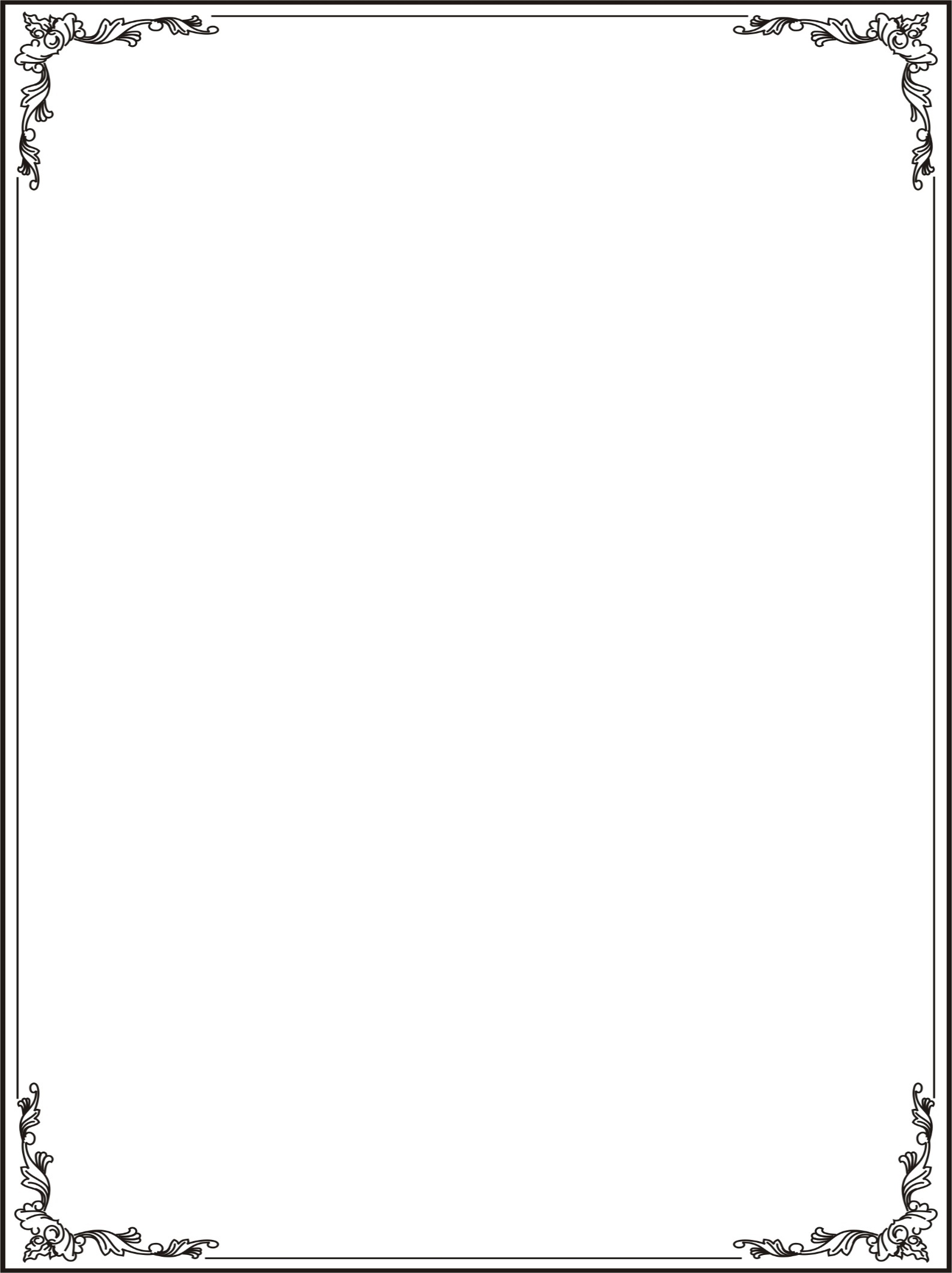
**

HỌC VIỆN KỸ THUẬT MẬT MÃ

**KHOA AN TOÀN THÔNG TIN**



BÁO CÁO

**CÔNG NGHỆ BLOCKCHAIN**

**Đề tài:**

**Tìm hiểu và triển khai ứng dụng các tiêu chuẩn Ethereum Request for Comments**

***Nhóm sinh viên thực hiện:*** Vũ Tiến Đạt AT170609

Lê Xuân Hinh AT170418

Nguyễn Hữu Nam AT170434

Nhóm 02

Hà Nội, 2024

# MỤC LỤC

[MỤC LỤC 1](#_Toc169630888)

[LỜI MỞ ĐẦU 2](#_Toc169630889)

[DANH MỤC TỪ VIẾT TẮT 3](#_Toc169630890)

[DANH MỤC HÌNH ẢNH 4](#_Toc169630891)

[CHƯƠNG 1. TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN 5](#_Toc169630892)

[1.1 Giới thiệu chung 5](#_Toc169630893)

[1.1.1 Khái niệm 5](#_Toc169630894)

[1.1.2 Vai trò 5](#_Toc169630895)

[1.2 Các kỹ thuật mật mã học trong blockchain 6](#_Toc169630896)

[1.2.1 Mật mã khóa công khai 6](#_Toc169630897)

[1.2.2 Hàm băm 6](#_Toc169630898)

[1.2.3 Chữ ký số 7](#_Toc169630899)

[1.2.4 Kiến trúc cơ bản của hệ thống blockchain 10](#_Toc169630900)

[1.2.5 Cơ chế hoạt động của hệ thống Blockchain 14](#_Toc169630901)

[1.3 Tổng kết chương 1 15](#_Toc169630902)

[CHƯƠNG 2. ETHEREUM VÀ CÁC TIÊU CHUẨN EHEREUM REQUEST FOR COMMENTS 18](#_Toc169630903)

[2.1 Giới thiệu chung về nền tảng blockcahin Ethereum 18](#_Toc169630904)

[2.1.1 Khái niệm 18](#_Toc169630905)

[2.1.2 Ethereum Virtual Machine 18](#_Toc169630906)

[2.1.3 Ethereum và hợp đồng thông minh. 20](#_Toc169630907)

[2.2 Các tiêu chuẩn Ethereum Request for Comments- ERC 24](#_Toc169630908)

[2.2.1 Tiêu chuẩn ERC-20 (Token Standard) 24](#_Toc169630909)

[2.2.2 Tiêu chuẩn ERC-721 (Non-Fungible Token Standard) 27](#_Toc169630910)

[2.2.3 Tiêu chuẩn ERC-777 (Token Standard) 30](#_Toc169630911)

[2.2.4 Tiêu chuẩn ERC-1155 (Multi-Token Standard) 33](#_Toc169630912)

[CHƯƠNG 3. TRIỂN KHAI ỨNG DỤNG CÁC TIÊU CHUẨN ERC 40](#_Toc169630913)

[3.1 Triển khai ứng dụng ERC-20 40](#_Toc169630914)

[3.1.1 Xây dựng Smart Contract theo ERC-20 40](#_Toc169630915)

[3.1.2 Triển khai hợp đồng KMACoin Tiến hành Deploy hợp đồng trên mạng thử nghiệm Sepolia 41](#_Toc169630916)

[3.2 Triển khai ứng dụng ERC-721 44](#_Toc169630917)

[3.2.1 Xây dựng Smart Contract theo chuẩn ERC-721 44](#_Toc169630918)

[3.2.2 Triển khai hợp đồng theo chuẩn ERC-721 47](#_Toc169630919)

[3.3 Triển khai ứng dụng ERC-1155 49](#_Toc169630920)

[3.4 Tổng kết chương 3 57](#_Toc169630921)

[TỔNG KẾT 58](#_Toc169630922)

[TÀI LIỆU THAM KHẢO 59](#_Toc169630923)

# LỜI MỞ ĐẦU

**1. Tính cấp thiết của đề tài.**

Trong thời đại số hóa ngày nay, blockchain đang trở thành một công nghệ quan trọng trong việc tạo ra các ứng dụng phi tập trung và đảm bảo tính bảo mật trong giao dịch trực tuyến. Blockchain đã được sử dụng trong nhiều lĩnh vực, từ tài chính đến y tế và cả quản lý chuỗi cung ứng.

Xu hướng phát triển của blockchain là tăng cường tính bảo mật và độ tin cậy trong các giao dịch trực tuyến. Nó hỗ trợ việc tạo ra các ứng dụng phi tập trung, giúp giảm thiểu chi phí và thời gian xử lý giao dịch, đồng thời tăng tính minh bạch và khả năng kiểm tra trong các giao dịch.

Các tiêu chuẩn ERC là các tiêu chuẩn được đề xuất bởi cộng đồng Ethereum để quản lý các token trên nền tảng Ethereum. Các tiêu chuẩn này đảm bảo tính đồng nhất và đáng tin cậy trong các sản phẩm và dịch vụ blockchain, giúp tăng tính tương thích giữa các sản phẩm và dịch vụ tại các công ty khác nhau, giúp tạo ra một hệ sinh thái blockchain đồng nhất và hiệu quả.

**2. Mục tiêu nghiên cứu của đề tài**

Triển khai ứng dụng các tiêu chuẩn ERC. Triển khai các hợp đông thông minh dựa trên các tiêu chuẩn ERC-20. ERC-721 và ERC-1155.

# DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| P2P | Peer to peer |
| DNS | domain name system |
| EVM | Ethereum Virtual Machine |
| ROM | Read-Only Memory |

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Mật mã khóa công khai 7](#_Toc169630815)

[Hình 1.2 Hàm băm 8](#_Toc169630816)

[Hình 1.3 Các bước thực hiện chữ ký số 9](#_Toc169630817)

[Hình 1.4 Các bước giải mã chữ ký số 10](#_Toc169630818)

[Hình 1.5 Mô hình mạng ngang hàng 11](#_Toc169630819)

[Hình 1.6 Block 13](#_Toc169630820)

[Hình 1.7 Cơ chế hoạt động của Blockchain 16](#_Toc169630821)

[Hình 2.1 Cơ chế hoạt động của EVM 20](#_Toc169630822)

[Hình 2.2 Phí gas 21](#_Toc169630823)

[Hình 3.1 Tạo Interface ERC-20 41](#_Toc169630824)

[Hình 3.2 Xây dựng hợp đồng dựa trên chuẩn ERC-20 41](#_Toc169630825)

[Hình 3.3 Kết nối với ví metamask 42](#_Toc169630826)

[Hình 3.4 Xác nhận hợp đồng với metamask 42](#_Toc169630827)

[Hình 3.5 Kiểm tra hợp đồng trên ethereum scan 43](#_Toc169630828)

[Hình 3.6 Nhập địa chỉ Token đã tạo vào Metamask 43](#_Toc169630829)

[Hình 3.7 Nhập token thành công trên metamask 44](#_Toc169630830)

[Hình 3.8 Gửi Token cho 1 tài khoản khác 44](#_Toc169630831)

[Hình 3.9 Kiểm tra trạng thái gửi token 45](#_Toc169630832)

[Hình 3.10 Deploy hợp đồng thwo chuẩn ERC-721 48](#_Toc169630833)

[Hình 3.11 Tạo 1 NFT qua hàm safeMint 49](#_Toc169630834)

[Hình 3.12 Kiểm tra số lượng NFT có trong ví 49](#_Toc169630835)

[Hình 3.13 Kiểm tra chủ sở hữu của NFT “0” 49](#_Toc169630836)

[Hình 3.14 Chuyển NFT ”0” từ ví 1 sang ví 2 50](#_Toc169630837)

[Hình 3.15 Kiểm tra NFT “0” ở ví 2 50](#_Toc169630838)

[Hình 3.16 Viết đoạn mã Solidity cho smart contract 51](#_Toc169630839)

[Hình 3.17 Biên dịch smart contract 52](#_Toc169630840)

[Hình 3.18 Triển khai smart contract 52](#_Toc169630841)

[Hình 3.19 Kết nối Remix với ví metamask 53](#_Toc169630842)

[Hình 3.20 Trả phí gas cho hợp đồng 53](#_Toc169630843)

[Hình 3.21 Kiểm tra hợp đồng đã triển khai 54](#_Toc169630844)

[Hình 3.22 Kiểm tra token ID = 6 trên ví gửi 54](#_Toc169630845)

[Hình 3.23 Kiểm tra token ID = 6 trên ví nhận 55](#_Toc169630846)

[Hình 3.24 Chuyển 5 token có ID = 6 56](#_Toc169630847)

[Hình 3.25 Trả fee để xác nhận giao dịch 56](#_Toc169630848)

[Hình 3.26 Kiểm tra giao dịch 57](#_Toc169630849)

[Hình 3.27 Kiểm tra token ở ví nhận 57](#_Toc169630850)

[Hình 3.28 Kiểm tra token ở ví gửi 57](#_Toc169630851)

# TỔNG QUAN VỀ CÔNG NGHỆ BLOCKCHAIN

## Giới thiệu chung

### Khái niệm

Theo bách khoa toàn thư mở Wikipedia, Blockchain được định nghĩa: “Blockchain (chuỗi khối), tên ban đầu block chain là một cơ sở dữ liệu phân tán lưu trữ thông tin trong các khối thông tin được liên kết với nhau bằng mã hóa và mở rộng theo thời gian. Mỗi khối thông tin đều chứa thông tin về thời gian khởi tạo và được liên kết tới khối trước đó, kèm một mã thời gian và dữ liệu giao dịch. Blockchain được thiết kế để chống lại sự thay đổi của dữ liệu: Một khi dữ liệu đã được mạng lưới chấp nhận thì sẽ không có cách nào thay đổi được nó.”.

### Vai trò

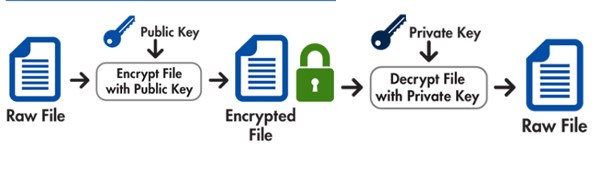
Nếu như trước đấy, các công nghệ cơ sở dữ liệu truyền thống đặt ra nhiều thách thức trong việc ghi lại các giao dịch tài chính. Chẳng hạn như hãy xét trường hợp bán một tài sản. Sau khi đã giao tiền, quyền sở hữu tài sản được chuyển cho người mua. Cả người mua và người bán đều có thể từng người ghi lại các giao dịch tiền tệ, nhưng không nguồn nào là đáng tin cậy. Người bán có thể dễ dàng khẳng định rằng họ chưa nhận được tiền ngay cả khi họ đã nhận được và người mua cũng có thể phản bác rằng họ đã chuyển tiền ngay cả khi họ chưa thanh toán. Sự xuất hiện của blockchain đóng vai trò thực hiện và ghi lại các giao dịch của người dùng và đồng bộ trên các máy tại các điểm nút trên hệ thống mạng blockchain. Nói một cách khác, giao dịch sẽ được thực hiện mà không cần một bên thứ ba xen vào.

Trái với mô hình mạng Client-Server, dữ liệu được lưu trên một máy chủ rất dễ bị quản trị viên hay người có thẩm quyền lạm dụng quền thay đổi nội dung dữ liệu, công nghệ blockchain là một hệ thống mạng máy tính dựa trên mô hình mạng phi tập trung, mỗi máy tính của người dùng tham gia mạng đóng vai trò là một điểm nút trên hệ thống mạng. Trên hệ thống mạng blockchain, các máy tính đóng vai trò là như nhau, cùng đồng bộ dữ liệu và không có máy chủ. Chính vì vậy, việc gian lận, sửa đổi dữ liệu trên hệ thống mạng blockchain rất khó để thực hiện và gần như là bất khả thi.

## Các kỹ thuật mật mã học trong blockchain

### Mật mã khóa công khai

Mật mã khóa công khai hay mã hóa khóa bất đối xứng sử dụng hai khóa có quan hệ toán học với nhau, tức là một khóa này được hình thành từ khóa kia: Người muốn nhận bản mã tạo ra một khóa mật (private key) và từ khóa mật tính ra khóa công khai (public key) với một thủ tục không phức tạp, còn việc tìm khóa mật khi biết khóa công khai là bài toán khó giải được. Khóa công khai sẽ đưa đến cho người gởi bản tin qua kênh công cộng và bản tin được mã hóa bằng khóa công khai. Bản mã truyền đến người nhận và nó được giải mã bằng khóa mật.



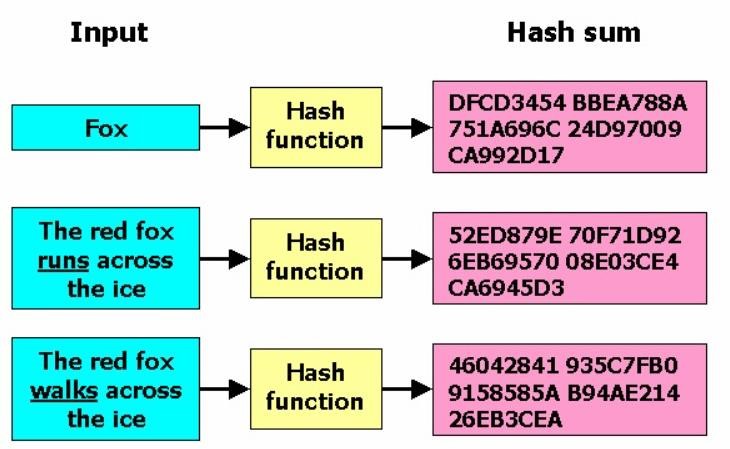
Hình 1.1 Mật mã khóa công khai

Hệ mật mã khóa công khai có vai trò vô cùng quan trọng trong công nghệ blockchain. Trong hệ thống mạng blockchain không có máy chủ. Chính vì vậy, khi một người dùng mới tham gia mạng sẽ tự tạo cho mình cặp khóa công khai và bí mật này. Để đảm bảo tính an toàn cho cặp khóa, độ dài của chúng phải được đảm bảo, thường là 256 bit, giúp chống lại các cuộc tấn công vét cạn mà dù là máy tính lượng tử cũng là bất khả thi.

Mật mã khóa công khai là một yếu tố cơ bản của công nghệ blockchain - nó là công nghệ cơ bản cho ví và giao dịch. Khi người dùng tạo ví trên blockchain, họ đang tạo cặp khóa công khai-riêng tư. Địa chỉ của ví là một chuỗi được tạo ra từ khóa công khai. Địa chỉ này được công khai cho tất cả mọi người và có thể được sử dụng để kiểm tra số dư trong ví đó hoặc gửi tiền vào đó. Khóa cá nhân được liên kết với ví là cách chứng minh quyền sở hữu và kiểm soát ví. Đó là cách duy nhất để gửi tiền ra khỏi nó và một khóa cá nhân bị mất có nghĩa là số tiền bên trong sẽ bị mắc kẹt ở đó mãi mãi.

