



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CÁC HỆ THỐNG PHÂN TÁN

Thiết kế và triển khai hệ thống P2P File Sharing sử dụng Pekko Framework

Giảng viên: Kim Ngọc Bách

Nhóm 10:

Hoàng Thanh Hằng - B24CHHT068

Trần Văn Quyền - B24CHHT090

Giáp Thị Huê - B24CHHT075

Hoàng Phương Hoa - B24CHHT075

Posts and Telecommunications
Institute of Technology



A. Tổng quan đề tài và mục tiêu

Bối cảnh và đặt vấn đề

- Sự bùng nổ của dữ liệu và nhu cầu trao đổi thông tin ngày càng cao.
- Các hệ thống tập trung (Client-Server) truyền thống bộc lộ nhiều hạn chế về **khả năng mở rộng, điểm nghẽn đơn lẻ và chi phí vận hành cao.**
- Hệ thống P2P được chọn làm giải pháp vì tính **phân tán, chịu lỗi và tối ưu tài nguyên.**



A. Tổng quan đề tài và mục tiêu

Mục tiêu chính

- Nghiên cứu và ứng dụng Pekko Framework để xây dựng hệ thống P2P.
- Thiết kế kiến trúc phi tập trung, module hóa, có khả năng mở rộng.
- Triển khai các chức năng cốt lõi: chia sẻ, tìm kiếm và tải tệp.
- Thực hiện demo và đánh giá tính khả thi, hiệu quả của hệ thống

Phạm vi:

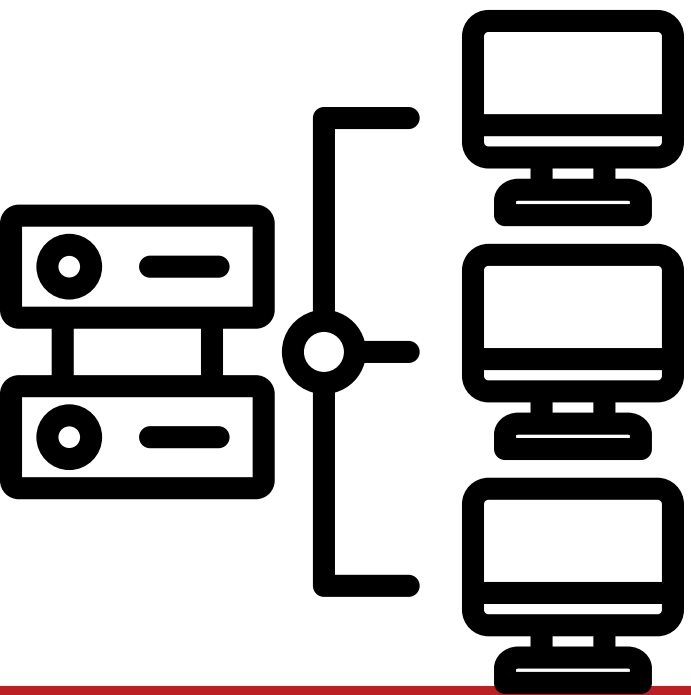
- Tập trung vào các chức năng cơ bản trên nền tảng Pekko, không đi sâu vào các tính năng nâng cao như mã hóa, xác thực hay GUI.



