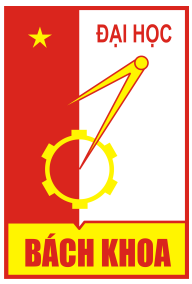
Trường Đại học Bách Khoa Hà Nội

Viện Toán ứng dụng và Tin học



BÁO CÁO THỰC TẬP KĨ THUẬT

Cấp phát nội dung số trên chuỗi khối sử dụng hệ thống tập tin liên hành tinh

Doanh nghiệp:

Division 6 - *Công ty cổ phần Rikkeisoft*

Thực hiện:

Đỗ Minh Tuấn - *20185419*

Hà Nội, 2022

Nhận xét của doanh nghiệp hướng dẫn

Mục đích và nội dung của đề tài

Kết quả đạt được

Ý thức làm việc của sinh viên thực hiện

Hà Nội, ngày ... tháng ... năm ...

Đại diện doanh nghiệp

(Ký và ghi rõ họ tên)

Lời cảm ơn

Khoảng thời gian được làm việc trong môi trường doanh nghiệp đem lại cho em rất nhiều kinh nghiệm, cũng như những kiến thức chuyên ngành quý giá. Em xin gửi lời cảm ơn tới toàn thể anh, chị tại *Công ty cổ phần Rikkeisoft*[1], đặc biệt là anh *Trần Văn Luật* (*Sub-Leader* của *Division 6*), mọi người đã giúp đỡ em rất nhiều trong quá trình học tập tại đây. Em xin cảm ơn mọi người.

Tuy một học kỳ là ngắn ngủi, nhưng với định hướng rõ ràng, quả thực, nó mang lại cho em quá là nhiều kiến thức, đưa em đến những chân trời mới về công nghệ. Sau mỗi thành tựu mà ai đó đạt được, sẽ thật thiếu sót khi không nhìn vào sự nỗ lực của chính họ. Bản thân em cũng đã dành ra khá nhiều công sức tìm hiểu, thử nghiệm, triển khai, và mang đến kết quả phù hợp nhất cho đề tài mà mình đã chọn. Em tự tin với chất lượng của bài báo cáo này.

[1]https://rikkeisoft.com/

Mục lục

Chương 1

Tổng quan về bài toán cấp phát nội dung số

Đã từ lâu, các doanh nghiệp cũng như các cá nhân bắt đầu lựa chọn các nền tảng đám mây cho mục đích lưu trữ dữ liệu. Những nhà cung cấp dịch vụ phổ biến hiện nay có thể kể đến như Google với *Google Drive*, Microsoft với *OneDrive*, DropBox, vân vân. Các dịch vụ này tương đối quen thuộc với hầu hết mọi người, hỗ trợ lên tới *15GiB* dữ liệu cho người dùng không trả phí (miễn phí).

Do phổ biến như vậy, các doanh nghiệp dần triển khai làm việc trên các dịch vụ này. Dễ dàng nhận thấy rằng, những chứng nhận (như chứng chỉ, văn bằng, huy chương của các cuộc thi, vân vân) đã có thể lưu trữ và phân phối trên nền tảng số. Nhiều doanh nghiệp thậm chí đã chấm dứt cấp phát dưới dạng vật lý (giấy tờ, các vật liệu giả kim hoặc kim loại, vân vân) để giảm thiểu chi phí sản xuất. Cũng có nhiều doanh nghiệp tự xây dựng hệ thống lưu trữ đám mây cho riêng mình để thuận tiện quản lý.

Tuy nhiên, việc phụ thuộc vào một nhà cung cấp dịch vụ cũng đem đến rủi ro. Với sự gia tăng ngày càng nhiều của các vụ tấn công nhằm vào các nhà cung cấp dịch vụ đám mây, "khách hàng" chưa hoàn toàn yên tâm trong việc lưu trữ dữ liệu. Không ít các cuộc tấn công như vậy khiến dữ liệu bị sai lệch, không thể

khôi phục lại được, hoặc có thể khôi phục lại nhưng dữ liệu không được như ban đầu. Rất nhiều giải pháp khác được thảo luận nhằm đảm bảo cho việc lưu trữ nội dung trên nền tảng số trở lên an toàn hơn.