### Hàm băm

Hàm băm (Hash function) là một hàm mật mã học được sử dụng để băm các giá trị đầu vào, biến đầu vào là một nội dung có kích thước, độ dài bất kỳ rồi sử dụng những thuật toán, công thức toán học để biến thành đầu ra tiêu chuẩn có độ dài nhất định, ví dụ: với hàm băm SHA-1 giá trị đầu ra luôn là 160 bit và SHA256 giá trị đầu ra là 256 bit. Đây là quá trình một chiều, tức là không thể từ giá trị băm tìm ra được giá trị đầu vào ban đầu.



Hình 1.2 Hàm băm

Trong công nghệ blockchan hàm băm đóng vai trò quan trọng trong việc đảm bảo tính xác thực tính toàn vẹn dữ liệu: Hàm băm được sử dụng để xác thực tính toàn vẹn của dữ liệu trong mỗi khối trong chuỗi blockchain. Mỗi khối được băm và kết quả của hàm băm được lưu trữ trong trường băm của khối. Giá trị băm của khối trước đó thường được sử dụng trong quá trình tạo khối mới để tạo liên kết giữa các khối trong chuỗi blockchain. Giá trị băm của khối trước đó được thêm vào khối mới như một phần của dữ liệu đầu vào cho hàm băm. Điều này tạo ra sự kết nối và sự liên kết giữa các khối trong chuỗi. Bất kỳ thay đổi nào trong dữ liệu của khối đều sẽ dẫn đến một giá trị băm khác, cho thấy rằng dữ liệu đã bị thay đổi. Bên cạnh đó, hàm băm được sử dụng để xác minh tính toàn vẹn của giao dịch trong một khối. Thay vì lưu trữ toàn bộ dữ liệu giao dịch trong mỗi khối, chỉ cần lưu trữ giá trị băm của dữ liệu giao dịch. Khi cần kiểm tra tính toàn vẹn của giao dịch, hàm băm có thể được sử dụng để so sánh với giá trị băm đã lưu trữ trong khối.

### Chữ ký số

##### Tạo cặp khóa cho RootCA

Chữ ký số là một kết quả quan trọng của mật mã khóa công khai, về định nghĩa chữ ký số là một tập con của chữ ký điện tử. Chữ ký điện tử được tạo lập dưới dạng từ, chữ, số, ký hiệu, âm thanh hoặc các hình thức khác bằng phương tiện điện tử, gắn liền hoặc kết hợp một cách lô gíc với thông điệp dữ liệu, có khả năng xác nhận người ký thông điệp dữ liệu và xác nhận sự chấp thuận của người đó đối với nội dung thông điệp dữ liệu được ký.

Chữ ký số theo Khoản 6, Điều 3, Nghị định 130/2018/NĐ-CP ngày 27/9/2018 được định nghĩa như sau:

"Chữ ký số" là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã không đối xứng, theo đó, người có được thông điệp dữ liệu ban đầu và khóa công khai của người ký có thể xác định được chính xác:

-Việc biến đổi nêu trên được tạo ra bằng đúng khóa bí mật tương ứng với khóa công khai trong cùng một cặp khóa

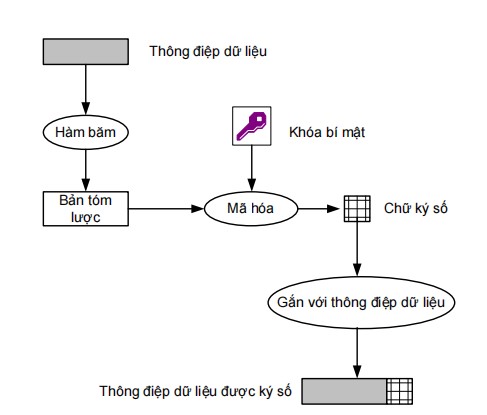
-Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên.

Chữ ký số hoạt động dựa trên sự kết hợp của một cặp khóa – private key và public key. Các bước thực hiện chữ ký số như sau:

B1: Người gửi tạo một tin nhắn, tài liệu hoặc giao dịch. Để tạo chữ ký số, người gửi sẽ sử dụng một thuật toán mã hóa để tính toán một đoạn mã băm (hash) duy nhất cho nội dung này.

B2: Người gửi sử dụng private key của mình để mã hóa mã băm đó, tạo thành chữ ký số. Quá trình mã hóa này sử dụng một thuật toán mã hóa đối xứng (symmetric encryption) hoặc mã hóa không đối xứng (asymmetric encryption) để tạo ra chữ ký.

B3: Người gửi gửi tài liệu hoặc giao dịch, cùng với chữ ký số của mình, đến người nhận.

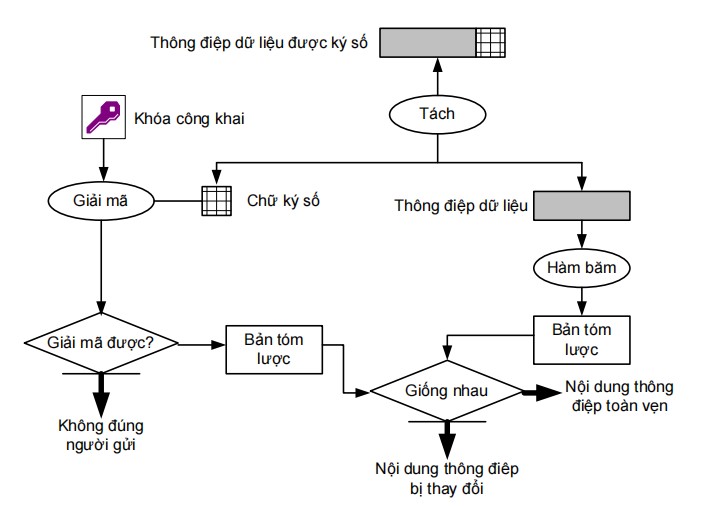


Hình 1.3 Các bước thực hiện chữ ký số

B4: Người nhận sử dụng khóa công khai của người gửi để giải mã chữ ký số, để lấy ra mã băm ban đầu. Sau đó, người nhận sử dụng cùng thuật toán mã hóa đã được sử dụng bởi người gửi, để tính toán lại mã băm.

B5: Nếu mã băm được tính toán bởi người nhận trùng khớp với mã băm ban đầu, thì chữ ký số được xác minh là hợp lệ, tức là nội dung của tài liệu hoặc giao dịch không bị thay đổi và chữ ký số được tạo ra bởi chính người gửi đó.

Quá trình này đảm bảo tính toàn vẹn của tài liệu hoặc giao dịch và đảm bảo rằng chỉ người gửi mới có thể tạo ra chữ ký số hợp lệ cho nội dung đó, do Private key chỉ được biết bởi người gửi. Ngoài ra, chữ ký số cũng giúp đảm bảo tính bảo mật và không thể phủ nhận (non-repudiation) trong trường hợp tranh chấp, vì người gửi không thể từ chối việc tạo ra chữ ký số đó.



Hình 1.4 Các bước giải mã chữ ký số

##### Vai trò của chữ ký số trong blockchain

Chữ ký số là một yếu tố quan trọng trong blockchain để xác thực tính toàn vẹn và chống lại sự gian lận. Trong blockchain, các giao dịch được đóng gói lại thành các khối và sau đó được liên kết với nhau để tạo thành Blockchain. Mỗi khối chứa thông tin về các giao dịch, bao gồm cả chữ ký số của người gửi và người nhận.

Chữ ký số được tạo ra bằng cách sử dụng một cặp khóa – Public Key và Private Key khai. Private key chỉ được biết bởi người gửi, trong khi Public key có thể được chia sẻ công khai. Khi người gửi tạo chữ ký số bằng private key, người nhận có thể sử dụng Public key để xác minh tính toàn vẹn của chữ ký.

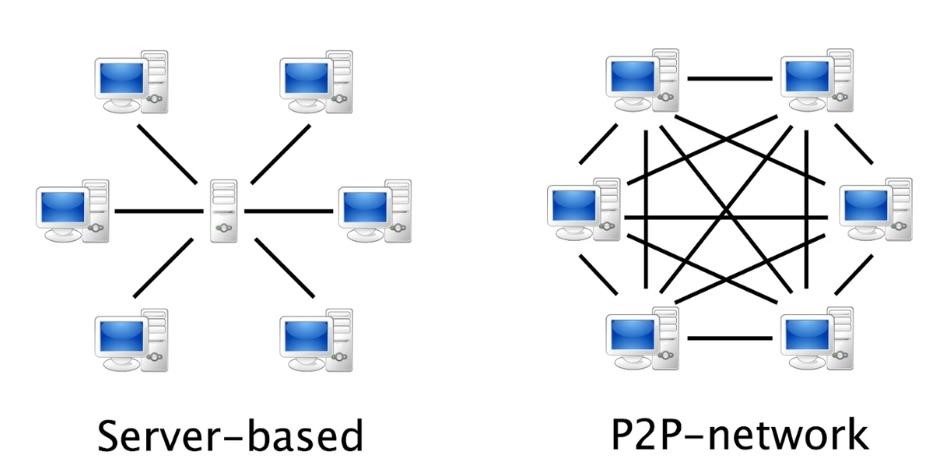
Trong blockchain, chữ ký số giúp đảm bảo rằng mỗi giao dịch được xác thực và không bị thay đổi khi được thêm vào khối. Khi một khối mới được tạo ra, nó sẽ có một mã băm duy nhất và chữ ký số của khối trước đó. Nếu bất kỳ giao dịch nào trong khối mới được thay đổi, mã băm của khối sẽ thay đổi và chữ ký số của khối trước đó sẽ không còn hợp lệ nữa. Do đó, hệ thống blockchain có thể phát hiện và từ chối bất kỳ giao dịch giả mạo nào được thêm vào Block.

Ngoài ra, chữ ký số cũng giúp đảm bảo tính riêng tư và bảo mật cho người dùng blockchain. Khi tạo một giao dịch, người dùng chỉ cần sử dụng private key của mình để tạo chữ ký số, trong khi khóa công khai của họ không cần được tiết lộ cho ai .

### Kiến trúc cơ bản của hệ thống blockchain

##### Mô hình mạng ngang hàng P2P

Mạng peer to peer (P2P) là một kiến trúc ứng dụng phân tán nhằm phân vùng nhiệm vụ hoặc khối lượng công việc giữa các peer. Các peer là những thiết bị tham gia trong ứng dụng có đặc quyền như nhau. Chúng tạo thành một mạng lưới các node ngang hàng.



Hình 1.5 Mô hình mạng ngang hàng

Trong hệ thống mạng blockchain, điểm nút là một máy tính chạy phần mềm trạm blockchain, các điểm nút có thể tự do kết nối mạng blockchain. Mỗi mạng blockchain thường duy trì một số DNS seed server cung cấp các địa chỉ truy cập ban đầu. Các điểm nút có thể trao đổi địa chỉ các nút lân cận. Sau khi một nút mạng kết nối, hoạt động đồng bộ dữ liệu được thực hiện.

##### Giao dịch - Transaction:

Giao dịch (transaction) trong blockchain là một hành động hoặc sự trao đổi thông tin và giá trị giữa các bên trong một mạng lưới blockchain. Giao dịch thường được sử dụng để đại diện cho sự chuyển đổi, trao đổi, hoặc lưu trữ dữ liệu và tài sản số trên blockchain. Mỗi giao dịch trong blockchain bao gồm các thông tin quan trọng như:

**Địa chỉ nguồn (sender address):** Địa chỉ ví điện tử của bên gửi giao dịch.

**Địa chỉ đích (recipient address):** Địa chỉ ví điện tử của bên nhận giao dịch.

**Số lượng (amount):** Số lượng tài sản số được gửi hoặc nhận trong giao dịch.

**Dữ liệu (data):** Dữ liệu bổ sung được gắn kết với giao dịch, có thể là tin nhắn, thông tin thêm, hoặc hợp đồng thông minh (smart contract).

**Chữ ký số (digital signature):** Chữ ký số của bên gửi giao dịch để xác minh và chứng thực giao dịch.

Mỗi giao dịch trong blockchain được ghi lại và đóng góp vào việc xây dựng chuỗi khối (blockchain). Các giao dịch được xác minh và lưu trữ trong các khối, tạo thành một lịch sử giao dịch không thể thay đổi được. Quá trình xác minh giao dịch trong blockchain thường liên quan đến sử dụng thuật toán mã hóa và chữ ký số để đảm bảo tính toàn vẹn và bảo mật của giao dịch.

Giao dịch trong blockchain rất quan trọng trong việc xác định quyền sở hữu, truy xuất lịch sử và ghi lại các hoạt động trên mạng lưới blockchain. Nó đóng vai trò quan trọng trong các ứng dụng tài chính, tiền điện tử, hợp đồng thông minh và nhiều lĩnh vực khác sử dụng công nghệ blockchain.

##### Sổ cái phân tán - Decentralized Ledger:

Sổ cái phân tán (Decentralized Ledger) là một hệ thống ghi chép và lưu trữ thông tin giao dịch và dữ liệu mà không được tập trung vào một thực thể duy nhất. Thay vì có một bên trung gian trung tâm kiểm soát và duy trì sổ cái, thông tin được phân tán và được lưu trữ trên nhiều nút (nodes) trong mạng lưới.

Trong một sổ cái phân tán, mỗi giao dịch và sự thay đổi đều được ghi lại dưới dạng một khối (block) và được liên kết với nhau để tạo thành một chuỗi khối (blockchain). Mỗi khối chứa thông tin về các giao dịch đã xảy ra trong một khoảng thời gian cụ thể và có một mã xác thực (hash) của khối trước đó. Điều này tạo ra một liên kết chặt chẽ giữa các khối và đảm bảo tính toàn vẹn của dữ liệu.

Sổ cái phân tán không phụ thuộc vào một bên trung gian duy nhất để kiểm soát và xác minh thông tin giao dịch. Thay vào đó, thông tin được kiểm tra và xác minh bởi một mạng lưới các nút (nodes) phân tán, trong đó mỗi nút có bản sao đồng bộ của toàn bộ sổ cái. Các nút trong mạng lưới thực hiện các thuật toán phân tán để đạt được sự đồng thuận về trạng thái của sổ cái và cập nhật dữ liệu mới.

Ưu điểm của sổ cái phân tán bao gồm tính toàn vẹn, an ninh, khả năng chống thay đổi và khả năng truy xuất dữ liệu từ mọi nút trong mạng. Bởi vì thông tin không được lưu trữ tập trung, sổ cái phân tán cũng đặc biệt khó bị tấn công hoặc bị làm giả mạo. Điều này làm cho sổ cái phân tán trở thành một công nghệ quan trọng trong các hệ thống blockchain và các ứng dụng liên quan.

##### Khối - Block:

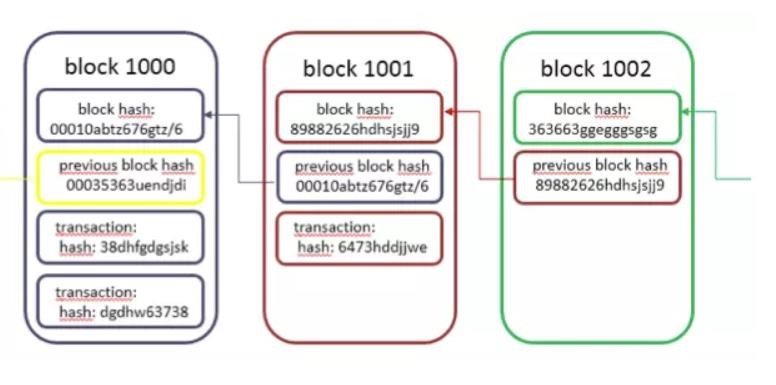
Khối (block) trong blockchain là một thành phần cơ bản của hệ thống blockchain. Nó là một đơn vị dữ liệu chứa thông tin về các giao dịch đã xảy ra trong mạng lưới blockchain. Mỗi khối bao gồm các thành phần sau:

**Mã xác thực (Hash):** Mã xác thực là một giá trị số học duy nhất được tạo ra thông qua việc áp dụng một thuật toán băm (hashing algorithm) vào dữ liệu trong khối. Mã xác thực đại diện cho nội dung của khối và đảm bảo tính toàn vẹn của dữ liệu. Nó cũng được sử dụng để liên kết khối hiện tại với khối trước đó trong chuỗi khối (blockchain).

**Giao dịch (Transactions):** Mỗi khối chứa một tập hợp các giao dịch đã xảy ra trong mạng lưới blockchain trong một khoảng thời gian nhất định. Giao dịch này có thể là chuyển tiền, truyền dữ liệu, thực hiện hợp đồng thông minh hoặc bất kỳ hành động nào khác mà mạng lưới blockchain hỗ trợ.

**Mã xác minh của khối trước (Previous Block's Hash):** Mỗi khối trong blockchain được liên kết với khối trước đó thông qua mã xác thực của khối trước. Điều này tạo ra sự liên kết chặt chẽ giữa các khối và đảm bảo tính toàn vẹn của chuỗi khối.

**Thông tin thời gian (Timestamp):** Khối cũng chứa thông tin về thời điểm tạo khối, được ghi dưới dạng một con số hoặc dữ liệu dạng thời gian.



Hình 1.6 Block

Các khối trong blockchain được tạo ra thông qua quá trình khai thác (mining) hoặc xác minh giao dịch (transaction verification) tùy thuộc vào thuật toán được sử dụng trong mạng lưới. Mỗi khối mới được thêm vào blockchain sau khi được xác minh và chấp nhận bởi một số lượng đủ các nút trong mạng. Tính toàn vẹn và tính nhất quán của blockchain được đảm bảo bởi việc xây dựng và kết nối các khối với nhau theo quy tắc liên kết mã xác thực. Quá trình này tạo ra một lịch sử giao dịch không thể thay đổi và làm cho blockchain trở thành một cơ sở dữ liệu phân tán và đáng tin cậy cho các ứng dụng blockchain.

##### Cơ chế đồng thuận:

Cơ chế đồng thuận (consensus mechanism) là một phương thức hoạt động trong hệ thống blockchain để đạt được sự thống nhất và đồng nhất giữa các nút (nodes) trong mạng lưới. Nó giúp đảm bảo rằng tất cả các nút đồng ý về trạng thái của blockchain và các giao dịch cụ thể. Cơ chế đồng thuận quan trọng để giải quyết vấn đề "Bài toán hai tù nhân" (Two Generals' Problem) trong môi trường phân tán, trong đó các bên phải đồng thuận về một hành động cụ thể mà không thể tin tưởng hoàn toàn lẫn nhau.

