

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA: CÔNG NGHỆ THÔNG TIN

Nguyễn Hải Tuyên - 21127474  
Trần Minh Đạt - 21127570

# HỆ THỐNG UY TÍN XÁC THỰC HỒ SƠ VÀ ĐÁNH GIÁ KỸ NĂNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN  
CHƯƠNG TRÌNH CHẤT LƯỢNG CAO

GIẢNG VIÊN HƯỚNG DẪN

PGS. TS. Nguyễn Đình Thúc

*Tp. Hồ Chí Minh, tháng 8 năm 2025*

# Lời cảm ơn

# Mục lục

<b>1</b>	<b>Giới thiệu chung</b>	<b>5</b>
1.1	Đặt vấn đề . . . . .	5
1.2	Mục tiêu . . . . .	6
1.3	Các nghiên cứu liên quan . . . . .	7
1.3.1	Nghiên cứu trong nước . . . . .	7
1.3.2	Nghiên cứu ngoài nước . . . . .	7
1.4	Cách tiếp cận . . . . .	9
1.4.1	Các thành phần công nghệ chính . . . . .	9
1.4.2	Các chủ thể tương tác . . . . .	10
1.4.3	Các tập dữ liệu . . . . .	10
1.4.4	Kịch bản tương tác giữa các chủ thể . . . . .	11
1.5	Đóng góp của đề tài . . . . .	13
<b>2</b>	<b>Các hệ thống và công nghệ liên quan</b>	<b>15</b>
2.1	ReGreT . . . . .	15
2.1.1	Chiều cá nhân . . . . .	15
2.1.2	Chiều xã hội . . . . .	16
2.1.3	Chiều bản thể học . . . . .	18
2.2	EigenTrust . . . . .	19
2.2.1	Giá trị tin cậy cục bộ . . . . .	20
2.2.2	Chuẩn hóa giá trị tin cậy cục bộ . . . . .	20
2.2.3	Tổng hợp các giá trị tin cậy cục bộ . . . . .	21
2.2.4	Giải thích về xác suất . . . . .	21
2.2.5	Phiên bản EigenTrust cơ bản . . . . .	22
2.2.6	EigenTrust phân tán . . . . .	23
2.3	Slashdot . . . . .	24

2.3.1	Karma . . . . .	24
2.3.2	Cơ chế kiểm duyệt do cộng đồng đảm nhiệm . . .	25
2.3.3	Phối hợp giữa cộng đồng và kỹ thuật . . . . .	25
<b>3</b>	<b>Mô hình uy tín hệ thống</b>	<b>27</b>
3.1	Hồ sơ uy tín . . . . .	27
3.2	Cơ chế cập nhật uy tín . . . . .	29
3.2.1	Đóng góp thử thách . . . . .	29
3.2.2	Kiểm duyệt chất lượng thử thách . . . . .	31
3.2.3	Tham gia thử thách . . . . .	32
3.2.4	Chấm điểm giải pháp . . . . .	32
3.3	Cấu thành hồ sơ uy tín . . . . .	33
3.4	Vai trò của hồ sơ uy tín . . . . .	34
3.4.1	Uy tín chuyên môn . . . . .	34
3.4.2	Uy tín toàn cục . . . . .	35
<b>4</b>	<b>Hệ thống SkillChain</b>	<b>37</b>
4.1	Yêu cầu chức năng chính . . . . .	37
4.1.1	Chức năng của người dùng . . . . .	37
4.1.2	Chức năng của nhà tuyển dụng . . . . .	41
4.1.3	Chức năng của hệ thống . . . . .	42
4.2	Yêu cầu phi chức năng . . . . .	43
4.3	Kiến trúc hệ thống . . . . .	44
4.3.1	Thành phần giao diện người dùng . . . . .	45
4.3.2	Tương tác với blockchain . . . . .	45
4.3.3	Hợp đồng thông minh và hạ tầng blockchain . . . .	46
4.3.4	Các dịch vụ bên ngoài . . . . .	46
4.4	Mô hình dữ liệu . . . . .	47
4.4.1	Quan hệ giữa User, Challenge, ModeratorReview, ReviewPool, Pot và ChallengeRevenue . . . . .	48
4.4.2	Quan hệ giữa User, Challenge, Solution, EvaluationPool và Evaluation . . . . .	50
4.4.3	Quan hệ giữa User, RecruiterProfile, Job, Application và Meeting . . . . .	52

4.4.4	Quan hệ giữa RecruiterProfile, User, PaymentRecord và Reputation . . . . .	54
4.5	Chi tiết các hợp đồng thông minh . . . . .	55
4.5.1	Danh sách các hợp đồng chính . . . . .	55
4.5.2	ReputationManager . . . . .	56
4.5.3	ChallengeManager . . . . .	57
4.5.4	ModeratorEscrow . . . . .	61
4.5.5	ChallengeCostManager . . . . .	64
4.5.6	SolutionManager . . . . .	67
4.5.7	RecruiterSubscription . . . . .	69
<b>5</b>	<b>Kết quả</b>	<b>70</b>
<b>6</b>	<b>Kết luận</b>	<b>71</b>

# Chương 1

## Giới thiệu chung

### 1.1 Đặt vấn đề

Trong quá trình tuyển dụng nguồn nhân lực hiện nay đối với các ngành nghề, đặc biệt là Công nghệ thông tin, việc xác thực trình độ học vấn, kỹ năng và kinh nghiệm của một cá nhân vẫn còn gặp nhiều khó khăn. Hiện nay, các nhà tuyển dụng và doanh nghiệp thường dựa vào chứng chỉ, bằng cấp hoặc thông tin do ứng viên cung cấp để đánh giá năng lực. Tuy nhiên, những thông tin này **có thể bị làm giả** hoặc không phản ánh chính xác thực lực của ứng viên.

Bên cạnh đó, nhiều cá nhân có kỹ năng thực tế nhưng lại không có cách nào để chứng minh năng lực của mình ngoài những hồ sơ truyền thống. Điều này làm hạn chế cơ hội phát triển và nâng cao giá trị của bản thân trong lĩnh vực. Vì vậy, cần có một giải pháp minh bạch, khách quan và đáng tin cậy để xác thực trình độ của mỗi cá nhân, đồng thời hỗ trợ doanh nghiệp đánh giá năng lực của ứng viên một cách thuận lợi và chính xác hơn.

Theo đó, ý tưởng về một **Hệ thống uy tín xác thực hồ sơ và đánh giá kỹ năng** được phát triển dựa trên công nghệ blockchain, cung cấp một nền tảng giúp cá nhân có thể chứng minh năng lực của mình một cách minh bạch, công khai và không thể thay đổi. Hệ thống cho phép người dùng tham gia các thử thách để kiểm tra và chứng minh kỹ năng, đồng thời áp dụng cơ chế đánh giá phi tập trung để đảm bảo tính khách quan.

Thay vì chỉ dựa vào văn bằng, chứng chỉ hay lời khai của ứng viên, hệ

thống này sẽ ghi nhận kết quả đánh giá được tích lũy lên blockchain và hệ thống lưu trữ phi tập trung, giúp tạo ra một hồ sơ uy tín đáng tin cậy, có thể sử dụng trên nhiều nền tảng khác nhau. Cộng đồng chuyên gia và nhà tuyển dụng có thể tham gia vào quá trình đánh giá để đảm bảo chất lượng, đồng thời giúp thúc đẩy sự phát triển của một hệ sinh thái xác thực năng lực trong lĩnh vực Công nghệ thông tin.

## 1.2 Mục tiêu

Đề tài nhằm xây dựng một hệ thống uy tín phi tập trung ứng dụng công nghệ blockchain để xây dựng và xác thực kỹ năng và hồ sơ cá nhân, bước đầu giới hạn trong lĩnh vực Công nghệ thông tin. Hệ thống sẽ cung cấp một môi trường và phương thức đánh giá minh bạch, khách quan và không thể thay đổi, giúp cá nhân chứng minh năng lực thực tế một cách đáng tin cậy, đồng thời hỗ trợ tổ chức và doanh nghiệp trong việc xác thực trình độ ứng viên.

Hệ thống hướng đến việc **tạo lập một môi trường đánh giá công bằng**, nơi mà năng lực của cá nhân được thể hiện thông qua kết quả thực tế, thay vì chỉ dựa vào chứng chỉ hoặc hồ sơ tự khai. Bên cạnh đó, việc ứng dụng blockchain giúp đảm bảo dữ liệu được bảo mật, minh bạch và có thể truy xuất dễ dàng, góp phần nâng cao độ tin cậy và hiệu quả trong quy trình tuyển dụng và đánh giá nhân sự.

Ngoài việc giải quyết bài toán xác thực hồ sơ, hệ thống còn tạo động lực để cá nhân **phát triển kỹ năng liên tục**, khi họ có thể tham gia các thử thách để cải thiện năng lực và nhận được sự công nhận từ cộng đồng. Với tiềm năng mở rộng, mô hình này có thể được áp dụng cho nhiều lĩnh vực khác, đóng góp vào xu hướng phát triển của các giải pháp phi tập trung trong tương lai.

## 1.3 Các nghiên cứu liên quan

### 1.3.1 Nghiên cứu trong nước

Bài nghiên cứu "Quản Lý Định Danh Phi Tập Trung"[1] là xuất phát điểm quan trọng của đề tài. Ý tưởng chính của bài nghiên cứu là việc xây dựng một thống quản lý định danh số nơi người dùng (chủ thể định danh) có toàn quyền kiểm soát đối với thông tin định danh của mình, thay vì phụ thuộc vào các nhà cung cấp dịch vụ (Service Provider). Bài nghiên cứu đã đề xuất việc sử dụng công nghệ blockchain để xây dựng hệ thống định danh, bao gồm hai chủ thể chính: người dùng (chủ thể định danh) và nhà cung cấp dịch vụ. Dữ liệu định danh của người dùng được lưu trữ cục bộ trên chính thiết bị của người dùng dưới dạng mã hóa và tham chiếu đến dữ liệu được lưu trên blockchain. Để quản lý dữ liệu mã hóa, cả chủ thể định danh và nhà cung cấp dịch vụ cần sử dụng giao thức tương tác DataTX. Chủ thể định danh có toàn quyền đối với dữ liệu của mình, có thể quản lý quyền truy cập thông qua giao thức tương tác AccessTx.

Bài nghiên cứu cũng đã định nghĩa về uy tín (reputation) như sau: "Uy tín được hình thành và biến đổi qua những thành công hay thất bại thực thể khi thực thi các nhiệm vụ cụ thể"[1, 2]. Có thể hiểu rằng, uy tín không phải là thực thể bất biến, mà luôn thay đổi theo thời gian và từng ngữ cảnh cụ thể. Không chỉ dừng lại ở một khái niệm lý thuyết đơn thuần, bài nghiên cứu đã đề xuất một hướng tiếp cận cụ thể để tính toán và số hóa khái niệm trừu tượng này bằng cách dựa vào hành vi (behavior) của các Node trong mạng, bao gồm người dùng (chủ thể định danh) và nhà cung cấp dịch vụ. Niềm tin của một thực thể (Node) được thiết lập là "giá trị kỳ vọng về hành vi tốt của Node trong tương lai"[1]. Kỳ vọng này chính là xác suất  $p$  trong phân phối Bernoulli, được tính bằng cách đếm số hành động của Node để tính xấp xỉ xác suất. [3, 1]

### 1.3.2 Nghiên cứu ngoài nước

Chúng tôi đã tham khảo từ các bài báo của dự án Rebooting the Web Of Trust (RWOT) [4, 5, 6]. Các công trình này đã đưa ra một khung khái niệm (conceptual framework) cơ bản cho một hệ thống uy tín, đặc biệt



trong việc đánh giá kỹ năng của một cá nhân khi tham gia vào mạng.

Bài báo [4] đã đề xuất một quy trình chung gồm các bước thao tác của các thực thể trong một hệ thống uy tín:

- Mỗi cá nhân tham gia vào mạng bằng một mã định danh phi tập trung (DID - Distributed Identifier).
- Chủ thể cần xây dựng uy tín sẽ tạo một tuyên bố (Assertion) và ký bằng khóa bí mật (Private Key) của mình.
- Để tăng tính thuyết phục của tuyên bố, chủ thể có thể tạo một bằng chứng (Evidence) và ký bằng khóa bí mật của mình, sau đó bổ sung tuyên bố ban đầu bằng cách tham chiếu đến bằng chứng vừa tạo.
- Các chủ thể khác tạo một bài đánh giá (Evaluation), ký bằng khóa bí mật của mình và tham chiếu đến tuyên bố của chủ thể cần xây dựng uy tín. Một bài đánh giá có thể ủng hộ hoặc thách thức tuyên bố gốc, có thể tham chiếu đến các bằng chứng (Evidence) để tăng tính thuyết phục.
- Cộng đồng sẽ tham gia bình chọn cho các bài đánh giá. Họ có thể lựa chọn đồng tình hoặc không đồng tình với bài đánh giá.

Bài báo [5] đã chỉ ra các yếu tố cốt lõi cần xem xét khi xây dựng một hệ thống uy tín phi tập trung, bao gồm bối cảnh (Context); sự tham gia của cộng đồng (Participation); sự đồng thuận của người dùng (User Consent); tính bảo mật (Confidentiality), khả năng tạo giá trị (Value Generation); hiệu suất hệ thống (Performance); tính bền vững của hệ thống (Sustainability); vòng đời của các tuyên bố (Claim Lifecycle); tính phục hồi sau các cuộc tấn công mạng (Resilience) và khía cạnh pháp lý (Legal).

Và bài báo [6] đã tập trung vào việc tìm kiếm giải pháp để số hóa khái niệm trừu tượng uy tín. Uy tín của một chủ thể phụ thuộc vào rất nhiều yếu tố và ngữ cảnh khác nhau, vì vậy việc biến những dữ liệu thô đầu vào thành dữ liệu đầu ra nhất quán, có thể xử lý được (actionable output) là một phần quan trọng trong một hệ thống uy tín. Cụ thể hơn, bài báo đã đưa ra một quy trình tuần tự, bao gồm việc xác định dữ liệu đầu ra (output), xác định dữ liệu thô đầu vào (raw input), xác định chất

lượng đầu vào và biên độ lỗi, chuẩn hóa dữ liệu để đưa về một dạng nhất quán, và cuối cùng là xử lý dữ liệu.

Nhìn chung, các nghiên cứu, bài báo mà chúng tôi đã tham khảo từ trong và ngoài nước đã định hình một ý tưởng sơ khai về một hệ thống uy tín phi tập trung. Các nghiên cứu, bài báo đã đề xuất những quy trình xử lý, cách thức tương tác giữa các chủ thể trong hệ thống, kiến trúc hệ thống, cũng như các khía cạnh cần xem xét trong một hệ thống uy tín. Qua đó, đã tạo tiền đề cho bài nghiên cứu này về việc xây dựng và triển khai một hệ thống uy tín trong thực tiễn.

## 1.4 Cách tiếp cận

Từ ý tưởng rút ra từ các nghiên cứu và bài báo đã tiến hành theo hướng đề tài này, chúng tôi có hướng tiếp cận tới hệ thống này như sau:

### 1.4.1 Các thành phần công nghệ chính

- Blockchain: sổ cái phi tập trung, có vai trò lưu trữ dữ liệu quan trọng trên chuỗi (on-chain) như thông tin người dùng, danh sách thử thách, danh sách giải pháp, chỉ số uy tín, quyền truy cập vào dữ liệu của người dùng,... Dữ liệu có thể chứa các tham chiếu (CID) của các dữ liệu lưu ngoài chuỗi (off-chain).
- Smart Contract: hợp đồng thông minh tương tác với blockchain để quản lý dữ liệu on-chain, đồng thời thực thi các tiến trình quan trọng như: tính toán chỉ số uy tín, cập nhật kết quả lưu trên chuỗi, thực hiện các biện pháp thưởng phạt theo quy ước.
- Kho lưu trữ dữ liệu phi tập trung (dStorage): dịch vụ lưu trữ phi tập trung dùng để lưu trữ dữ liệu người dùng, nội dung của các *thử thách*, các *giải pháp* của nó, và các dữ liệu lớn cần lưu trữ ngoài chuỗi.

### 1.4.2 Các chủ thể tương tác

- *Người đóng góp (Contributor)*: là những người dùng tạo các thử thách, có trách nhiệm đóng góp ngân hàng thử thách của hệ thống.
- *Người kiểm duyệt (Moderator)*: là những người có trách nhiệm kiểm duyệt các thử thách do người đóng góp tạo ra, thực hiện phân loại, đánh giá, và kiểm định chất lượng của thử thách. Để trở thành người kiểm duyệt, người dùng phải thỏa mãn các điều kiện cụ thể được quy định bởi hệ thống.
- *Người dùng (User)*: là những người có nhu cầu xây dựng uy tín, có thể tìm kiếm các thử thách, tham gia và đưa ra các giải pháp cho thử thách tương ứng.
- *Người đánh giá (Evaluator)*: là những người tham gia vào quá trình đánh giá giải pháp mà người dùng đưa ra cho một thử thách.
- *Phòng tuyển dụng của các công ty (Companies' recruitment department)*: là đội ngũ tuyển dụng của các công ty có nhu cầu tuyển dụng những ứng viên phù hợp.

### 1.4.3 Các tập dữ liệu

- *Ngân hàng thử thách*: kho tàng các thử thách được đóng góp bởi các Contributor, mỗi thử thách sẽ có tiêu đề, mô tả, loại thử thách, lượng token treo thưởng, v.v.
- *Bộ kết quả kiểm duyệt*: một thử thách sẽ nhận được nhiều điểm số từ nhiều Moderator khác nhau trước khi nó được công khai cho User tham gia.
- *Bộ danh sách giải pháp*: một thử thách sẽ có nhiều giải pháp khác nhau được cung cấp bởi nhiều User khác nhau.
- *Bộ kết quả đánh giá*: một giải pháp sẽ nhận được nhiều điểm số từ nhiều Evaluator khác nhau.