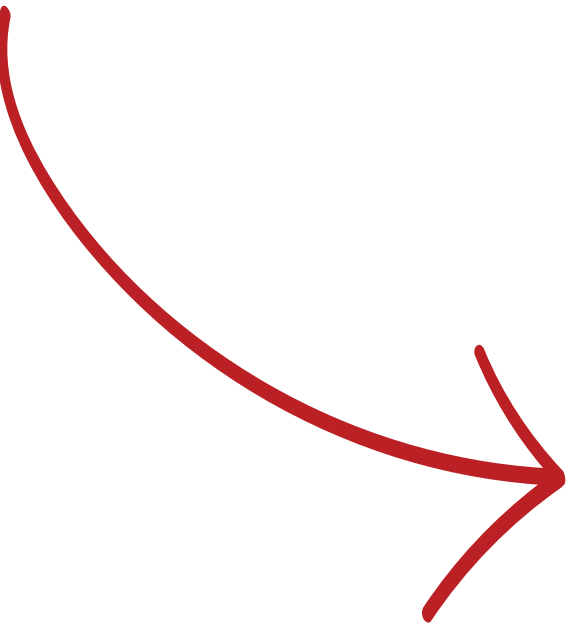
Công nghệ nền tảng

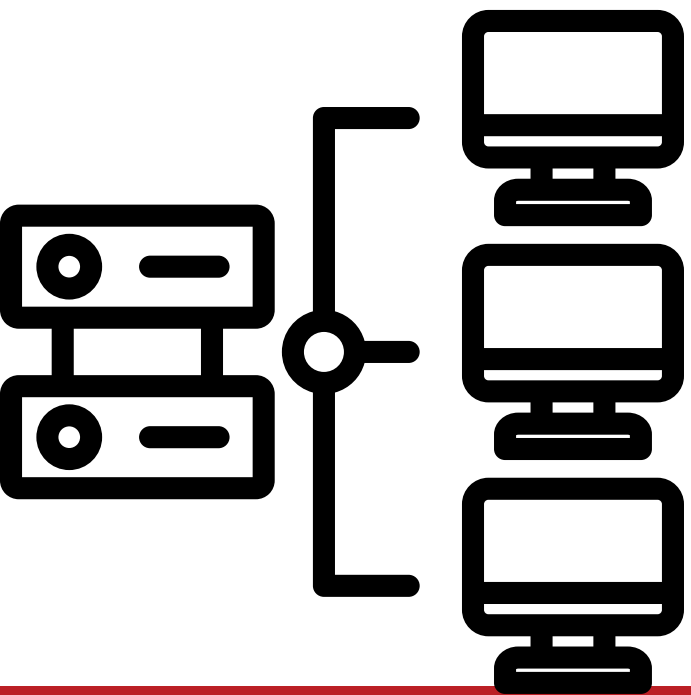
PEKKO FRAMEWORK

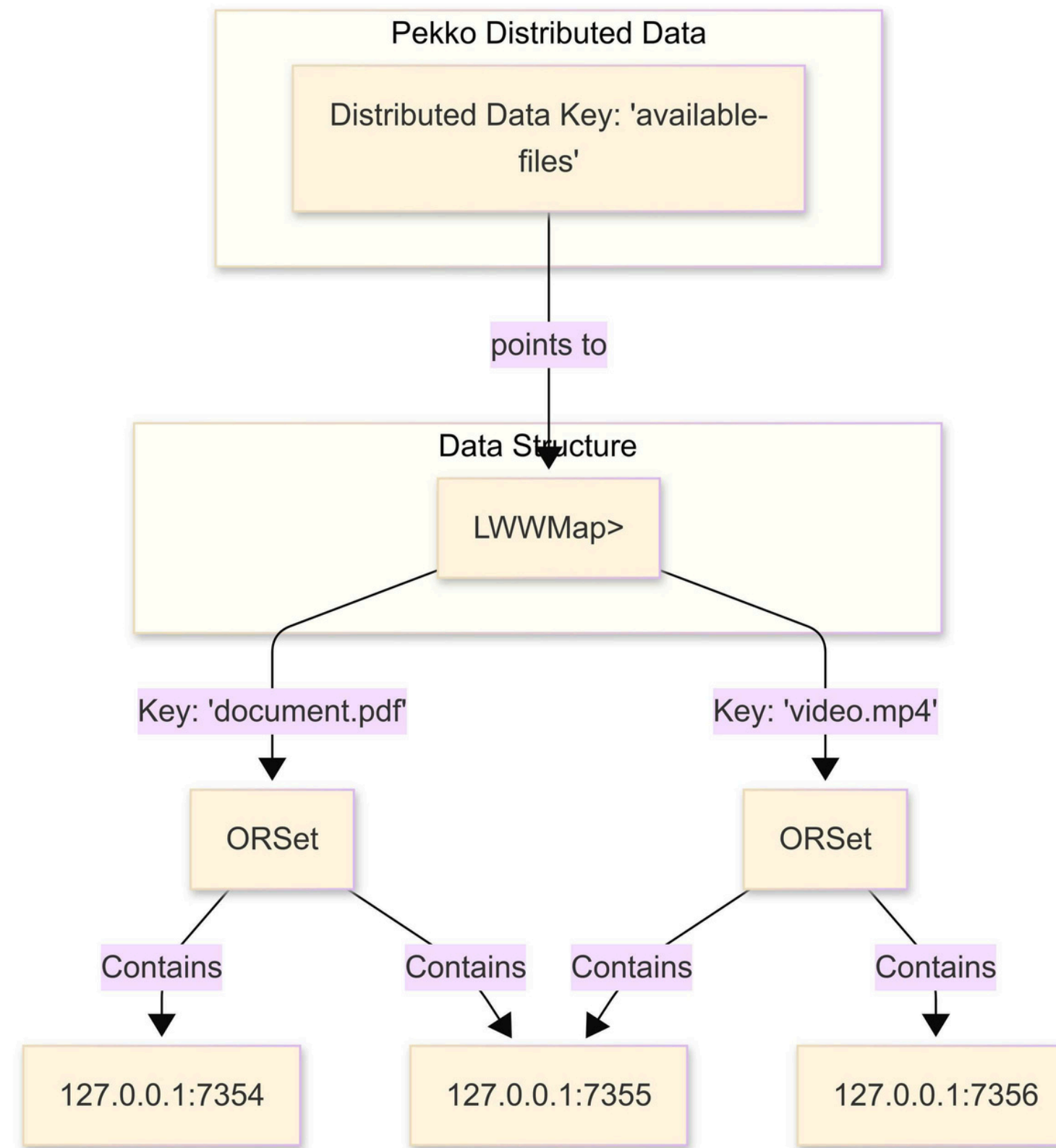
- Một framework mạnh mẽ dựa trên **Actor Model** để xây dựng các ứng dụng phân tán, song song, và có khả năng chịu lỗi.
- **Mô hình Actor**: Là đơn vị tính toán cơ bản, hoạt động độc lập và giao tiếp bằng tin nhắn. Mô hình này giúp đơn giản hóa việc xử lý đồng thời (concurrency) và song song (parallelism).
- **Pekko Cluster**: Quản lý một nhóm các **ActorSystem** hoạt động như một cụm thống nhất. Đây là nền tảng cho việc khám phá peer và quản lý thành viên của mạng P2P.