Có nhiều cơ chế đồng thuận khác nhau được sử dụng trong các hệ thống blockchain, mỗi cơ chế có những quy tắc và thuật toán riêng. Dưới đây là một số cơ chế đồng thuận phổ biến:

**Proof of Work (PoW):** Cơ chế đồng thuận PoW được sử dụng trong Bitcoin và nhiều mạng lưới blockchain khác. Người tham gia mạng phải giải một bài toán tính toán phức tạp để chứng minh rằng họ đã tiêu tốn một lượng công việc (thời gian và tài nguyên tính toán) để đề xuất một khối mới. Quá trình khai thác (mining) trong PoW được sử dụng để tạo ra khối mới và đồng thuận trạng thái của blockchain.

**Proof of Stake (PoS):** Cơ chế đồng thuận PoS dựa trên sự đóng góp vốn (stake) của các tham gia viên trong mạng. Thay vì phải giải một bài toán tính toán phức tạp, người tham gia mạng sẽ đặt cược một số lượng tiền mã hóa của họ như một đồng cọc và dựa vào sự chọn ngẫu nhiên để lựa chọn người tạo khối mới. Các người tham gia có càng nhiều tiền mã hóa và cọc càng lớn thì càng có cơ hội lớn hơn để được chọn và đóng góp vào quá trình đồng thuận.

**Delegated Proof of Stake (DPoS):** DPoS là một biến thể của PoS, trong đó người dùng bỏ phiếu để chọn ra một số lượng nhất định các đại diện (delegates) đại diện cho mạng. Các đại diện này sẽ có trách nhiệm tạo khối mới và đồng thuận trạng thái của blockchain. DPoS tập trung vào việc cung cấp tốc độ xử lý nhanh hơn và hiệu suất cao hơn so với PoW và PoS.

**Proof of Authority (PoA):** PoA là một cơ chế đồng thuận trong đó các quyền lực (authorities) được ủy quyền để tạo và xác minh các khối trong mạng. Những quyền lực này có trách nhiệm về tính toàn vẹn và bảo mật của mạng và thường được xác định trước bởi quản trị viên.

Mỗi cơ chế đồng thuận có ưu điểm và nhược điểm riêng. Sự lựa chọn cơ chế đồng thuận thích hợp phụ thuộc vào các yêu cầu và mục tiêu của hệ thống blockchain cụ thể. Một cơ chế đồng thuận hiệu quả là quan trọng để đảm bảo tính toàn vẹn, an toàn và sự tin cậy của mạng lưới blockchain.

### Cơ chế hoạt động của hệ thống Blockchain

Hoạt động cơ bản của một hệ thống blockchain là quá trình sinh khối (block) và thêm khối vào chuỗi khối móc xích của hệ thống. Các bước cơ bản trong quá trình này được minh họa trong hình 1.6.

**Gom nhóm giao dịch:** Đầu tiên, các giao dịch mới được gửi đến hệ thống và được gom nhóm lại để tạo thành một khối. Các giao dịch này có thể bao gồm việc chuyển tiền, thực hiện hợp đồng thông minh hoặc các hoạt động khác phụ thuộc vào mục đích của blockchain cụ thể.

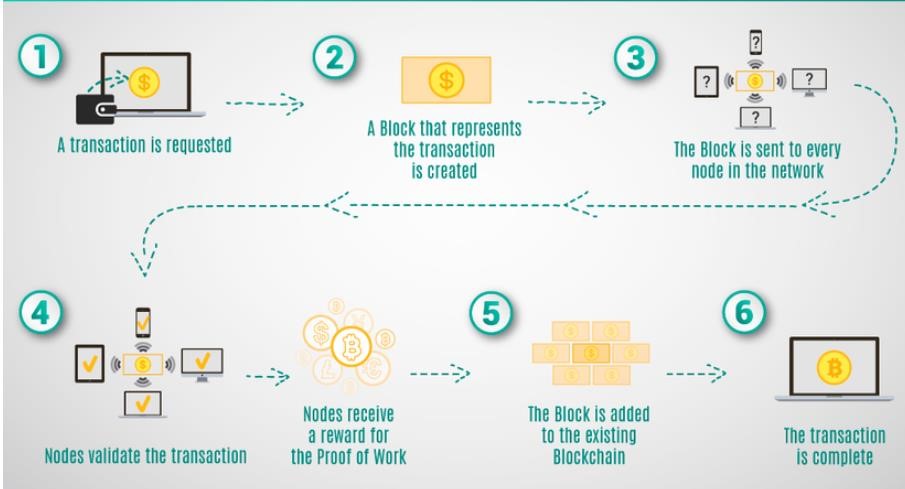
**Tạo khối:** Trước khi một khối mới được tạo, các nút trong mạng lưới blockchain thực hiện việc xác minh tính hợp lệ của các giao dịch trong khối. Quá trình xác minh bao gồm kiểm tra chữ ký số, kiểm tra số dư, kiểm tra điều kiện thông minh và xác định tính hợp lệ của giao dịch. Sau khi các giao dịch đã được xác minh, một khối mới được tạo ra. Khối bao gồm thông tin về các giao dịch, mã xác thực của khối trước đó (previous block's hash), và một số thông tin khác như timestamp (thời gian tạo khối), nonce (số ngẫu nhiên) và thông tin bổ sung.

**Các khối được gửi tới mọi nút trên hệ thống mạng.**

**Xác minh và đồng thuận:** Khi một người khai thác tìm ra giá trị nonce phù hợp và tạo được khối mới, khối đó được truyền qua mạng lưới blockchain để các nút khác xác minh tính hợp lệ của nó. Các nút kiểm tra mã xác thực, xác minh bằng chứng công việc (proof of work) và đồng thuận rằng khối mới được thêm vào chuỗi khối chính xác. Trong các hệ thống blockchain sử dụng cơ chế đồng thuận Proof of Work, quá trình tạo khối cần tiếp tục với giai đoạn khai thác (mining). Người tham gia mạng (người khai thác) phải giải một bài toán tính toán phức tạp để tìm ra một giá trị nonce sao cho mã xác thực của khối mới thỏa mãn một số điều kiện nhất định. Quá trình này đòi hỏi sự tiêu tốn năng lượng tính toán và làm cho việc tạo khối trở nên khó khăn.

**Thêm vào chuỗi khối:** Khi khối mới đã được xác minh và đồng thuận, nó được thêm vào chuỗi khối (blockchain) và trở thành một phần của lịch sử giao dịch của hệ thống. Quá trình này đảm bảo tính toàn vẹn và liên kết giữa các khối trong blockchain.

**Quá trình tạo khối**: một quá trình liên tục trong hệ thống blockchain, trong đó các giao dịch mới được gom nhóm và thêm vào chuỗi khối theo một cách tuần tự và đồng thuận. Cơ chế đồng thuận được sử dụng trong quá trình này đảm bảo tính an toàn, tin cậy và phân quyền của hệ thống.



Hình 1.7 Cơ chế hoạt động của Blockchain

## Tổng kết chương 1

Hệ thống blockchain là một công nghệ phân tán và bảo mật, đưa ra một phương pháp mới để lưu trữ và quản lý thông tin mà không cần sự trung gian của bên thứ ba. Các tính chất chung được rút ra về hệ thống blockchain:

**Tính phân tán:** Blockchain hoạt động trên một mạng lưới phân tán của các nút (nodes). Thay vì có một máy chủ trung tâm, dữ liệu được lưu trữ và quản lý trên nhiều nút khác nhau trên mạng. Điều này giúp đảm bảo tính phân tán, khó tấn công và độ tin cậy cao của hệ thống.

**Chuỗi khối:** Dữ liệu trong blockchain được tổ chức thành các khối (blocks) liên kết với nhau thông qua mã xác thực của khối trước đó. Mỗi khối chứa thông tin về giao dịch, mã xác thực, timestamp và thông tin khác. Chuỗi khối giúp đảm bảo tính toàn vẹn của dữ liệu và lịch sử giao dịch.

**Cơ chế đồng thuận:** Blockchain sử dụng các cơ chế đồng thuận như Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS) hoặc Proof of Authority (PoA) để đạt được sự thống nhất và đồng thuận giữa các nút trong mạng. Cơ chế này đảm bảo tính toàn vẹn và bảo mật của dữ liệu trong hệ thống.

**Bảo mật:** Blockchain sử dụng mã hóa và các thuật toán bảo mật để đảm bảo tính an toàn của dữ liệu. Mỗi giao dịch trong blockchain được ký điện tử để xác thực nguồn gốc và ngăn chặn sửa đổi trái phép. Các thuật toán băm (hashing) và chữ ký số được sử dụng để đảm bảo tính toàn vẹn và xác minh.

**Tin cậy và phân quyền:** Hệ thống blockchain cho phép tất cả các thành viên trong mạng có thể kiểm tra và xác minh dữ liệu. Không có một bên thứ ba trung gian và tất cả các giao dịch và quyết định được đưa ra theo quy tắc được đồng thuận. Điều này tạo ra tính tin cậy và tính phân quyền trong hệ thống.

**Ứng dụng đa dạng:** Blockchain không chỉ được áp dụng trong lĩnh vực tiền điện tử như Bitcoin, mà còn được sử dụng trong nhiều lĩnh vực khác như giao dịch tài chính, quản lý chuỗi cung ứng, quản lý quyền sở hữu, bỏ phiếu điện tử, chứng thực dữ liệu và nhiều ứng dụng khác.

Hệ thống blockchain mang lại sự an toàn và đáng tin cậy trong việc lưu trữ, quản lý và trao đổi thông tin. Tuy nhiên, nó cũng tồn tại những nhực điểm rõ ràng như:

**Khối lượng dữ liệu lớn:** Blockchain lưu trữ toàn bộ lịch sử giao dịch trên mạng, và điều này đồng nghĩa với việc kích thước của chuỗi khối tăng lên theo thời gian. Việc lưu trữ và xử lý khối lượng dữ liệu lớn có thể trở thành một vấn đề đối với các nút mạng và làm cho quá trình đồng bộ hóa chậm hơn.

**Tốc độ xử lý:** Quá trình xác minh và ghi các giao dịch trên blockchain có thể mất thời gian, đặc biệt đối với các hệ thống sử dụng cơ chế đồng thuận Proof of Work (PoW). Thời gian xử lý chậm có thể hạn chế khả năng mở rộng và hiệu suất của hệ thống.

**Chi phí cao:** Một số hệ thống blockchain có chi phí cao cho việc xử lý và ghi giao dịch. Ví dụ, trong hệ thống sử dụng PoW, việc tiêu tốn năng lượng tính toán và điện năng để giải quyết bài toán tính toán phức tạp có thể gây ra chi phí đáng kể.

**Mất quyền kiểm soát dữ liệu:** Mặc dù tính phân quyền là một ưu điểm của blockchain, nó cũng đồng nghĩa với việc mất quyền kiểm soát dữ liệu của một bên thứ ba trung gian. Điều này có thể là một thách thức trong việc tuân thủ các quy định và chính sách, đặc biệt trong các lĩnh vực có yêu cầu nghiêm ngặt về tuân thủ và quản lý dữ liệu.

**Độ tin cậy phụ thuộc vào số lượng nút:** Độ tin cậy của hệ thống blockchain phụ thuộc vào số lượng nút tham gia trong mạng. Nếu một số lượng lớn nút bị tấn công hoặc không chính xác, điều này có thể ảnh hưởng đến tính toàn vẹn và đáng tin cậy của hệ thống.

**Khó khăn thay đổi:** Một khi dữ liệu đã được ghi vào blockchain, nó khó khăn để sửa đổi hoặc xóa bỏ. Điều này có thể tạo ra những thách thức về quản lý lỗi và thay đổi dữ liệu trong trường hợp cần thiết.

Tóm lại, công nghê blockchain là một lĩnh vực vô cùng tiềm năng, có thể áp dụng trong nhiều lĩnh vực không chỉ là an toàn thông tin hay tiền mã hóa mà còn trong các lĩnh vực yêu cầu cao về độ tin cậy và tính nhất quán của dữ liệu như: y tế, giáo dục, công nghệ thực phẩm, thương mại,...

# ETHEREUM VÀ CÁC TIÊU CHUẨN EHEREUM REQUEST FOR COMMENTS

## Giới thiệu chung về nền tảng blockcahin Ethereum

### Khái niệm

Ethereum là một nền tảng blockchain công cộng và cũng là một đồng tiền điện tử (cryptocurrency) có tên gọi là Ether (ETH). Nó được tạo ra bởi Vitalik Buterin vào năm 2013 và ra mắt chính thức vào năm 2015. Ethereum không chỉ là một đồng tiền điện tử, mà còn là một môi trường phát triển và triển khai các ứng dụng phi tập trung (Decentralized Applications - DApps). Nó cung cấp một nền tảng để xây dựng các hợp đồng thông minh (smart contracts) và các ứng dụng phi tập trung sử dụng công nghệ blockchain.

Ethereum cung cấp một môi trường phát triển mạnh mẽ và linh hoạt, với ngôn ngữ lập trình Solidity, để phát triển các ứng dụng blockchain. Nó cũng có một cộng đồng lớn và phát triển đa dạng của những nhà phát triển, doanh nghiệp và người dùng cuối. Mục tiêu của Ethereum là xây dựng một nền tảng phân quyền và phi tập trung, cho phép việc thực hiện các giao dịch và ứng dụng phi tập trung một cách an toàn, minh bạch và không thể tác động.

### Ethereum Virtual Machine

Ethereum là một chuỗi khối với một máy tính được nhúng trong đó. Nó là nền tảng để xây dựng các ứng dụng và tổ chức theo cách phi tập trung, không cần cấp phép và chống kiểm duyệt.

Trong toàn hệ thống Ethereum, có một máy tính chính tắc duy nhất (được gọi là Máy ảo Ethereum hoặc EVM) có trạng thái mà mọi người trên mạng Ethereum đều đồng ý. Mọi người tham gia vào mạng Ethereum (mọi nút Ethereum) đều giữ một bản sao trạng thái của máy tính này. Ngoài ra, bất kỳ người tham gia nào cũng có thể gửi yêu cầu cho máy tính này thực hiện tính toán tùy ý. Bất cứ khi nào một yêu cầu như vậy được phát đi, những người tham gia khác trên mạng sẽ xác minh, xác thực và thực hiện ("thực thi") tính toán. Việc thực thi này gây ra thay đổi trạng thái trong EVM, thay đổi này được cam kết và lan truyền trên toàn bộ mạng.

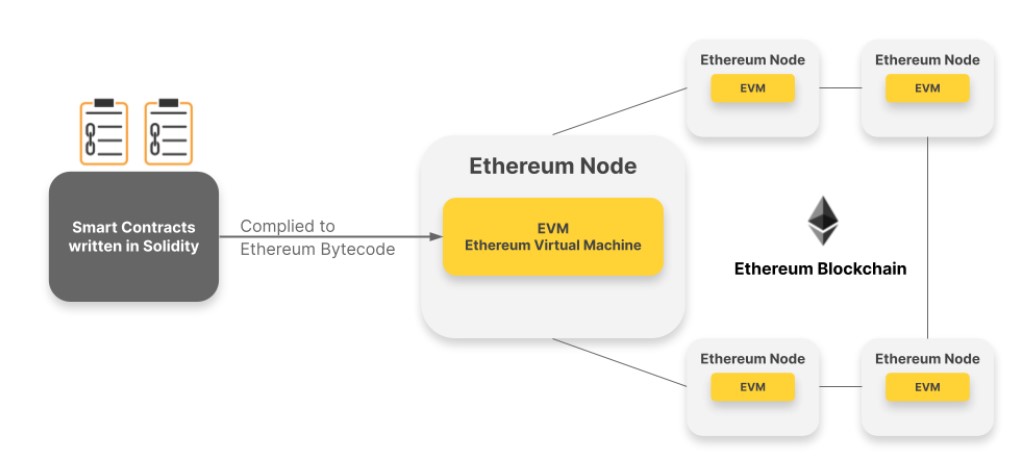
Yêu cầu tính toán được gọi là yêu cầu giao dịch; bản ghi của tất cả các giao dịch và trạng thái hiện tại của EVM được lưu trữ trên chuỗi khối, do đó được tất cả các nút lưu trữ và đồng ý.

EVM là một máy trạng thái Turing hoàn chỉnh, bởi vì tất cả các bước thực thi được giới hạn trong một lượng hữu hạn các bước tính toán. Đây là điều khác biệt so với Bitcoin khi trên bitcoin thì Stack Machine chỉ là máy Turing không hoàn chỉnh. EVM được thiết kế theo kiến trúc stack-based, tất cả đều được lưu trong stack, word size là 256-bit. Các thành phần lưu trữ thông tin trên EVM được chia ra thành 3 phần:

-Một mã ROM cố định, không thể thay đổi: Được load cùng byte code của smart contract khi xử lý contract.

-Một bộ nhớ ngắn hạn: Khi muốn lưu trên Solidity thì dùng từ khóa memory.

-Một bộ nhớ dài hạn: Khi muốn lưu trên Solidity thì dùng từ khóa storage.



Hình 2.1 Cơ chế hoạt động của EVM

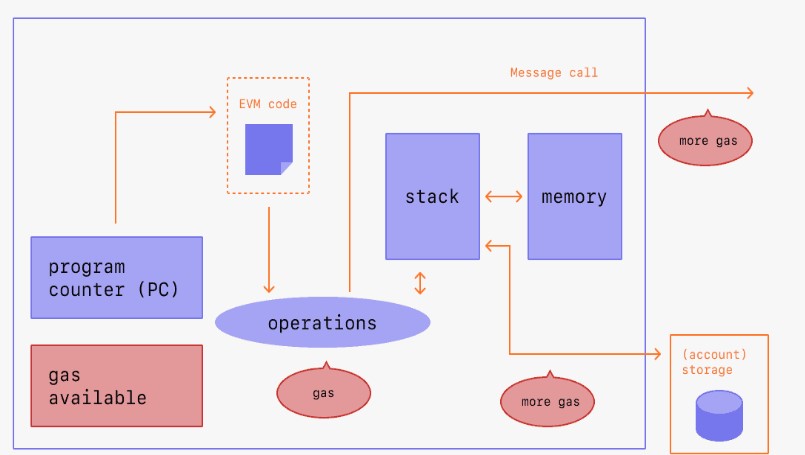
Cơ chế hoạt động của EVM gồm các bước sau:

**Biên dịch:** Mã nguồn của hợp đồng thông minh được viết bằng một ngôn ngữ lập trình như Solidity và sau đó được biên dịch thành bytecode, một ngôn ngữ máy ảo mà EVM có thể hiểu.

**Triển khai:** Bytecode của hợp đồng thông minh được triển khai lên blockchain Ethereum thông qua việc tạo một giao dịch đặc biệt được gọi là giao dịch triển khai (deployment transaction). Giao dịch này chứa bytecode của hợp đồng và được gửi đến mạng lưới để được thực thi. Thực thi: EVM làm nhiệm vụ thực thi bytecode của hợp đồng thông minh. Nó chạy qua từng lệnh bytecode và thực hiện các hoạt động như tính toán, lưu trữ dữ liệu và tương tác với các tài khoản khác trên blockchain.