- Danh sách bài tuyển dụng: các bài tuyển dụng được đăng tải bởi các đội ngũ tuyển dụng, mỗi bài tuyển dụng sẽ có tiêu đề, mô tả công việc, yêu cầu về điểm uy tín, v.v.
- Bộ thông tin ứng tuyển: một bài tuyển dụng sẽ được nhiều User khác nhau ứng tuyển.
- Danh sách các cuộc họp trực tuyến: những buổi gặp mặt trực tuyến được lên lịch bởi đội ngũ tuyển dụng dành cho một User cho một bài tuyển dụng nhất định.
- Hồ sơ người dùng:
  - Thông tin cá nhân, chẳng hạn như địa chỉ ví, tên, email, ảnh đại diện, v.v.
  - Chỉ số uy tín của người dùng trên hệ thống
  - Lịch sử tham gia thử thách
- Thông tin giao dịch token: phản ánh việc đóng góp và tham gia thử thách, đánh giá giải pháp, đăng bài tuyển dụng và ứng tuyển, và sự biến động về chỉ số uy tín.

#### 1.4.4 Kịch bản tương tác giữa các chủ thể

- **Tạo thử thách**
  - Người đóng góp (Contributor) tạo và đăng tải các thử thách lên hệ thống. Nội dung của thử thách được lưu trữ và chia sẻ lên dStorage.
  - Mỗi thử thách phải thuộc về một loại thử thách nhất định (ví dụ: Khoa học máy tính, An ninh mạng, Phát triển phần mềm, v.v. )
  - Hệ thống lưu nội dung của thử thách ở dStorage. Smart Contract cập nhật danh sách các thử thách trên chuỗi chứa các CID tham chiếu đến nội dung của thử thách ở dStorage.

- Các thử thách vừa tạo ở trạng thái chưa kiểm duyệt, cần phải thông qua ý kiến của những người kiểm duyệt (Moderator).
  - Sau giai đoạn kiểm duyệt, nếu thử thách đạt mức độ và được hội đồng tán thành sẽ chuyển sang trạng thái đã kiểm duyệt. Những người cần xây dựng uy tín có thể tham gia vào thử thách.
- **Kiểm duyệt thử thách**
    - Những người kiểm duyệt (Moderator) đọc nội dung của thử thách, kiểm tra chất lượng và mức độ phù hợp của thử thách. Sau đó, những người kiểm duyệt tiến hành bỏ phiếu về các thông tin như: phân loại của thử thách, độ khó, mức độ phù hợp ...
    - Smart Contract tổng hợp đánh giá của những Moderator, dựa trên một công thức tính đã quy định sẵn để tổng hợp ra kết quả cuối cùng.
    - Smart Contract cập nhật thông tin liên quan của thử thách đã lưu trên chuỗi.
  - **Tham gia thử thách và đánh giá**
    - Người dùng (User) trả một khoản token nhất định cho Contributor để tham gia thử thách.
    - Hệ thống tự động tạo một không gian làm việc nơi User có thể tạo giải pháp của mình.
    - User thiết lập một phiên đánh giá với số lượng tối đa Người đánh giá (Evaluator) cố định có thể tham gia chấm điểm. Evaluator có thể đánh giá bằng cách tham gia vào phiên.
    - Sau khi hoàn thành đánh giá, mỗi Evaluator sẽ nộp số điểm.
    - Smart Contract thu thập tất cả điểm số từ Evaluator, áp dụng thuật toán tính điểm để xác định số điểm cuối cùng.
    - Những Evaluator có điểm số gần với kết quả chính xác nhất sẽ nhận thêm điểm uy tín cho bản thân. Evaluator có điểm số

lệch xa với kết quả nhất (hoặc biên độ lệch vượt quá giới hạn đã quy định sẵn) sẽ bị trừ đi điểm uy tín.

- Smart Contract sau đó cập nhật kết quả lên blockchain để đảm bảo tính minh bạch và không thể chỉnh sửa.
- Điểm số của giải pháp sẽ làm thay đổi chỉ số uy tín của User trong một loại thử thách cụ thể.

- **Đăng bài tuyển dụng và ứng tuyển**

- Đội ngũ tuyển dụng của các công ty tạo và đăng tải các bài tuyển dụng kèm theo yêu cầu về chỉ số uy tín của ứng viên.
- Ứng viên tiến hành ứng tuyển vào các bài tuyển dụng.
- Nhà tuyển dụng lúc này có thể xem được chỉ số uy tín của ứng viên và có thể truy cập lịch sử tham gia thử thách.
- Smart Contract cập nhật dữ liệu quyền truy cập lưu trên chuỗi (nếu có).
- Nhà tuyển dụng có thể lên lịch các cuộc họp trực tuyến để phỏng vấn ứng viên.
- Đội ngũ tuyển dụng xem xét và quyết định tuyển dụng đối với User. Nếu có, công ty cần trả một khoản token cho hệ thống.

## 1.5 Đóng góp của đề tài

Đề tài mang lại những đóng góp chính sau:

- **Đề xuất một mô hình hệ thống uy tín phi tập trung** ứng dụng công nghệ blockchain để xác thực kỹ năng và hồ sơ cá nhân một cách minh bạch, khách quan và không thể thay đổi. Mô hình này hướng đến việc xây dựng một chuẩn đánh giá năng lực mới trong lĩnh vực Công nghệ thông tin.
- **Thiết kế kiến trúc hệ thống đầy đủ** với các thành phần chính như Smart Contract, blockchain, dịch vụ lưu trữ phi tập trung (dStorage), cùng luồng tương tác rõ ràng giữa các chủ thể trong hệ thống

như người dùng, người đóng góp, người kiểm duyệt, người đánh giá và nhà tuyển dụng.

- **Xây dựng cơ chế đánh giá dựa trên cộng đồng (community-driven assessment)** kết hợp với hệ thống tính điểm uy tín nhằm đảm bảo tính công bằng, chống gian lận và khuyến khích sự tham gia tích cực từ nhiều bên liên quan.
- **Triển khai một hệ thống minh họa (prototype)** hoạt động được ở môi trường cục bộ, bao gồm giao diện người dùng, backend tích hợp Smart Contract, khả năng lưu trữ dữ liệu phi tập trung và cơ chế đánh giá thử thách.
- **Đóng góp vào hướng nghiên cứu về hệ thống đánh giá phi tập trung và quản lý danh tiếng**, mở ra khả năng mở rộng mô hình cho các lĩnh vực khác ngoài Công nghệ thông tin, phù hợp với xu hướng ứng dụng công nghệ blockchain trong quản lý dữ liệu và xác thực hồ sơ.

## Chương 2

# Các hệ thống và công nghệ liên quan

### 2.1 ReGreT

ReGreT [7] là một mô hình đánh giá uy tín trong xã hội nhiều tác nhân (multi-agent society), được xây dựng nhằm mô phỏng cách con người hình thành và sử dụng danh tiếng trong các tương tác xã hội. Mô hình này mở rộng so với các mô hình trước bằng cách đưa ra ba chiều chính (dimension) của uy tín: chiều cá nhân, chiều xã hội, và chiều bản thể học (ontological).

#### 2.1.1 Chiều cá nhân

Đây là mức độ cơ bản nhất của uy tín, dựa trên trải nghiệm trực tiếp giữa hai tác nhân. Khi hai tác nhân tương tác, mỗi bên ghi nhận *kết quả* (outcomes) từ tương tác đó, bao gồm những gì đã được thỏa thuận và những gì thực sự xảy ra. Dựa vào những kết quả này, mỗi tác nhân hình thành *ấn tượng cá nhân* (impressions), từ đó tính ra *uy tín chủ quan* (subjective reputation). Bên cạnh đó, cũng sẽ một *độ tin cậy* (reliability) đối với uy tín chủ quan này. Những ấn tượng gần thời điểm hiện tại hơn sẽ được ưu tiên cao hơn thông qua hàm trọng số thời gian.

Từ đó, ReGreT định nghĩa cách tính uy tín trực tiếp của một tác nhân



$a$  đối với tác nhân  $b$  (chiều cá nhân) như sau:

$$R_{a \rightarrow b}(\text{subject})$$

### 2.1.2 Chiều xã hội

ReGreT mô hình hóa hiện tượng phổ biến trong xã hội con người: một cá nhân mang theo uy tín của nhóm mà họ thuộc về. Nếu không có đủ thông tin trực tiếp, ta có thể dựa vào uy tín nhóm để ước lượng hành vi của một tác nhân. Tương tự như cách một cá nhân có thể bị ảnh hưởng bởi uy tín của nhóm mà mình thuộc về, bản thân cá nhân đó cũng sẽ dựa vào *trải nghiệm* (experiences) của những người trong chính nhóm của mình để bổ sung và củng cố hiểu biết cá nhân về một thực thể. Nói cách khác, những gì mà các thành viên trong nhóm từng trải qua với một thực thể cụ thể (hoặc với nhóm của thực thể đó) sẽ góp phần định hình và làm phong phú thêm nhận định của mỗi thành viên trong nhóm.

Do đó, để tính giá trị uy tín của mình đối với một tác nhân theo chiều xã hội, ReGreT mở rộng thêm ba nguồn thông tin mới. Bên cạnh tương tác trực tiếp với chính tác nhân, giờ đây phải xem xét thêm tương tác với các thành viên trong nhóm mà tác nhân đó thuộc về, thông tin mà nhóm của mình đối với tác nhân đó, và cuối cùng là thông tin mà nhóm của mình đối với nhóm của tác nhân đó.

#### 2.1.2.1 Trải nghiệm cá nhân

Giả sử ta tính giá trị uy tín của một tác nhân  $a$  thuộc về nhóm  $\mathcal{A}$  đối với tác nhân  $b$  thuộc về nhóm  $\mathcal{B}$ . Đầu tiên, ta đã biết uy tín trực tiếp giữa hai tác nhân như sau:

$$R_{a \rightarrow b}(\text{subject})$$

Tiếp theo, sự tương tác của  $a$  đối với các thành viên khác của nhóm  $\mathcal{B}$  được biểu diễn như sau:

$$R_{a \rightarrow \mathcal{B}}(\text{subject}) = \sum_{b_i \in \mathcal{B}} \omega^{ab_i} \cdot R_{a \rightarrow b_i}(\text{subject})$$

trong đó,  $\sum_{b_i \in \mathcal{B}} \omega^{ab_i} = 1$ . Vì đang trong trường hợp uy tín chủ quan,

chúng ta cần phương thức để thể thiện độ tin cậy của uy tín này:

$$RL_{a \rightarrow \mathcal{B}}(subject) = \sum_{b_i \in \mathcal{B}} \omega^{ab_i} \cdot RL_{a \rightarrow b_i}(subject)$$

### 2.1.2.2 Trải nghiệm nhóm

Một khi đã có được trải nghiệm của cá nhân  $a$ , ta sẽ xem xét đến trải nghiệm của nhóm  $\mathcal{A}$  đối với tác nhân  $b$  và cả nhóm  $\mathcal{B}$ .

Đầu tiên, ta tính uy tín chủ quan của nhóm  $\mathcal{A}$  đối với tác nhân  $b$  cùng với độ tin cậy của uy tín như sau:

$$R_{\mathcal{A} \rightarrow b}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i b} \cdot R_{a_i \rightarrow b}(subject)$$

$$RL_{\mathcal{A} \rightarrow b}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i b} \cdot RL_{a_i \rightarrow b}(subject)$$

trong đó,  $\sum_{a_i \in \mathcal{A}} \omega^{a_i b} = 1$ .

Tương tự, để biết được uy tín của nhóm  $\mathcal{A}$  đối với nhóm  $\mathcal{B}$ , ta tính như sau:

$$R_{\mathcal{A} \rightarrow \mathcal{B}}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i \mathcal{B}} \cdot R_{a_i \rightarrow \mathcal{B}}(subject)$$

$$RL_{\mathcal{A} \rightarrow \mathcal{B}}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i \mathcal{B}} \cdot RL_{a_i \rightarrow \mathcal{B}}(subject)$$

trong đó,  $\sum_{a_i \in \mathcal{A}} \omega^{a_i \mathcal{B}} = 1$ .

### 2.1.2.3 Tổng hợp tất cả thông tin lại với nhau

Cuối cùng, ReGreT định nghĩa cách tính uy tín của tác nhân  $a$  đối với tác nhân  $b$  theo chiều xã hội như sau:

$$\begin{aligned} SR_{a \rightarrow b}(subject) = & \xi_{ab} \cdot R_{a \rightarrow b}(subject) + \\ & \xi_{a\mathcal{B}} \cdot R_{a \rightarrow \mathcal{B}}(subject) + \\ & \xi_{\mathcal{A}b} \cdot R_{\mathcal{A} \rightarrow b}(subject) + \\ & \xi_{\mathcal{A}\mathcal{B}} \cdot R_{\mathcal{A} \rightarrow \mathcal{B}}(subject) \end{aligned}$$

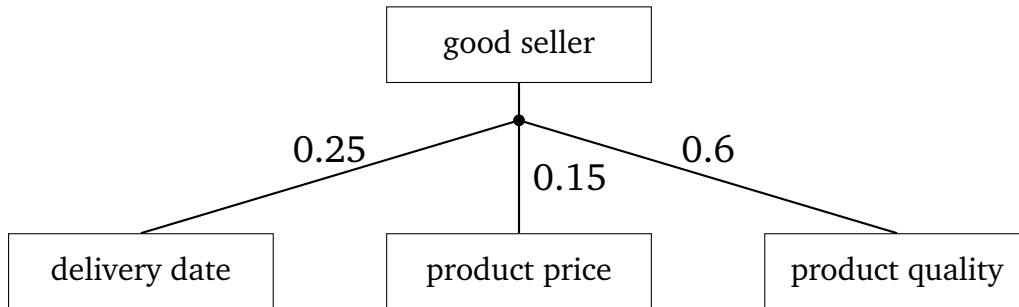
$$\begin{aligned}
SRL_{a \rightarrow b}(subject) = & \xi_{ab} \cdot RL_{a \rightarrow b}(subject) + \\
& \xi_{aB} \cdot RL_{a \rightarrow B}(subject) + \\
& \xi_{Ab} \cdot RL_{A \rightarrow b}(subject) + \\
& \xi_{AB} \cdot RL_{A \rightarrow B}(subject)
\end{aligned}$$

trong đó,  $\xi_{ab} + \xi_{aB} + \xi_{Ab} + \xi_{AB} = 1$ .

### 2.1.3 Chiều bản thể học

Trong hai chiều cá nhân và xã hội, mỗi lần đánh giá uy tín chỉ tập trung vào một khía cạnh đơn lẻ. Tuy nhiên, trong thực tế, các khía cạnh này thường liên quan đến nhau và cần được kết hợp cùng *trọng số* của từng khía cạnh lại để hình thành một khái niệm uy tín phức tạp hơn – đó chính là mục tiêu của **chiều bản thể học**.

Ở chiều này, mô hình ReGreT sử dụng cấu trúc đồ thị để mô tả mối quan hệ giữa các khía cạnh khác nhau của uy tín. Ví dụ, *một người bán hàng tốt* (good seller) có thể bao gồm các yếu tố: giao hàng nhanh, giá cả hợp lý, và chất lượng sản phẩm cao, ta có cấu trúc đồ thị bản thể như hình 2.1 để mô hình hóa mối quan hệ giữa các khía cạnh trong ví dụ này.



Hình 2.1: Cấu trúc đồ thị bản thể cho “một người bán hàng tốt”

Vì vậy, khi muốn tính một uy tín phức hợp theo chiều bản thể học, một tác nhân cần tính uy tín của từng khía cạnh liên quan. Mỗi khía cạnh này có thể lại là một nút trong một đồ thị con, nơi nó cũng được cấu thành từ các khía cạnh nhỏ hơn nữa.

Đối với những nút cuối cùng trong đồ thị (các khía cạnh cơ bản nhất của hành vi, gọi là *atomic aspect*), thì uy tín của chúng được tính dựa trên chiều cá nhân và chiều xã hội.

Sau khi có điểm uy tín cho từng nút con, điểm uy tín của một nút bất kỳ  $i$  trong đồ thị bản thể sẽ được tính bằng cách kết hợp các giá trị của các nút con của nó theo một công thức:

$$OR_{a \rightarrow b}(i) = \sum_{j \in \text{children}(i)} w_{ij} \cdot OR_{a \rightarrow b}(j)$$

$$ORL_{a \rightarrow b}(i) = \sum_{j \in \text{children}(i)} w_{ij} \cdot ORL_{a \rightarrow b}(j)$$

trong đó,  $OR_{a \rightarrow b}(j) = SR_{a \rightarrow b}(j)$  khi  $j$  là một khía cạnh cơ bản.

Đối với ví dụ ở hình 2.1, ta có thể tính giá trị uy tín của  $b$  (người bán hàng tốt) từ góc nhìn của  $a$  bằng công thức:

$$\begin{aligned} OR_{a \rightarrow b}(\text{good seller}) &= 0.25 \cdot SR_{a \rightarrow b}(\text{delivery date}) + \\ &0.15 \cdot SR_{a \rightarrow b}(\text{product price}) + \\ &0.6 \cdot SR_{a \rightarrow b}(\text{product quality}) \end{aligned}$$

## 2.2 EigenTrust

Mạng chia sẻ tập tin ngang hàng (P2P) đang ngày càng trở nên phổ biến nhờ ưu điểm không cần máy chủ trung tâm, có khả năng mở rộng tốt và cung cấp nhiều dữ liệu đa dạng. Tuy nhiên, đặc tính ẩn danh và mở của P2P cũng khiến hệ thống này dễ bị lạm dụng, không ai chịu trách nhiệm rõ ràng cho nội dung mà họ chia sẻ. Kết quả là tệp tin giả mạo, virus, hoặc nội dung bị chỉnh sửa có thể được phát tán rộng rãi, các peer độc hại có thể dễ dàng lừa người dùng tải về nội dung sai.