Chương 2

Chuỗi khối và mạng Ethereum

2.1 Tổng quan về chuỗi khối

Một chuỗi khối là một sổ cái điện tử *phân tán*[1], *phi tập trung*[2], bao gồm các bản ghi được gọi là *khối* (block) thường được dùng để ghi lại các *giao dịch* (transaction) qua các máy tính. Nói một cách dễ hiểu, chuỗi khối là một cơ chế cơ sở dữ liệu tiên tiến cho phép chia sẻ thông tin minh bạch trong một mạng lưới. Cơ sở dữ liệu chuỗi khối lưu trữ dữ liệu trong các khối được liên kết với nhau trong một chuỗi. Dữ liệu có sự nhất quán theo trình tự thời gian vì bạn không thể xóa hoặc sửa đổi chuỗi mà không có sự đồng thuận từ mạng lưới. Do đó, chuỗi khối được coi như một sổ cái không thể chỉnh sửa hay biến đổi để theo dõi các thông tin theo thời gian.

[1]Distributed
[2]Decentralized

2.2 Mạng Ethereum

2.2.1 Lịch sử

Ethereum là một nền tảng mã nguồn mở dựa trên chuỗi khối, hỗ trợ *Hợp đồng thông minh*[3]. Ethereum khá nổi với *đồng tiền mã hoá*[4] của nó với tên gọi là *Ether* (ký hiệu: ETH). Dựa vào sự phân tán của công nghệ chuỗi khối, Ethereum khá an toàn, và cũng nhờ bảo mật cao nên giá trị của đồng tiền ETH tích luỹ ngày càng lớn trên thị trường tiện điện tử.

Bắt đầu ý tưởng từ năm 2013 bởi lập trình viên *Vitalik Buterin* và một số cộng sự, công việc phát triển Ethereum được vận hành và kêu gọi vốn từ cộng đồng vào năm sau đó. Mạng Ethereum chính thức "lên sóng" vào ngày 30 tháng 7 năm 2015.

2.2.2 Kiến trúc

Máy ảo Ethereum

Máy ảo Ethereum (EVM) là một môi trường chạy các hợp đồng thông minh Ethereum. Định nghĩa chính thức của EVM được quy định trong Ethereum Yellow Paper của Gavin Wood. Nó được hoàn toàn cô lập từ mạng, hệ thống tập tin và các quá trình khác của hệ thống máy chủ. Mỗi nút Ethereum trong mạng chạy một EVM và thực hiện các hướng dẫn giống nhau. Ethereum Virtual Machines đã được lập trình trong C++, Go, Haskell, Java, Python, Ruby, Rust và WebAssembly (hiện đang được phát triển).

Hợp đồng thông minh

Nền tảng Ethereum còn hỗ trợ mạng lưới các *ứng dụng phi tập trung*[5]. Mạng này vận hành xoay quanh các hợp đồng thông minh. Phần lớn các ứng dụng sử dụng

[3]Smart contract
[4]Cryptocurrency
[5]Decentralized applications (DApps)

hợp đồng thông minh để liên kết với công nghệ chuỗi khối. Có thể nói, hợp đồng thông minh chính là nhân tố trung tâm của nền tảng Ethereum.

Hợp đồng thông minh là *hợp đồng tự thực thi*[6] với các điều khoản được viết bởi các dòng lệnh hay các đoạn mã lập trình. Các đoạn mã này tồn tại khắp các nút trong mạng chuỗi khối, điều hành sự thực thi các giao dịch, và không thể thay đổi. Hợp đồng thông minh mang đến các giao dịch đáng tin cậy, sự đồng ý với các điều khoản trong hợp đồng tới các bên "ẩn danh" mà không cần qua một bên trung gian hay một cơ chế thực thi bên ngoài.

Trên Ethereum, các đoạn mã của hợp đồng thông minh được viết bằng ngôn ngữ lập trình *Solidity* hoặc *Vyper*. Solidity là ngôn ngữ lập trình hướng đối tượng bậc cao dựa theo *C++*, *JavaScript*, *Python*, và được thiết kế để tích hợp được với *Máy ảo Ethereum*[7] (EVM). Vyper là ngôn ngữ đang trong quá trình thử nghiệm.

Tài khoản

Mỗi tài khoản Ethereum được đại diện bởi 20 ký tự. Các thông số sau được lưu trong dữ liệu trạng thái (state) của Ethereum cho mỗi tài khoản:

- Số nonce, để đảm bảo mỗi giao dịch chỉ được xử lý một lần.
- Số dư tài khoản.
- Mã nguồn hợp đồng (nếu có).
- Phần lưu trữ của tài khoản (mặc định là trống).