**Tương tác với tài khoản:** EVM có thể tương tác với các tài khoản khác trên blockchain Ethereum thông qua việc gửi và nhận thông điệp. Thông điệp này có thể gửi từ hợp đồng thông minh hiện tại đến một hợp đồng khác hoặc một địa chỉ tài khoản bên ngoài. Quá trình tương tác này có thể thực hiện các giao dịch, truy xuất dữ liệu hoặc gọi các hàm trong hợp đồng khác.

**Gas và chi phí:** Mỗi hoạt động trong EVM được tính toán và tiêu tốn một lượng nguồn tài nguyên gọi là gas. Gas được định giá bằng Ether và đóng vai trò là một đơn vị đo lường cho chi phí thực hiện một hoạt động. Người gửi giao dịch phải trả gas để thực thi các hoạt động và chi phí này sẽ được chuyển đến những người đào khối và duy trì mạng lưới. Vì mỗi giao dịch Ethereum yêu cầu tài nguyên tính toán để thực hiện nên mỗi giao dịch đều yêu cầu một khoản phí. Gas đề cập đến phí cần thiết để thực hiện giao dịch trên Ethereum, bất kể giao dịch thành công hay thất bại. Phí gas được thanh toán bằng tiền tệ bản địa của Ethereum, ether (ETH). Giá gas được biểu thị bằng gwei, bản thân nó là một mệnh giá của ETH - mỗi gwei bằng 0,000000001 ETH (10 -9 ETH). Ví dụ: thay vì nói rằng gas của bạn tốn 0,000000001 ether, bạn có thể nói gas của bạn tốn 1 gwei. Bản thân từ 'gwei' có nghĩa là 'giga-wei' và nó bằng 1.000.000.000 wei.



Hình 2.2 Phí gas

### Ethereum và hợp đồng thông minh.

##### Hợp đồng thông minh

Hợp đồng thông minh (smart contract) là một chương trình được viết trong ngôn ngữ lập trình và được triển khai trên mạng lưới blockchain. Nó là một hợp đồng tự thực hiện, không cần sự can thiệp của bên thứ ba và tuân theo các điều khoản và điều kiện được xác định trước.

Một hợp đồng thông minh được xây dựng dựa trên nguyên tắc của mã nguồn mở và công nghệ blockchain. Nó hoạt động như một hợp đồng truyền thống, nhưng được thực thi tự động và có thể kiểm tra được trên blockchain. Hợp đồng thông minh có thể thực hiện các hành động tự động khi các điều kiện được xác định trước được đáp ứng. Ví dụ, một hợp đồng thông minh có thể được sử dụng để thực hiện việc chuyển tiền tự động từ một bên sang bên khác khi các điều kiện nhất định được thỏa mãn, như thời gian hoặc sự xác nhận từ các bên tham gia. Nó cũng có thể thực hiện các hoạt động phức tạp hơn như quản lý quyền sở hữu tài sản kỹ thuật số, ghi chép giao dịch, đảm bảo tính toàn vẹn dữ liệu và rất nhiều ứng dụng khác.

Hợp đồng thông minh là một đặc điểm quan trọng của nền tảng Ethereum. Ethereum cho phép việc triển khai và thực thi các hợp đồng thông minh trên blockchain của nó bằng cách sử dụng ngôn ngữ Solidity. Solidity cho phép các nhà phát triển xây dựng các hợp đồng thông minh với các tính năng như quản lý tài sản kỹ thuật số, chuyển tiền tự động, điều kiện và ràng buộc, sự kiện và hơn thế nữa. Khi một hợp đồng thông minh được triển khai trên mạng Ethereum, nó trở thành một phần của blockchain và có thể được gọi và tương tác bởi các tài khoản khác trên mạng. Các giao dịch và tương tác với hợp đồng thông minh được thực hiện bằng cách gửi các giao dịch đặc biệt đến địa chỉ của hợp đồng.

Trên thực tế, những người tham gia không viết mã mới mỗi khi họ muốn yêu cầu tính toán trên EVM. Thay vào đó, các nhà phát triển ứng dụng tải các chương trình (đoạn mã có thể tái sử dụng) vào trạng thái EVM và người dùng đưa ra yêu cầu thực thi các đoạn mã này với các tham số khác nhau. Chúng ta có thể gọi các chương trình được tải lên và thực thi bởi mạng là hợp đồng thông minh.

Ở mức rất cơ bản, người dùng có thể coi hợp đồng thông minh giống như một loại máy bán hàng tự động: một tập lệnh, khi được gọi với các tham số nhất định, sẽ thực hiện một số hành động hoặc tính toán nếu các điều kiện nhất định được thỏa mãn. Ví dụ: hợp đồng thông minh của nhà cung cấp đơn giản có thể tạo và chỉ định quyền sở hữu tài sản kỹ thuật số nếu người gọi gửi ETH cho một người nhận cụ thể.

Bất kỳ nhà phát triển nào cũng có thể tạo hợp đồng thông minh và công khai nó trên mạng, sử dụng chuỗi khối làm lớp dữ liệu của nó, với một khoản phí trả cho mạng. Sau đó, bất kỳ người dùng nào cũng có thể gọi hợp đồng thông minh để thực thi mã của nó, một lần nữa với một khoản phí trả cho mạng.

Do đó, với các hợp đồng thông minh, các nhà phát triển có thể xây dựng và triển khai các ứng dụng và dịch vụ phức tạp tùy ý hướng tới người dùng như: thị trường, công cụ tài chính, trò chơi,...

##### Tiền mã hóa

Tiền kỹ thuật số (hoặc tiền điện tử) là tiền hoặc các tài sản tương đương tiền được lưu trữ, quản lý và trao đổi trên các hệ thống thiết bị điện tử kỹ thuật số, đặc biệt là qua mạng internet. Tiền kỹ thuật số có thể được lưu trữ trong cơ sở dữ liệu phân tán trên Internet, trong cơ sở dữ liệu máy tính điện tử, trong tệp kỹ thuật số hoặc trong thẻ có giá trị lưu trữ. Ví dụ về các loại tiền tệ kỹ thuật số bao gồm tiền mã hóa, tiền ảo, tiền tệ kỹ thuật số ngân hàng trung ương và e-Cash.

Tiền mã hóa là một loại tiền điện tử, được tạo ra dựa trên kỹ thuật mật mã. Theo bách khoa toàn thư mở Wikipedia: Tiền mã hóa (cryptocurrency) là một dạng của tiền mật tính điện tử hoặc tiền kỹ thuật số, tiền ảo, là một tài sản kỹ thuật số được thiết kế để làm việc như là một trung gian trao đổi mà sử dụng mật mã để đảm bảo các giao dịch của nó, để kiểm soát việc tạo ra các đơn vị bổ sung và để xác minh việc chuyển giao tài sản. Mật tính (cryptographic computation) là từ được tạm dịch từ các thuật toán mã hóa tư liệu thông tin kỹ thuật số để bảo toàn nội dung và chữ ký của những tư liệu thông tin đó. Tiền mã hóa được phân loại như là một tập con của các loại tiền kỹ thuật số và cũng được phân loại là một tập con của các loại tiền tệ thay thế (Altcoin). Tuy nhiên danh từ tiền ảo không nên được sử dụng quá phổ biến vì nó chứa đựng một hàm ngữ mang tính chất phi thực tế và không có tính năng hoặc công dụng thật ngoài đời sống.

Ether (ETH) là loại tiền mã hóa được sử dụng cho nhiều thứ trên mạng Ethereum. Về cơ bản, đây là hình thức thanh toán phí giao dịch duy nhất được chấp nhận và sau The Merge , ETH được yêu cầu xác thực và đề xuất các khối trên Mainnet. Ether cũng được sử dụng như một hình thức thế chấp chính trong thị trường cho vay DeFi , như một đơn vị tài khoản trong thị trường NFT, như khoản thanh toán kiếm được khi thực hiện dịch vụ hoặc bán hàng hóa trong thế giới thực, v.v.

Ethereum cho phép các nhà phát triển tạo các ứng dụng phi tập trung (dapps) , tất cả đều chia sẻ một nhóm sức mạnh tính toán. Nhóm chia sẻ này là hữu hạn, vì vậy Ethereum cần một cơ chế để xác định ai sẽ sử dụng nó. Mặt khác, một dapp có thể vô tình hoặc cố ý tiêu thụ tất cả tài nguyên mạng, điều này sẽ chặn những người khác truy cập vào nó.

Đồng Ether hỗ trợ cơ chế định giá cho sức mạnh tính toán của Ethereum. Khi người dùng muốn thực hiện giao dịch, họ phải trả ether để giao dịch của họ được công nhận trên chuỗi khối. Các chi phí sử dụng này được gọi là phí gas và phí gas phụ thuộc vào lượng điện toán cần thiết để thực hiện giao dịch và nhu cầu điện toán trên toàn mạng tại thời điểm đó.

Do đó, ngay cả khi một dapp độc hại gửi một vòng lặp vô hạn, giao dịch cuối cùng sẽ hết ether và chấm dứt, cho phép mạng trở lại bình thường.

##### Khái niệm Token

Token (Token) là một thuật ngữ được sử dụng trong lĩnh vực công nghệ blockchain và tiền mã hóa. Nó đề cập đến một đơn vị giá trị hoặc quyền sở hữu có thể được đại diện bằng một tài sản kỹ thuật số, thường được xác định và giao dịch trên mạng lưới blockchain.Token có thể đại diện cho nhiều loại tài sản hoặc quyền sở hữu khác nhau:

**Tiền mã hóa:** Token có thể được sử dụng như tiền mã hóa, tương tự như đồng tiền truyền thống. Ví dụ, Bitcoin (BTC) và Ethereum (ETH) là những ví dụ về token tiền mã hóa.

**Quyền sở hữu:** Token có thể đại diện cho quyền sở hữu trong một tài sản nhất định. Ví dụ, token có thể đại diện cho quyền sở hữu một tài sản truyền thống như bất động sản, cổ phiếu công ty, hàng hóa và nhiều tài sản khác.

**Quyền truy cập:** Token có thể được sử dụng để đại diện cho quyền truy cập vào một dịch vụ hoặc nền tảng cụ thể. Ví dụ, token có thể được sử dụng để truy cập vào một ứng dụng phi tập trung (dApp) hoặc mạng xã hội phi tập trung.

Token thường được tạo và quản lý bằng cách sử dụng các giao thức và chuẩn tiêu chuẩn như ERC-20 (Ethereum Request for Comments 20) trên mạng lưới Ethereum hoặc các giao thức tương tự trên các nền tảng blockchain khác. Các token có thể được giao dịch, chuyển nhượng và sử dụng theo các quy tắc được định nghĩa trong smart contract và giao thức tương ứng.

Hiện nay trên thị trường có hàng ngàn loại token được phát hành do các công ty/dự án khác nhau phát hành. Mỗi loại đều có những cách thức sử dụng riêng và giải quyết một vấn đề cụ thể. Tuy nhiên, nếu xét về tính năng, chúng ta có thể chia token ra làm 2 loại đó là utility token và security token:

**Utility token:** Đây là loại token tiện ích. Nó được tạo ra để phục vụ cho một dự án mới với mục tiêu và tính năng rất rõ ràng, cụ thể. Ví dụ token được phát hành bởi dự án Dock.io có thể sử dụng để thanh toán, bình chọn; còn BNB token của Binance thì có tính năng giảm phí giao dịch,…

**Security token:** Security token còn có tên gọi khác là token chứng khoán. Nó là một dạng cổ phiếu điện tử được tạo ra dưới hình thức token. Người sở hữu security token sẽ được hưởng mức cổ tức tương ứng với số cổ phần của họ trong dự án. Ngoài ra, nếu nắm giữ trong tay loại token này, bạn còn có quyền được tham gia bầu chọn, biểu quyết một số hoạt động của dự án.

## Các tiêu chuẩn Ethereum Request for Comments- ERC

Ethereum Request for Comments (ERC) là một chuẩn đề xuất được sử dụng trong mạng lưới Ethereum để định nghĩa các giao thức, chuẩn và quy tắc cho việc triển khai các token và các hợp đồng thông minh trên Ethereum. ERC là một tiền tố cho Ethereum Request for Comments, được mượn từ RFC (Request for Comments) trong lĩnh vực các giao thức internet.

Các chuẩn ERC được đề xuất và đánh số theo thứ tự liên tục khi có sự phát triển và mở rộng trong cộng đồng Ethereum. Mỗi chuẩn ERC mô tả các yêu cầu, quy tắc và giao thức cụ thể mà các hợp đồng thông minh hoặc token phải tuân thủ để đảm bảo tính tương thích và giao tiếp được với các ứng dụng và hệ thống khác trong mạng lưới Ethereum

### Tiêu chuẩn ERC-20 (Token Standard)

##### Giới thiệu chung ERC-20

ERC-20 là một chuẩn tiêu chuẩn cho các token fungible (có thể thay thế nhau) trong mạng lưới Ethereum. ERC-20 được đề xuất để định nghĩa các quy tắc và giao thức chung cho việc phát hành và quản lý các token trong mạng lưới Ethereum. Chuẩn này giúp đảm bảo tính tương thích và tương tác giữa các token và các ứng dụng, ví và sàn giao dịch trên Ethereum.

ERC-20 (Ethereum Request for Comments 20), được đề xuất bởi Fabian Vogelsteller vào tháng 11 năm 2015, là Tiêu chuẩn mã thông báo triển khai API cho mã thông báo trong Hợp đồng thông minh. Một token ERC-20 phải tuân thủ các yêu cầu và quy tắc sau đây:

**Tổng cung cấp cố định:** Một token ERC-20 có tổng cung cố định, tức là số lượng token đã được phát hành không thể thay đổi sau khi triển khai.

**Chuyển đổi:** Token ERC-20 cho phép người dùng chuyển đổi token giữa các tài khoản khác nhau trong mạng lưới Ethereum.

**Số lượng:** Mỗi token ERC-20 được phân chia thành các đơn vị nhỏ hơn gọi là "đơn vị token" và được đếm bằng số nguyên không âm.

**Giao diện chuẩn:** Các token ERC-20 phải triển khai các phương thức chuẩn như **balanceOf** (số dư tài khoản), **transfer** (chuyển token), **approve** (cho phép một địa chỉ khác rút token từ tài khoản của bạn), và **transferFrom** (chuyển token từ một địa chỉ khác).  Sự kiện: Các sự kiện được sử dụng để ghi lại các hoạt động quan trọng như chuyển token, phê duyệt và rút token.

Chuẩn ERC-20 đã trở thành một trong những chuẩn phổ biến nhất và được sử dụng rộng rãi trong cộng đồng Ethereum. Đa số các token dựa trên Ethereum hiện nay đều tuân thủ chuẩn ERC-20 để đảm bảo tính tương thích và tương tác dễ dàng với các ứng dụng và dịch vụ khác trong mạng lưới Ethereum.

##### Các phương thức trong ERC-20

Trong chuẩn ERC-20, có một số function quan trọng cần được triển khai trong smart contract của token ERC-20 để đáp ứng các yêu cầu và quy tắc của chuẩn này. Dưới đây là một số function chính trong ERC-20:

**name():** tên token. **symbol():** mã token. **decimal():** xác định số lượng chữ số thập phân sau dấu phẩy của token. Nó cho phép token ERC-20 được phân chia thành các đơn vị nhỏ hơn, giúp điều chỉnh độ chính xác của số lượng token.Tham số decimals trong chuẩn ERC-20 xác định số lượng chữ số thập phân của token. Thông thường, decimals được đặt là 18, tương ứng với chuẩn Ethereum và số chữ số thập phân phổ biến nhất. Điều này cho phép token ERC-20 được hiển thị và giao dịch với độ chính xác cao.

**totalSupply():** Function này trả về tổng số lượng token đã được tạo ra trong hợp đồng.

**balanceOf(address \_owner):** Function này trả về số lượng token mà địa chỉ (\_owner) sở hữu.

**transfer(address \_to, uint256 \_value):** Function này cho phép chuyển số lượng token (\_value) từ địa chỉ người gửi đến địa chỉ (\_to).

**transferFrom(address \_from, address \_to, uint256 \_value):** Function này cho phép chuyển số lượng token (\_value) từ địa chỉ (\_from) đến địa chỉ (\_to). Điều kiện là người gửi (\_from) đã được ủy quyền trước đó bởi chủ sở hữu để thực hiện việc chuyển tiền.

**approve(address \_spender, uint256 \_value):** Function này cho phép địa chỉ (\_spender) được ủy quyền chuyển số lượng token (\_value) từ địa chỉ người ủy quyền.

**allowance(address \_owner, address \_spender):** Function này trả về số

lượng token mà địa chỉ (\_spender) được ủy quyền chuyển từ địa chỉ (\_owner).

Các function này tạo ra các cơ chế cơ bản cho việc chuyển đổi và quản lý token trong chuẩn ERC-20. Các function này cung cấp tính tương thích và tương tác giữa các tài khoản và các ứng dụng khác trong mạng lưới Ethereum

##### Ứng dụng của ERC-20

Trong thực tế, ứng dụng của ERC-20 rất đa dạng và phong phú. Dưới đây là một số ví dụ về các ứng dụng thực tế của ERC-20:

**Token giao dịch:** Các token ERC-20 được sử dụng như tiền tệ kỹ thuật số để thực hiện giao dịch trong hệ sinh thái blockchain Ethereum. Chúng có thể được gửi, nhận và giao dịch trên các sàn giao dịch tiền mã hóa và được sử dụng để trả phí hoặc trao đổi giá trị trong các ứng dụng blockchain.

**ICOs (Initial Coin Offerings):** ERC-20 được sử dụng rộng rãi để phát hành token trong quá trình ICO. Những dự án mới muốn gây quỹ cho các dự án của mình có thể tạo ra và phân phối token ERC-20 trong quá trình ICO để thu hút người đầu tư.

**Stablecoins:** Nhiều stablecoin (tiền mã hóa liên kết với giá trị ổn định) cũng được xây dựng dựa trên chuẩn ERC-20. Stablecoin ERC-20 giữ giá trị ổn định bằng cách liên kết với các tài sản truyền thống như USD hoặc vàng. Ví dụ như USDT (Tether) và USDC (USD Coin) là hai ví dụ nổi tiếng.

**Token sử dụng trong dApps:** Nhiều ứng dụng phi tập trung (dApps) trên Ethereum sử dụng token ERC-20 để thực hiện các chức năng cụ thể trong hệ thống của họ. Ví dụ, token có thể được sử dụng để trao đổi giữa người dùng trong một nền tảng xã hội phi tập trung hoặc để trao đổi giữa các thành viên trong một hệ thống phi tập trung.