PEKKO FRAMEWORK

- 
- **Pekko Distributed Data (CRDTs):** Cung cấp các cấu trúc dữ liệu phân tán không yêu cầu đồng bộ tức thì, đảm bảo tính **nhất quán cuối cùng (Eventual Consistency)**. Rất phù hợp để lưu trữ metadata tệp tin (tên, kích thước, hash) một cách phi tập trung.
 - **Pekko Streams:** Một API để xử lý luồng dữ liệu bất đồng bộ. Đây là công cụ lý tưởng để **truyền tải tệp tin lớn** một cách hiệu quả, chia nhỏ tệp thành các chunk và gửi đi một cách tự động



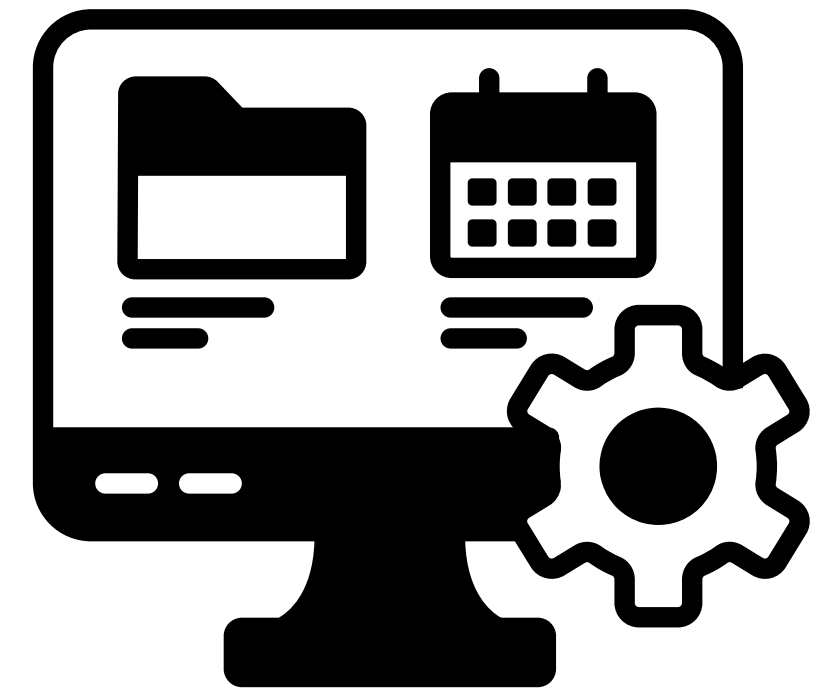


Hình 01: Biểu đồ Cấu trúc dữ liệu phân tán

Kiến trúc hệ thống và các thành phần chính

Tổng quan kiến trúc:

- Kiến trúc **Hybrid P2P** và **Structured P2P**.
- Sử dụng **Pekko Cluster** làm nền tảng cho việc tự động khám phá và gia nhập mạng của các node.
- **Truyền tải tệp tin trực tiếp** giữa các peer (Pure P2P) sau khi đã tìm thấy nhau. Dữ liệu về tệp tin được phân tán và đồng bộ bằng **Pekko Distributed Data**.