Các giao dịch giữa các tài khoản được trả tiền bằng Ether. Có hai loại tài khoản: Tài khoản ngoại vi được quản lý bởi khóa riêng tư, và tài khoản hợp đồng được quản lý bởi mã hợp đồng. Tài khoản ngoại vi không chứa mã hợp đồng, có thể gửi thông điệp đi bằng cách tạo và ký kết một giao dịch, giống như tài khoản

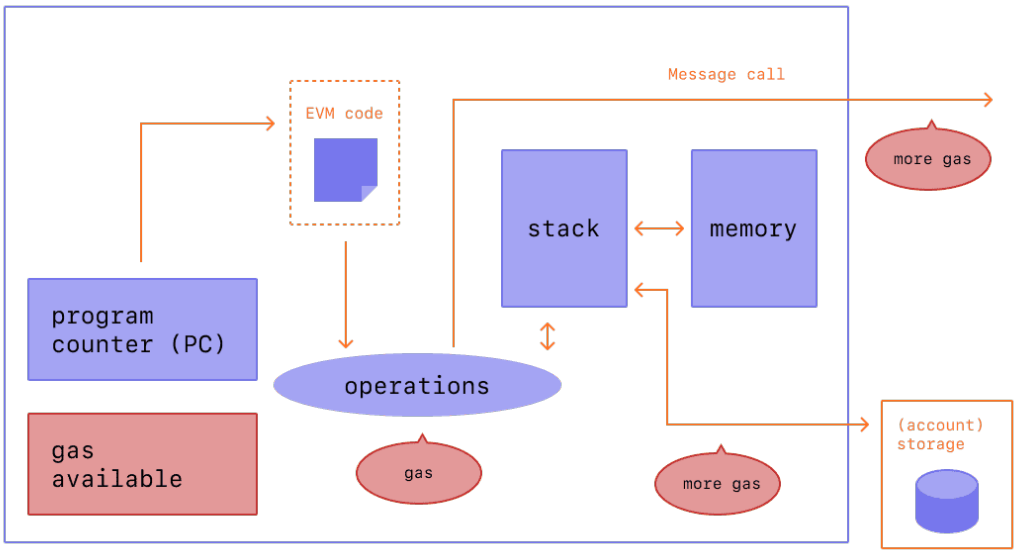
[6]Self-executed contract
[7]Ethereum Virtual Machine - EVM

Bitcoin. Về phía tài khoản hợp đồng, mỗi khi nó nhận được 1 thông điệp, mã hợp đồng sẽ chạy và cho phép đọc và ghi vào phần lưu trữ của nó, kèm theo việc gửi thông điệp đi và tạo ra hợp đồng khác lần lượt.

Lưu ý rằng "hợp đồng" trong Ethereum không phải là một cái gì đó phải "hoàn thành" hoặc "tuân thủ". Thay vào đó, nó giống như các "thực thể tự trị" sống bên trong môi trường Ethereum, luôn thực hiện một đoạn mã cụ thể khi được tác động bởi một thông điệp hoặc giao dịch, và có quyền kiểm soát trực số Ether và dữ liệu trong phần lưu trữ của nó.

Chi phí giao dịch

Gas là đơn vị thể hiện cho khối lượng tính toán để thực hiện một hành động nào đó trên mạng Ethereum. Do mỗi giao dịch trên mạng Ethereum đều cần tài nguyên tính toán để được thực thi, vì thế mà nó phát sinh ra *chi phí giao dịch*. Khi đó, *gas* thể hiện chi phí để thực hiện giao dịch thành công trên mạng.



Gas được trả bằng đồng *ether* (ETH). *Giá gas*[8] có đơn vị là *gwei*, mỗi *gwei* tương

[8]Gas price

ứng với một phần một tỷ của một *ether*: 1 *gwei* $= 10^{-9}$ *ether*. Vì vậy, thay vì nói chi phí giao dịch là $0,000000001$ *ether*, ta có thể nói giao dịch đó tiêu tốn 1 *gwei*. Ngoài ra, 1 *gwei* chính là một tỷ *wei*; *wei* (được đặt tên theo *Wei Dai* - nhà khoa học máy tính nổi tiếng đưa ra lý thuyết về thanh toán bằng tiền mã hoá) là đơn vị nhỏ nhất trên Ethereum.

Chương 3

Hệ thống tập tin phân tán IPFS

3.1 Tổng quan về IPFS

IPFS là viết tắt của từ Interplanetary File System, một hệ thống tập tin phân tán ngang hàng kết nối tất cả các thiết bị máy tính với nhau. Cụ thể hơn, nó sẽ phân phối dữ liệu được lưu trữ theo hình thức P2P, hay còn gọi là mạng ngang hàng (mạng đồng đẳng).