**Token đại diện cho tài sản:** ERC-20 cũng được sử dụng để biểu diễn và giao dịch các tài sản truyền thống như bất động sản, cổ phiếu, hàng hóa và nhiều tài sản khác trên blockchain Ethereum. Token đại diện cho tài sản này giúp dễ dàng giao dịch và chia sẻ giữa các bên mà không cần trung gian truyền thống.

### Tiêu chuẩn ERC-721 (Non-Fungible Token Standard)

##### Giới thiệu chung ERC-721

ERC-721 là một chuẩn token NFT (Non-Fungible Token) được sử dụng phổ biến ngày nay, nó là một chuẩn token phức tạp hơn ERC20, với nhiều phần mở rộng tùy chỉnh. ERC-721 được tạo ra bởi bốn nhà phát triển là William Entriken, Dieter Shirley, Jacob Evans và Nastassia Sachs vào năm 2018.

Fungible Token (FT) có thể hiểu là mỗi token là giống hệt nhau, có thể thay thế cho nhau và có thể phân chia được. Các tài sản đáng tin cậy được sử dụng hàng ngày như đô la Mỹ, Bitcoin, Ethereum, việt nam đồng,... đều có thể xem là các tài sản có thể thay thế được cho nhau.

Trong khi đó, Non-Fungible Token (NFT) là một loại token đại diện cho một tài sản độc nhất. Thuộc tính này có nghĩa là mỗi token là duy nhất và không thể được hoán đổi cho một token khác. NFT mở rộng khái niệm “không thể thay thế” bằng cách tận dụng blockchain Ethereum để đại diện cho các tài sản vật lý hoặc tài sản kỹ thuật số duy nhất. Quyền sở hữu NFT được xác thực và theo dõi bằng cách sử dụng một blockchain công khai. NFT có nhiều trường hợp sử dụng. Nó có thể là một tài sản kỹ thuật số trong một trò chơi, một tác phẩm nghệ thuật mã hoá có khả năng sưu tầm hoặc thậm chí là một vật thể trong thế giới thực như bất động sản,...

Ta thấy ERC-721 và ERC-20 là hai chuẩn đối lập và có những tính chất cơ bản:

**Độc nhất và không thể thay thế:** Mỗi token ERC-721 là duy nhất và không thể thay đổi bằng bất kỳ token nào khác. Điều này cho phép việc đại diện và giao dịch các tài sản kỹ thuật số phi fungible có giá trị và tính cá nhân hóa cao.

**Tính chất sở hữu:** ERC-721 cho phép ghi nhận quyền sở hữu của người dùng đối với các tài sản kỹ thuật số. Điều này rất hữu ích trong việc tạo ra và quản lý các tài sản ảo, game item hoặc các phiên bản số hóa của tài sản thực.

**Ứng dụng đa dạng:** ERC-721 được sử dụng rộng rãi trong lĩnh vực các ứng dụng tài sản kỹ thuật số phi fungible. Ví dụ bao gồm trò chơi trên blockchain, nền tảng nghệ thuật số, thị trường tài sản ảo và các dự án kỹ thuật số phi fungible khác.

**Tiềm năng tạo ra giá trị:** Tính độc nhất và không thể thay thế của các token ERC-721 đã tạo ra cơ hội để tạo ra giá trị và thúc đẩy các hoạt động thương mại trong lĩnh vực tài sản kỹ thuật số phi fungible.

##### Các phương thức trong ERC-721

**balanceOf(address \_owner):** Function này trả về số lượng token ERC-721 mà địa chỉ (\_owner) sở hữu.

**ownerOf(uint256 \_tokenId):** Function này trả về địa chỉ của chủ sở hữu của một token ERC-721 cụ thể (\_tokenId).

**safeTransferFrom(address \_from, address \_to, uint256 \_tokenId, bytes data):** Function này cho phép chuyển nhượng một token ERC-721 từ địa chỉ (\_from) sang địa chỉ (\_to). Nếu (\_to) là một smart contract, nó phải thực hiện hàm onERC721Received để xác nhận việc chuyển nhượng. Tham số (\_data) có thể chứa dữ liệu tùy chọn.

**transferFrom(address \_from, address \_to, uint256 \_tokenId):** Function này cho phép chuyển nhượng một token ERC-721 từ địa chỉ (\_from) sang địa chỉ (\_to). Nó giống như safeTransferFrom, nhưng không yêu cầu xác nhận từ smart contract nhận token (\_to). Thay vào đó, nó chỉ yêu cầu sự ủy quyền trước đó từ chủ sở hữu hiện tại của token (\_from).

**safeTransferFrom(address \_from, address \_to, uint256 \_tokenId**): Function này tương tự như transferFrom, nhưng nó yêu cầu xác nhận từ smart contract nhận token (\_to). Smart contract này phải triển khai interface ERC721Receiver và thực hiện hàm onERC721Received để xác nhận việc chuyển nhượng token.

**approve(address \_approved, uint256 \_tokenId):** Function này cho phép địa chỉ (\_approved) được ủy quyền quyền sở hữu của một token ERC-721 cụ thể (\_tokenId).

**getApproved(uint256 \_tokenId):** Function này trả về địa chỉ đã được ủy quyền quyền sở hữu của một token ERC-721 cụ thể (\_tokenId).

**setApprovalForAll(address \_operator, bool \_approved):** Function này cho phép địa chỉ (\_operator) được ủy quyền quyền sở hữu của tất cả các token ERC-721 của người dùng (\_msgSender()). Tham số (\_approved) xác định xem quyền ủy quyền có được cấp hay không.

**isApprovedForAll(address \_owner, address \_operator):** Function này xác định xem địa chỉ (\_operator) có được ủy quyền quyền sở hữu tất cả các token ERC-721 của địa chỉ (\_owner) hay không.

##### Ứng dụng của ERC-721

Với đặc tính độc nhất không thể thay thế và tính sở hữu cao của mình, ERC721 là một chuẩn được áp dụng trong nhiều lĩnh vực như trò chơi NFT, thị trường giao dịch ảo hay thậm trí là các tác phẩm nghệ thuật điện tử. Một số những sản phẩm tiêu biểu như:

Trò chơi trên blockchain:

**CryptoKitties:** Là một trò chơi nổi tiếng trên blockchain Ethereum, nơi người dùng có thể sở hữu và chăm sóc các con mèo ảo độc đáo. Mỗi con mèo được biểu diễn bằng một token ERC-721 và có tính độc nhất, mang lại giá trị và tính cá nhân hóa cho từng con mèo.

**Decentraland:** Là một thế giới ảo phi tập trung được xây dựng trên blockchain Ethereum. Mỗi mảnh đất ảo trong Decentraland được đại diện bằng một token ERC-721, cho phép người dùng sở hữu, giao dịch và xây dựng trên đó.

-Nền tảng nghệ thuật số:

**SuperRare:** Là một nền tảng trưng bày và giao dịch các tác phẩm nghệ thuật kỹ thuật số. Mỗi tác phẩm nghệ thuật được đại diện bằng một token ERC-721, cho phép người dùng sở hữu và giao dịch chúng.

**NBA Top Shot:** Là một nền tảng nghệ thuật số cho việc thu thập và giao dịch các phiên bản giới hạn của highlight NBA. Mỗi highlight được biểu diễn bằng một token ERC-721, mang lại tính độc nhất và giá trị sưu tập cho người dùng.

Thị trường tài sản ảo:

**Decentraland Marketplace:** Là một thị trường nội bộ trong Decentraland, nơi người dùng có thể mua, bán và giao dịch các mảnh đất ảo và tài sản khác trong thế giới ảo.

**Axie Infinity:** Là một trò chơi blockchain nổi tiếng, nơi người dùng có thể sở hữu và giao dịch các con Axie (loại sinh vật ảo). Mỗi con Axie được biểu diễn bằng một token ERC-721, mang lại giá trị sưu tập và tính độc nhất cho từng con.

Sở hữu bất động sản:

**RealT:** Là một nền tảng sở hữu bất động sản ảo trên blockchain Ethereum. Mỗi tài sản bất động sản (như căn hộ, nhà) được đại diện bằng một token ERC-721, cho phép người dùng sở hữu và giao dịch chúng.

Phiếu mua hàng (vouchers):

**Crypto Vouchers:** Là một dự án sử dụng chuẩn ERC-721 để tạo và quản lý các phiếu mua hàng ảo. Mỗi phiếu mua hàng được biểu diễn bằng một token ERC-721, cho phép người dùng mua, bán và sử dụng các ưu đãi và sản phẩm được đại diện bởi các phiếu này.

### Tiêu chuẩn ERC-777 (Token Standard)

##### Giới thiệu chung ERC-777

ERC-777 là một tiêu chuẩn token trên blockchain Ethereum được đề xuất để nâng cấp cho tiêu chuẩn token phổ biến nhất là ERC-20. Đây là tiêu chuẩn được tạo ra vào năm 2017 bởi Jacques Dafflon, Jordi Baylina (hiện đang là lãnh đạo tại Polygon zkEVM) và Thomas Shabibi. Nó được giới thiệu để cải thiện một số chức năng của token ERC-20. Đồng thời, tiêu chuẩn mới sẽ tiết kiệm phí gas và có tính tương tác tốt hơn, nhưng cũng có tính phức tạp hơn và không được tương thích hoàn toàn với một số ứng dụng và ví.

ERC-777 được thiết kế nhằm cải thiện và tối ưu hơn so với chuẩn ERC-20 trong một số khía cạnh quan trọng. Một số điểm mà ERC-777 tối ưu hóa so với ERC-20:

**Tương tác trực tiếp với hợp đồng thông minh (Smart Contract Interoperability):** ERC-777 cho phép tương tác trực tiếp với các hợp đồng thông minh khác trên blockchain Ethereum. Điều này giúp tiện lợi hơn trong việc triển khai và tích hợp các dịch vụ, ứng dụng và hợp đồng thông minh khác.

**Chuyển tiếp (Forwarding):** ERC-777 cho phép chuyển tiếp (forwarding) các hành động từ một địa chỉ ví (wallet) hoặc một hợp đồng thông minh khác. Điều này mang lại tính linh hoạt và tiện lợi trong việc quản lý và thực hiện các hoạt động liên quan đến token.

**Tính bảo mật cao hơn:** ERC-777 cung cấp một lớp bảo vệ bổ sung bằng cách đảm bảo rằng các hành động không hợp lệ như chuyển tiếp (forwarding) hay lừa đảo không xảy ra. Điều này giúp tránh nhầm lẫn và rủi ro bảo mật trong việc giao dịch và sử dụng token.

**Đơn vị định lượng tùy chỉnh (Granularity):** ERC-777 cho phép tùy chỉnh đơn vị định lượng (granularity) của token. Điều này cho phép chia nhỏ và chuyển đổi token thành các đơn vị nhỏ hơn, tạo ra tính linh hoạt và tiện lợi hơn trong việc sử dụng và giao dịch token.

**Tương thích ngược (Backward Compatibility):** ERC-777 được thiết kế để tương thích ngược với chuẩn ERC-20. Điều này có nghĩa là các token ERC-20 có thể được nâng cấp lên thành ERC-777 mà không làm mất đi tính tương thích với các dịch vụ và ứng dụng đã tồn tại.

Những tối ưu này giúp ERC-777 mang lại nhiều lợi ích và khả năng mở rộng hơn so với ERC-20, tạo ra một chuẩn token thông minh và tiện lợi hơn cho các ứng dụng và dịch vụ trên blockchain Ethereum

##### Các phương thức trong ERC-777

Tiêu chuẩn ERC-777 gồm các phương thức:

**name()**: Trả về tên của token.

**symbol()**: Trả về ký hiệu đại diện cho token.

**totalSupply()**: Trả về tổng số lượng token đã được phát hành.

**balanceOf(address \_owner)**: Trả về số lượng token hiện có trong tài khoản của một địa chỉ cụ thể.

**granularity()**: Trả về giá trị đơn vị tối thiểu mà token có thể được chia nhỏ.

**defaultOperators()**: Trả về danh sách các địa chỉ được ủy quyền mặc định để thực hiện các hành động cho người dùng.

**send(address \_to, uint256 \_amount, bytes \_data)**: Gửi một số lượng token cho một địa chỉ cụ thể và chuyển tiếp dữ liệu tuỳ ý thông qua tham số **\_data**.

**operatorSend(address \_from, address \_to, uint256 \_amount, bytes \_data, bytes \_operatorData)**: Gửi một số lượng token từ một địa chỉ **\_from** đến một địa chỉ **\_to**, cho phép người dùng ủy quyền và chuyển tiếp dữ liệu liên quan.

**authorizeOperator(address \_operator)**: Ủy quyền một địa chỉ **\_operator** để thực hiện các hành động cho người dùng.

**revokeOperator(address \_operator)**: Thu hồi ủy quyền của một địa chỉ **\_operator**.

**isOperatorFor(address \_operator, address \_tokenHolder)**: Kiểm tra xem một địa chỉ **\_operator** có được ủy quyền để thực hiện các hành động cho một địa chỉ **\_tokenHolder** không.

**operatorSendTo(address \_from, address \_to, uint256 \_amount, bytes \_data, bytes \_operatorData)**: Tương tự như phương thức **operatorSend()**, nhưng gửi token trực tiếp đến một địa chỉ **\_to** mà không cần thông qua người dùng **\_from**.

Các phương thức này giúp quản lý và thực hiện các hoạt động liên quan đến token ERC-777, bao gồm chuyển giao token, ủy quyền và xác thực các địa chỉ được thực hiện các hành động trên token

##### Ứng dụng của ERC-777

Chuẩn ERC-777 mang đến một số ưu điểm và tính năng mới so với chuẩn ERC-20, mở ra nhiều ứng dụng tiềm năng trong các lĩnh vực khác nhau:

**Token tài sản:** ERC-777 có thể được sử dụng để biểu diễn và quản lý các tài sản kỹ thuật số, bao gồm cổ phiếu, trái phiếu, bất động sản và nhiều hơn nữa. Với tính năng ủy quyền mạnh mẽ, ERC-777 cho phép cấp quyền và kiểm soát các hành động trên tài sản, tạo ra một môi trường an toàn và minh bạch cho việc giao dịch và sở hữu tài sản.

**Tiền điện tử và thanh toán:** ERC-777 có thể được sử dụng như một đơn vị tiền tệ và phương tiện thanh toán trên mạng Ethereum. Tính năng tương tác trực tiếp với hợp đồng thông minh và khả năng chuyển tiếp giúp tạo ra một hệ thống thanh toán dễ dàng, nhanh chóng và an toàn trên blockchain.

**Tích hợp trong ứng dụng và dịch vụ:** Chuẩn ERC-777 cho phép tích hợp dễ dàng vào các ứng dụng và dịch vụ blockchain khác, bao gồm ví điện tử, sàn giao dịch, trò chơi và hệ thống thanh toán. Tính năng tương tác với hợp đồng thông minh và khả năng phòng ngừa lỗi giúp tăng tính linh hoạt và bảo mật trong việc sử dụng token.

**Quyền biểu quyết và ủy quyền:** ERC-777 cung cấp khả năng biểu quyết và ủy quyền trong các quyết định liên quan đến token. Người dùng có thể ủy quyền cho các địa chỉ khác thực hiện các hành động cho token mà không cần chuyển giao quyền sở hữu. Điều này hữu ích trong việc tổ chức bỏ phiếu, quản lý quỹ và quản lý dự án.

**Quản lý danh mục đầu tư:** ERC-777 cho phép tạo ra các token đại diện cho các tài sản tài chính và quản lý danh mục đầu tư. Những token này có thể được chia sẻ, giao dịch và kiểm tra giá trị một cách dễ dàng trên blockchain.  Phân phối token: ERC-777 có thể được sử dụng để phân phối token trong các chiến dịch công cộng (ICO) hoặc trong các chương trình phần thưởng, khuyến mãi và quỹ cộng đồng. Tính năng chuyển tiếp và tính bảo mật cao giúp tăng tính minh bạch và đáng tin cậy trong quá trình phân phối.

### Tiêu chuẩn ERC-1155 (Multi-Token Standard)

##### Giới thiệu chung ERC-1155

Chuẩn ERC-1155 là một chuẩn token đa chức năng trên blockchain Ethereum. Nó được giới thiệu nhằm cung cấp một giải pháp linh hoạt cho việc phát hành và quản lý các loại token khác nhau trên cùng một hợp đồng thông minh.

ERC-1155 cho phép phát hành không chỉ các token đơn lẻ như trong chuẩn ERC-20 và ERC-721, mà còn cung cấp khả năng phát hành các "class" (lớp) của token. Mỗi lớp token có thể có nhiều phiên bản (instances) và mỗi phiên bản có thể được phân biệt với một số chỉ mục (index). Điều này tạo ra một mô hình linh hoạt để quản lý và giao dịch các loại token đa dạng, bao gồm cả các phiên bản và các lớp token. Các tính năng và ưu điểm của chuẩn ERC-1155 bao gồm:

**Tiết kiệm chi phí:** Với ERC-1155, người dùng chỉ cần triển khai một hợp đồng thông minh duy nhất để quản lý nhiều loại token. Điều này giúp giảm thiểu chi phí triển khai và tiết kiệm tài nguyên mạng Ethereum.

**Tiết kiệm thời gian:** Chuẩn ERC-1155 giúp giảm thời gian và công sức cần thiết để tạo ra và quản lý các loại token khác nhau. Thay vì triển khai và duy trì nhiều hợp đồng thông minh riêng biệt, ERC-1155 cho phép quản lý tất cả các loại token từ một nơi.

**Tính linh hoạt:** ERC-1155 cho phép phát hành và quản lý các loại token đa dạng, bao gồm cả các phiên bản của token và các lớp token. Điều này tạo ra tính linh hoạt cao cho việc tạo ra các ứng dụng, trò chơi, thẻ bài, tài sản kỹ thuật số và nhiều hơn nữa.

**Tiết kiệm lưu trữ:** Với ERC-1155, chỉ cần lưu trữ một hợp đồng thông minh duy nhất, các token đa dạng và phiên bản có thể được quản lý. Điều này giúp giảm tải lưu trữ và tăng tính hiệu quả trong việc quản lý tài sản kỹ thuật số.

**Tương thích ngược:** Chuẩn ERC-1155 được thiết kế để tương thích ngược với các chuẩn token khác như ERC-20 và ERC-721. Điều này có nghĩa là các token có thể được nâng cấp từ các chuẩn cũ sang ERC-1155 mà không làm mất tính tương thích với các ứng dụng và dịch vụ hiện có.