**EigenTrust** [8] được giới thiệu là một thuật toán giảm thiểu khả năng tải về các tập tin độc hại bằng cách gán mỗi peer một *giá trị tin cậy toàn cục* (global trust value) duy nhất, dựa trên lịch sử tải lên của peer đó. Mỗi peer trong mạng cùng tham gia tính toán điểm tin cậy của nhau theo cách phân tán và đối xứng. Khi tải tập tin về, các peer sẽ ưu tiên chọn nguồn tải là những peer có điểm tin cậy cao, giúp giảm đáng kể số lượng tập tin không xác thực.

### 2.2.1 Giá trị tin cậy cục bộ

Mỗi peer trong hệ thống mạng P2P cho phép theo dõi lịch sử giao dịch của nhau và đánh giá lẫn nhau sau mỗi giao dịch. Ví dụ, mỗi khi peer  $i$  tải tập tin từ peer  $j$ , nó có thể gán điểm tích cực ( $tr(i, j) = 1$ ) nếu hài lòng, hoặc điểm tiêu cực ( $tr(i, j) = -1$ ) nếu gặp vấn đề như tập tin bị hỏng, giả mạo hoặc quá trình tải thất bại. Từ các đánh giá này, một *giá trị tin cậy cục bộ* (local trust value) có thể được tính bằng cách cộng các điểm đánh giá mà peer  $i$  đã gán cho  $j$ , biểu diễn là  $s_{ij} = \sum tr_{ij}$ .

Ngoài ra, có thể biểu diễn giá trị này bằng cách lưu số lượng giao dịch hài lòng  $sat(i, j)$  và không hài lòng  $unsat(i, j)$ , rồi tính:

$$st_{ij} = sat(i, j) - unsat(i, j)$$

### 2.2.2 Chuẩn hóa giá trị tin cậy cục bộ

Trước tiên, cần chuẩn hóa các giá trị tin cậy cục bộ  $sat_{ij}$  để đảm bảo tính công bằng. Nếu không chuẩn hóa, một peer độc hại có thể đánh giá tích cực cho các peer độc hại khác và đánh giá tiêu cực cho các peer có uy tín cao. Để chuẩn hóa, ta chỉ xét các đánh giá tích cực, và tính xác suất chọn mỗi peer  $j$  khi được peer  $i$  tin tưởng như sau:

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

Cách làm này đảm bảo các giá trị  $c_{ij}$  nằm trong khoảng  $[0, 1]$  và có thể được hiểu như một xác suất trong mô hình ngẫu nhiên (chú ý: nếu  $\sum_j \max(s_{ij}, 0) = 0$  thì  $c_{ij}$  sẽ không xác định, vấn đề này sẽ được giải quyết sau đó). Tuy nhiên, có một số hạn chế: nếu  $c_{ij} = c_{ik}$ , ta chỉ biết peer  $j$  và  $k$  được  $i$  đánh giá ngang nhau, chứ không biết là tốt hay trung bình. Dù vậy, việc chuẩn hóa theo cách này giúp quá trình tính toán hiệu quả hơn và dễ dàng áp dụng mô hình xác suất.

### 2.2.3 Tổng hợp các giá trị tin cậy cục bộ

Sau khi chuẩn hóa, mục tiêu là tổng hợp các đánh giá cục bộ để thu được cái nhìn rộng hơn về độ tin cậy của từng peer. Điều này được thực hiện bằng cách áp dụng **transitive trust** (tạm dịch: niềm tin chuyển tiếp) – nghĩa là peer  $i$  sẽ hỏi ý kiến những người mà nó tin tưởng, và kết hợp đánh giá của họ với trọng số tương ứng.

Quá trình tổng hợp này dẫn đến một biểu thức dạng:

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

trong đó,  $t_{ik}$  thể hiện niềm tin của peer  $i$  đối với  $k$  sau khi hỏi bạn bè của nó ( $j$ ), mỗi người bạn của  $i$  lại đánh giá  $k$  một cách riêng ( $c_{jk}$ ). Ta có thể biểu diễn biểu thức trên thành ma trận: nếu ta gọi  $C$  là ma trận  $[c_{ij}]$  và  $\vec{t}_i$  là vector chứa các giá trị  $t_{ik}$ , khi đó ta có  $\vec{t}_i = C^T \vec{c}_i$ .

Ma trận trên có thể mở rộng khi peer  $i$  hỏi thêm những người bạn của bạn mà  $i$  đã hỏi ( $(C^T)^2, (C^T)^3, \dots, (C^T)^n$ ), và khi  $n$  đủ lớn, vector  $\vec{t}_i$  sẽ hội tụ về một vector *duy nhất cho mỗi peer  $i$*  – đó chính là vector riêng chính của ma trận tin cậy chuẩn hóa  $C$ . Nói cách khác,  $\vec{t}$  là *vector tin cậy toàn cục* (global trust vector) trong mô hình này, và mỗi phần tử của nó,  $\vec{t}_j$ , định lượng mức độ tin cậy của hệ thống nói chung đặt vào peer  $j$ .

### 2.2.4 Giải thích về xác suất

Cách hiểu xác suất cung cấp một góc nhìn trực quan cho quá trình lan truyền uy tín trong mạng. Tưởng tượng có một tác nhân ngẫu nhiên đi lang thang trong mạng P2P để tìm peer đáng tin. Tại mỗi bước, khi đang đứng ở peer  $i$ , nó sẽ chuyển đến peer  $j$  với xác suất là  $c_{ij}$  – tức là mức độ mà  $i$  tin tưởng  $j$ . Quá trình này tạo thành một **xích Markov** [9] với ma trận chuyển xác suất là ma trận tin cậy chuẩn hóa  $C$ . Sau nhiều bước di chuyển, tác nhân sẽ thường xuyên rơi vào những peer đáng tin hơn – và xác suất dừng lại ở mỗi peer chính là mức độ danh tiếng toàn cục.

Điều này giống như mô hình “Random Surfer” [10] trong PageRank: dù đi lung tung, người dùng sẽ thường ghé lại các trang web đáng tin và phổ biến hơn.

## 2.2.5 Phiên bản EigenTrust cơ bản

Trong phiên bản đơn giản nhất, ta giả sử có một máy chủ trung tâm lưu trữ tất cả giá trị  $c_{ij}$ , mục tiêu chính là tính được giá trị tin cậy toàn cục  $\vec{t} = C^T \vec{e}$ , với vector khởi đầu là  $\vec{e}$  thể hiện phân bố xác suất đồng đều của toàn bộ  $m$  peer, tức  $e_i = 1/m$ . Chi tiết tại thuật toán 1.

---

### Algorithm 1 Thuật toán EigenTrust đơn giản

---

```

 $\vec{t}^{(0)} = \vec{e}$ 
repeat
     $\vec{t}^{(k+1)} = C^T \vec{t}^{(k)}$ 
     $\delta = ||\vec{t}^{(k+1)} - \vec{t}^{(k)}||$ 
until  $\delta < \epsilon$ 

```

---

Tuy nhiên, có ba vấn đề thực tiễn mà thuật toán 1 không thể xử lý: nhóm peer có uy tín sẵn, những peer không hoạt động, và các peer độc hại hợp tác với nhau.

**Nhóm peer có uy tín sẵn.** Một hệ thống mạng P2P chỉ có một vài peer là có uy tín cao, thường là những người đầu tiên tham gia hệ thống, chẳng hạn như những nhà phát triển hay những người dùng truy cập sớm. Ta định nghĩa tập hợp  $P$  là các peer có uy tín sẵn (pre-trusted peers) và  $\vec{p}$  là vector khởi đầu thay cho  $\vec{e}$ , với:

$$p_i = \begin{cases} 1/|P|, & \text{nếu } i \in P \\ 0, & \text{ngược lại} \end{cases}$$

điều này giúp hội tụ nhanh hơn, cũng như loại bỏ được các peer độc hại.

**Những peer không hoạt động.** Nếu peer  $i$  chưa từng tương tác với ai, thì không thể tính được  $c_{ij}$  (do không thể chia hết cho 0). Khi đó, ta sẽ sử dụng  $p_j$  thay thế cho  $c_{ij}$ , tức là nếu peer  $i$  không tương tác với ai, không tin tưởng ai thì sẽ chọn peer có uy tín sẵn. Ta có thể điều chỉnh lại các tính  $c_{ij}$  như sau:

$$c_{ij} = \begin{cases} \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}, & \text{nếu } \sum_j \max(s_{ij}, 0) \neq 0 \\ p_j, & \text{ngược lại} \end{cases}$$

**Các peer độc hại hợp tác với nhau.** Có khả năng sẽ xuất hiện nhiều

nhóm peer độc hại mà chúng biết lẫn nhau, bọn chúng sẽ đánh giá cao lẫn nhau và đánh giá thấp những peer có uy tín. Để ngăn chặn điều này, ta trộn thêm  $\vec{p}$  vào mỗi vòng lặp:

$$\vec{t}^{(k+1)} = (1 - a)C^T\vec{t}^{(k)} + a\vec{p}$$

trong đó,  $a$  là một hằng số nhỏ hơn 1. Việc này giúp “kéo” giá trị  $\vec{c}_i$  của các peer về phía nhóm  $P$  nhằm phá vòng lặp tin tưởng nội bộ của nhóm độc hại. Cuối cùng, ta hoàn thiện thuật toán EigenTrust cơ bản như sau:

---

**Algorithm 2** Thuật toán EigenTrust cơ bản

---

```

 $\vec{t}^{(0)} = \vec{p}$ 
repeat
     $\vec{t}^{(k+1)} = C^T\vec{t}^{(k)}$ 
     $\vec{t}^{(k+1)} = (1 - a)\vec{t}^{(k+1)} + a\vec{p}$ 
     $\delta = ||\vec{t}^{(k+1)} - \vec{t}^{(k)}||$ 
until  $\delta < \epsilon$ 

```

---

### 2.2.6 EigenTrust phân tán

Ở đây, bài toán được mở rộng để không cần máy chủ trung tâm, mà mỗi peer tự tính toán uy tín của chính mình bằng thông tin từ những peer từng tương tác với nó. Yêu cầu đầu tiên của bài toán này là mỗi peer  $i$  phải lưu trữ vector tin cậy cục bộ  $\vec{c}_i$  và giá trị tin cậy toàn cục  $t_i$  của chính nó. Mỗi peer có thể tính giá trị tin cậy toàn cục của mình như sau:

$$t_i^{(k+1)} = (1 - a)(c_{1i}t_1^{(k)} + \dots + c_{ni}t_n^{(k)}) + ap_i$$

Từ đó, ta có được thuật toán EigenTrust phân tán cơ bản được thể hiện ở thuật toán 3. Vì mỗi peer chỉ tương tác với một số peer nhỏ (không phải toàn mạng), việc tính toán và truyền tin cũng nhẹ nhàng và hiệu quả. Trong trường hợp một mạng lưới có nhiều peer hoạt động mạnh, ta có thể duy trì những lợi ích trên bằng cách giới hạn số lượng giá trị tin cậy cục bộ  $c_{ij}$  mà mỗi peer có thể báo cáo.



---

**Algorithm 3** Thuật toán EigenTrust phân tán

---

**Definitions**

- $A_i$ : set of peers which have downloaded files from peer  $i$
- $B_i$ : set of peers from which peer  $i$  has downloaded files

**Algorithm**

Each peer  $i$  do {

Query all peers  $j \in A_i$  for  $t_j^0 = p_j$ ;

**repeat**

    Compute  $t_i^{(k+1)} = (1 - a)(c_{1i}t_1^{(k)} + c_{2i}t_2^{(k)} + \dots + c_{ni}t_n^{(k)}) + ap_i$

    Send  $c_{ij}t_i^{(k+1)}$  to all peers  $j \in B_i$

    Compute  $\delta = |t_i^{(k+1)} - t_i^{(k)}|$

    Wait for all peers of  $j \in A_i$  to return  $c_{ji}t_j^{(k+1)}$

**until**  $\delta < \epsilon$

}

---

## 2.3 Slashdot

**Slashdot** [11] là một trong những nền tảng trực tuyến lâu đời, nổi bật với mô hình vận hành như một diễn đàn công cộng thực sự – nơi người dùng không chỉ đọc tin tức mà còn có thể bình luận, thảo luận và tranh luận về các chủ đề liên quan đến công nghệ, phần mềm mã nguồn mở, khoa học, và xã hội kỹ thuật số. Điều làm nên sự đặc biệt của Slashdot không nằm ở bản thân nội dung, mà ở cách mà cộng đồng và hệ thống kỹ thuật cùng phối hợp để duy trì trật tự, chất lượng, và tính dân chủ của các cuộc đối thoại.

### 2.3.1 Karma

Trái tim của cơ chế tự điều chỉnh trên Slashdot là hệ thống **karma** [12] – một công cụ định lượng uy tín và mức độ đóng góp của người dùng trong cộng đồng. Karma không phải là một loại điểm số đơn thuần, mà là một dạng “vốn xã hội kỹ thuật số”. Nó phản ánh việc người dùng có đưa ra những bình luận hữu ích, đóng góp xây dựng hay không.

Người dùng sẽ tăng karma nếu bình luận của họ được đánh giá cao bởi những người kiểm duyệt tạm thời (moderators); hoặc nội dung họ cung cấp có chất lượng, mang lại thông tin mới, sâu sắc hoặc hữu ích.

Ngược lại karma giảm đi khi người dùng đăng bình luận gây tranh

cãi, xúc phạm, thiếu nội dung hoặc không liên quan; hoặc bị nhiều người đánh giá tiêu cực hoặc báo cáo vì hành vi không phù hợp.

Điểm số karma không hiển thị công khai cụ thể, nhưng nó tác động rõ rệt đến trải nghiệm và quyền hạn của người dùng:

- Mức karma cao: người dùng có thể được lựa chọn làm moderator, có ảnh hưởng xã hội cao, bình luận thường được ưu tiên hiển thị.
- Mức karma trung bình: người dùng có thể tham gia thảo luận, gửi bình luận, tương tác bình thường.
- Mức karma thấp: bình luận của người dùng có thể bị ẩn, quyền đăng bài hoặc đánh giá bị giới hạn.

### **2.3.2 Cơ chế kiểm duyệt do cộng đồng đảm nhiệm**

Người dùng có karma cao sẽ được chọn ngẫu nhiên làm moderator trong một khoảng thời gian ngắn. Vai trò của họ là:

- Đọc các bình luận mới trên bài viết
- Đánh giá chất lượng mỗi bình luận bằng một thang điểm (thường từ -1 đến +5)
- Gắn nhãn cho bình luận như: Insightful (sâu sắc), Offtopic (lạc đề), Funny (vui nhộn), Troll (gây rối), v.v.

Đặc biệt, người được trao quyền kiểm duyệt không phải quản trị viên cố định, mà luôn là người dùng bình thường được hệ thống chọn dựa trên mức karma – điều này tạo nên một cơ chế kiểm duyệt dân chủ và linh hoạt.

### **2.3.3 Phối hợp giữa cộng đồng và kỹ thuật**

Cơ chế karma và moderation của Slashdot là sự phối hợp giữa hai yếu tố kỹ thuật và xã hội:

- Về mặt kỹ thuật, hệ thống được thiết kế để tự động chọn moderator, xử lý hiển thị bình luận theo điểm đánh giá, và giới hạn quyền của người dùng có karma thấp.
- Về mặt xã hội, người dùng tự điều chỉnh hành vi của mình thông qua phần thưởng và hình phạt từ cộng đồng – dưới dạng điểm karma. Điều này không chỉ giảm thiểu gánh nặng cho quản trị hệ thống mà còn tăng cường sự tham gia và tinh thần trách nhiệm của người dùng trong việc gìn giữ môi trường thảo luận lành mạnh.

## Chương 3

# Mô hình uy tín hệ thống

Trong một hệ thống phi tập trung, nơi phần lớn hoạt động xoay quanh các thử thách, **uy tín người dùng đóng vai trò then chốt** trong việc đánh giá mức độ đóng góp, năng lực kiểm duyệt thử thách cũng như độ tin cậy khi chấm điểm cho các giải pháp. Trên cơ sở đó, hệ thống xây dựng một hồ sơ uy tín cho mỗi người dùng, phản ánh vai trò và mức độ đóng góp của họ trong cộng đồng, đồng thời hỗ trợ các bên tuyển dụng trong việc đánh giá năng lực ứng viên.

Hệ thống áp dụng một mô hình uy tín tích lũy, được xác lập dựa trên hành vi on-chain minh bạch, nhằm đảm bảo tính công bằng, khả năng truy xuất và chống thao túng từ người dùng.

### 3.1 Hồ sơ uy tín

**Hồ sơ uy tín** là nền tảng cho mọi hoạt động của người dùng trong hệ thống. Hồ sơ này bao gồm hai thành phần chính: *uy tín chuyên môn* (Domain Reputation) và *uy tín toàn cục* (Global Reputation).

- **Uy tín chuyên môn** là điểm uy tín phân theo từng lĩnh vực chuyên môn của thử thách. Mỗi người dùng có thể có mức uy tín khác nhau ở từng lĩnh vực tùy theo lịch sử hoạt động của họ.
- **Uy tín toàn cục** phản ánh tổng mức độ hành vi của người dùng xoay quanh các thử thách trong toàn bộ hệ thống.