Trong đó, các hoạt động của IPFS chủ yếu dựa vào khả năng tính toán băng thông của tất cả các máy tham gia chứ không tập trung vào một phần nhỏ các máy chủ trung tâm như giao thức HTTP. Nói cách khác, IPFS là mạng lưới chuyển phát nội dung hoàn toàn phi tập trung cho phép quản lý và lưu trữ dữ liệu một cách linh hoạt. Mỗi máy tính tham gia trong mạng lưới đảm nhận nhiệm vụ download và upload dữ liệu mà không cần sự can thiệp của máy chủ trung tâm.

3.2 Kiến trúc của IPFS

3.2.1 Đối tượng IPFS

Một đối tượng IPFS (IPFS Object hay còn được ký hiệu là IPLD) là một cấu trúc dữ liệu với hai trường:

- `Data`: Dữ liệu nhị phân không có cấu trúc có kích thước nhỏ hơn 256kB.
- `Links`: Chứa các liên kết (`Link`) đến các đối tượng IPFS khác.

Cấu trúc của `Link` gồm ba trường:

- `Name`: Tên của liên kết.
- `Hash`: Hàm băm của đối tượng IPFS được liên kết tới.
- `Size`: Kích thước tích luỹ của đối tượng IPFS được liên kết tới, bao gồm cả các liên kết sau đó nữa.

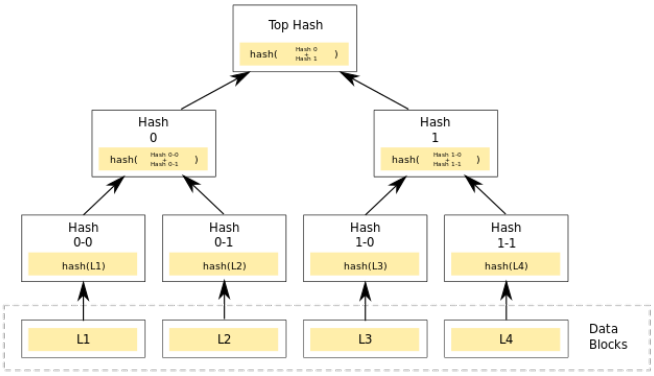
Trong đó, trường `Size` chủ yếu được sử dụng cho việc tối ưu hoá mạng P2P.

3.2.2 Merkle-DAG

DAG (Directed Acyclic Graph) là một dạng đồ thị có hướng, trong đó mỗi nút sẽ liên kết với các nút khác và không cho phép tạo thành chu trình có hướng. Một nút mà không là con của một nút nào khác trong đồ thị được gọi là nút gốc.

Merkle-DAG là một DAG trong đó mỗi nút có một định danh (identity hay id) là kết quả của việc mã hoá nội dung của nút đó. Điều này mang lại một số lưu ý:

- Các nút con phải được sinh trước thì các nút cha mới có id để liên kết tới.
- Mỗi nút trong Merkle-DAG là một nút gốc của một Merkle-DAG con nào đó.
- Các nút trong Merkle-DAG là không thể thay đổi. Bất kỳ thay đổi nào trong một nút sẽ làm thay đổi id của nút đó, và ảnh hưởng đến tất cả các nút khác.



Hình 3.1: Ví dụ về một Merkle-DAG.

3.2.3 Hệ thống tập tin

IPFS dễ dàng biểu diễn một hệ thống các tập tin và thư mục.

Các tập tin nhỏ

Một tập tin nhỏ được định nghĩa có kích thước nhỏ hơn 256kB, được biểu thị bằng một đối tượng IPFS với trường `Data` chứa nội dung của nó và trường `Links` là một danh sách rỗng.

Do tên tập tin không phải là một phần của đối tượng IPFS nên nếu có hai tập tin có cùng nội dung, chúng sẽ được biểu diễn bởi cùng một đối tượng IPFS.

Các tập tin lớn

Một tập tin lớn được định nghĩa có kích thước không dưới 256kB, được biểu thị bởi một Merkle-DAG của các đối tượng IPFS sao cho mỗi đối tượng có kích thước dữ liệu nhỏ hơn 256kB.

3.3 Cơ chế hoạt động

Đầu tiên mọi dữ liệu sẽ được mã hoá và được lưu dưới dạng mã băm (còn gọi là đối tượng IPFS). Ý tưởng chủ đạo là nếu trình duyệt của bạn muốn truy cập một trang nào đó trên IPFS thì chỉ cần đưa ra mã băm rồi mạng sẽ tìm máy có lưu trữ dữ liệu khớp với mã băm và sau đó tải dữ liệu, trang đó về từ máy tính đấy về cho bạn.

