ra một hệ sinh thái mới về các dapps sử dụng các NFTs.

Đồng token đại diện cho item có giá trị vật lý thực tế cụ thể. Ví dụ: tác phẩm nghệ thuật, vật phẩm game,… các tài sản vật lý khác có thể dùng đồng token đại diện giá trị luận lý.

**EIPs**

Các cải tiến nhà phát hành Ethereum đưa ra sau mỗi đợt hardfork chia mạng lưới blockchain ở mainnet.

**Smart Contract (Hợp đồng thông minh)**

Trong phần trước, chúng ta đã thảo luận cách chúng ta có thể viết các ứng dụng trên ngôn ngữ lập trình Solidity và deploy nó tới Ethereum blockchain. Trong Ethereum, các ứng dụng này được gọi là Smart Contract. Chắc hẳn đang thắc mắc tại sao các ứng dụng này được gọi là smart contract. Nói chung, hợp đồng là một thỏa thuận bằng văn bản giữa hay hoặc nhiều bên và được buộc thi hành bởi pháp luật. Nếu chúng ta lấy hợp đồng bằng văn bản này và dịch nó thành code và deploy tới blockchain, chúng ta có các hợp đồng điện tử. Nhưng cái hay của code này trên blockchain là nó có thể khiến thỏa thuận giữa các bên có hiệu lực và đó là lí do chúng được gọi là smart contract. Code này làm cách nào để thỏa thuận có hiệu lực? Một khi một hợp đồng được deploy tới blockchain, nó không thể hoặc bị dừng lại hay thay đổi. Đó là cách thỏa thuận được thi hành. Hãy cùng xem một ví dụ đơn giản. Giả sử chúng ta muốn xây một ứng dụng Crowdsale. Có một người tạo lập (creator) muốn huy động $10,000 để tạo ra một sản phẩm và có các khách hàng tiềm năng quan tâm tới sản phẩm này và có thiện chí trả trước cho nó. Thỏa thuận là, mỗi khách hàng (contributor) sẽ đóng góp từ $10 - $10,000 và nếu mục đích của họ được thỏa mãn thì số tiền sẽ được gửi tới creator. Nếu không, số tiền sẽ được hoàn lại cho các contributor. Đây là một trường hợp lí tưởng để sử dụng smart contract để thực hiện thỏa thuận giữa creator và các supporter. Việc thực hiện logic này trên một ứng dụng tập trung cũng không đáng kể. Các lợi ích của việc thực hiện trên smart contract là:

1. Khoản tiền được gửi bởi các contributor nằm trong smart contract. Cả contributor hay creator đều không được lấy số tiền đó.

2. Code có thể được thực hiện để số tiền được gửi cho creator ngay khi mục tiêu đạt được.

3. Code có thể được thực hiện để bất kì ai cũng có thể yêu cầu hoàn lại tiền cho các khách hàng nếu như mục tiêu không đạt được trong một khung thời gian nhất định. Code này không thể bị can thiệp một khi đã được deploy tới blockchain.