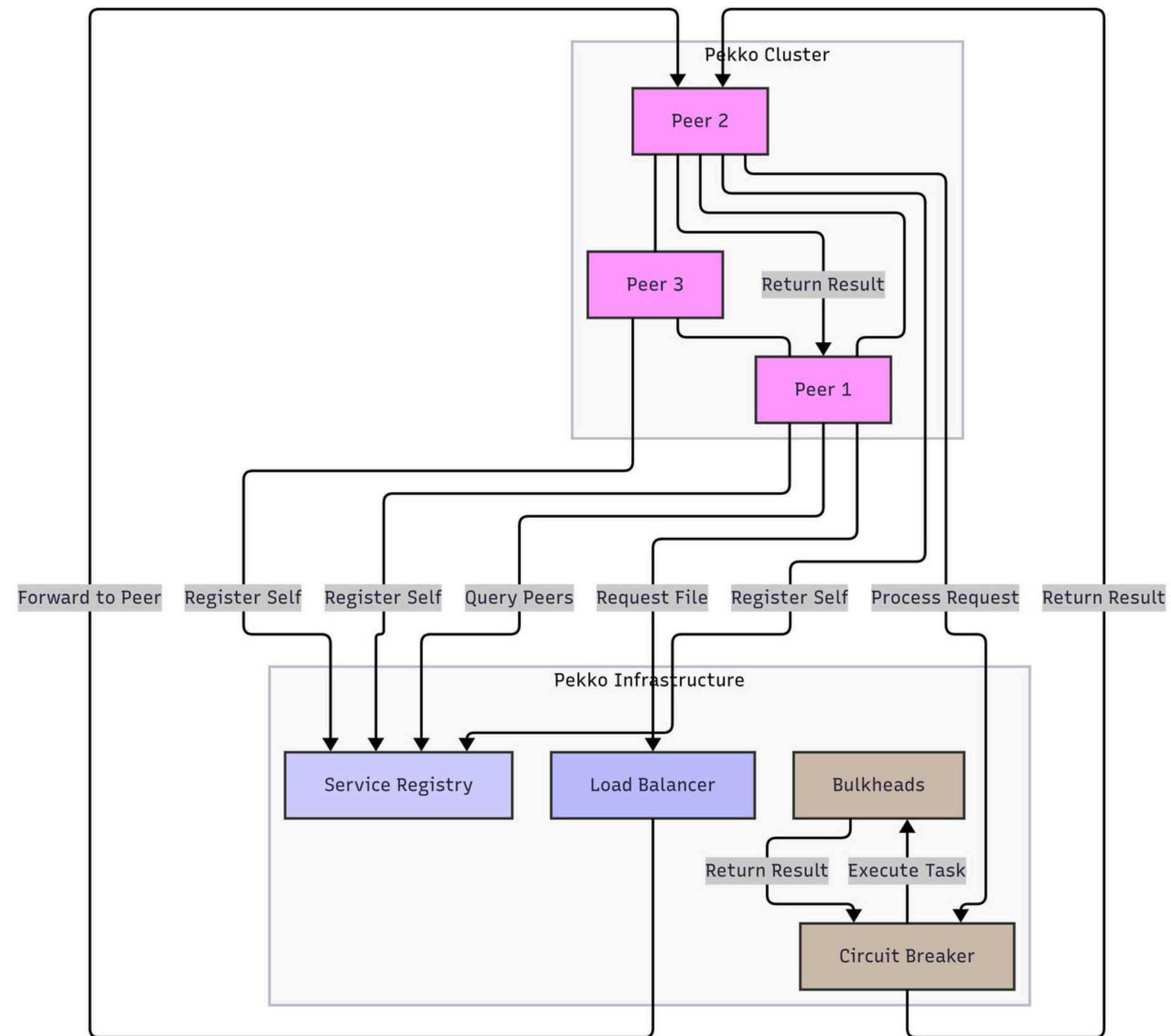


Kiến trúc hệ thống và các thành phần chính

Các thành phần chính:

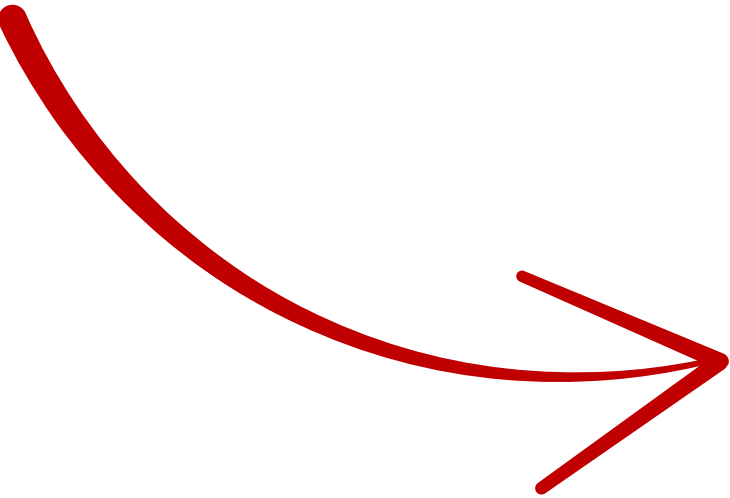
- **Pekko Cluster**: Trái tim của hệ thống, chịu trách nhiệm quản lý mạng lưới P2P.
- **FileShareGuardian**: Actor điều phối chính trên mỗi node, xử lý các yêu cầu từ người dùng và phân luồng công việc.
- **DistributedDataCoordinator**: Sử dụng Pekko Distributed Data để lưu trữ và truy vấn thông tin tệp tin được chia sẻ trên toàn mạng.
- **LocalFileManager**: Quản lý các tệp tin tại node cục bộ, bao gồm việc đọc và ghi tệp.
- **FileTransferActor**: Các worker actors được tạo ra để thực hiện việc tải lên (upload) và tải xuống (download) tệp tin, tận dụng Pekko Streams.