Cách thức hoạt động của IPFS sẽ tương tự như BitTorrent, đồng nghĩa với mỗi máy tính tham gia trong mạng lưới của nó sẽ đảm nhận cả việc tải xuống lẫn tải lên dữ liệu mà không cần có sự có mặt của một máy chủ trung tâm.

Tổng quan, cách hoạt động của IPFS sẽ có 2 phần chính:

- Xác định tệp có địa chỉ nội dung (giá trị băm của tệp đó).
- Tìm dữ liệu được lưu trữ và tải xuống: khi bạn có đoạn hash của tệp hay trang cần tải, mạng sẽ tìm và kết nối tới máy tốt nhất để tải dữ liệu xuống cho bạn.

3.4 Điểm độc đáo của IPFS

Nếu được triển khai đúng, IPFS mang lại tiềm năng lớn nhờ cải thiện được tốc độ truyền tải dữ liệu, tránh sự phụ thuộc vào các máy chủ và tiết kiệm chi phí.

3.4.1 Tránh sự phụ thuộc vào máy chủ

Trong các mô hình Máy khách - Máy chủ như HTTP, khi các máy chủ đang gặp phải sự cố thì chúng sẽ không thể hồi đáp thông tin cho người dùng. Đây cũng là vấn đề lớn nhất mà giao thức HTTP gặp phải khi nó phụ thuộc vào một máy chủ tập trung, điều mà nó không thể cải thiện cũng như khắc phục.

Với IPFS, nó hoàn toàn bỏ qua khái niệm máy chủ, mà chỉ quan tâm tới nội dung tìm kiếm. Điều này không chỉ giúp chúng ta rút ngắn con đường tới thông tin, mà lại không lo gặp phải các máy chủ kém chất lượng, kém tin cậy.

3.4.2 Mô hình phi tập trung

Với một mô hình tập trung, số lượng lớn dữ liệu được tập trung trong tay một số tên tuổi lớn trong lĩnh vực như Facebook, Amazon, Google,... Điều này vô tình khiến chúng trở thành tâm điểm cho các tin tặc tấn công. Trong lịch sử, không ít lần chúng ta chứng kiến những vụ rò rỉ thông tin liên quan đến những tên tuổi lớn.

Với mô hình phi tập trung của IPFS, các vấn đề này hoàn toàn được khắc phục và không còn chế độ quản lý phân cấp. Các dữ liệu được lưu trữ phân tán và không có một máy chủ tập trung để tấn công, càng nhiều người tham gia vào IPFS thì mạng sẽ càng bảo mật và khó có thể thao túng hơn.

3.4.3 Giảm bớt chi phí

Ưu điểm tiếp theo của mô hình IPFS đó là giảm bớt chi phí đối với cả người cung cấp nội dung và người dùng thông thường. IFPS sẽ cho phép đoạn video trên được tải hoàn toàn về mạng nội bộ IFPS dù bạn là ai và đang ở đâu. Do đó loại bỏ sự cần thiết của hàng loạt trạm kết nối và máy chủ Internet, giúp chi phí tổng thể giảm một cách rõ rệt.

Chương 4

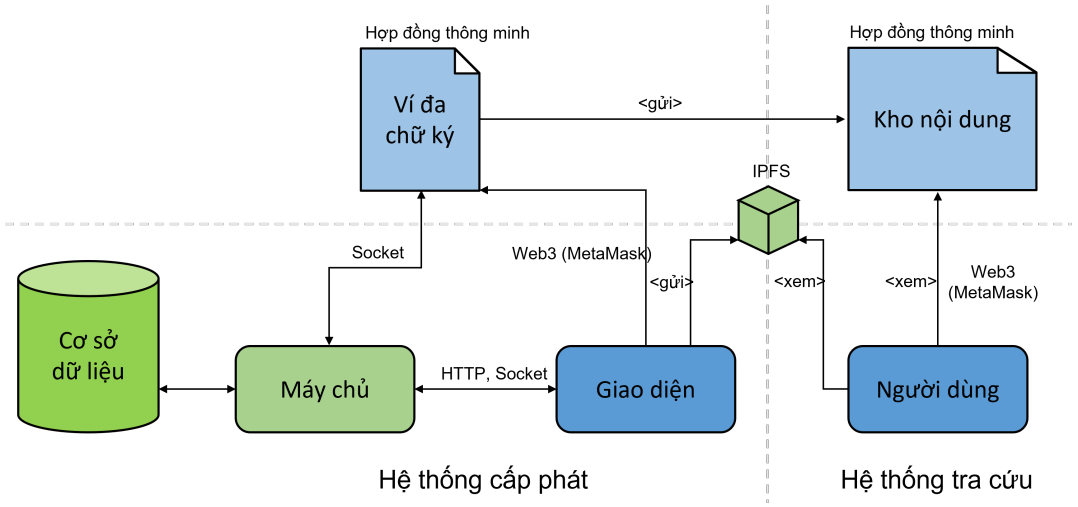
Cấp phát nội dung số trên mạng Ethereum kết hợp IPFS