ERC-1155 mang lại sự linh hoạt và tiện lợi trong việc phát hành, quản lý và giao dịch các loại token đa dạng trên blockchain Ethereum. Nó cung cấp một chuẩn token tiên tiến cho các ứng dụng, trò chơi, thẻ bài, tài sản kỹ thuật số và nhiều lĩnh vực khác

##### Các phương thức trong ERC-1155

1. Chức năng Batch Transfers

Batch transfer (chuyển đổi hàng loạt) là một tính năng quan trọng trong các hợp đồng ERC-1155, cho phép người dùng chuyển đổi nhiều token cùng một lúc từ nhiều địa chỉ nguồn đến một địa chỉ đích. Thay vì thực hiện nhiều giao dịch riêng lẻ, batch transfer giúp tiết kiệm thời gian và gas cost (phí giao dịch) bằng cách thực hiện tất cả các giao dịch trong một giao dịch duy nhất. Trong batch transfers (chuyển đổi hàng loạt) của hợp đồng ERC-1155, các function chính bao gồm:

**safeBatchTransferFrom:** Phương thức này cho phép chuyển đổi hàng loạt các token từ một địa chỉ nguồn đến một địa chỉ đích. Cú pháp của phương thức là:

function safeBatchTransferFrom( address \_from, address \_to, uint256[] memory \_ids, uint256[] memory \_amounts, bytes memory \_data

) external;

Trong đó:

\_from: Địa chỉ nguồn, là địa chỉ chủ sở hữu token.

\_to: Địa chỉ đích, là địa chỉ mà các token được chuyển đến.

\_ids: Mảng chứa danh sách các ID của các token cần chuyển đổi.

\_amounts: Mảng chứa số lượng tương ứng của các token cần chuyển đổi.

\_data: Dữ liệu bổ sung (tùy chọn) được truyền kèm với giao dịch.

**batchTransferFrom:** Phương thức này cho phép chuyển đổi hàng loạt các token từ một địa chỉ nguồn đến một địa chỉ đích, tương tự như safeBatchTransferFrom, nhưng không có kiểm tra an toàn. Cú pháp của phương thức là:

|  |
| --- |
| function batchTransferFrom(  address \_from, address \_to, uint256[] memory \_ids, uint256[] memory \_amounts  ) external; |

Trong đó:

from: Địa chỉ nguồn, là địa chỉ chủ sở hữu token.

\_to: Địa chỉ đích, là địa chỉ mà các token được chuyển đến.

\_ids: Mảng chứa danh sách các ID của các token cần chuyển đổi.

\_amounts: Mảng chứa số lượng tương ứng của các token cần chuyển đổi.

Cả hai phương thức trên cho phép chuyển đổi hàng loạt các token trong một lần giao dịch, giúp tối ưu hóa thời gian và phí giao dịch. Người dùng có thể chỉ định danh sách các ID và số lượng tương ứng của các token để chuyển đổi.

1. Chức năng Batch Balance

Batch balance (số dư hàng loạt) không phải là một phương thức chính thức trong hợp đồng ERC-1155. Tuy nhiên, chúng ta có thể triển khai một phương thức tùy chỉnh để tính toán số dư hàng loạt của các token trong hợp đồng ERC-1155.

|  |
| --- |
| function balanceOfBatch( address[] calldata \_owners,  uint256[] calldata \_ids  ) external view returns (uint256[] memory); |

Trong đó:

**\_owners**: Mảng các địa chỉ chủ sở hữu token cần lấy số dư.

**\_ids**: Mảng các ID của các token cần lấy số dư.

Phương thức này trả về một mảng **uint256[]** chứa số dư tương ứng của các token cho từng địa chỉ chủ sở hữu.

1. Chức năng Batch Approval

Batch Approval hay ủy quyền hàng loạt cũng là một chức năng mở rộng trong tiêu chuẩn ERC-1155, nó bao gồm 2 phương thức chính:

**setApprovalForAll:** Phương thức này được sử dụng để ủy quyền hoặc thu hồi ủy quyền cho một người khác (operator) trên toàn bộ các token của một chủ sở hữu (owner). Cú pháp của phương thức như sau:

|  |
| --- |
| function setApprovalForAll(  address \_operator, bool \_approved  ) external; |

Trong đó:

**\_operator**: Địa chỉ của người được ủy quyền hoặc thu hồi ủy quyền.

**\_approved**: Giá trị boolean (**true** hoặc **false**) để chỉ định xem ủy quyền được cấp cho **\_operator** hay được thu hồi.

**isApprovedForAll**: Phương thức này được sử dụng để kiểm tra xem một người (operator) có được ủy quyền bởi một chủ sở hữu (owner) hay không trên toàn bộ các token. Cú pháp của phương thức như sau:

|  |
| --- |
| function isApprovedForAll(  address \_owner, address \_operator  ) external view returns (bool); |

Trong đó:

\_owner: Địa chỉ của chủ sở hữu (owner) của các token.

\_operator: Địa chỉ của người được kiểm tra ủy quyền.

Phương thức isApprovedForAll trả về giá trị boolean (true hoặc false) để chỉ ra xem \_operator có được ủy quyền bởi \_owner hay không trên toàn bộ các token.

1. Hàm onERC1155Received

**onERC1155Received** là một hàm (function) tùy chọn mà bạn có thể triển khai trong hợp đồng của mình để nhận token ERC-1155 và thực hiện các xử lý phù hợp. Đây là một phần của giao thức chuẩn và được gọi tự động khi một token ERC-1155 được chuyển đến hợp đồng của bạn. Cú pháp của hàm onERC1155Received như sau:

|  |
| --- |
| function onERC1155Received(  address \_operator, address \_from, uint256 \_id, uint256 \_amount, bytes calldata \_data  ) external returns (bytes4); |

Trong đó:

\_operator: Địa chỉ của người thực hiện giao dịch chuyển token.

\_from: Địa chỉ của người gửi token.  \_id: ID của token ERC-1155.

\_amount: Số lượng token nhận được.

\_data: Dữ liệu tùy chọn được gửi kèm theo giao dịch.

Hàm onERC1155Received trả về một giá trị bytes4 để xác nhận rằng hàm đã được triển khai và có thể xử lý việc nhận token.

Bằng cách triển khai hàm onERC1155Received trong hợp đồng của bạn, bạn có thể tùy chỉnh và xử lý việc nhận token ERC-1155 theo ý muốn.

##### Ứng dụng của ERC-1155

ERC-1155 mang lại nhiều lợi ích như tính linh hoạt, tiết kiệm chi phí và tương thích cao, làm cho nó trở thành một lựa chọn hấp dẫn trong việc tạo, quản lý và giao dịch các tài sản kỹ thuật số trong các ứng dụng thực tế.

-Trong lĩnh vực game:

ERC-1155 được sử dụng rộng rãi để tạo và quản lý các phiên bản số hóa của vật phẩm trong trò chơi:

**Đa dạng vật phẩm:** ERC-1155 cho phép tạo ra nhiều loại vật phẩm khác nhau trong một hợp đồng duy nhất. Điều này rất hữu ích trong việc tạo ra một hệ sinh thái game phong phú với nhiều loại vật phẩm có giá trị và tính năng khác nhau.

**Tiết kiệm chi phí gas:** So với việc tạo một hợp đồng riêng biệt cho mỗi loại vật phẩm, ERC-1155 giúp tiết kiệm chi phí gas trong việc triển khai và giao dịch vật phẩm. Điều này đặc biệt quan trọng khi có hàng ngàn hoặc hàng triệu vật phẩm được tạo ra trong game.

**Quản lý linh hoạt:** ERC-1155 cung cấp các chức năng quản lý linh hoạt cho vật phẩm trong game. Nó cho phép xác định số lượng hạn chế hoặc không hạn chế cho mỗi loại vật phẩm, quản lý việc chuyển đổi, giao dịch và sở hữu của người chơi.

**Tích hợp dễ dàng:** Nhờ chuẩn hóa, các token ERC-1155 có thể dễ dàng tích hợp vào các nền tảng game hiện có. Điều này giúp giảm đáng kể công sức và thời gian tích hợp vật phẩm số hóa vào trò chơi.

**Tương tác và sáng tạo:** ERC-1155 cung cấp các khả năng tương tác và sáng tạo cho vật phẩm trong game. Người chơi có thể kết hợp, chế tạo và tạo ra những vật phẩm mới từ các phiên bản khác nhau của ERC-1155. Điều này mở ra nhiều khả năng tạo ra trải nghiệm tương tác độc đáo trong game.

-Vật phẩm và bảo hành trong thương mại điện tử:

Với ERC-1155, người dùng có thể tạo ra các token đại diện cho các vật phẩm trong thương mại điện tử như sản phẩm điện tử, quần áo, đồ trang sức và nhiều hơn nữa. Mỗi token có một ID duy nhất và có thể được tùy chỉnh với các thuộc tính và thông tin liên quan đến vật phẩm như tên, mô tả, hình ảnh, giá trị và trạng thái bảo hành.

Ngoài ra, ERC-1155 cũng hỗ trợ tính năng bảo hành cho các vật phẩm trong thương mại điện tử. Người dùng có thể gắn kết thông tin bảo hành cho từng token, bao gồm thời gian bảo hành, điều kiện và quyền lợi của khách hàng. Người dùng có thể kiểm tra trạng thái bảo hành của vật phẩm và yêu cầu sửa chữa hoặc đổi trả nếu cần thiết.

Điểm mạnh của ERC-1155 trong việc quản lý vật phẩm và bảo hành trong thương mại điện tử là khả năng tiết kiệm chi phí và tối ưu hóa việc lưu trữ trên blockchain. Thay vì tạo một token riêng biệt cho mỗi vật phẩm và bảo hành, ERC1155 cho phép gộp chung nhiều loại token trong cùng một hợp đồng thông minh, giúp tiết kiệm không gian lưu trữ và giảm chi phí giao dịch.

Đồng thời, việc sử dụng ERC-1155 cũng cung cấp tính linh hoạt cho các nhà phát triển và doanh nghiệp trong việc tạo ra các ứng dụng và nền tảng thương mại điện tử phức tạp hơn, bao gồm việc quản lý vật phẩm, bảo hành, giao dịch và theo dõi trạng thái của các sản phẩm trong môi trường blockchain.

**Trong việc cấp chứng chỉ số và bằng cấp**:

ERC-1155 cung cấp một khung công nghệ mạnh mẽ để quản lý và giao dịch chứng chỉ số và bằng cấp trên blockchain Ethereum. Bằng cách sử dụng ERC1155, bạn có thể tạo ra các token đại diện cho các chứng chỉ số và bằng cấp, bao gồm các bằng cấp, văn bằng, chứng chỉ nghề và các khóa học hoàn thành, cho phép việc xác thực và chia sẻ thông tin về thành tích và năng lực của cá nhân hoặc tổ chức một cách minh bạch và bảo mật

##### Tổng kết chương 2

Trong chương 2, những khái niệm, kiến thức cơ bản về nền tảng Ethereum và hợp đồng thông minh đã được trình bày. Qua đó, chúng ta thấy được mạng Ethereum có thể được coi là một siêu máy tính cho phép mọi người dùng trên mạng tải và chạy chương trình của riêng mình. Các giao dịch điện tử được thực hiện một cách công bằng và minh bạch với các hợp đồng điệng tử áp dụng các tiêu chuẩn ERC.

Chương này đã đề cập đến bốn tiêu chuẩn ERC là 20. 721, 777 và 1155. Trong đó, mỗi tiêu chuẩn đều có những ưu và nhược điểm của riêng mình. ERC20 là tiêu chuẩn về các token có thể thay thế lẫn nhau thì ERC-721 tạo ra các token độc nhất không thể thay thế gọi là NFT. ERC-777 là bản nâng cấp của ERC-20 giúp cải thiện tính năng bảo mật, tính tương tác, giảm phí gas. ERC-1155 có thẻ coi như là tiêu chuẩn tổng hợp của ERC-20 và ERC-721 khi kế thừa các đặc tính tiêu biểu của chúng.

Tuy nhiên, để có thể hiểu rõ hơn về cách hoạt động của các tiêu chuẩn này, chúng ta cần phải có những thực nghiệm về chúng.

# TRIỂN KHAI ỨNG DỤNG CÁC TIÊU CHUẨN ERC

## Triển khai ứng dụng ERC-20

ERC-20 là một chuẩn dùng để tạo và quản lý các Fungible Token. Phần

thực nghiệm này sẽ xây dựng một smart contract theo tiêu chuẩn ERC-20 để tạo

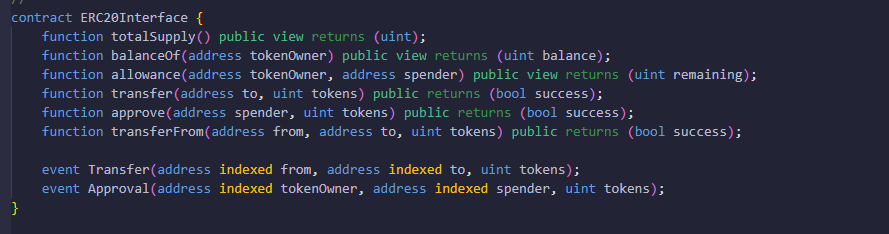
một token là KMACoin. Smart contract sẽ được viết bằng Solidity- ngôn ngữ lập

trình bậc cao chuyên dùng để viết Smart Contract. Token sẽ được triển khai trên

mạng thử nghiệm Sepolia của Ethereum.

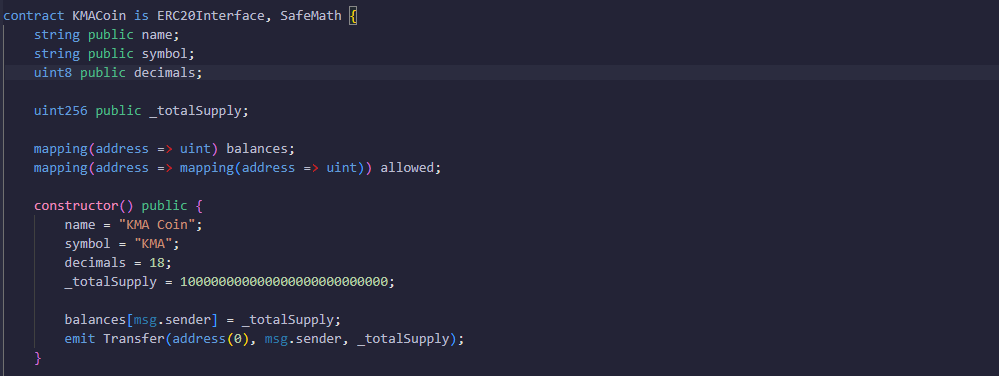
### Xây dựng Smart Contract theo ERC-20

-Tạo Interface ERC-20



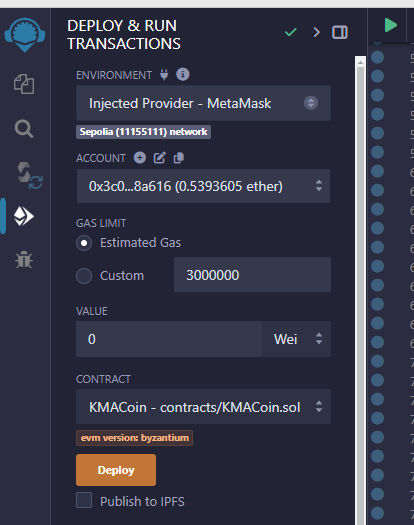
Hình 3.1 Tạo Interface ERC-20

-Xây dựng hợp đồng dựa trên chuẩn ERC-20



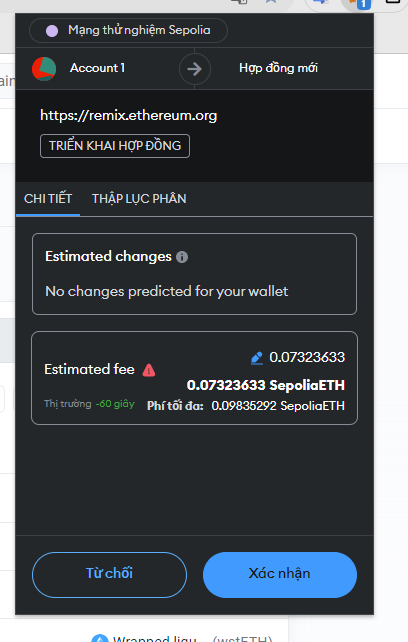
Hình 3.2 Xây dựng hợp đồng dựa trên chuẩn ERC-20

### Triển khai hợp đồng KMACoin Tiến hành Deploy hợp đồng trên mạng thử nghiệm Sepolia



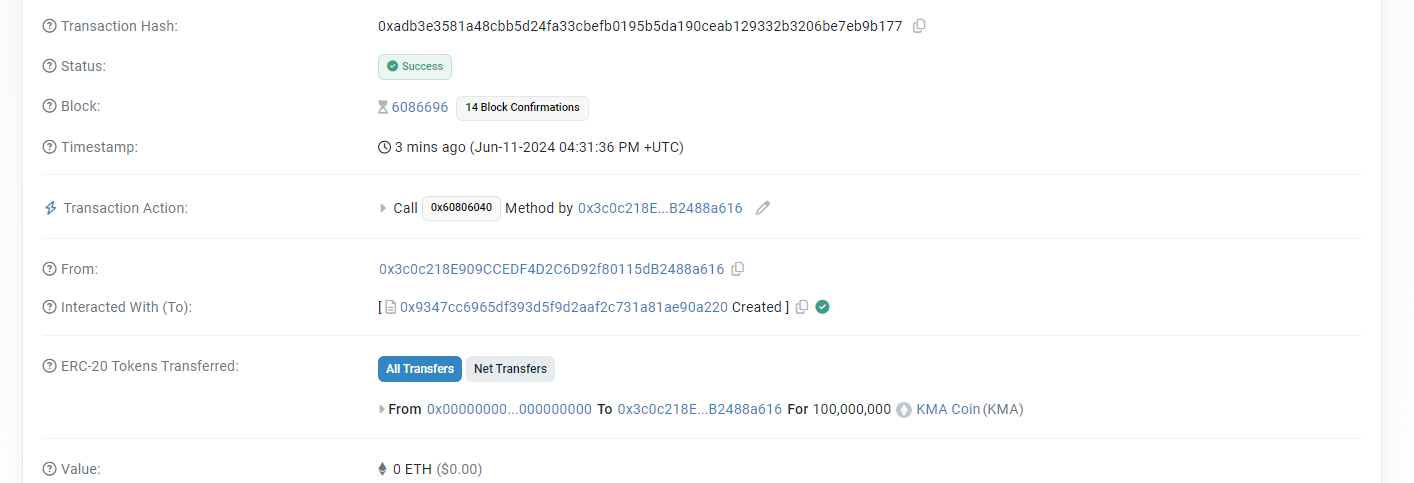
Hình 3.3 Kết nối với ví metamask

-Xác nhận hợp đồng trên Meta Mask



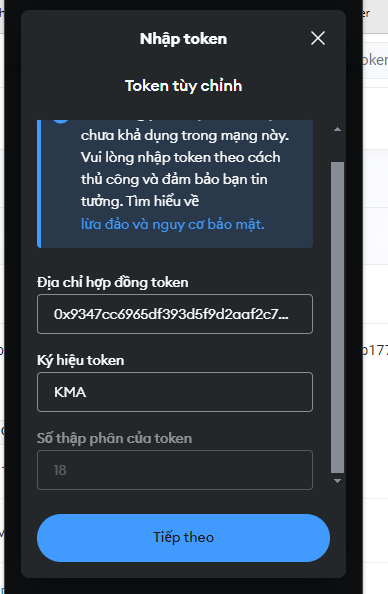
Hình 3.4 Xác nhận hợp đồng với metamask

-Kiểm tra hợp đồng trên ethereum scan và ta thấy hợp đồng được thực hiện thành công.