Hệ thống hiện tại định nghĩa 14 lĩnh vực chuyên môn, tương ứng với các loại thử thách, bao gồm:

1. Khoa học máy tính cơ bản (Computer Science Fundamentals)
2. Phát triển phần mềm (Software Development)
3. Hệ thống và mạng máy tính (Systems and Networking)
4. An ninh mạng (Cybersecurity)
5. Khoa học và phân tích dữ liệu (Data Science and Analytics)
6. Quản trị cơ sở dữ liệu (Database Administration)
7. Đảm bảo và kiểm tra chất lượng (Quality Assurance and Testing)
8. Quản lý dự án (Project Management)
9. Trải nghiệm người dùng và thiết kế (User Experience and Design)
10. Phân tích nghiệp vụ (Business Analysis)
11. Trí tuệ nhân tạo (Artificial Intelligence)
12. Chuỗi khối và tiền mã hóa (Blockchain and Cryptocurrency)
13. Quản trị mạng (Network Administration)
14. Điện toán đám mây (Cloud Computing)

Mỗi khi người dùng thực hiện các hành động như đóng góp thử thách, giải thử thách, hoặc tham gia đánh giá giải pháp của thử thách, hệ thống sẽ cập nhật điểm uy tín tương ứng ở cả hai cấp độ: chuyên môn và toàn cục, đối với tất cả các bên tham gia hoạt động.

Ví dụ, nếu một người dùng giải thành công một thử thách thuộc lĩnh vực Cybersecurity, và một nhóm người đánh giá hoàn thành việc chấm điểm cho giải pháp đó, thì hệ thống sẽ căn cứ vào điểm số cuối cùng để điều chỉnh điểm uy tín của từng cá nhân. Cả người giải lẫn các người đánh giá liên quan sẽ được cập nhật điểm uy tín chuyên môn trong lĩnh vực Cybersecurity và điểm uy tín toàn cục.

Các thành phần này được lưu trữ on-chain thông qua hợp đồng thông minh, đảm bảo tính minh bạch, bất biến và truy xuất được toàn bộ lịch sử hoạt động liên quan đến uy tín của từng người dùng.

## 3.2 Cơ chế cập nhật uy tín

Trong hệ thống, có bốn hoạt động chính liên quan đến thử thách có ảnh hưởng trực tiếp đến điểm uy tín của người dùng: đóng góp thử thách, kiểm duyệt chất lượng thử thách, tham gia giải thử thách và chấm điểm giải pháp. Mỗi khi người dùng hoàn thành một trong các hoạt động trên, hệ thống sẽ tính toán một *điểm biến thiên hoạt động*  $\Delta R$ , từ đó điều chỉnh điểm uy tín chuyên môn trong lĩnh vực tương ứng và điểm uy tín toàn cục của người dùng.

Mỗi hoạt động ảnh hưởng đến uy tín của người dùng đều đi kèm với một *ngưỡng điểm hoạt động* ký hiệu là  $T$ , là một hằng số nằm trong khoảng từ 0 đến 100. Ngưỡng này được sử dụng để phân định hai trường hợp: hoạt động của người dùng đạt yêu cầu (được thưởng điểm) hoặc không đạt yêu cầu (bị trừ điểm uy tín).

Dựa trên mức độ vượt trội hoặc sai lệch so với ngưỡng, hệ thống tính toán điểm biến thiên uy tín  $\Delta R$  tương ứng. Để điều chỉnh mức thưởng và phạt một cách linh hoạt, mỗi hoạt động sẽ được gán hai hằng số:

- $L_{\text{reward}}$ : Mức thưởng tối đa – đại diện cho giá trị dương lớn nhất của  $\Delta R$  nếu người dùng thực hiện tốt hoạt động.
- $L_{\text{penalty}}$ : Mức phạt tối đa – đại diện cho giá trị âm lớn nhất (về độ lớn) của  $\Delta R$  nếu người dùng thực hiện kém so với yêu cầu.

### 3.2.1 Đóng góp thử thách

Mỗi khi người dùng đóng góp một thử thách mới, hệ thống sẽ tính *điểm đóng góp*  $R_{\text{contribution}}$ .

### 3.2.1.1 Độ khó của thử thách

Độ khó của một thử thách thuộc một trong ba mức dễ, trung bình và khó. Mỗi độ khó sẽ có trọng số  $W_{\text{difficulty}}$  tương ứng hỗ trợ tính *điểm biến thiên đóng góp*  $\Delta R_{\text{contribution}}$ , cụ thể:

$$W_{\text{difficulty}} = \begin{cases} 1 & , \text{nếu độ khó là dễ} \\ 1.2 & , \text{nếu độ khó là trung bình} \\ 1.4 & , \text{nếu độ khó là khó} \end{cases}$$

### 3.2.1.2 Tính điểm biến thiên

Sau khi được xác nhận và đánh giá bởi một nhóm người kiểm duyệt, mỗi thử thách sẽ có điểm chất lượng  $S_{\text{challenge}}$  và độ khó cuối cùng. Điểm chất lượng sẽ được so sánh với *ngưỡng điểm đóng góp*  $T_{\text{contribution}}$  để tính  $\Delta R_{\text{contribution}}$  theo các công thức khác nhau:

- Nếu  $0 \leq S_{\text{challenge}} \leq T_{\text{contribution}}$ :

$$\Delta R_{\text{contribution}} = L_{\text{penalty}} * \gamma * \left( \frac{S_{\text{challenge}}}{T_{\text{contribution}}} - 1 \right)$$

- Nếu  $T_{\text{contribution}} < S_{\text{challenge}} \leq 100$ :

$$\Delta R_{\text{contribution}} = L_{\text{reward}} * \gamma * W_{\text{difficulty}} * \frac{S_{\text{challenge}} - T_{\text{contribution}}}{100 - T_{\text{contribution}}}$$

trong đó,  $\gamma \in [0, 1]$  là hằng số tỉ lệ toàn cục thể hiện mức độ quan trọng của hoạt động đóng góp trong hệ thống.

### 3.2.1.3 Ghi nhận điểm đóng góp mới

Sau khi có điểm biến thiên, hệ thống sẽ ghi nhận điểm đóng góp mới:

$$R_{\text{contribution}} = R_{\text{contribution}} + \Delta R_{\text{contribution}}$$

### 3.2.2 Kiểm duyệt chất lượng thử thách

Người kiểm duyệt thử thách sẽ có *điểm kiểm duyệt*  $R_{\text{moderation}}$ .

#### 3.2.2.1 Độ lệch điểm chất lượng

Một người kiểm duyệt sẽ để lại *điểm kiểm duyệt thử thách*  $s_{\text{moderate}}$  cho một thử thách. Sau khi tổng hợp điểm cuối cùng từ nhóm người kiểm duyệt, ta sẽ có điểm chất lượng thử thách  $S_{\text{challenge}}$  và độ khó cuối cùng, từ đó xác định được độ lệch điểm chất lượng  $D_{\text{moderation}}$ :

$$D_{\text{moderation}} = |S_{\text{challenge}} - s_{\text{moderate}}|$$

#### 3.2.2.2 Tính điểm biến thiên

Độ lệch điểm chất lượng sẽ được so sánh với *ngưỡng điểm kiểm duyệt*  $T_{\text{moderation}}$  để tính  $\Delta R_{\text{moderation}}$  theo các công thức khác nhau:

- Nếu  $0 \leq D_{\text{moderation}} \leq T_{\text{moderation}}$ :

$$\Delta R_{\text{moderation}} = L_{\text{reward}} * \phi * W_{\text{difficulty}} * \left(1 - \frac{D_{\text{moderation}}}{T_{\text{moderation}}}\right)$$

- Nếu  $T_{\text{moderation}} < D_{\text{moderation}} \leq 100$ :

$$\Delta R_{\text{moderation}} = L_{\text{penalty}} * \phi * \frac{T_{\text{moderation}} - D_{\text{moderation}}}{100 - T_{\text{moderation}}}$$

trong đó,  $\phi \in [0, 1]$  là hằng số tỉ lệ toàn cục thể hiện mức độ quan trọng của hoạt động kiểm duyệt trong hệ thống.

#### 3.2.2.3 Ghi nhận điểm kiểm duyệt mới

Sau khi có điểm biến thiên, hệ thống sẽ ghi lại điểm kiểm duyệt mới:

$$R_{\text{moderation}} = R_{\text{moderation}} + \Delta R_{\text{moderation}}$$



### 3.2.3 Tham gia thử thách

Khi người dùng tham gia giải một thử thách và nộp giải pháp, hệ thống sẽ tính *điểm tham gia*  $R_{\text{participation}}$ .

#### 3.2.3.1 Tính điểm biến thiên

Khi người dùng tạo ra *giải pháp* dành cho một thử thách, một nhóm người đánh giá sẽ tham gia vào phiên đánh giá giải pháp này và tính được *điểm giải pháp*  $S_{\text{solution}}$  cuối cùng. Điểm giải pháp sẽ được so sánh với *ngưỡng điểm tham gia*  $T_{\text{participation}}$  để tính  $\Delta R_{\text{participation}}$  theo các công thức khác nhau:

- Nếu  $0 \leq S_{\text{solution}} \leq T_{\text{participation}}$ :

$$\Delta R_{\text{participation}} = L_{\text{penalty}} * \alpha * \left( \frac{S_{\text{solution}}}{T_{\text{participation}}} - 1 \right)$$

- Nếu  $T_{\text{participation}} < S_{\text{solution}} \leq 100$ :

$$\Delta R_{\text{participation}} = L_{\text{reward}} * \alpha * W_{\text{difficulty}} * \frac{S_{\text{solution}} - T_{\text{participation}}}{100 - T_{\text{participation}}}$$

trong đó,  $\alpha \in [0, 1]$  là hằng số tỉ lệ toàn cục thể hiện mức độ quan trọng của hoạt động tham gia thử thách.

#### 3.2.3.2 Ghi nhận điểm tham gia mới

Sau khi có điểm biến thiên, hệ thống sẽ ghi lại điểm tham gia mới:

$$R_{\text{participation}} = R_{\text{participation}} + \Delta R_{\text{participation}}$$

### 3.2.4 Chấm điểm giải pháp

Cuối cùng, người tham gia đánh giá sẽ có *điểm đánh giá*  $R_{\text{evaluation}}$ .

### 3.2.4.1 Độ lệch điểm đánh giá

Một người đánh giá sẽ để lại *điểm đánh giá giải pháp*  $s_{\text{evaluate}}$  cho một giải pháp. Sau khi tổng hợp điểm cuối cùng từ nhóm người đánh giá, ta sẽ có điểm giải pháp  $S_{\text{solution}}$  cuối cùng, từ đó xác định được độ lệch điểm đánh giá  $D_{\text{evaluation}}$ :

$$D_{\text{evaluation}} = |S_{\text{solution}} - s_{\text{evaluate}}|$$

### 3.2.4.2 Tính điểm biến thiên

Độ lệch điểm đánh giá sẽ được so sánh với *ngưỡng điểm đánh giá*  $T_{\text{evaluation}}$  để tính  $\Delta R_{\text{evaluation}}$  theo các công thức khác nhau:

- Nếu  $0 \leq D_{\text{evaluation}} \leq T_{\text{evaluation}}$ :

$$\Delta R_{\text{evaluation}} = L_{\text{reward}} * \beta * W_{\text{difficulty}} * \left(1 - \frac{D_{\text{evaluation}}}{T_{\text{evaluation}}}\right)$$

- Nếu  $T_{\text{evaluation}} < D_{\text{evaluation}} \leq 100$ :

$$\Delta R_{\text{evaluation}} = L_{\text{penalty}} * \beta * \frac{T_{\text{evaluation}} - D_{\text{evaluation}}}{100 - T_{\text{evaluation}}}$$

trong đó,  $\beta \in [0, 1]$  là hằng số tỉ lệ toàn cục thể hiện mức độ quan trọng của hoạt động đánh giá trong hệ thống.

### 3.2.4.3 Ghi nhận điểm đánh giá mới

Sau khi có điểm biến thiên, hệ thống sẽ ghi lại điểm đánh giá mới:

$$R_{\text{evaluation}} = R_{\text{evaluation}} + \Delta R_{\text{evaluation}}$$

## 3.3 Cấu thành hồ sơ uy tín

Uy tín chuyên môn  $R_{\text{domain}}$  của người dùng trong một lĩnh vực cụ thể được tính bằng tổng điểm uy tín thu được từ bốn loại hoạt động liên

quan đến thử thách, đã được trình bày ở Mục 3.2, bao gồm: đóng góp thử thách, kiểm duyệt chất lượng thử thách, tham gia giải thử thách và chấm điểm giải pháp. Cụ thể:

$$R_{\text{domain}} = R_{\text{contribution}} + R_{\text{moderation}} + R_{\text{participation}} + R_{\text{evaluation}}$$

Mỗi thành phần trong công thức trên chỉ bao gồm điểm số phát sinh từ những thử thách thuộc cùng một lĩnh vực chuyên môn (ví dụ: chỉ tính các hoạt động trong lĩnh vực Cybersecurity nếu đang xét).

**Uy tín toàn cục** được tính bằng tổng điểm uy tín chuyên môn trong tất cả các lĩnh vực thử thách mà hệ thống hỗ trợ:

$$R_{\text{global}} = R_{\text{domain}_1} + R_{\text{domain}_2} + \dots + R_{\text{domain}_n}$$

trong đó,  $n$  là tổng số lĩnh vực chuyên môn được định nghĩa trong hệ thống.

## 3.4 Vai trò của hồ sơ uy tín

### 3.4.1 Uy tín chuyên môn

Uy tín chuyên môn của người dùng trong một lĩnh vực cụ thể thể hiện mức độ hiểu biết, kinh nghiệm và mức độ đáng tin cậy của họ đối với loại thử thách tương ứng. Đây là một chỉ số quan trọng phản ánh mức độ đóng góp có giá trị của người dùng và đồng thời được sử dụng như một cơ sở để phân quyền thực hiện các chức năng nâng cao trong hệ thống.

Cụ thể, hệ thống sử dụng chỉ số uy tín chuyên môn để kiểm soát và điều hướng hành vi người dùng như sau:

- Người dùng chỉ được phép tạo thử thách trong một lĩnh vực nếu đạt mức uy tín chuyên môn tối thiểu trong lĩnh vực đó. Điều này đảm bảo rằng người đóng góp có đủ năng lực chuyên môn để thiết kế thử thách phù hợp, thực tiễn và có chất lượng.
- Quyền tham gia kiểm duyệt chất lượng thử thách cũng chỉ được trao cho những người dùng có uy tín chuyên môn cao. Cơ chế này

giúp ngăn chặn các đánh giá sai lệch, thiếu khách quan hoặc không đủ chuyên môn, từ đó giữ gìn chuẩn mực nội dung trên toàn hệ thống.

- Trong bối cảnh tuyển dụng, các nhà tuyển dụng có thể tham khảo hồ sơ uy tín chuyên môn để đánh giá mức độ phù hợp của ứng viên đối với các vị trí yêu cầu kỹ năng cụ thể. Uy tín cao trong một lĩnh vực chuyên môn giúp ứng viên tăng khả năng được chú ý và đánh giá cao.
- Tương tự, quyền tham gia đánh giá các giải pháp trong một thử thách cũng bị giới hạn cho những người dùng có đủ uy tín chuyên môn, nhằm đảm bảo công bằng, tránh thiên vị và hạn chế hành vi phá hoại trong quá trình đánh giá.

### 3.4.2 Uy tín toàn cục

Uy tín toàn cục phản ánh mức độ đáng tin cậy tổng thể của người dùng trong toàn bộ hệ thống, không phụ thuộc vào bất kỳ lĩnh vực chuyên môn cụ thể nào. Chỉ số này được sử dụng để phân tầng quyền hạn thực hiện các chức năng quan trọng, từ đó đảm bảo rằng các hành vi đóng góp, đánh giá, và kiểm duyệt đến từ những cá nhân có lịch sử hoạt động minh bạch và tích cực.

Quyền hạn của người dùng trong hệ thống được mở rộng theo mức độ uy tín toàn cục mà họ đạt được:

- Người dùng có uy tín thấp (thường là người mới hoặc chưa có nhiều đóng góp) chỉ được phép tham gia giải thử thách, như một bước khởi đầu để xây dựng hồ sơ uy tín cá nhân.
- Người dùng có mức uy tín trung bình sẽ được cấp quyền đóng góp thử thách mới vào hệ thống, với điều kiện đồng thời thỏa mãn yêu cầu về uy tín chuyên môn trong lĩnh vực tương ứng.
- Người dùng đạt mức uy tín khá sẽ được phép tham gia đánh giá các giải pháp của người dùng khác, đóng vai trò phản biện tích cực trong hệ sinh thái học tập cộng tác.

- Người dùng có uy tín cao sẽ được trao quyền tham gia kiểm duyệt chất lượng các thử thách do cộng đồng đóng góp, đóng vai trò như một lớp kiểm soát đảm bảo chất lượng và chuẩn hóa nội dung.

Việc áp dụng mô hình phân tầng dựa trên uy tín toàn cục không chỉ nâng cao hiệu quả quản lý người dùng mà còn tạo động lực tích cực thúc đẩy sự tham gia có trách nhiệm, hướng đến một cộng đồng học tập phi tập trung nhưng đáng tin cậy và bền vững.

## Chương 4

# Hệ thống SkillChain

**SkillChain** là một hệ thống ứng dụng mô hình uy tín phi tập trung dựa trên công nghệ chuỗi khối. Hệ thống cung cấp một môi trường làm việc xoay quanh các **thử thách**, cho phép người dùng tham gia để rèn luyện kỹ năng, trau dồi chuyên môn, củng cố kiến thức và đồng thời xây dựng hồ sơ uy tín một cách minh bạch, công khai và bất biến. Bên cạnh đó, hệ thống thiết lập một cơ chế đánh giá công bằng, khách quan và được dẫn dắt bởi cộng đồng chuyên gia, nhằm đảm bảo rằng năng lực của người dùng được **ghi nhận đúng thực chất**, qua đó thúc đẩy động lực phát triển kỹ năng một cách bền vững và liên tục.

Ngoài việc hỗ trợ người học thể hiện năng lực thông qua hành động cụ thể, SkillChain còn cung cấp **công cụ xác thực đáng tin cậy** cho các nhà tuyển dụng trong việc đánh giá trình độ ứng viên – vượt ra khỏi giới hạn của văn bằng, chứng chỉ hay khai báo chủ quan trong các mô hình tuyển dụng truyền thống.