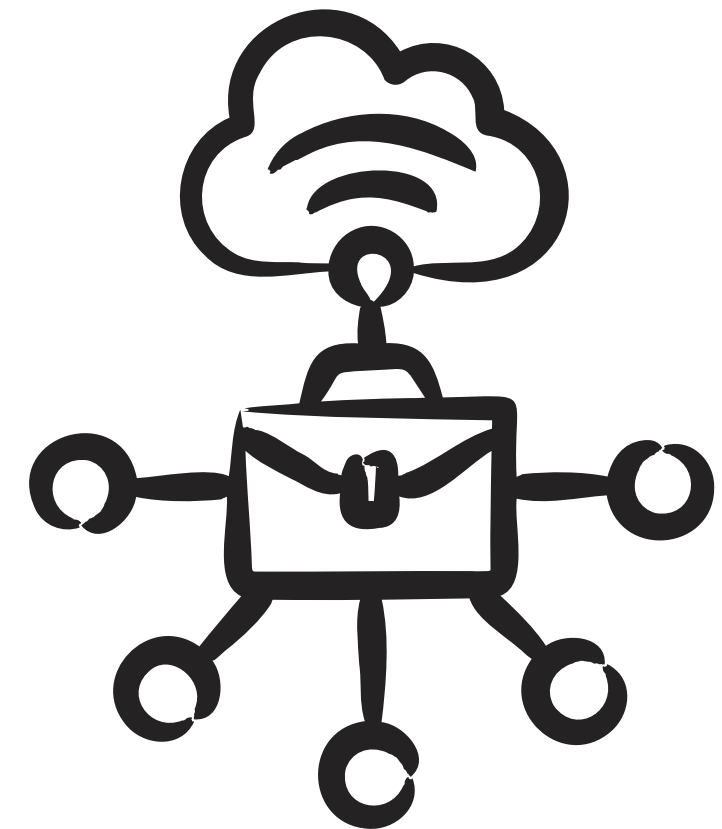


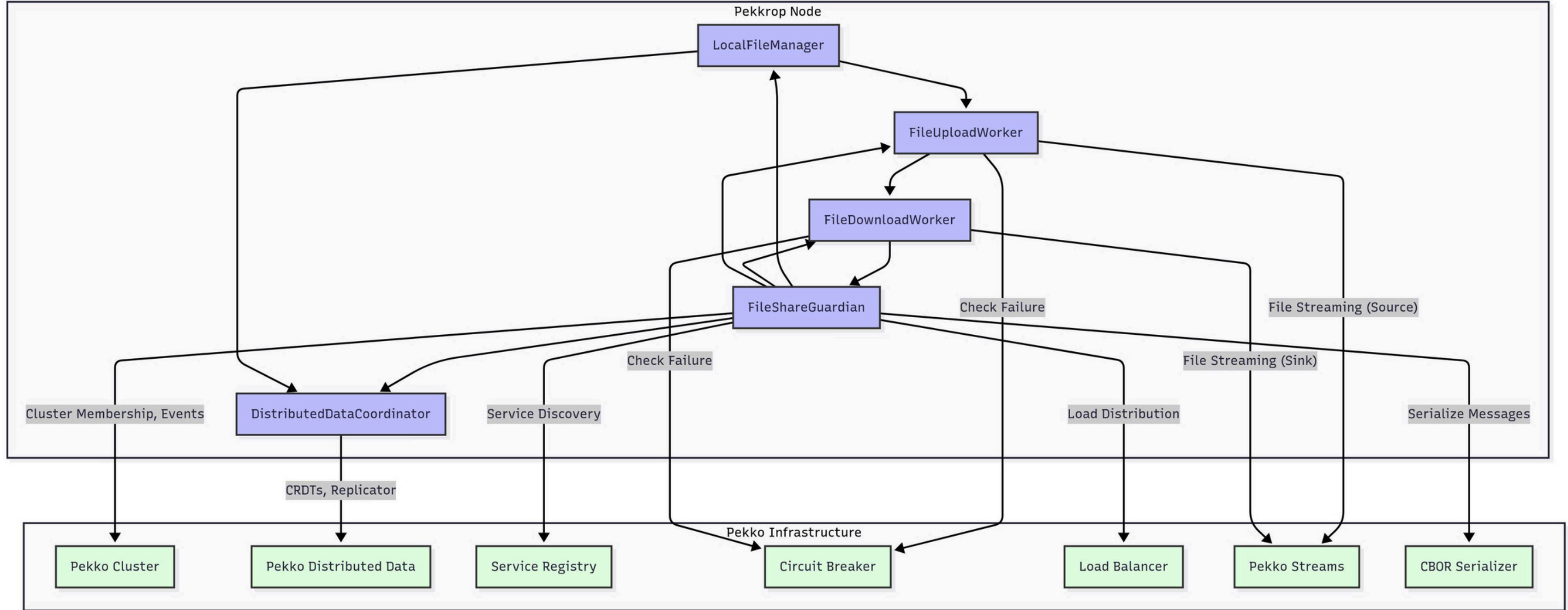


Hình 02: Sơ đồ Kiến trúc tổng thể hệ thống P2P File Sharing

Các mẫu thiết kế phân tán và phụ trợ

- 
- **API Gateway (Phân tán):** Mỗi peer hoạt động như một gateway, định tuyến các yêu cầu nội bộ đến các Actor chuyên trách.
 - **Bulkheads (Vách ngăn):** Cô lập tài nguyên để lỗi ở một thành phần không lan truyền ra toàn hệ thống.
 - **Circuit Breaker (Cầu dao):** Phát hiện và cách ly các dịch vụ từ xa bị lỗi liên tiếp, ngăn chặn các cuộc gọi không cần thiết.