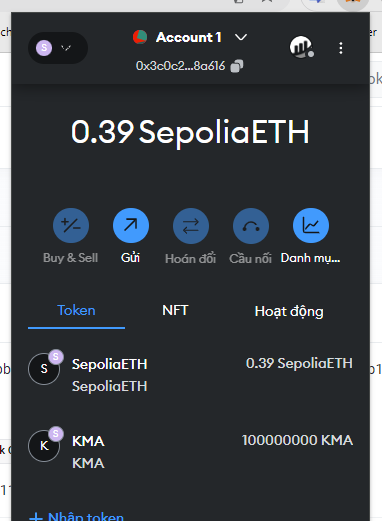
Hình 3.5 Kiểm tra hợp đồng trên ethereum scan

-Nhập địa chỉ Token đã tạo vào Meta Mask



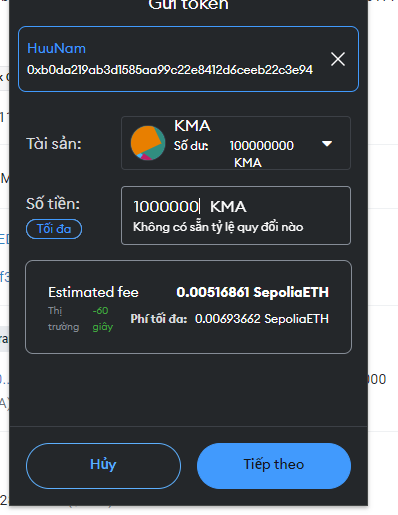
Hình 3.6 Nhập địa chỉ Token đã tạo vào Metamask

-Nhập Token thành công trên Meta Mask.

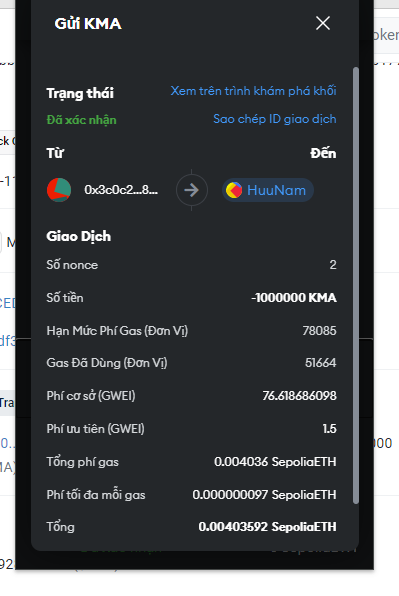


Hình 3.7 Nhập token thành công trên metamask

-Gửi thử Token KMA cho một tài khoản khác



Hình 3.8 Gửi Token cho 1 tài khoản khác



Hình 3.9 Kiểm tra trạng thái gửi token

## Triển khai ứng dụng ERC-721

### Xây dựng Smart Contract theo chuẩn ERC-721

Nhiệm vụ: Sử dụng ngôn ngữ Solidity để tạo ra một contract, có chức năng tạo ra, trao đổi và quản lí NFT (Non-Fungible Token).

Ngôn ngữ sử dụng để lập trình là Solidity.

Trình biên dịch được sử dụng là Remix IDE giúp biên dịch Sodility contract thành bytecode, chạy trên nền tảng Ethereum.

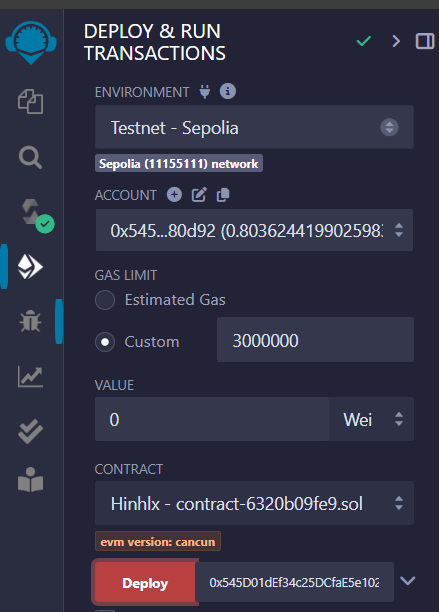
Mã nguồn đã xây dựng:

|  |
| --- |
| // SPDX-License-Identifier: MIT  // Compatible with OpenZeppelin Contracts ^5.0.0  pragma solidity ^0.8.20;  import "@openzeppelin/contracts@5.0.2/token/ERC721/ERC721.sol";  import "@openzeppelin/contracts@5.0.2/token/ERC721/extensions/ERC721Enumerable.sol";  import "@openzeppelin/contracts@5.0.2/token/ERC721/extensions/ERC721URIStorage.sol";  import "@openzeppelin/contracts@5.0.2/token/ERC721/extensions/ERC721Pausable.sol";  import "@openzeppelin/contracts@5.0.2/access/Ownable.sol";  contract Hinhlx is ERC721, ERC721Enumerable, ERC721URIStorage, ERC721Pausable, Ownable {  uint256 private \_nextTokenId;  constructor(address initialOwner)  ERC721("hinhlx", "hlx")  Ownable(initialOwner)  {}  function pause() public onlyOwner {  \_pause();  }  function unpause() public onlyOwner {  \_unpause();  }  function safeMint(address to, string memory uri) public onlyOwner {  uint256 tokenId = \_nextTokenId++;  \_safeMint(to, tokenId);  \_setTokenURI(tokenId, uri);  }  function \_update(address to, uint256 tokenId, address auth)  internal  override(ERC721, ERC721Enumerable, ERC721Pausable)  returns (address)  {  return super.\_update(to, tokenId, auth);  }  function \_increaseBalance(address account, uint128 value)  internal  override(ERC721, ERC721Enumerable)  {  super.\_increaseBalance(account, value);  }  function tokenURI(uint256 tokenId)  public  view  override(ERC721, ERC721URIStorage)  returns (string memory)  {  return super.tokenURI(tokenId);  }  function supportsInterface(bytes4 interfaceId)  public  view  override(ERC721, ERC721Enumerable, ERC721URIStorage)  returns (bool)  {  return super.supportsInterface(interfaceId);  }  } |

### Triển khai hợp đồng theo chuẩn ERC-721

Tiến hành compile, nếu không có lỗi thì sẽ có một dấu tích cạnh biểu tượng

như hình sau:



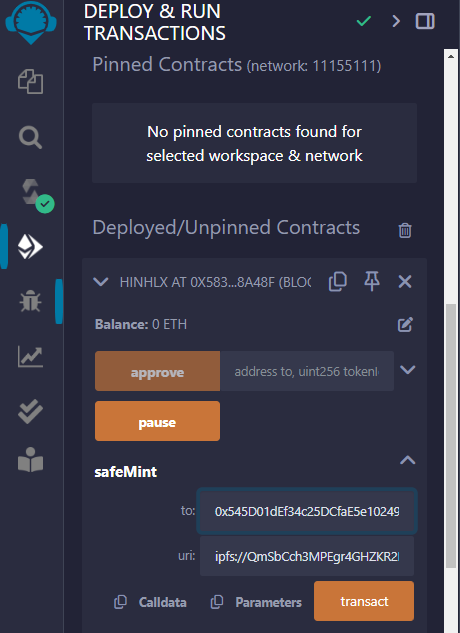
Hình 3.10 Deploy hợp đồng thwo chuẩn ERC-721

Địa chỉ ví thứ nhất: 0x545D01dEf34c25DCfaE5e10249F6fcAdA1280d92

Địa chỉ ví thứ 2 là: 0xf7aEd7A0F253781d1832a878dB5725e7Bca7E549

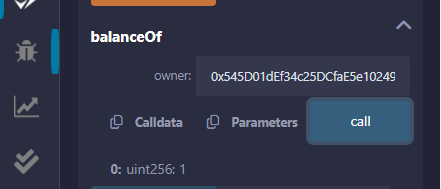
Tiến hành Deploy, sẽ sinh ra địa chỉ contract và các function bên dưới:

-Tiến hành tạo một NFT qua hàm safeMint, Tiến hành nhập giá trị địa chỉ của ví sở hữu NFT, uri và nhấn transact:



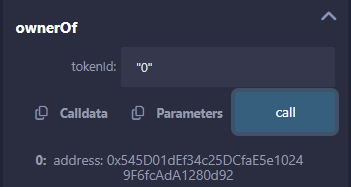
Hình 3.11 Tạo 1 NFT qua hàm safeMint

-Kiểm tra số lượng NFT có trong ví:



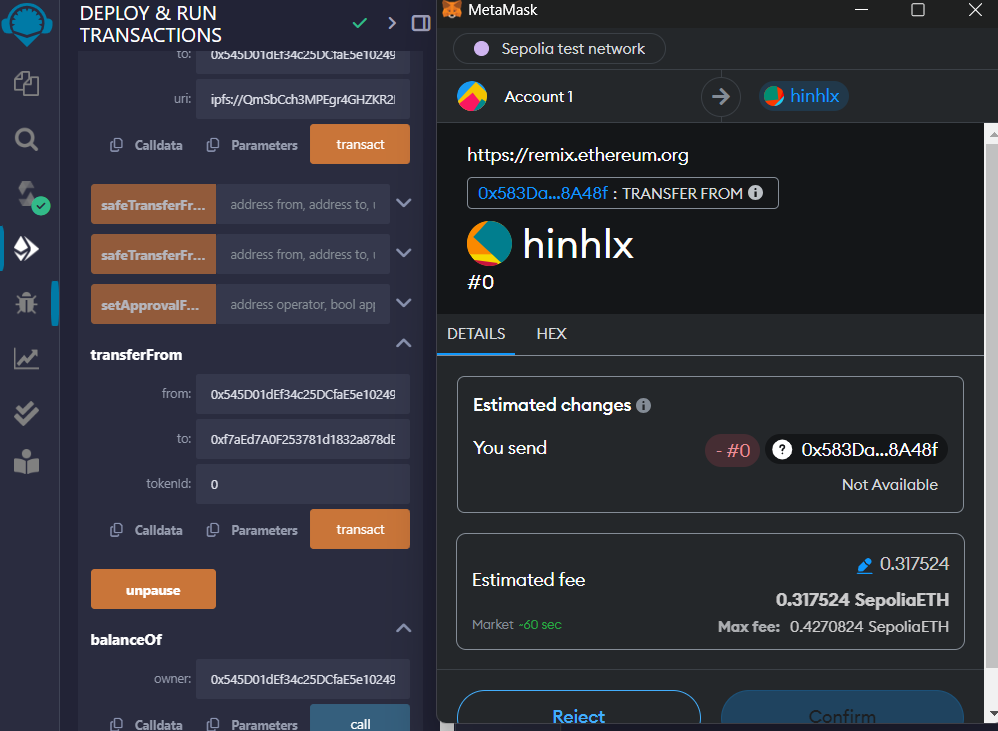
Hình 3.12 Kiểm tra số lượng NFT có trong ví

-Kiểm tra chủ sở hữu của NFT “0”:



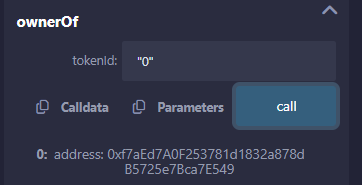
Hình 3.13 Kiểm tra chủ sở hữu của NFT “0”

-Chuyển NFT “0” từ địa chỉ ví thứ 1 sang địa chỉ ví thứ 2



Hình 3.14 Chuyển NFT ”0” từ ví 1 sang ví 2

-Kiểm tra lại cho thấy NFT “0” đã ở ví thứ 2



Hình 3.15 Kiểm tra NFT “0” ở ví 2

## Triển khai ứng dụng ERC-1155

Ý tưởng thực hiện : Phát triển smart contract dựa theo tiêu chuẩn ERC-1155 để tạo nên 4 token ảo sau đó có thể giao dịch các token này cho những người dùng khác trong mạng.

Công cụ sử dụng : Solidity, Remix, Metamask

-Đầu tiên là khởi tạo dự án bằng cách truy cập vào https://remix.ethereum.org/. Để xây dựng và triển khai một smartcontract ta cần viết một đoạn mã Solidity cho smartcontract trên trình soạn thảo Remix. Tại mục File explorer ta tạo 1 file mới và đặt tên là LOL.sol

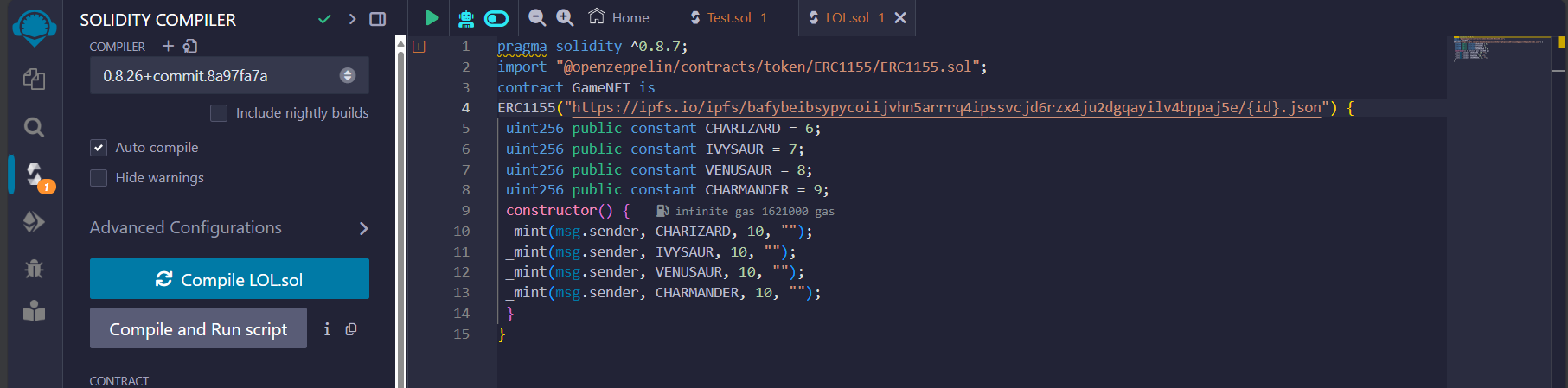


Hình 3.16 Viết đoạn mã Solidity cho smart contract

-Hợp đồng được triển khai theo mã nguồn sau:

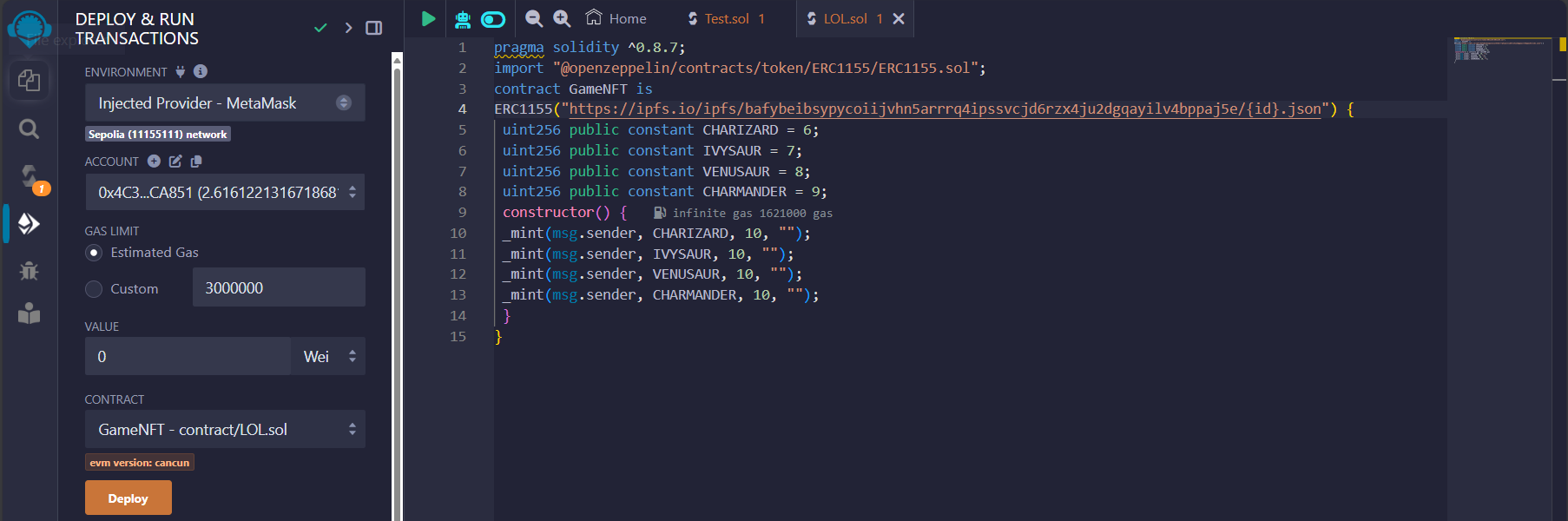
|  |
| --- |
| pragma solidity ^0.8.7;  import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol";  contract GameNFT is ERC1155("https://ipfs.io/ipfs/bafybeibsypycoiijvhn5arrrq4ipssvcjd6rzx4ju2dg qayilv4bppaj5e/{id}.json") {  uint256 public constant CHARIZARD = 6;  uint256 public constant IVYSAUR = 7;  uint256 public constant VENUSAUR = 8;  uint256 public constant CHARMANDER = 9;  constructor() { \_mint(msg.sender, CHARIZARD, 10, "");  \_mint(msg.sender, IVYSAUR, 10, "");  \_mint(msg.sender, VENUSAUR, 10, "");  \_mint(msg.sender, CHARMANDER, 10, "");  }  } |

-Sau khi viết xong mã chương trình ,bấm vào nút "Compile" để biên dịch smart contract.