## 4.1 Yêu cầu chức năng chính

### 4.1.1 Chức năng của người dùng

**Người dùng** là những cá nhân có nhu cầu xây dựng uy tín, chứng minh kỹ năng, trau dồi kiến thức và tìm kiếm cơ hội việc làm. Một người dùng phổ thông có thể thực hiện các chức năng sau:

- **Truy cập và quản lý thông tin cá nhân, hồ sơ uy tín:** Người dùng

có thể truy cập vào trang thông tin cá nhân, tại đây họ được phép cập nhật các thông tin liên quan đến hồ sơ người dùng. Đồng thời, người dùng có thể xem chi tiết hồ sơ uy tín của bản thân, bao gồm điểm uy tín toàn cục và điểm uy tín chuyên môn theo từng lĩnh vực, tương ứng với các loại thử thách đã được liệt kê tại mục 3.1.

- **Xem danh sách thử thách đang hiện hành trên hệ thống:** Người dùng có thể xem danh sách các thử thách đang có trên hệ thống, bao gồm cả thông tin chi tiết về thử thách và người đóng góp tương ứng.
- **Tham gia thực hiện thử thách:** Sau khi lựa chọn một thử thách phù hợp, người dùng sẽ phải trả một khoản token cho người đóng góp để được tham gia (khoản token sẽ được tính toán bởi hệ thống). Hệ thống sẽ cung cấp một không gian làm việc riêng nhằm giải quyết yêu cầu của thử thách và cho phép tùy chỉnh giải pháp. Tuy nhiên, người đóng góp sẽ không thể tham gia thực hiện chính thử thách do mình đóng góp.
- **Nộp và gửi giải pháp:** Sau khi hoàn thành, người dùng có thể nộp và gửi giải pháp lên hệ thống để được đánh giá. Lúc này, giải pháp sẽ chuyển sang trạng thái *chờ đánh giá* và không thể chỉnh sửa thêm. Bản thân người dùng không thể tham gia đánh giá giải pháp của chính mình.
- **Xem danh sách thử thách đã và đang thực hiện:** Người dùng có thể xem danh sách các thử thách mà mình đã hoặc đang tham gia cùng với các giải pháp tương ứng. Đối với những thử thách đã được đánh giá, hệ thống sẽ hiển thị danh sách người đánh giá và điểm số chi tiết của từng người.
- **Xem các bài tuyển dụng đang hiện hành:** Người dùng có thể xem danh sách các bài đăng tuyển dụng đang được đăng tải bởi các nhà tuyển dụng trên hệ thống, bao gồm đầy đủ thông tin chi tiết của từng bài đăng.
- **Tham gia ứng tuyển và phỏng vấn:** Người dùng có thể nộp đơn

ứng tuyển cho các vị trí tuyển dụng nếu đáp ứng yêu cầu đề ra trong bài đăng. Sau khi ứng tuyển, người dùng sẽ chờ phản hồi từ phía nhà tuyển dụng. Trong trường hợp được chọn vào danh sách rút gọn, người dùng có thể tham gia phỏng vấn trực tuyến khi được nhà tuyển dụng lên lịch.

- **Xem danh sách các bài tuyển dụng đã và đang ứng tuyển:** Người dùng có thể theo dõi các bài tuyển dụng mà mình đã hoặc đang ứng tuyển, kèm theo trạng thái hiện tại của từng hồ sơ ứng tuyển (ví dụ: đang chờ phản hồi, đã được mời phỏng vấn, bị từ chối, v.v.).

Hệ thống còn mở rộng thêm các chức năng chuyên biệt theo vai trò cụ thể mà một người dùng phổ thông có thể sở hữu.

#### 4.1.1.1 Người đóng góp

**Người đóng góp** là người dùng có vai trò tạo và thiết kế thử thách, đóng góp chúng vào ngân hàng thử thách chung của hệ thống. Một người đóng góp có thể thực hiện các chức năng sau:

- **Tạo thử thách mới ở trạng thái nháp:** Mỗi thử thách bao gồm các thông tin bắt buộc như tiêu đề, loại thử thách, mô tả chi tiết, và mức token treo thưởng. Khi tạo thử thách, hệ thống cung cấp một không gian làm việc cho người đóng góp, cho phép chỉnh sửa nội dung trong giai đoạn nháp.
- **Gửi thử thách để kiểm duyệt:** Sau khi hoàn thiện nội dung, người đóng góp có thể gửi thử thách lên hệ thống để kiểm duyệt. Từ thời điểm này, thử thách chuyển sang trạng thái *chờ kiểm duyệt* và không còn được chỉnh sửa. Người đóng góp không thể tham gia vào quá trình kiểm duyệt thử thách của chính mình.
- **Xem danh sách thử thách đã tạo:** Người đóng góp có thể xem danh sách các thử thách mà mình đã đóng góp. Đối với các thử thách đã được kiểm duyệt, có thể xem thông tin người kiểm duyệt, điểm đánh giá và mức thưởng token của từng người. Nếu thử thách đã được chấp nhận, người đóng góp cũng có thể xem danh sách người tham gia cùng với phần lợi nhuận thu được từ thử thách.



#### 4.1.1.2 Người kiểm duyệt

**Người kiểm duyệt** là người dùng có vai trò đánh giá và phê duyệt chất lượng các thử thách được gửi từ phía người đóng góp. Một người kiểm duyệt có thể thực hiện các chức năng sau:

- **Xem danh sách thử thách đang chờ kiểm duyệt:** Người kiểm duyệt có thể xem danh sách các thử thách đang ở trạng thái chờ phê duyệt, bao gồm thông tin chi tiết về thử thách và người đóng góp tương ứng.
- **Tham gia kiểm duyệt thử thách:** Sau khi lựa chọn một thử thách phù hợp, người kiểm duyệt có thể đăng ký tham gia vào quá trình đánh giá. Hệ thống sẽ cung cấp một không gian làm việc riêng để thực hiện quá trình kiểm duyệt.
- **Thực hiện kiểm duyệt thử thách:** Trong không gian làm việc, người kiểm duyệt có thể đánh giá nội dung thử thách và chấm điểm theo thang chuẩn của hệ thống. Sau khi gửi đánh giá, kết quả sẽ được ghi nhận vĩnh viễn và không thể chỉnh sửa.
- **Xem danh sách thử thách đã và đang kiểm duyệt:** Người kiểm duyệt có thể theo dõi danh sách các thử thách mà mình đã hoặc đang tham gia đánh giá. Đối với các thử thách đã hoàn tất kiểm duyệt, hệ thống sẽ hiển thị thông tin về các kiểm duyệt viên khác, điểm số đánh giá cũng như phần thưởng token được phân bổ cho từng người.

#### 4.1.1.3 Người đánh giá

**Người đánh giá** là những người dùng có vai trò đánh giá và chấm điểm các giải pháp được gửi từ người dùng nói chung. Một người đánh giá có thể thực hiện các chức năng sau:

- **Xem danh sách giải pháp đang chờ đánh giá:** Người đánh giá có thể xem danh sách các giải pháp đang ở trạng thái *chờ đánh giá*. Tại thời điểm này, họ chỉ có thể xem nội dung thử thách và thông tin người gửi, nhưng chưa thể truy cập nội dung giải pháp.

- **Tham gia đánh giá giải pháp:** Sau khi lựa chọn một giải pháp phù hợp, người đánh giá có thể đăng ký tham gia vào quá trình đánh giá. Hệ thống sẽ cung cấp một không gian làm việc riêng để tiến hành việc đánh giá. Tuy nhiên, người đóng góp của thử thách không thể tham gia đánh giá giải pháp.
- **Thực hiện đánh giá giải pháp:** Trong không gian làm việc, người đánh giá sẽ được quyền xem nội dung giải pháp và tiến hành chấm điểm. Sau khi gửi đánh giá, kết quả sẽ được ghi nhận vĩnh viễn và không thể chỉnh sửa.
- **Xem danh sách giải pháp đã và đang đánh giá:** Người đánh giá có thể theo dõi danh sách các giải pháp mà mình đã hoặc đang tham gia đánh giá. Đối với các thử thách đã hoàn tất quá trình đánh giá, hệ thống sẽ hiển thị thông tin về các người đánh giá khác và điểm số tương ứng.

#### 4.1.2 Chức năng của nhà tuyển dụng

**Nhà tuyển dụng** là chủ thể có nhu cầu tìm kiếm ứng viên, xác thực kỹ năng và đánh giá năng lực thông qua mô hình uy tín phi tập trung của hệ thống SkillChain. Một nhà tuyển dụng có thể thực hiện các chức năng sau:

- **Tạo bài tuyển dụng ở trạng thái nháp:** nhà tuyển dụng có thể khởi tạo một bài đăng tuyển dụng với các thông tin bắt buộc như: tiêu đề, mô tả công việc, yêu cầu công việc, khoảng thời gian làm việc, mức lương, thời điểm kết thúc ứng tuyển, và ngưỡng yêu cầu về uy tín chuyên môn. Ngoài ra, có thể bổ sung các thông tin tùy chọn như địa điểm làm việc hoặc yêu cầu về uy tín toàn cục. Bài đăng có thể được chỉnh sửa ở bất kỳ trạng thái nào.
- **Đăng bài tuyển dụng:** Sau khi hoàn thiện nội dung, nhà tuyển dụng có thể đăng bài lên hệ thống, qua đó cho phép người dùng đủ điều kiện ứng tuyển.

- **Xem thông tin ứng tuyển:** Đối với mỗi bài tuyển dụng, nhà tuyển dụng có thể xem danh sách các ứng viên đã ứng tuyển, bao gồm thông tin cá nhân và các chỉ số uy tín phù hợp với yêu cầu đề ra. Sau khi đánh giá, nhà tuyển dụng có thể thay đổi trạng thái ứng tuyển của từng ứng viên (ví dụ: từ “đang chờ” sang “đã đánh giá”).
- **Lập lịch phỏng vấn:** Đối với các ứng viên nằm trong danh sách rút gọn, nhà tuyển dụng có thể tạo lịch hẹn phỏng vấn dưới hình thức buổi họp trực tuyến để đánh giá sâu hơn trước khi đưa ra quyết định cuối cùng.

### 4.1.3 Chức năng của hệ thống

Bên cạnh các chức năng được phân tách theo vai trò người dùng, hệ thống SkillChain cũng đảm nhận một tập hợp các chức năng nền tảng, đóng vai trò duy trì toàn bộ quá trình vận hành, lưu trữ, đánh giá và phân phối thông tin trong nền tảng. Cụ thể, hệ thống chịu trách nhiệm thực hiện các tác vụ sau:

- **Lưu trữ và cập nhật dữ liệu:** Hệ thống có nhiệm vụ lưu trữ các dữ liệu do người dùng và nhà tuyển dụng tạo ra (bao gồm: thông tin cá nhân, thử thách, giải pháp, bài tuyển dụng, v.v.) trên cả nền tảng on-chain và off-chain. Bên cạnh đó, hệ thống cũng chịu trách nhiệm cập nhật các dữ liệu khi có thay đổi hợp lệ từ phía người dùng.
- **Truy xuất dữ liệu theo yêu cầu:** Hệ thống xử lý các yêu cầu truy vấn từ người dùng và nhà tuyển dụng, bao gồm việc lấy dữ liệu từ blockchain hoặc từ các nguồn lưu trữ off-chain, nhằm hiển thị thông tin một cách nhanh chóng và chính xác.
- **Quản lý phiên kiểm duyệt thử thách:** Hệ thống thu thập điểm kiểm duyệt từ tất cả các người kiểm duyệt tham gia một phiên kiểm duyệt, tính toán điểm chất lượng cuối cùng của thử thách, từ đó phân bổ phần thưởng token phù hợp và cập nhật điểm uy tín của từng người kiểm duyệt và người đóng góp.

- **Tính toán khoản token tham gia thử thách:** Hệ thống xác định mức phí token mà người tham gia cần chi trả để tham gia một thử thách, dựa trên thông tin cấu hình thử thách.
- **Quản lý phiên đánh giá giải pháp:** Hệ thống tổng hợp các điểm đánh giá từ các người đánh giá, tính toán điểm trung bình cuối cùng của giải pháp, và từ đó cập nhật điểm uy tín của cả người thực hiện giải pháp và người đánh giá.
- **Thông kê quá trình tuyển dụng:** Hệ thống tổng hợp dữ liệu từ các bài đăng, hồ sơ ứng tuyển, lịch phỏng vấn và kết quả tuyển dụng để cung cấp báo cáo thông kê cho nhà tuyển dụng, giúp đánh giá hiệu quả và chất lượng quy trình tuyển dụng của họ.

## 4.2 Yêu cầu phi chức năng

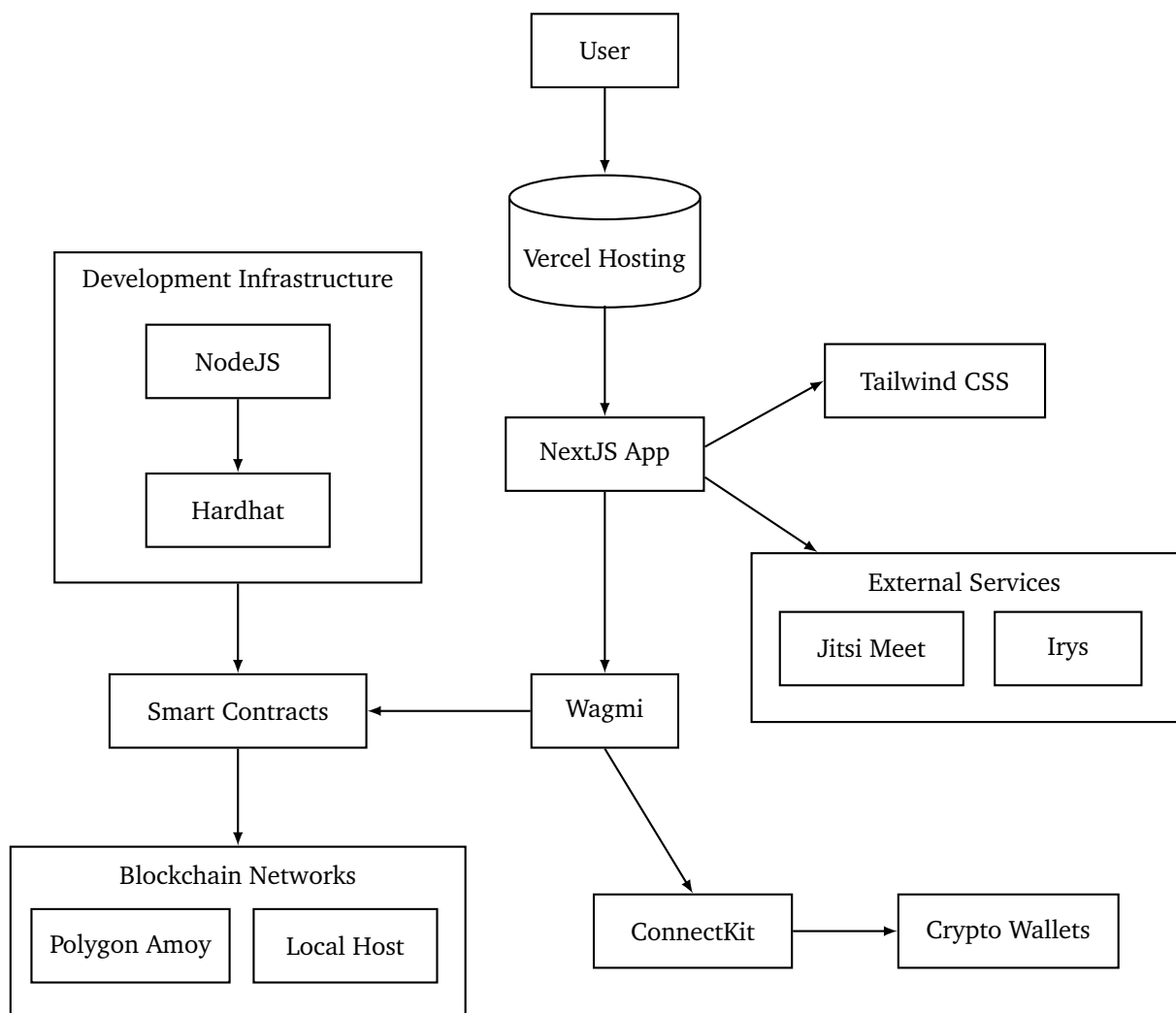
Bên cạnh các chức năng chính, hệ thống SkillChain cần đáp ứng một số yêu cầu phi chức năng để đảm bảo hiệu suất, khả năng sử dụng, bảo mật và độ tin cậy trong quá trình vận hành. Các yêu cầu này được liệt kê như sau:

- **Tính minh bạch và bất biến:** Mọi hành vi quan trọng của người dùng như đóng góp thử thách, tham gia giải thử thách, đánh giá giải pháp hay kiểm duyệt chất lượng thử thách đều phải được ghi nhận on-chain nhằm đảm bảo tính minh bạch, công khai và không thể chỉnh sửa sau khi ghi nhận.
- **Tính bảo mật:** Thông tin tài khoản người dùng và dữ liệu nhạy cảm (ví dụ: dữ liệu giải pháp lưu trên nền tảng off-chain) cần được bảo vệ khỏi truy cập trái phép. Hệ thống phải xác thực danh tính người dùng thông qua ví điện tử và chỉ cấp quyền tương ứng nếu có yêu cầu truy cập.
- **Tính khả dụng:** Hệ thống phải đảm bảo hoạt động liên tục, sẵn sàng phục vụ người dùng 24/7. Trong trường hợp một số dịch vụ off-chain bị gián đoạn, hệ thống vẫn cần đảm bảo dữ liệu on-chain có thể được truy xuất độc lập.

