**PHÂN TÍCH MÔ HÌNH 3 KHỐI**

Mô hình 3 khối (3-tier architecture) là một kiến trúc phần mềm được sử dụng để xây dựng các ứng dụng phức tạp và hiệu quả. Nó chia ứng dụng thành ba phần chính (khối) với mỗi khối có một nhiệm vụ cụ thể. Dưới đây là phân tích chi tiết về mô hình 3 khối:

* Khối Giao diện người dùng (User Interface Tier):

Khối này là tầng giao diện với người dùng cuối. Nó là nơi người dùng tương tác trực tiếp với ứng dụng.

Nhiệm vụ chính của khối này là hiển thị thông tin cho người dùng và thu thập dữ liệu từ họ thông qua giao diện người dùng (UI).

Đây có thể là ứng dụng web, ứng dụng di động hoặc giao diện đồ họa trên máy tính.

* Khối Logic Ứng dụng (Application Logic Tier):

Khối này chứa logic xử lý chính của ứng dụng. Nó điều phối các yêu cầu từ giao diện người dùng và thực hiện các nhiệm vụ xử lý dữ liệu.

Các nhiệm vụ bao gồm kiểm tra tính hợp lệ của dữ liệu, thực hiện tính toán, truy vấn cơ sở dữ liệu, và nhiều tác vụ xử lý khác.

Đây là nơi logic kinh doanh của ứng dụng được triển khai, và nó không phụ thuộc vào giao diện người dùng cụ thể. Khối này cung cấp sự phân chia giữa giao diện và dữ liệu.

* Khối Dữ liệu (Data Tier):

Khối này chứa dữ liệu của ứng dụng. Dữ liệu có thể được lưu trữ trong cơ sở dữ liệu quan hệ, hệ thống tệp tin, hoặc các hệ thống lưu trữ dữ liệu khác.

Khối này cung cấp các dịch vụ truy cập và quản lý dữ liệu. Các yêu cầu từ khối Logic Ứng dụng được chuyển đến đây để truy xuất hoặc cập nhật dữ liệu.

Đối với ứng dụng có thể mở rộng, có thể sử dụng nhiều hệ thống cơ sở dữ liệu hoặc bộ nhớ cache để cải thiện hiệu suất.

Mô hình 3 khối tách biệt các phần chức năng của ứng dụng thành các lớp độc lập. Điều này có nhiều lợi ích, bao gồm:

* Tính Dễ Quản Lý: Mô hình này tách biệt logic ứng dụng, giao diện người dùng và dữ liệu, làm cho ứng dụng dễ quản lý và bảo trì.
* Tính Mở Rộng: Các khối có thể mở rộng độc lập với nhau, cho phép tăng cường hiệu suất và khả năng mở rộng.
* Tính Bảo Mật: Mô hình này cho phép triển khai biện pháp bảo mật ở mỗi khối một cách riêng lẻ.

Tuy nhiên, nó cũng có nhược điểm, bao gồm sự phức tạp trong việc truyền dữ liệu giữa các khối và đôi khi dẫn đến overhead trong việc phát triển ứng dụng.