4.1 Đề xuất giải pháp

Với sự phát triển của công nghệ chuỗi khối, người ta nhận ra tiềm năng của nó trong việc giải quyết các bài toán về lưu trữ dữ liệu. Không phải nói quá, chuỗi khối giải quyết quá tốt vấn đề tính toàn vẹn của dữ liệu dựa trên các thuật toán mã hoá. Đặc biệt, tương tác với chuỗi khối ngày càng trở nên đơn giản với *DApp* - các ứng dụng phi tập trung với giao diện thân thiện, dễ dùng, đồng thời tốc độ truy xuất thông tin chấp nhận được, việc ứng dụng chuỗi khối càng được ưa chuộng. Việc lưu trữ nội dung trên chuỗi khối đảm bảo được dữ liệu không thể bị chỉnh sửa, mọi người đều dễ dàng truy cập. Đó là ưu điểm khi lưu trữ nội dung số không phụ thuộc vào một số nhà cung cấp nhất định, nhưng cũng là thách thức cho các cơ sở cấp phát trong vấn đề xây dựng hệ thống tích hợp với mạng chuỗi khối một cách hợp lý và tốn không quá nhiều chi phí.

4.2 Xây dựng hệ thống

Với yêu cầu của bài toán, ta cần xây dựng hệ thống cấp phát cho các doanh nghiệp, và hệ thống tra cứu thông tin nội dung cho người có nhu cầu tra cứu.



Hình 4.1: Sơ đồ tổng quan các hệ thống và mạng chuỗi khối

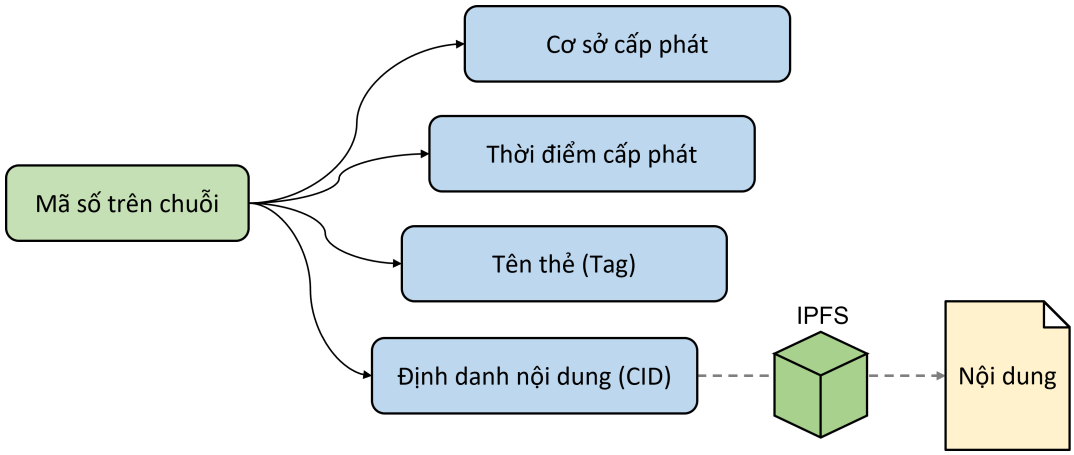
Những hệ thống này sẽ tương tác với các hợp đồng thông minh trên mạng chuỗi khối, bao gồm *Kho nội dung* và *Ví đa chữ ký*. Trong bài báo cáo này, em xin phép tập trung trình bày về phần thiết kế các hợp đồng thông minh được triển khai cùng hệ thống.

4.2.1 Kho nội dung

Kho nội dung là một hợp đồng thông minh nắm giữ thông tin các nội dung được lưu trữ trên mạng chuỗi khối. Thông tin được tổ chức theo cấu trúc dạng cây, mỗi mã số nội dung trên mạng chuỗi khối sẽ tương ứng với các thông tin liên quan đến nội dung đã được cấp phát.