- **Khả năng mở rộng:** Hệ thống cần thiết kế để có thể mở rộng số lượng và số loại thử thách, người dùng và bài tuyển dụng trong tương lai mà không ảnh hưởng đến hiệu năng hoặc trải nghiệm người dùng.
- **Khả năng tương thích:** Ứng dụng cần tương thích với các trình duyệt phổ biến (Chrome, Firefox, Safari), ví điện tử (MetaMask, Coinbase) và nhiều loại thiết bị điện tử (máy tính, máy tính bảng, điện thoại thông minh).
- **Khả năng sử dụng:** Giao diện người dùng cần trực quan, rõ ràng và dễ sử dụng đối với người dùng mới, đồng thời hỗ trợ đa vai trò với trải nghiệm nhất quán.
- **Hiệu suất hệ thống:** Thời gian phản hồi cho các thao tác phổ biến như truy xuất danh sách thử thách, nộp giải pháp, hay đánh giá thử thách không được vượt quá 3 giây trong điều kiện bình thường. Hệ thống cần tối ưu hóa việc đồng bộ dữ liệu giữa blockchain và phần giao diện.
- **Khả năng kiểm thử và bảo trì:** Hệ thống cần được xây dựng theo hướng mô-đun, dễ dàng kiểm thử và nâng cấp. Các thành phần frontend, backend, smart contract và dữ liệu lưu trữ cần được tách biệt rõ ràng.

## 4.3 Kiến trúc hệ thống

Hệ thống SkillChain được xây dựng theo mô hình ứng dụng phi tập trung, kết hợp giữa các công nghệ frontend hiện đại, hợp đồng thông minh trên blockchain và các dịch vụ tương tác ngoài chuỗi. Kiến trúc tổng thể bao gồm bốn thành phần chính: giao diện người dùng, tương tác ví và blockchain, hệ thống hợp đồng thông minh và các dịch vụ bên ngoài. Hình 4.1 minh họa kiến trúc tổng quát của hệ thống.



Hình 4.1: Sơ đồ kiến trúc hệ thống

### 4.3.1 Thành phần giao diện người dùng

Giao diện người dùng được phát triển bằng framework **NextJS** và triển khai trên nền tảng **Vercel**, hỗ trợ hiệu năng cao và dễ dàng mở rộng. Giao diện sử dụng **Tailwind CSS** để xây dựng UI hiện đại, tối ưu cho trải nghiệm người dùng. Toàn bộ tương tác của người dùng với hệ thống (tham gia thử thách, gửi giải pháp, truy xuất hồ sơ uy tín...) đều diễn ra thông qua lớp này.

### 4.3.2 Tương tác với blockchain

Để frontend tương tác với blockchain, hệ thống sử dụng thư viện **Wagmi** kết hợp với **ConnectKit** giúp tích hợp ví điện tử như MetaMask hoặc

CoinBase. Các chức năng như xác thực người dùng qua ví, gửi giao dịch on-chain, theo dõi sự kiện blockchain đều được xử lý tại đây.

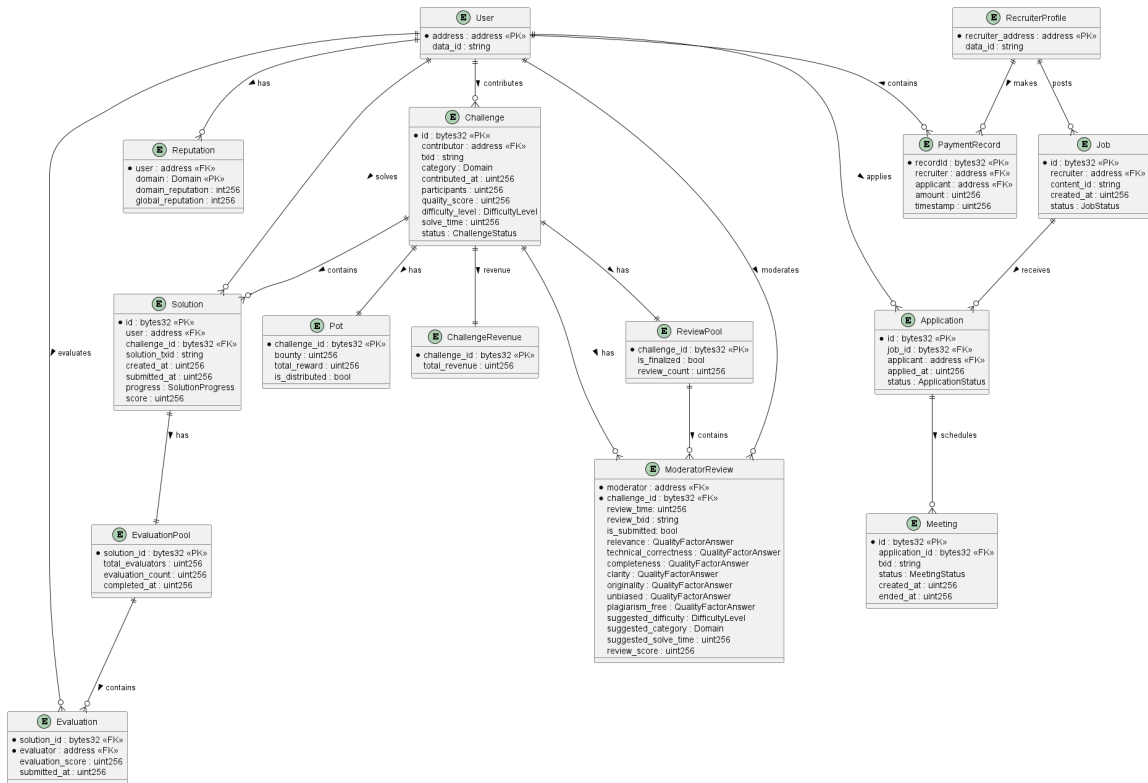
### 4.3.3 Hợp đồng thông minh và hạ tầng blockchain

Các hợp đồng thông minh được phát triển bằng **Solidity**, biên dịch và triển khai thông qua **Hardhat**. Trong quá trình phát triển và thử nghiệm, hệ thống hỗ trợ triển khai trên mạng cục bộ và mạng thử nghiệm **Polygon Amoy**. Các chức năng chính của smart contract bao gồm: quản lý thử thách, giải pháp và thông tin tuyển dụng, quản lý thông tin cá nhân và hồ sơ uy tín của người dùng và nhà tuyển dụng, kiểm soát truy cập.

### 4.3.4 Các dịch vụ bên ngoài

Một số loại dữ liệu không phù hợp để lưu trực tiếp trên chuỗi như nội dung thử thách hoặc giải pháp sẽ được lưu trữ trên **Irys**, với CID được lưu lại trên blockchain để dễ dàng tham chiếu. Ngoài ra, hệ thống còn tích hợp **Jitsi Meet** để phục vụ việc tổ chức các buổi phỏng vấn trực tuyến giữa nhà tuyển dụng và ứng viên, đảm bảo tính riêng tư và tiện lợi.

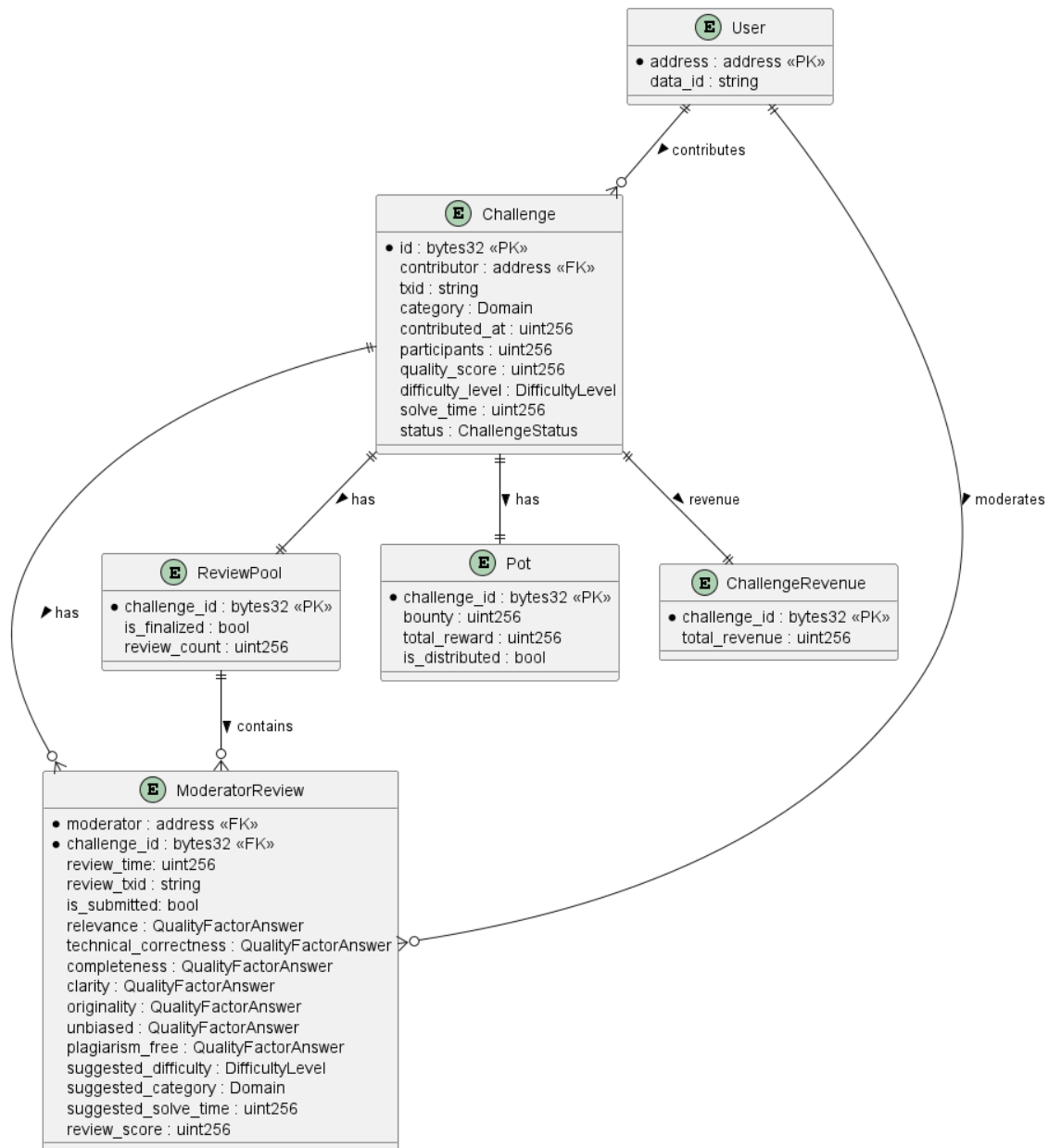
## 4.4 Mô hình dữ liệu



Hình 4.2: Sơ đồ mô hình dữ liệu (ERD) của hệ thống SkillChain



#### 4.4.1 Quan hệ giữa User, Challenge, ModeratorReview, ReviewPool, Pot và ChallengeRevenue



Hình 4.3: Mô hình dữ liệu (ERD) giữa các thực thể User, Challenge, ModeratorReview, ReviewPool, Pot và ChallengeRevenue

##### Mô tả các thực thể

- **User:** Đại diện cho tài khoản người dùng trên hệ thống, với address làm khóa chính và data\_id là biến tham chiếu dữ liệu off-chain.
- **Challenge:** Mỗi thử thách được xác định bởi id theo kiểu dữ liệu bytes32, bao gồm các trường thông tin: tham chiếu đến dữ liệu off-

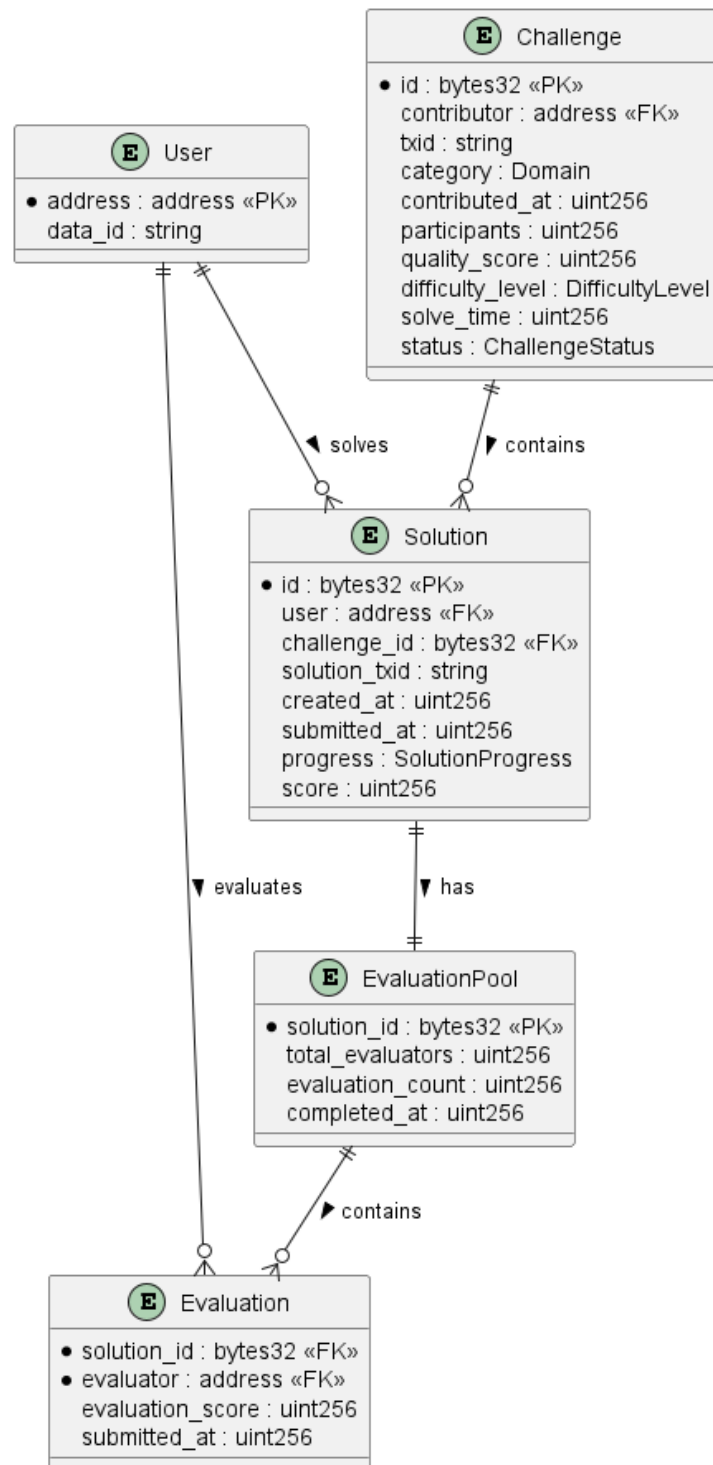
chain, lĩnh vực, thời điểm đóng góp, số lượng người đã tham gia, điểm chất lượng, độ khó thời gian giải ước tính và trạng thái. Khóa ngoại contributor liên kết đến User.address.

- **ModeratorReview:** Lưu trữ lượt đánh giá của moderator với khóa chính ghép (moderator, challenge\_id), bao gồm thời gian, tham chiếu đến dữ liệu off-chain, các câu trả lời về yếu tố chất lượng, đề xuất siêu dữ liệu (metadata) mới và điểm chất lượng đề xuất.
- **ReviewPool:** Quản lý các bài đánh giá cho mỗi thử thách. Sử dụng challenge\_id làm khóa chính và cờ is\_finalized để đánh dấu khi đối tượng này (pool) đã trải qua quá trình tổng hợp các bài đánh giá hay chưa.
- **Pot:** Quản lý quỹ thưởng của thử thách, bao gồm số token người đóng góp chi, tổng quỹ thưởng và cờ đánh dấu quỹ đã phân phát đến những người kiểm duyệt hay chưa. Khóa chính là challenge\_id.
- **ChallengeRevenue:** Theo dõi tổng doanh thu của mỗi thử thách, với challenge\_id làm khóa chính.

## Mối quan hệ chính

- Một **User** có thể đóng góp nhiều **Challenge**.
- Mỗi **Challenge** tạo ra một **ReviewPool** duy nhất và có thể có nhiều lượt **ModeratorReview**.
- Một **ModeratorReview** luôn liên kết với một **ReviewPool** và một **Challenge**.
- Một **User** (với vai trò moderator) có thể thực hiện nhiều **ModeratorReview**.
- Mỗi **Challenge** có một **Pot** thưởng và một **ChallengeRevenue**.