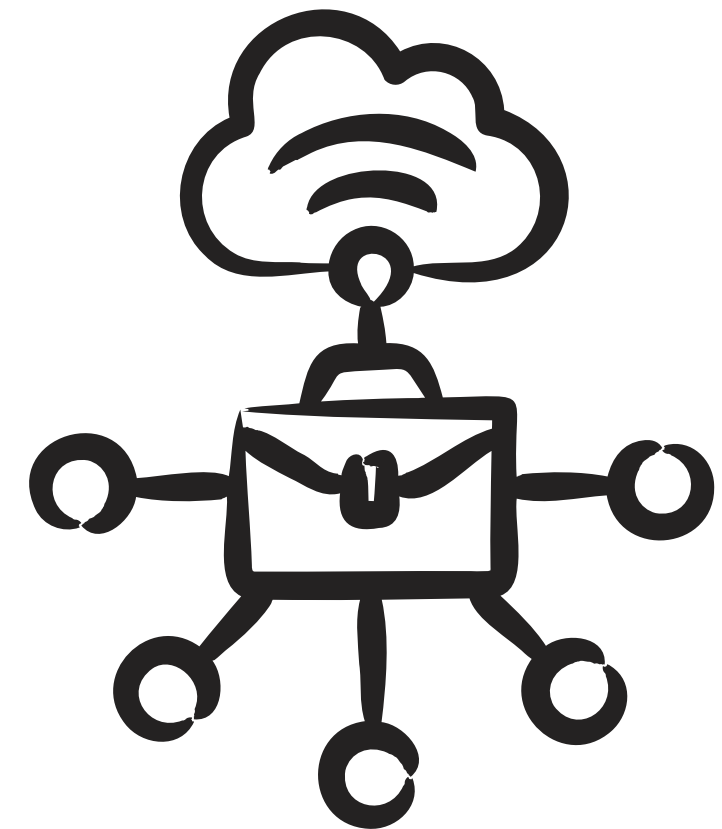


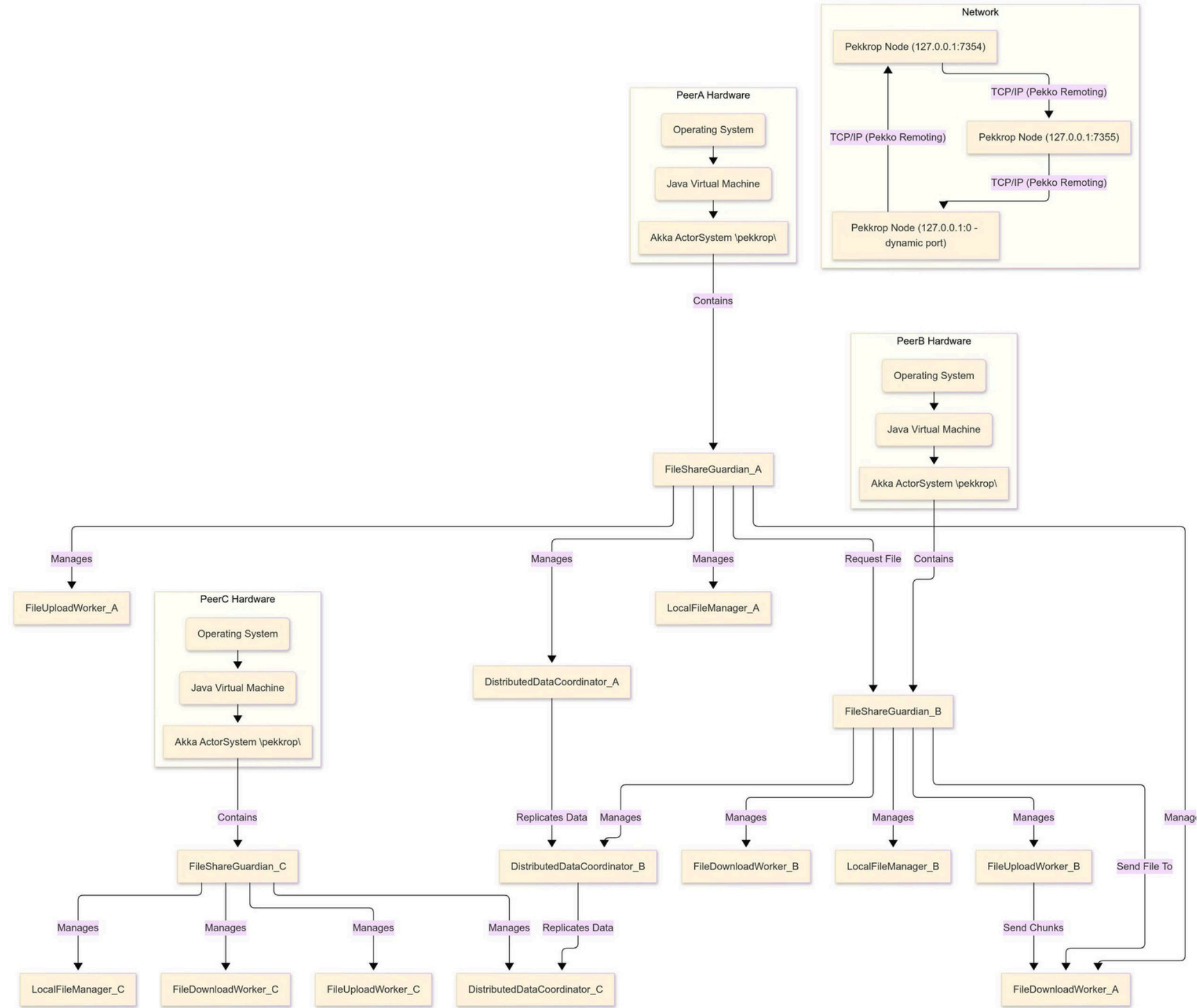


Hình 03: Biểu đồ thành phần hệ thống (Component Diagram)

Các mẫu thiết kế phân tán và phụ trợ

- **Fallback (Dự phòng):** Cung cấp một phản hồi thay thế khi lỗi xảy ra, hoạt động cùng với Circuit Breaker.
- **Load Balancing (Cân bằng tải):** Phân phối yêu cầu tải tập đến các peer đang sở hữu tệp, tối ưu hóa tốc độ.
- **Service Registry (Thư mục dịch vụ):** Được triển khai bằng Pekko Receptionist, cho phép các Actor tự động đăng ký và khám phá lẫn nhau.
- **HOCON:** Định dạng cấu hình linh hoạt, dễ đọc được sử dụng để thiết lập các thông số của Pekko Cluster.