Để lưu trữ nội dung trên mạng chuỗi khối, các cơ sở cấp phát cần sử dụng một địa chỉ ví để tương tác với hợp đồng thông minh được triển khai trên mạng đó.

Mỗi cơ sở phát hành nội dung sẽ có một địa chỉ ví xác định, là chữ ký đại diện cho cơ sở cấp phát nội dung đó.



Hình 4.2: Cấu trúc lưu trữ thông tin trong *kho nội dung*

Kho nội dung cung cấp một số chức năng liên quan đến lưu thông tin, tra cứu thông tin nội dung. Cụ thể, ba chức năng hiện có ở đây là:

- Lưu thông tin nội dung
- Xem thông tin nội dung
- Khoá nội dung

Đối với việc *lưu thông tin*, địa chỉ ví của cơ sở (hay người) gửi yêu cầu lưu thông tin nội dung sẽ được lấy làm *Cơ sở cấp phát*, và thông tin đi kèm sẽ được sử dụng làm *Tên thẻ* cho nội dung. Như vậy, sẽ không có tình huống một cơ sở cấp phát phát hành nội dung với địa chỉ của cơ sở khác, trường hợp nhầm lẫn do vô ý hoặc có chủ đích không thể xảy ra.

Để *xem thông tin*, người tra cứu chỉ cần cung cấp mã số nội dung đã được cấp phát. Những thông tin được hợp đồng thông minh trả về bao gồm cả định danh nội dung trên mạng IPFS, hệ thống tra cứu sẽ tự động lấy nội dung nội dung

(bao gồm cả phụ lục nội dung) dựa trên định danh này.

Đối với trường hợp nội dung đã được cấp phát nhưng thông tin bị sai hoặc không phù hợp, phía cơ sở cấp phát nội dung có thể lựa chọn *khoá nội dung* lại. Người tra cứu không thể *xem thông tin* đối với những nội dung đã bị khoá.

4.2.2 Ví đa chữ ký

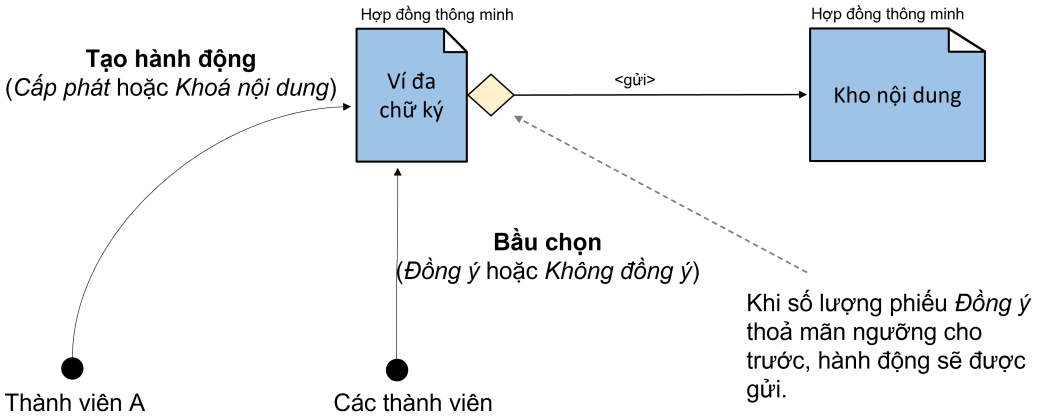
Ví đa chữ ký là một hợp đồng thông minh với mục đích tăng tính bảo mật cho quá trình tương tác thay đổi thông tin nội dung trên chuỗi khối.

Với việc "đẩy" thông tin nội dung lên mạng chuỗi khối một cách thông thường, mỗi cơ sở cấp phát sử dụng địa chỉ ví của một cá nhân đại diện để tương tác, hoặc lựa chọn một địa chỉ ví và sử dụng chung cho cá nhân trong cơ sở. Điều này đảm bảo mỗi cơ sở cấp phát chỉ có một địa chỉ duy nhất. Tuy nhiên, khi nhiều cá nhân cùng dùng một địa chỉ ví, khả năng mất cắp tài sản liên kết với địa chỉ này càng lớn, đặc biệt khi nó còn được sử dụng trong các giao dịch khác có giá trị về tài chính (như địa chỉ sở hữu tiền mã hoá với giá trị cao trên các *sàn giao dịch*[1], hay liên kết với các *DApp* khác). Do đó, một cơ chế giúp giảm thiểu khả năng nhiều người cùng sở hữu một địa chỉ ví và có thể sử dụng địa chỉ ví để xác thực thông tin cơ sở cấp phát nội dung là vô cùng cần thiết. *Ví đa chữ ký* ra đời để giải quyết vấn đề này.