#### 4.4.2 Quan hệ giữa User, Challenge, Solution, EvaluationPool và Evaluation



Hình 4.4: Mô hình dữ liệu (ERD) giữa các thực thể User, Challenge, Solution, EvaluationPool và Evaluation

##### Mô tả các thực thể

- **Solution**: Giải pháp mà người dùng gửi cho một thử thách, với

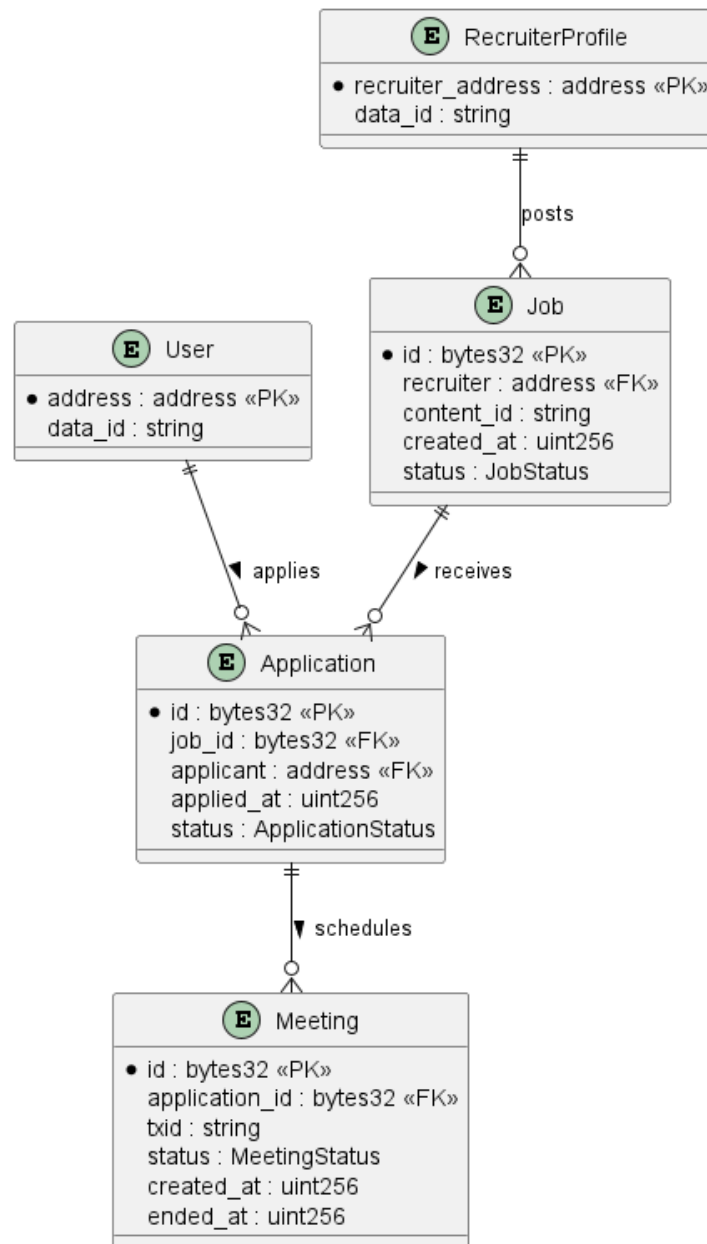
id làm khóa chính. Các thuộc tính bao gồm user (khóa ngoại đến User.address), challenge\_id (khóa ngoại đến Challenge.id), tham chiếu đến dữ liệu off-chain solution\_txid, thời gian tạo (created\_at), thời gian nộp (submitted\_at), tiến độ (progress) và điểm đánh giá trung bình (score).

- **EvaluationPool:** Quản lý những người đánh giá tham gia vào phiên đánh giá của một giải pháp và các bài đánh giá tương ứng, bao gồm các thuộc tính solution\_id làm khóa chính, tổng số lượng người đánh giá, số lượng bài đánh giá và thời điểm hoàn thành quá trình tổng hợp đánh giá (completed\_at).
- **Evaluation:** Mỗi lượt đánh giá của một người đánh giá đối với một giải pháp, khóa chính ghép là (solution\_id, evaluator). Các thuộc tính bao gồm điểm đánh giá đề xuất cho giải pháp và thời gian nộp.

### Mối quan hệ chính

- Một **User** có thể giải nhiều **Solution** (solves).
- Một **Challenge** có thể chứa nhiều **Solution** (contains).
- Mỗi **Solution** gắn với đúng một **EvaluationPool** (has).
- **EvaluationPool** có thể chứa nhiều lượt **Evaluation** (contains).
- Một **User** trong vai trò người có thể thực hiện nhiều bài đánh giá **Evaluation** (evaluates).

### 4.4.3 Quan hệ giữa User, RecruiterProfile, Job, Application và Meeting



Hình 4.5: Mô hình dữ liệu (ERD) giữa các thực thể User, RecruiterProfile, Job, Application và Meeting

#### Mô tả các thực thể

- **RecruiterProfile:** Hồ sơ nhà tuyển dụng, với `recruiter_address` là khóa chính và `data_id` là biến tham chiếu nội dung chi tiết được trữ off-chain.
- **Job:** Công việc do nhà tuyển dụng đăng, xác định bởi `id` kiểu

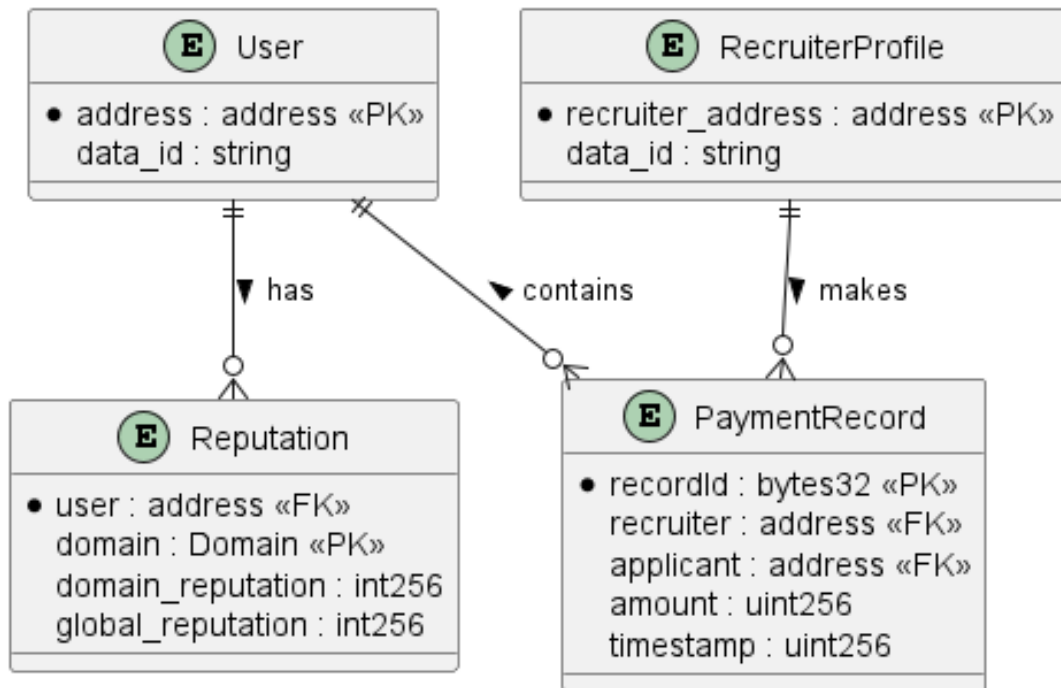
bytes32. Thuộc tính bao gồm `recruiter`, `content_id` (tham chiếu nội dung mô tả công việc được lưu trữ off-chain), thời gian tạo (`created_at`) và trạng thái (`status`).

- **Application:** Đơn ứng tuyển cho một công việc, với `id` làm khóa chính. Gồm `job_id` (khóa ngoại đến `Job.id`), `applicant` (khóa ngoại đến `User.address`), thời gian nộp (`applied_at`) và trạng thái (`status`).
- **Meeting:** Cuộc hẹn phỏng vấn cho một đơn ứng tuyển, với `id` làm khóa chính. Các thuộc tính gồm `application_id` (khóa ngoại đến `Application.id`), tham chiếu đến dữ liệu off-chain (`txid`), trạng thái (`status`), thời điểm bắt đầu (`created_at`) và kết thúc (`ended_at`).

### Mối quan hệ chính

- Một **RecruiterProfile** có thể đăng nhiều **Job** (*posts*).
- Một **Job** có thể nhận nhiều **Application** (*receives*).
- Một **User** có thể nộp nhiều **Application** (*applies*).
- Mỗi **Application** có thể lên lịch nhiều **Meeting** (*schedules*).

#### 4.4.4 Quan hệ giữa RecruiterProfile, User, PaymentRecord và Reputation



Hình 4.6: Mô hình dữ liệu (ERD) giữa các thực thể RecruiterProfile, User, PaymentRecord và Reputation

##### Mô tả các thực thể

- **PaymentRecord**: Bản ghi thanh toán phí tuyển dụng mỗi khi nhà tuyển dụng tuyển một ứng cử viên, xác định bởi `recordId` kiểu `bytes32`. Các thuộc tính bao gồm `recruiter` là khóa ngoại đến `RecruiterProfile.recruiter_address`, `applicant` là khóa ngoại đến `User.address`, `amount` (số tiền) và `timestamp` (thời gian giao dịch).
- **Reputation**: Điểm uy tín của người dùng theo từng lĩnh vực, với khóa chính tổng hợp (`user`, `domain`). Thuộc tính bao gồm điểm uy tín chuyên môn `domain_reputation` và điểm uy tín toàn cục `global_reputation`.

##### Mối quan hệ chính

- Một **RecruiterProfile** có thể tạo nhiều **PaymentRecord** (*makes*).

- Một **User** có thể xuất hiện trong nhiều **PaymentRecord** với vai trò ứng viên (*contains*).
- Một **User** có thể có nhiều **Reputation** theo các lĩnh vực khác nhau (*has*).

## 4.5 Chi tiết các hợp đồng thông minh

### 4.5.1 Danh sách các hợp đồng chính

Hệ thống SkillChain được tổ chức thành nhiều hợp đồng thông minh độc lập theo nguyên tắc phân tách trách nhiệm. Mỗi hợp đồng đảm nhiệm một vai trò riêng trong hệ thống và được triển khai độc lập để tăng khả năng bảo trì, tái sử dụng và kiểm soát truy cập. Dưới đây là danh sách các hợp đồng chính trong hệ thống:

- **UserDataManager**: Quản lý thông tin cá nhân và định danh người dùng.
- **ReputationManager**: Lưu trữ và cập nhật hồ sơ uy tín của người dùng, bao gồm điểm uy tín toàn cục và điểm uy tín chuyên môn theo từng lĩnh vực. Hợp đồng này là trung tâm của hệ thống uy tín.
- **ChallengeManager**: Quản lý vòng đời của thử thách, từ khởi tạo, kiểm duyệt đến cho phép người dùng tham gia.
- **ModerationEscrow**: Điều phối việc phân phối phần thưởng token cho những người kiểm duyệt thử thách sau khi quá trình kiểm duyệt hoàn tất.
- **ChallengeCostManager**: Tính toán mức phí mà người dùng cần chi trả để tham gia vào một thử thách cụ thể.
- **SolutionManager**: Quản lý quá trình người dùng thực hiện và gửi giải pháp cho thử thách, ghi nhận hành vi nộp bài, và xử lý luồng đánh giá bởi các người đánh giá.



- **RecruiterDataManager:** Quản lý thông tin tài khoản nhà tuyển dụng.
- **RecruiterSubscription:** Kiểm soát việc thanh toán phí đăng ký theo chu kỳ từ phía nhà tuyển dụng để có quyền đăng bài tuyển dụng trên hệ thống.
- **JobManager:** Lưu trữ và quản lý các bài đăng tuyển dụng.
- **JobApplicationManager:** Ghi nhận và quản lý thông tin ứng tuyển của người dùng cho từng bài đăng cụ thể.
- **MeetingManager:** Tổ chức và lưu trữ lịch họp trực tuyến giữa nhà tuyển dụng và ứng viên.
- **RoleManager:** Phân quyền hệ thống, kiểm soát ai được phép gọi các chức năng đặc biệt. Hợp đồng này đảm bảo an toàn và tách biệt vai trò trong hệ thống.

## 4.5.2 ReputationManager

Hợp đồng **ReputationManager** chịu trách nhiệm duy trì, cập nhật và điều chỉnh các chỉ số uy tín của người dùng theo từng lĩnh vực chuyên môn cũng như tổng điểm uy tín toàn cục. Đây là thành phần cốt lõi để xác định vai trò, quyền truy cập và độ tin cậy của người tham gia trên nền tảng SkillChain.

### 4.5.2.1 Chức năng chính

- Ghi nhận và duy trì điểm uy tín theo từng lĩnh vực chuyên môn và tổng uy tín toàn cục.
- Tính toán và cập nhật sự thay đổi uy tín dựa trên hành vi của người dùng trong các vai trò người giải thử thách, người đóng góp, người kiểm duyệt và người đánh giá giải pháp. Cơ chế cập nhật đã được giải thích chi tiết tại mục 3.2.
- Tự động cấp hoặc thu hồi vai trò tùy theo mức uy tín đạt được.

#### 4.5.2.2 Các hàm chức năng chính

- **updateSolvingProblemReputation**: cập nhật uy tín người giải thử thách sau khi giải pháp đã được đánh giá
- **updateEvaluateSolutionReputation**: cập nhật uy tín người đánh giá giải pháp sau khi phiên đánh giá kết thúc
- **updateContributionReputation**: cập nhật uy tín người đóng góp sau khi thử thách đã được duyệt xong
- **updateModerationReputation**: cập nhật uy tín người kiểm duyệt sau khi phiên kiểm duyệt hoàn tất

#### 4.5.2.3 Tương tác với hợp đồng khác

- **RoleManager**: Cấp hoặc thu hồi vai trò sau khi uy tín thay đổi
- **ChallengeManager**: Cập nhật uy tín sau khi phiên kiểm duyệt thử thách hoàn tất
- **SolutionManager**: Cập nhật uy tín sau khi phiên đánh giá giải pháp kết thúc

### 4.5.3 ChallengeManager

Hợp đồng **ChallengeManager** đóng vai trò trung tâm trong quá trình quản lý các thử thách kỹ năng trong hệ thống SkillChain. Hợp đồng này chịu trách nhiệm cho toàn bộ vòng đời của một thử thách – từ khởi tạo, đóng góp, kiểm duyệt cho đến tham gia thử thách và cập nhật dữ liệu uy tín.

#### 4.5.3.1 Chức năng chính

- Cho phép người đóng góp tạo và gửi thử thách cho bên kiểm duyệt
- Cho phép người kiểm duyệt tham gia kiểm duyệt và đánh giá chất lượng thử thách

- Hợp nhất kết quả kiểm duyệt theo mô hình có trọng số dựa trên uy tín, từ đó phân phối token cho bên kiểm duyệt và cập nhật uy tín cho người đóng góp và từng người kiểm duyệt
- Cho phép người dùng tham gia thử thách, hỗ trợ tính toán lượng token cần phải trả cho người đóng góp để tham gia

#### 4.5.3.2 Cấu trúc dữ liệu chính

- Challenge: chứa thông tin đầy đủ của một thử thách
- ModeratorReview: ghi nhận đánh giá của từng người kiểm duyệt đối với một thử thách
- ReviewPool: tập hợp những người kiểm duyệt tham gia vào một thử thách và trạng thái kiểm duyệt
- JoinedChallenges: ghi nhận thử thách mà người dùng đã và đang tham gia giải

#### 4.5.3.3 Các hàm chức năng chính

##### Quản lý thử thách

- createChallenge: Người dùng có đủ quyền đóng góp trong một lĩnh vực chuyên môn có thể khởi tạo thử thách nháp
- contributeChallenge: Người đóng góp chính thức gửi thử thách cho bên kiểm duyệt, đính kèm mức token treo thưởng.

##### Kiểm duyệt thử thách

- joinReviewPool: Cho phép người kiểm duyệt hợp lệ tham gia vào phiên kiểm duyệt
- submitModeratorReview: Gửi đánh giá chất lượng thử thách, tính điểm theo các yếu tố chất lượng
- finalizeChallenge: Khi đủ số lượng đánh giá, hợp nhất kết quả yếu tố chất lượng, xác định mức độ khó, tính thời gian giải ước tính, tính điểm chất lượng, cập nhật uy tín và phân phối token.

## Người dùng tham gia thử thách

- `userJoinChallenge`: Người dùng trả phí token, tham gia thử thách và khởi tạo không gian giải pháp
- `userCompleteChallenge`: Cập nhật số người hoàn thành

### 4.5.3.4 Cơ chế hợp nhất kết quả kiểm duyệt

Giả sử  $D$  là tập hợp mức độ khó của thử thách (dễ, trung bình, khó),  $F$  là tổng số lượng các yếu tố chất lượng dùng để đánh giá thử thách,  $M$  là tập hợp những người kiểm duyệt đánh giá một thử thách. Với mỗi người kiểm duyệt  $m \in M$ , ta có:

- $d_m$  là độ khó đã đề xuất
- $t_m$  là thời gian giải ước tính đã đề xuất
- $q_{mf}$  là đánh giá của người kiểm duyệt đối với một yếu tố chất lượng, mang một trong hai ý nghĩa: Đúng hoặc Sai,  $q_{mf} \in \{0, 1\}$
- $w_m$  là trọng số uy tín của người kiểm duyệt dựa trên điểm uy tín chuyên môn  $R_i$  của họ trong lĩnh vực của thử thách

$$w_m = R_i$$

### Xác định mức độ khó

- Để xác định mức độ khó của thử thách, hệ thống sử dụng cơ chế biểu quyết theo số đông dựa trên trọng số. Hệ thống lần lượt tính tổng trọng số  $W(d)$  của từng mức độ khó  $d \in D$  theo công thức:

$$W(d) = \sum_{m \in M} w_m \cdot \mathbf{1}(d_m = d)$$

- Sau cùng, mức độ khó nào có tổng trọng số cao nhất sẽ là mức độ khó của thử thách:

$$d_{\text{final}} = \arg \max_{d \in D} W(d)$$

## Tính thời gian giải ước tính

- Thời gian giải ước tính của thử thách là trung bình có trọng số của thời gian giải ước tính đã đề xuất  $t_m$  và trọng số uy tín  $w_m$  của người kiểm duyệt tương ứng:

$$T_{\text{final}} = \frac{\sum_{m=1}^M (w_m * t_m)}{\sum_{m=1}^M w_m}$$

## Tính điểm chất lượng

- Người kiểm duyệt đánh giá thử thách bằng cách trả lời  $F$  yếu tố chất lượng đã được hệ thống quy định sẵn. Sau khi nộp bài đánh giá, hệ thống tính điểm chất lượng đề xuất của từng người kiểm duyệt theo công thức:

$$q_m = \frac{100}{F} \sum_{f=1}^F q_{mf} \in [0, 100]$$

- Điểm chất lượng của thử thách được tổng hợp dựa trên điểm chất lượng đề xuất của những người kiểm duyệt tham gia vào phiên kiểm duyệt:

$$Q_{\text{challenge}} = \frac{\sum_{m=1}^M w_m q_m}{\sum_{m=1}^M w_m} \in [0, 100]$$

- Điểm chất lượng là yếu tố then chốt để quyết định thử thách đạt hay không đạt, dựa trên ngưỡng đạt  $Q_{\text{threshold}}$  :

$$\text{Kết quả} = \begin{cases} \text{Thử thách đạt,} & \text{nếu } Q_{\text{challenge}} \geq Q_{\text{threshold}} \\ \text{Thử thách không đạt,} & \text{nếu } Q_{\text{challenge}} < Q_{\text{threshold}} \end{cases}$$

- Theo thống nhất của nhóm phát triển, ngưỡng đạt ở mức sau là phù hợp với yêu cầu hiện tại của hệ thống, vừa đảm bảo các thử thách

có chất lượng tốt, vừa không quá khắt khe để khuyến khích việc đóng góp thử thách vào hệ thống:

$$Q_{\text{threshold}} = 80$$

#### 4.5.3.5 Tương tác với hợp đồng khác

- **ReputationManager**: Cập nhật uy tín sau khi phiên kiểm duyệt chất lượng thử thách kết thúc
- **RoleManager**: Kiểm soát quyền gọi hàm dựa trên chỉ số uy tín của người gọi
- **SolutionManager**: Phối hợp để quản lý thử thách mà người dùng tham gia
- **ChallengeCostManager**: Tính toán khoản token để tham gia thử thách

### 4.5.4 ModeratorEscrow

Hợp đồng **ModerationEscrow** đóng vai trò như một “kho quỹ ký gửi” để tạm giữ khoản token thưởng do người đóng góp nạp vào khi tạo thử thách. Sau khi thử thách được kiểm duyệt và hợp nhất kết quả, hợp đồng này sẽ tính toán và phân phối phần thưởng cho các người kiểm duyệt phù hợp, dựa trên mức độ chính xác của kết quả kiểm duyệt và uy tín chuyên môn.