Hình 04: Biểu đồ triển khai hệ thống (Deployment Diagram))

Luồng hoạt động của hệ thống

- **Định nghĩa các Protocol:** Các giao thức được định nghĩa rõ ràng để các Actor có thể giao tiếp hiệu quả. Toàn bộ thông điệp được tuần tự hóa bằng **CBOR (Concise Binary Object Representation)**: Định dạng tuần tự hóa dữ liệu nhị phân nhỏ gọn, được sử dụng để đóng gói các tin nhắn giữa các actors, giúp tiết kiệm băng thông và tăng tốc độ xử lý.
 - **ShareProtocol:** Các lệnh chính mà **FileShareGuardian** xử lý.
 - **DDProtocol:** Giao tiếp với **DistributedDataCoordinator**.
 - **LocalFileProtocol:** Tương tác với **LocalFileManager**.
 - **DownloadProtocol** và **UploadProtocol:** Các lệnh liên quan đến quá trình tải và chia sẻ tệp.



Luồng hoạt động của hệ thống

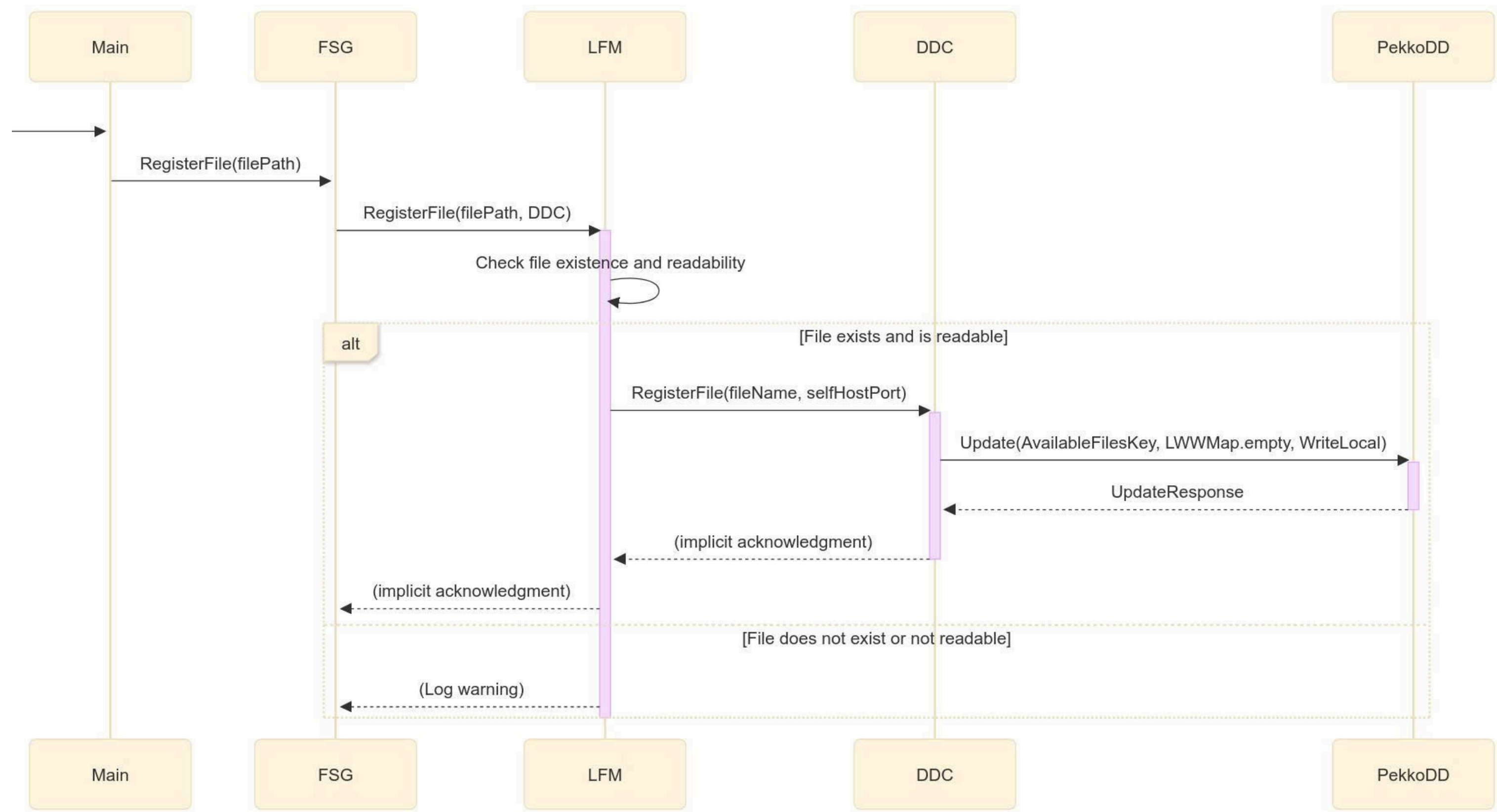
1. Tham gia mạng (Joining the Cluster):

- Một peer khởi động, cấu hình kết nối với các seed node.
- **Pekko Cluster** tự động xử lý quá trình gia nhập và thông báo cho các thành viên khác.