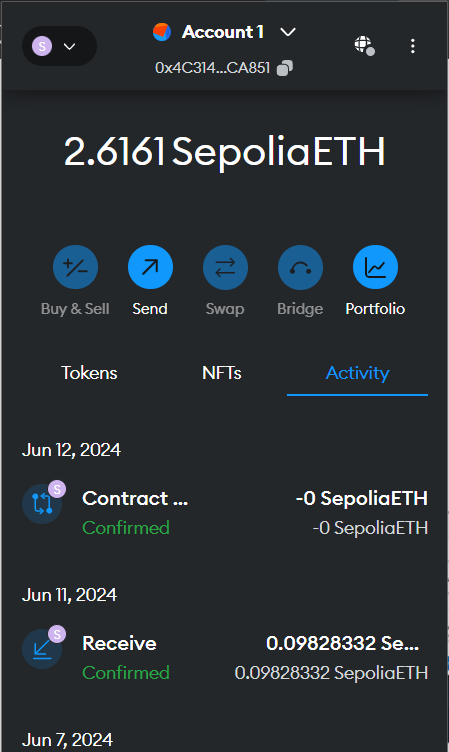
Hình 3.17 Biên dịch smart contract

-Sau khi biên dịch thành công, xuống phần “DEPLOY & RUN TRANSACTIONS” để có thể triển khai smart contract bằng cách sử dụng một trong các tùy chọn triển khai có sẵn trên Remix. Chọn môi trường “Injected Provider – MetaMask”



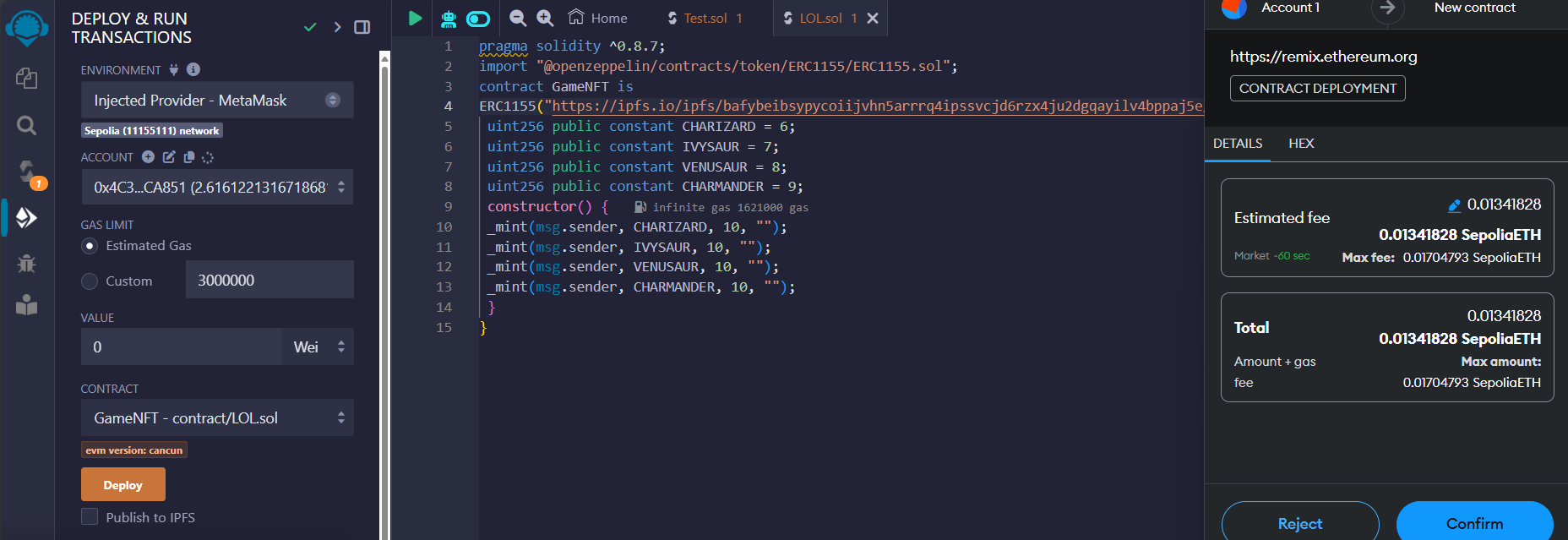
Hình 3.18 Triển khai smart contract

-Kết nối với ví Metamask .Ở đây sử dụng mạng thử nghiệm Sepolia.



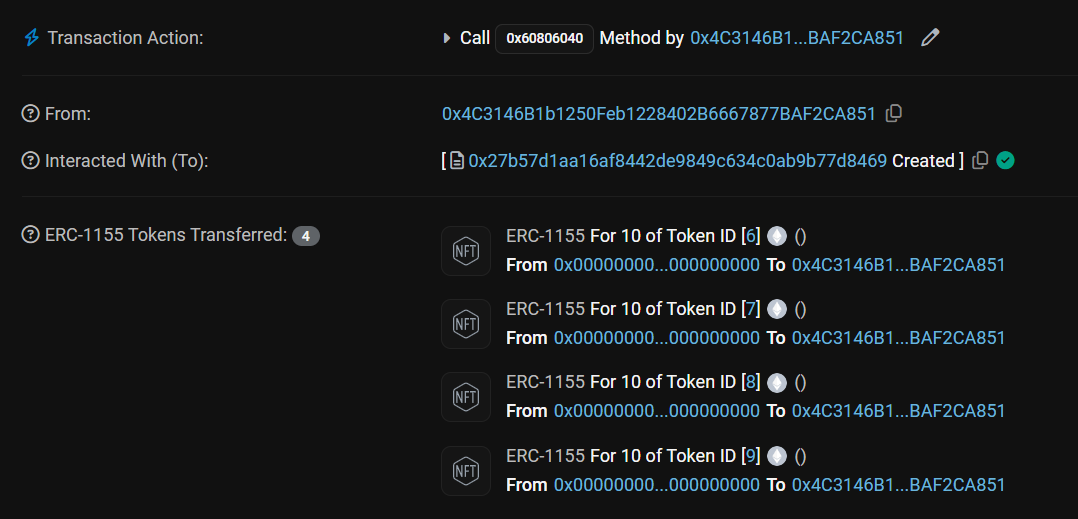
Hình 3.19 Kết nối Remix với ví metamask

-Sau khi Deploy ta phải trả phí gas để triển khai hợp đồng:



Hình 3.20 Trả phí gas cho hợp đồng

-Hợp đồng sau khi được triển khai có thể thấy được tại mạng thử nghiệm Sepolia: <https://sepolia.etherscan.io>

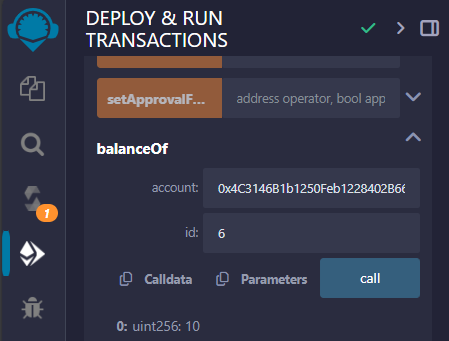


Hình 3.21 Kiểm tra hợp đồng đã triển khai

Đã tạo được 4 token có ID lần lượt là 6,7,8,9.

-Tiếp theo sẽ thực hiện chuyển token từ tài khoản đang sở hữu qua tài khoản chưa sở hữu token. Sử dụng hàm balanceOf(address \_owner, uint256 \_id): Trả về số lượng token có sẵn của một người dùng (\_owner) với ID tương ứng (\_id).

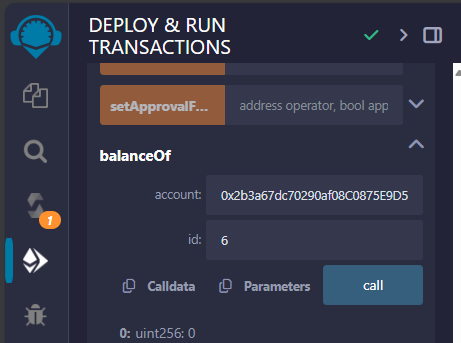
-Ta thực hiện kiểm tra xem số token có id = 6 mà tài khoản 0x4C3146B1b1250Feb1228402B6667877BAF2CA851 đang sở hữu là bao nhiêu bằng cách call hàm trên



Hình 3.22 Kiểm tra token ID = 6 trên ví gửi

Ta thấy ở đây tài khoản đang sở hữu 10 token có ID=6.

-Thực hiện kiểm tra số token có ID = 6 đang sở hữu trên tài khoản 0x2b3a67dc70290af08C0875E9D552e98f7a23e007

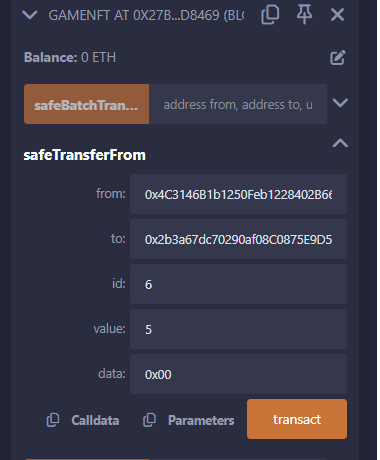


Hình 3.23 Kiểm tra token ID = 6 trên ví nhận

Ta thấy tk trên đang sở hữu 0 token có ID=6.

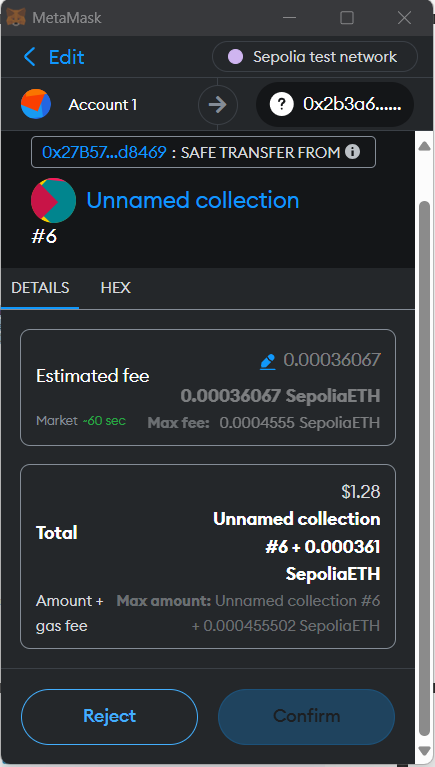
-Tiếp theo ta sẽ sử dụng hàm : safeTransferFrom(address \_from, address \_to, uint256 \_id, uint256 \_amount, bytes \_data): Chuyển token từ một người dùng (\_from) đến người dùng khác (\_to) với số lượng tương ứng (\_amount) và dữ liệu (metadata) liên quan (\_data).

-Thực hiện chuyển đổi 5 token từ ví gửi đến ví nhận



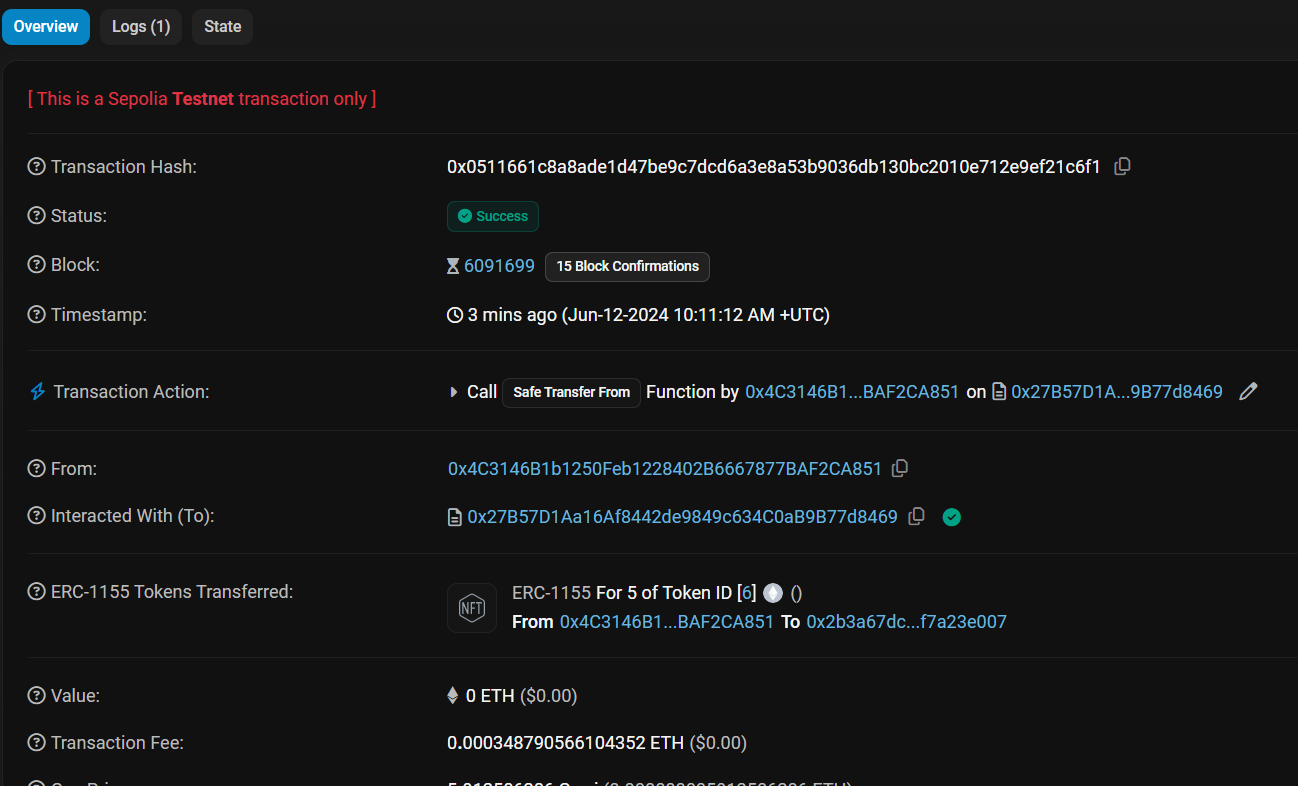
Hình 3.24 Chuyển 5 token có ID = 6

-Sau khi ấn transact sẽ hiện giao dịch:



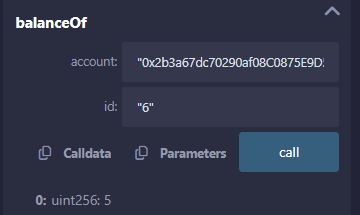
Hình 3.25 Trả fee để xác nhận giao dịch

-Ấn xác nhận để hoàn tất giao dịch. Sau khi giao dịch thành công , ta có thể kiểm tra giao dịch trên trang web : <https://sepolia.etherscan.io>



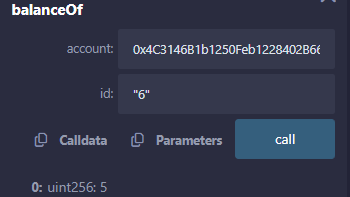
Hình 3.26 Kiểm tra giao dịch

-Tiến hành kiểm tra ví nhận token:



Hình 3.27 Kiểm tra token ở ví nhận

-Tiến hành kiểm tra ví gửi token:



Hình 3.28 Kiểm tra token ở ví gửi

Như vậy, quá trình chuyển 5 token có ID = 6 từ ví có địa chỉ 0x4C3146B1b1250Feb1228402B6667877BAF2CA851 đến ví có địa chỉ 0x2b3a67dc70290af08C0875E9D552e98f7a23e007 thành công và giao dịch được công khai trên mạng Sepolia Testnet

## Tổng kết chương 3

Chương 3 tiến hành xây dựng các hợp đồng thông minh dựa trên các tiêu chuẩn ERC-20, ERC-721 và ERC-1155 để tiến hành tạo và quản lý các token. Với hợp đồng xây dựng theo chuẩn ERC-20 sẽ tạo và quản lý các token có thể quy đổi và phân chia như một loại tiền điện tử, tuy nhiên việc đúc và phát hành đồng tiền này chỉ xảy ra một lần duy nhất nên không thể thay đổi đơn vị cũng như số lượng phát hành sau này. Với token được đúc và quản lý bởi hợp đồng được xây dựng trên chuẩn ERC-721, đây là một token độc nhất và không thể phân chia. Hợp đồng được xây dựng trên chuẩn 1155 tạo và quản lý nhiều loại token trên một smart contract.

# TỔNG KẾT

Kết thúc bài báo cáo, nhóm chúng em đã tìm hiểu được tổng quan về công nghệ blockchain nói chung và nền tảng blockchain Ethereum nói riêng. Các nội dung về khái niệm, cấu trúc hệ thống, cơ chế hoạt động của chúng đac được tổng hợp trong bà báo cáo. Bên cạnh đó, bài báo cáo đã trình bày các nội dung cơ bản về các tiêu chuẩn ERC-20, ERC-721, ERC-777, ERC-1155. Cho thấy cách hoạt động và khả năng ứng dụng của chúng trong thực tiễn qua demo về việc xây dựng các hợp đồng thông minh dựa trên các tiêu chuẩn ERC-20, ERC-721 và ERC1155.

Tuy nhiên, do sự hạn chế về kiến thức và khả năng tìm hiểu, nhóm chúng em nhận thấy bài báo cáo chưa được sâu dsawsc trong việc trình bày nội dung các tiêu chuẩn ERC và triển khai ứng dụng của chúng. Trong tương lai, nhóm chúng em sẽ tiếp tục tìm hiểu và trau dồi thêm kiến thức để có thể vận dụng các tiêu chuẩn này vào thực tiến.

Cuối cùng, nhóm chúng em xin gửi lời cảm ơn chân thành tới thầy Triệu

Anh Dũng đã nhiệt tình giảng dạy và hướng dẫn chúng em hoàn thành bà báo cáo này.

Chúng em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

1. Blockchain-based Smart Contracts -Applications and Challenges

Nhóm tác giả: Yining Hu, Madhusanka Liyanage, Ahsan Manzoor, Kanchana Thilakarathna

1. Smart Contract Upgradeability on the Ethereum Blockchain Platform:

An Exploratory Study

Nhóm tác giả: Ilham Qasse, Mohammad Hamdaqa, Björn Þór Jónsson

1. Tài liệu “ERC-20 Token Standard” (21/05/2023). Tác giả: Corwin Smith

Link: <https://ethereum.org/vi/developers/docs/standards/tokens/erc-20/>

1. Tài liệu “ERC-721 Non-Fungible Token Standard” (08/04/2023). Tác giả: Corwin Smith

Link: <https://ethereum.org/vi/developers/docs/standards/tokens/erc-721/>

1. Tài liệu “ERC-777 Token Standard” (30/09/2022). Tác giả: aul Wackerow

Link: <https://ethereum.org/vi/developers/docs/standards/tokens/erc-777/>

1. Tài liệu “ERC-1155 Multi-Token Standard”. (08/04/2023). Tác giả: Corwin Smith

Link: <https://ethereum.org/vi/developers/docs/standards/tokens/erc-1155/>

1. Topic “Blockchain” của thư viện mã nguồn mở Wikipedia. Link: <https://vi.wikipedia.org/wiki/Blockchain>