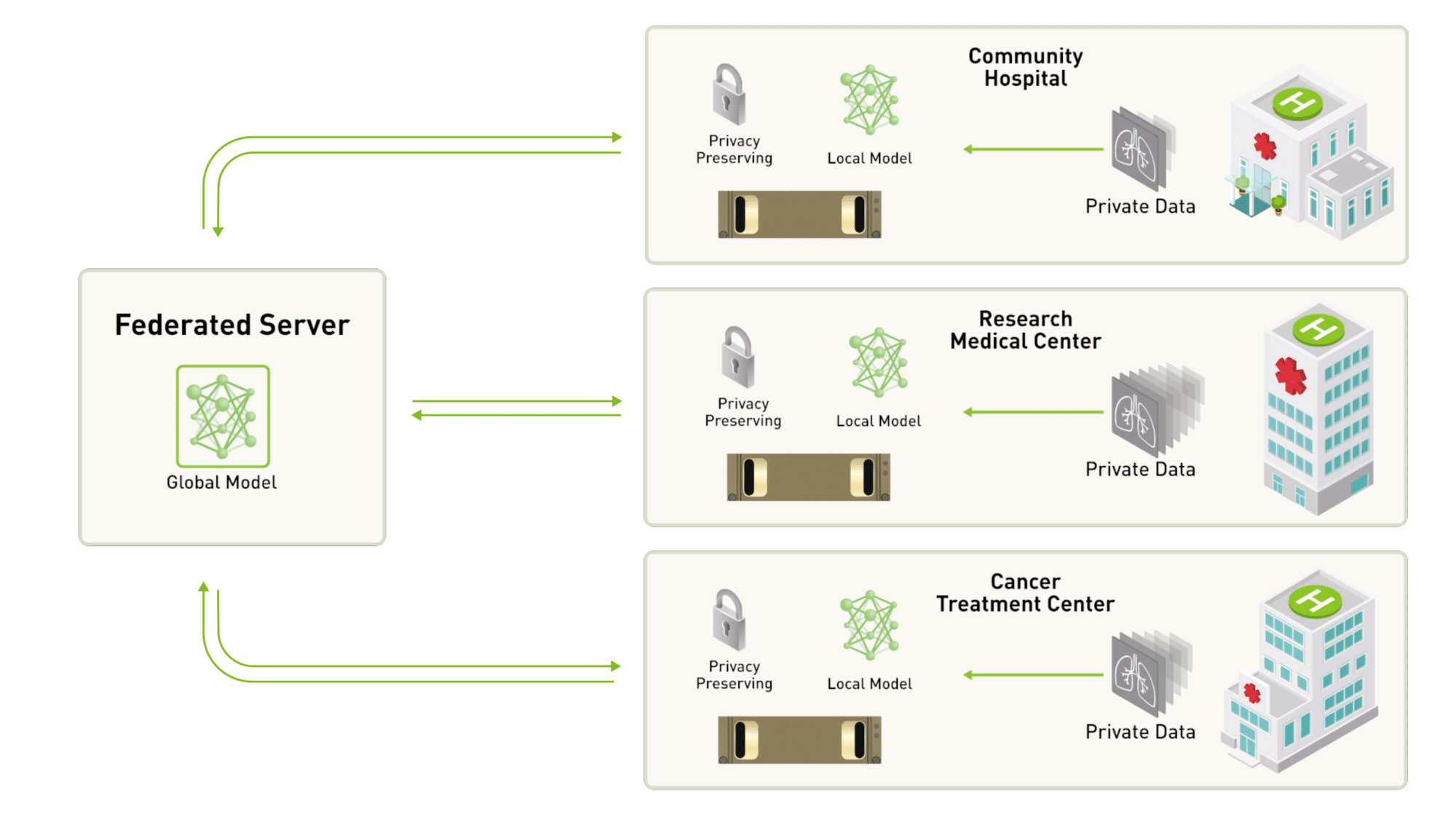
<https://inseclab.uit.edu.vn/tong-quan-mo-hinh-ket-hop-federated-learning-va-blockchain/>



Hoạt động

Dữ liệu được gửi lên là trọng số của mô hình thay vì dữ liệu gốc của người dùng.

Đảm bảo privacy, Giảm thiểu được tài nguyên tính toán cho máy chủ trung tâm.

Serve khởi tạo mô hình và gửi cho các máy trạm để huấn luyện trên tập dataset của mình.

Quá trình tổng hợp mô hình có thể sử dụng một số thuật toán phổ biến như **FedAVG** hoặc **FedSGD.**

Máy chủ trung tâm sẽ thực hiện lặp lại quá trình này đến khi nào đủ số round cần huấn luyện.

Vấn đề về đầu độc mô hình

node gửi trọng số sai

Cơ chế phát hiện:

1.Cosine Similarity

hoạt động bằng cách đo độ tương tự Cosine giữa trọng số của máy trạm i và server tổng hợp. Giá trị này được cho phép nếu nằm trong một ngưỡng nhất định, nếu ngoài ngưỡng này, trọng số của máy trạm i được xem là độc hại và sẽ không được dùng để tổng hợp ở máy chủ trung tâm.

2.Các node có điểm thấp dưới ngưỡng nào đó sẽ bị xem là độc hại và bị loại khỏi quá trình training (có kết hợp sharding).

1. Mỗi shard sẽ có một leader thực hiện tổng hợp trọng số, leader này sẽ chấm điểm các node trong shard đó dựa theo thời gian phản hồi trọng số huấn luyện.