#### 4.5.4.1 Chức năng chính

- Nhận và lưu trữ token bounty từ người đóng góp
- Ghi nhận danh sách người kiểm duyệt tham gia
- Đánh giá độ lệch giữa điểm số của từng người kiểm duyệt với điểm số trung bình cuối cùng
- Tính trọng số và phân phối phần thưởng công bằng dựa trên uy tín và mức độ chính xác

#### 4.5.4.2 Các hàm chức năng chính

##### Giao tiếp với người đóng góp

- `depositBounty`: Được gọi bởi **ChallengeManager** để lưu mức treo thưởng do người đóng góp cung cấp khi gửi thử thách.

##### Giao tiếp với người kiểm duyệt

- `syncModeratorsFromReviewPool`: Đồng bộ danh sách kiểm duyệt từ **ChallengeManager** vào **ModerationEscrow**
- `finalizeChallengePot`: Thực hiện phân phối token sau khi thử thách được duyệt xong

#### 4.5.4.3 Cơ chế phân phối token

Khi người đóng góp tạo một thử thách, họ cần chi một khoảng token  $B$  để thưởng cho những người kiểm duyệt. Sau quá trình tổng hợp kết quả kiểm duyệt đã trình bày ở phần trước, hệ thống đã tính được điểm chất lượng  $Q_{\text{challenge}}$  của thử thách. Giả sử  $M$  là tập hợp những người kiểm duyệt đánh giá một thử thách. Với mỗi người kiểm duyệt  $m \in M$ , ta có:

- $q_m$  là điểm chất lượng đề xuất của người kiểm duyệt
- $\Delta q_m$  là chênh lệch điểm số giữa điểm chất lượng  $Q_{\text{challenge}}$  của thử thách sau kiểm duyệt và điểm chất lượng đề xuất của người kiểm duyệt:

$$\Delta q_m = |q_m - Q|$$

- $R_m$  là điểm uy tín chuyên môn của người kiểm duyệt
- $B_m$  là số token mà người kiểm duyệt được thưởng (nếu đạt yêu cầu)

##### Phân loại dựa trên chênh lệch điểm chất lượng

- Dựa vào chênh lệch điểm chất lượng  $\Delta q_m$  của từng người kiểm duyệt và ngưỡng lệch điểm chất lượng  $q_{\text{threshold}}$ , hệ thống phân loại

họ thành hai nhóm: những người kiểm duyệt được thưởng và những người kiểm duyệt không được thưởng.

$$\text{Người kiểm duyệt} = \begin{cases} \text{Được thưởng} & \text{nếu } \Delta q_m \leq q_{\text{threshold}} \\ \text{Không được thưởng} & \text{nếu } \Delta q_m > q_{\text{threshold}} \end{cases}$$

- Theo thống nhất của nhóm phát triển, ngưỡng lệch điểm chất lượng sẽ ở mức sau:

$$q_{\text{threshold}} = 40$$

### Tính trọng số thưởng cho những người kiểm duyệt được thưởng

- Giả sử  $\beta$  là hệ số phản ánh mức độ ảnh hưởng của điểm uy tín chuyên môn đối với trọng số thưởng ( $\beta \in [0, 1]$ ),  $\alpha$  là hệ số phản ánh mức độ giảm giá trị thưởng theo chênh lệch điểm chất lượng, ( $\alpha \in (0, +\infty)$ ). Trọng số thưởng của người kiểm duyệt  $m$  được tính theo công thức:

$$w_m = R_m^\beta \cdot e^{-\alpha \Delta q_m}$$

- Công thức trọng số thưởng là một hàm kết hợp dựa trên hai yếu tố: điểm uy tín chuyên môn và chênh lệch điểm chất lượng của người kiểm duyệt (độ chính xác).
- Tham số  $\beta \in [0, 1]$  điều chỉnh mức độ ảnh hưởng của uy tín tới trọng số thưởng.
  - Nếu  $\beta$  lớn ( $\beta$  gần 1), điểm uy tín chuyên môn sẽ có ảnh hưởng nhiều hơn đến trọng số thưởng. Từ đó, người kiểm duyệt có điểm uy tín cao sẽ được nhận thưởng nhiều hơn.
  - Nếu  $\beta$  nhỏ ( $\beta$  gần 0), điểm uy tín chuyên môn sẽ có ảnh hưởng không đáng kể đến trọng số, nghĩa là yếu tố chính xác (thể hiện qua chênh lệch điểm chất lượng  $\Delta q_m$ ) sẽ chiếm vai trò chủ đạo.
- Tham số  $\alpha$  có tác dụng kiểm soát tốc độ giảm của trọng số thưởng dựa trên chênh lệch điểm chất lượng:



- Nếu  $\alpha$  lớn, trọng số thưởng sẽ giảm rất nhanh khi chênh lệch điểm chất lượng tăng lên, nghĩa là phần lớn token thưởng được phát cho những người đánh giá chính xác nhất.
- Nếu  $\alpha$  nhỏ, trọng số thưởng giảm chậm hơn khi chênh lệch điểm chất lượng tăng lên, nghĩa là số token thưởng sẽ được phân phát đều hơn cho những người kiểm duyệt.

### Phân phối phần thưởng

- Quỹ thưởng chính là số token thưởng  $B$  mà người đóng góp chi ra để thưởng cho những người kiểm duyệt
- Giả sử  $M_{\text{reward}}$  là tập hợp những người kiểm duyệt được thưởng. Phần thưởng mà mỗi người kiểm duyệt nhận được là:

$$B_m = \frac{w_m}{\sum_m^{M_{\text{reward}}} w_m} B$$

#### 4.5.4.4 Tương tác với hợp đồng khác

- **ReputationManager**: Truy xuất thông tin uy tín của kiểm duyệt viên
- **ChallengeManager**: Hỗ trợ phân phối thưởng khi phiên kiểm duyệt hoàn tất

### 4.5.5 ChallengeCostManager

Hợp đồng **ChallengeCostManager** đảm nhiệm việc tính toán, thu thập và chuyển giao token phí từ người tham gia thử thách đến người đóng góp thử thách. Nó giúp đảm bảo sự khuyến khích kinh tế rõ ràng cho người tạo nội dung và điều chỉnh giá trị tham gia dựa trên độ khó và chất lượng.

#### 4.5.5.1 Chức năng chính

- Tính phí tham gia thử thách

- Xử lý thanh toán từ người tham gia đến contributor khi đăng ký tham gia thử thách
- Ghi nhận tổng doanh thu và thông tin thanh toán theo từng thử thách

#### 4.5.5.2 Các hàm chức năng chính

- `_computeCost`: tính lượng token để tham gia thử thách
- `addTalentPayment`: gửi token từ người tham gia cho người đóng góp và ghi nhận người tham gia đối với thử thách này

#### 4.5.5.3 Công thức tính phí

Giả sử:

- $F_{\min}, F_{\max}$  là phí sàn và phí trần (native token - POL)
- $\alpha$  là hằng số phản ánh mức độ ảnh hưởng của độ mức độ khó đối với phí tham gia ( $0 < \alpha < 1$ )
- $\beta$  là hằng số phản ánh mức độ ảnh hưởng của chất lượng đối với phí tham gia ( $0 < \beta < 1$ )
- $W_{\text{difficulty}}$  là trọng số độ khó

$$W_{\text{difficulty}} = \begin{cases} 1 & , \text{nếu độ khó là dễ} \\ 2 & , \text{nếu độ khó là trung bình} \\ 3 & , \text{nếu độ khó là khó} \end{cases}$$

- $Q_{\text{challenge}}$  là điểm chất lượng của thử thách sau kiểm duyệt, được chuẩn hoá  $[0, 1]$
- $\gamma$  là hằng số phản ánh tỷ lệ hoàn vốn ( $0 < \gamma \leq 1$ )
- $B$  là số lượng token người đóng góp chi ra để thưởng cho những người kiểm duyệt

- $E$  là số lượt người dùng tham gia vào thử thách kỳ vọng
- $n_i$  là số lượt người dùng đã tham gia vào thử thách trước người chơi thứ  $i$
- $\beta$  là tham số phản ánh tốc độ suy giảm chi phí tham gia thử thách khi số lượng người chơi tăng lên

### Tính phí tham gia thử thách

- Phí tham gia thử gồm hai thành phần: phí cơ sở và phí hoàn vốn
- Phí cơ sở được tính bằng một hàm tuyến tính dựa trên phí sàn, độ khó của thử thách và điểm chất lượng của thử thách:

$$F_{\text{base}} = F_{\text{min}} + \alpha W_d + \beta Q$$

- Phí hoàn vốn là mức phí tính thêm cho những người đầu tiên tham gia vào thử thách, giảm dần về 0 khi càng nhiều người tham gia vào thử thách. Mức phí này hỗ trợ người đóng góp hoàn lại số vốn đã chi khi đóng góp thử thách:

$$F_{\text{bonus}} = \gamma \frac{B}{E} e^{-\beta n_i}$$

- Phí tham gia thử thách của người tham gia thứ  $i$  được tính bằng công thức sau, phụ thuộc vào phí cơ sở và phí hoàn vốn. Tuy nhiên, mức phí này bị ràng buộc giới hạn trên  $F_{\text{max}}$  và giới hạn dưới  $F_{\text{min}}$  để đảm bảo quyền lợi của người dùng khi tham gia thử thách:

$$F_i = \begin{cases} F_{\text{min}}, & \text{nếu } F_{\text{base}} + F_{\text{bonus}} < F_{\text{min}}, \\ F_{\text{base}} + F_{\text{bonus}}, & \text{nếu } F_{\text{min}} \leq F_{\text{base}} + F_{\text{bonus}} \leq F_{\text{max}}, \\ F_{\text{max}}, & \text{nếu } F_{\text{base}} + F_{\text{bonus}} > F_{\text{max}}. \end{cases}$$

**Các giá trị hằng số đề xuất** Bảng sau đây liệt kê các giá trị hằng số được đề xuất cho công thức tính phí tham gia thử thách:

Ký hiệu	Giá trị đề xuất
$F_{\text{base}}$	0.01 POL
$F_{\text{max}}$	0.05 POL
$\alpha$	0.6
$\beta$	0.8
$\gamma$	0.3
$E$	30

#### 4.5.5.4 Tương tác với hợp đồng khác

- **ChallengeManager**: Hỗ trợ tính phí tham gia thử thách
- **ModeratorEscrow**: Dùng để lấy mức treo thưởng, phục vụ cho tính phí động

### 4.5.6 SolutionManager

Hợp đồng **SolutionManager** chịu trách nhiệm ghi nhận toàn bộ quy trình người dùng tham gia thử thách, nộp giải pháp, đưa vào hệ thống đánh giá, cũng như tổng hợp kết quả đánh giá và cập nhật điểm uy tín.

#### 4.5.6.1 Chức năng chính

- Quản lý thông tin giải pháp của người dùng đối với mỗi thử thách
- Điều phối quá trình đánh giá giải pháp bằng cách tổ chức một phiên đánh giá gồm các người đánh giá
- Tính điểm trung bình có trọng số cho giải pháp, cập nhật tiến độ, lưu trữ điểm số và cập nhật điểm uy tín cho người giải và những người đánh giá

#### 4.5.6.2 Cấu trúc dữ liệu chính

- **Solution**: Chứa thông tin đầy đủ của một giải pháp
- **Evaluation**: Ghi nhận điểm số của người đánh giá đối với một giải pháp

- `EvaluationPool`: Tập hợp những người đánh giá tham gia vào một giải pháp và trạng thái đánh giá

#### 4.5.6.3 Các hàm chức năng chính

##### Người dùng tham gia thử thách

- `createSolutionBase`: Ghi nhận hành vi tham gia thử thách và khởi tạo không gian giải pháp

##### Nộp và chuyển trạng thái giải pháp

- `submitSolution`: Người dùng nộp bài (không chỉnh sửa được nữa).
- `putSolutionUnderReview`: Chuyển giải pháp sang trạng thái chờ đánh giá và khởi tạo một phiên đánh giá.

##### Quản lý phiên đánh giá

- `evaluatorJoinSolution`: Cho phép người đánh giá tham gia đánh giá giải pháp
- `evaluatorSubmitScore`: Gửi điểm đánh giá
- `finalizeEvaluation`: Khi ghi nhận đủ điểm đánh giá, tính toán điểm giải pháp cuối cùng và cập nhật uy tín cho người giải và những người đánh giá

#### 4.5.6.4 Cơ chế tổng hợp điểm đánh giá

Sau khi người dùng nộp bài giải cho một thử thách, một phiên đánh giá sẽ được kích hoạt. Một số chuyên gia đánh giá sẽ tham gia phiên này và đưa ra điểm số cho bài giải. Điểm số cuối cùng của bài giải được tính theo công thức sau:

$$S_{\text{solution}} = \frac{\sum_{i=1}^n (W_i \cdot S_i)}{\sum_{i=1}^n W_i}$$

Trong đó:

- $S_{\text{solution}}$ : Là điểm số cuối cùng của bài giải, với giá trị nằm trong khoảng  $[0, 100]$ .
- $s_i$ : Là điểm số mà người đánh giá thứ  $i$  chấm cho bài giải, với giá trị nằm trong khoảng  $[0, 100]$ .
- $n$ : Là tổng số người đánh giá tham gia phiên đánh giá.
- $W_i$ : Là trọng số thể hiện uy tín của người đánh giá thứ  $i$  trong **lĩnh vực của thử thách**, được xác định theo công thức:

$$W_i = R_i$$

Trong đó  $R_i$  là điểm uy tín chuyên môn của người đánh giá thứ  $i$  trong lĩnh vực của thử thách.

#### 4.5.6.5 Tương tác với hợp đồng khác

- **ReputationManager**: Cập nhật uy tín sau khi phiên đánh giá giải pháp kết thúc
- **ChallengeManager**: Phối hợp để quản lý thử thách mà người dùng đã tham gia
- **RoleManager**: Kiểm soát quyền gọi hàm

#### 4.5.7 RecruiterSubscription

## Chương 5

# Kết quả

## Chương 6

# Kết luận



# Tài liệu tham khảo

- [1] N. Đình Thúc, “Quản lý Định danh phi tập trung.” Tài liệu nội bộ được cung cấp bởi giảng viên hướng dẫn.
- [2] T. Grandison and M. Sloman, “A survey of trust in internet applications,” *IEEE Communications Surveys and Tutorials*, vol. 3, pp. 2–16, 01 2000.
- [3] N. Đình Thúc, Đặng Hải Vân, and L. Phong, *Thống kê máy tính*. Nhà xuất bản Khoa học và kỹ thuật, 2010.
- [4] C. Allen, T. Daubenschütz, M. Sporny, N. Thorp, H. Wood, G. Willen, and A. Voto, “Portable reputation toolkit use cases.” GitHub Repository, 2017. A White Paper from Rebooting the Web of Trust III Design Workshop.
- [5] A. C. de Crespigny, D. Khovratovich, F. Blondeau, K. Sok, P. Honigman, N. Alexopoulos, F. Petitcolas, and S. Conway, “Design considerations for decentralized reputation systems.” GitHub Repository, 2017. A White Paper from the Rebooting the Web of Trust IV Design Workshop.
- [6] A. Brock, K. Hamlin, G. R. Rachmany, and J. Lanc, “Reputation interpretation.” GitHub Repository, 2019. A white paper from Rebooting the Web of Trust IX.
- [7] J. Sabater and C. Sierra, “Regret: reputation in gregarious societies,” in *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS ’01, (New York, NY, USA), pp. 194–195, Association for Computing Machinery, 2001.

- [8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *Proceedings of the 12th International Conference on World Wide Web, WWW '03*, (New York, NY, USA), pp. 640–651, Association for Computing Machinery, 2003.
- [9] Wikipedia contributors, “Markov chain — Wikipedia, the free encyclopedia,” 2025. [Online; accessed 25-June-2025].
- [10] P. Chebolu and P. Melsted, “Pagerank and the random surfer model,” in *SODA*, vol. 8, pp. 1010–1018, 2008.
- [11] “Slashdot.” <https://slashdot.org/>, 1997. Lần cuối truy cập vào 25/06/2025.
- [12] N. Poor, “Mechanisms of an online public sphere: the website slashdot,” *Journal of Computer-Mediated Communication*, vol. 10, p. JCMC1028, 07 2017.

# Phụ lục