Không giống với các *hệ thống xác thực đa chữ ký*[2] khi ít nhiều phụ thuộc vào các cơ chế xác thực phức tạp, *ví đa chữ ký* sử dụng các tính năng, lợi thế của hợp đồng thông minh và mạng chuỗi khối. Ở *ví đa chữ ký*, mỗi hành động cần thực thi (ở đây là việc cấp phát nội dung) yêu cầu một số lượng nhất định sự đồng ý từ cá nhân. Địa chỉ ví của các cá nhân này đã được thêm vào danh sách "thành viên" ngay từ khi hợp đồng thông minh này được triển khai, và họ được coi như các "cổ đông" của "doanh nghiệp" cấp phát nội dung khi có "tiếng nói" trong các "hoạt

[1] Exchange
[2] Multi-signature authentication system

động" ở đây. Mỗi cơ sở cấp phát nội dung sử dụng một *ví đa chữ ký* duy nhất, và địa chỉ của hợp đồng thông minh này đại diện cho địa chỉ ví của cả cơ sở đó. Các nội dung cần được đẩy lên *kho nội dung* sẽ được một cá nhân trong cơ sở gửi lên "ví" này. Các thành viên khác trong cơ sở có thể xem thông tin các nội dung được gửi lên, và đưa ra biểu quyết "đồng ý" hay "không đồng ý" trên hợp đồng thông minh. Khi số lượng sự đồng ý đạt ngưỡng nhất định (được thiết lập từ đầu), các nội dung đó được đẩy lên "kho", và thông tin được lưu trữ trên mạng chuỗi khối.



Hình 4.3: Cơ chế bầu chọn trong hệ thống

4.3 Kết quả triển khai và đánh giá

Triển khai

Giao diện *hệ thống quản lý* được xây dựng với thư viện *React* (được phát triển bởi đội ngũ *Facebook* với sự đóng góp của cộng đồng) sử dụng ngôn ngữ lập trình *TypeScript*. Đồng thời, người dùng cần đồng ý kết nối ví mã hoá *MetaMask* với hệ thống để sử dụng các tính năng tương tác với hợp đồng thông minh. Phía máy chủ, *Node.js* được lựa chọn, ta sử dụng *Express* để tạo các *API*[3]. Thông tin cần thiết được lấy từ cơ sở dữ liệu, và việc trao đổi giữa máy chủ và giao diện được thực hiện qua kết nối *HTTP* và *Web Socket*, các cập nhật từ một người sẽ được thông báo ngay lập tức tới các cá nhân khác trong cùng cơ sở cấp phát nội dung.

Hệ thống tra cứu sử dụng cấu trúc *trang tĩnh*[4], triển khai trên *GitHub Pages* với mã nguồn công khai, cung cấp cho người dùng một công cụ tra cứu thông tin nhanh chóng, đáng tin cậy.

[3]Application Programming Interface
[4]Static web

Kết luận

Tính tới thời điểm này, chuỗi khối không còn gì là mới mẻ, nhưng còn rất nhiều vấn đề ta cần giải quyết để khai thác được tiềm năng tối đa của nó. BitCoin với mang đến sự thịnh hành của tiền mã hoá, Ethereum lại phổ biến hợp đồng thông minh và *DApp*, đem tới nhiều cái nhìn tích cực hơn về công nghệ này. Và rồi nay mai đây thôi, những giải pháp cải tiến mạng chuỗi khối tiếp tục ra đời, đồng thời sự phổ cập kiến thức về nó cũng sẽ được đẩy mạnh. Em tin rằng, không lâu nữa, rất nhiều bài toán sẽ có thêm lời giải hợp lý khi đi cùng chuỗi khối.

Tài liệu tham khảo

[1] Diniel Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 2017

[2] Investopedia, *Blockchain Definition: What You Need To Know*

[3] Investopedia, *Consensus Mechanism (Cryptocurrency) Definition*

[4] 101 Blockchains, *Blockchain Technology Explained: A Decentralized Ecosystem*

[5] R. C. Hansdah, *A Multisignature Scheme for Implementing Safe Delivery Rule in Group Communication Systems*

[6] Ethereum, *Ethereum Developement Documentation*

[7] ConsenSys Academy, *MultiSig. Wallet Excercise*

[8] ChainSafe, *Web3.js Documentation*

[9] Internet, *các thông tin về Chuỗi khối*