2. Nếu phản hồi trong thời gian cho phép sẽ được cộng điểm, không phản hồi hoặc quá thời gian sẽ bị trừ điểm. Ngoài ra, lịch sử phản hồi cũng được đánh trọng số để chấm điểm, các phản hồi trong quá khứ sẽ có trọng số nhỏ hơn phản hồi gần đây (ít quan trọng hơn).

Hơn nữa, cũng cần có cơ chế để các shard có thể chấm điểm chéo cho nhau, hoặc các leader có thể chia sẻ thông tin điểm ở shard của mình để đồng bộ điểm cho cả hệ thống.

Đảm bảo quyền riêng tư (privacy)

Có 2 cơ chế thường được sử dụng:

Differential Privacy: Kỹ thuật này thực hiện thêm nhiễu vào trong dữ liệu / trọng số, nhằm làm khó khăn cho kẻ tấn công trong trường hợp muốn rút trích một phần thông tin gốc. Nhược điểm của kỹ thuật này là làm giảm độ chính xác của mô hình do nhiễu.

Sử dụng kỹ thuật mật mã: Có thể kể đến như mã hóa đồng cấu (Homomorphic Encryption) cho phép tính toán trực tiếp trên dữ liệu bị mã hóa mà không cần thiết phải giải mã. Ví dụ: Ta có 2 thông điệp bản rõ là m1 và m2, hàm mã hóa Enc. Khi đó, Enc(m1) + Enc(m2) = m1 + m2 hoặc Enc(m1) \* Enc(m2) = m1 \* m2.

**BlockChain**

Hyperledger Fabric

Giao thức đồng thuận cũng có thể chia làm hai loại là

BFT-based (Byzantine Fault Tolerance-based)

PoX-based (Proof of X-based)

BFT-based (ví dụ như Practical Byzantine Fault Tolerance), mặc dù các cơ chế này hoạt động với độ bảo mật cao nhưng thường khả năng scale out (mở rộng) thấp do số lượng message phải gửi trong mạng là quá lớn.

Nakamoto sử dụng cơ chế **Proof of Work (PoW)** để xác định người nào sẽ đảm nhận quá trình tạo khối.

PoW là cơ chế giải đố, các thợ đào (miner) trong mạng cần phải giải một bài toán rất khó, người nào giải ra đầu tiên sẽ trở thành người đề xuất khối mới.

hạn chế của PoW (tốn tài nguyên, ảnh hưởng đến môi trường do phải giải quyết một bài toán rất khó)

**Proof of Stake (PoS).**

PoS hoạt động bằng cách các node trong mạng stake tiền để trở thành các validator, người nào stake càng nhiều tiền thì khả năng trở thành validator càng lớn.

Các validator này có nhiệm vụ xác thực các giao dịch, đảm bảo an toàn cho hệ thống, vì vậy nếu họ thực hiện các hành động sai trái thì sẽ bị phạt tiền và ngược lại thì sẽ nhận được tiền thưởng từ hệ thống.

Security PoW > PoS, fast: PoW < PoS

PoS được xem là ít an toàn hơn so với PoW nhưng thời gian sinh ra mỗi khối của PoS nhanh hơn PoW (10 phút) do đó TPS (transaction per second) của các blockchain dựa trên PoS thường cao hơn.

Security PoX < BFT-based, fast: PoX > BFT-based

Các cơ chế đồng thuận dựa trên PoX thường chạy nhanh hơn nhiều so với BFT-based nhưng lại không an toàn bằng vì các cơ chế này không đảm bảo đầy đủ tính chất Byzantine Fault Tolerance của một hệ thống phân tán mà chỉ đảm bảo một phần.

<https://inseclab.uit.edu.vn/ky-thuat-sharding-trong-blockchain/>

A diagram of a network

Description automatically generated

**Sharding**

Chia thành các node nhỏ để dễ trị.

Tăng tốc độ

Số lượng shard cần chia cũng là một vấn đề quan trọng vì số lượng shard càng nhiều thì sức mạnh tính toán cần để kiểm soát một shard càng ít và ngược lại thì khả năng scale sẽ không được tốt (đánh đổi giữa bảo mật và khả năng mở rộng).

**INTRA-CONSENSUS-SAFETY:** chia càng nhiều thì càng dễ bị tấn công, attacker chỉ cần tấn công một shared là chiếm điều khiển shared đó. P > PT/n\*(50%)

**CROSS-SHARD-ATOMICITY**: transaction có nguồn và đích không thuộc cùng một shard. Do đó, việc xử lý các transaction này thường phức tạp và tốn thời gian hơn.

Cơ chế Locking/Unlocking:

1. client sẽ gửi cross-shard transaction đến các input shards để thu thập bằng chứng chứng thực (tiền đủ không, đã Lock chưa, ...) từ các input shards.

2. Các input shards sẽ thực hiện khóa (Lock) tiền của client, đồng thời gửi các bằng chứng.

3. Nếu tất cả bằng chứng đều hợp lệ (Accepted) thì client sẽ gửi đến output shard để xử lý, thực hiện việc Unlock tiền để chuyển đến tài khoản đích.

4. Nếu bằng chứng không hợp lệ (Rejected) thì thực hiện việc Unlock tiền để trả về cho client, đảm bảo tính Atomicity.

Cơ chế Locking/Unlocking hoạt động trên các blockchain như Omniledger, RapidChain, ChainSpace, ....Ngoài Locking/Unlocking, còn có một số cơ chế như Eventual Atomicity hoặc Junking, .... Trong đó, Eventual Atomicity được sử dụng trong blockchain Monoxide cùng với cơ chế minning Chu-ko-nu.