2. Đăng ký và chia sẻ tệp (Registering Files):

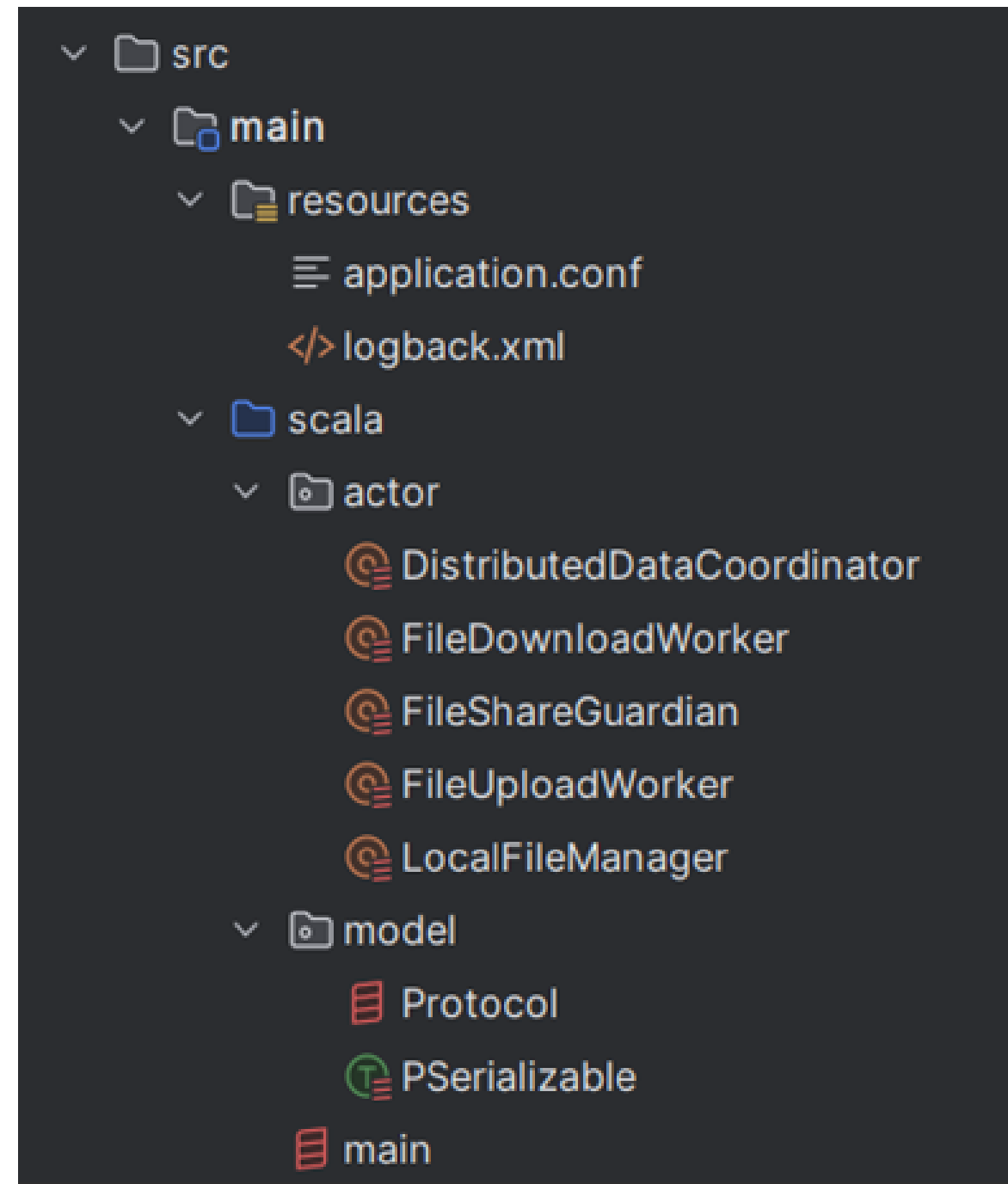
- Người dùng chạy lệnh `register <file_path>`.
- Lệnh được gửi đến **FileShareGuardian**.
- **FileShareGuardian** yêu cầu **LocalFileManager** đọc thông tin tệp, sau đó gửi thông tin này đến **DistributedDataCoordinator**.
- **DistributedDataCoordinator** sử dụng **Pekko Distributed Data** để cập nhật thông tin, sau đó thông tin này sẽ được đồng bộ tới tất cả các node trong mạng.





Hình 04: Biểu đồ tuần tự Biểu đồ tuần tự (Sequence Diagram) cho kịch bản "Yêu cầu tải tệp"

Demo Hệ thống



Hình 05: Cấu trúc thư mục dự án



Demo Hệ thống

- Mục tiêu demo: Minh chứng các luồng hoạt động chính và khả năng chịu lỗi.
- Kịch bản demo: in nghiên k cần cho vào slide
 1. **Khởi động nhiều node:** Cho thấy 3 cửa sổ terminal chạy 3 node P2P khác nhau, các node tự động kết nối và nhận diện nhau.
 2. **Chia sẻ tệp:** Trên Node 1, thực hiện *register data.txt*.
 3. **Kiểm tra tính đồng bộ:** Trên Node 2, thực hiện *list available files*. Chứng minh tệp *data.txt* đã xuất hiện.
 4. **Tải tệp:** Trên Node 3, thực hiện *request data.txt*. Hiển thị quá trình tải và tệp đã tải về thành công.
 5. **Minh chứng tính chịu lỗi:** Trong khi Node 3 đang tải tệp, hãy tắt Node 1. Hệ thống có thể chuyển sang tải từ một node khác (nếu có) hoặc thông báo lỗi một cách có kiểm soát.



CÁM ƠN CÁC BẠN ĐÃ LẮNG NGHE!