تحقیق درباره معماری‌های میکروسرویس و مونو لیتیک

این تحقیق به بررسی دو معماری نرم‌افزاری رایج، یعنی معماری میکروسرویس و مونو لیتیک، می‌پردازد و تفاوت‌های ساختاری، روش‌های احراز هویت، مقایسه قابلیت‌های مقیاس‌پذیری، هزینه‌های پیاده‌سازی، نگهداری و توسعه، و سایر نکات مهم مانند پیچیدگی، تست، استقرار، خطایابی و وابستگی‌ها را بررسی می‌کند. هدف از این تحقیق، شناخت عمیق‌تر این دو معماری و شناسایی کاربردهای هر کدام در شرایط مختلف است.

# تعریف هر دو معماری و تفاوت‌های ساختاری آنها1.

1.1 معماری میکروسرویس:  
در معماری میکروسرویس، سیستم به مجموعه‌ای از سرویس‌های کوچک تقسیم می‌شود که هرکدام به‌طور مستقل کار می‌کنند و می‌توانند به‌طور جداگانه توسعه داده شوند و مقیاس‌پذیر باشند. این سرویس‌ها معمولاً از طریق API ها با یکدیگر ارتباط دارند.  
1.2 معماری مونو لیتیک:  
در این نوع معماری، تمامی اجزای نرم‌افزار به‌طور یکپارچه در یک پروژه یا برنامه واحد ساخته می‌شوند. تمامی بخش‌های سیستم به هم متصل هستند و تغییرات در یک بخش ممکن است روی کل سیستم تأثیر بگذارد.  
  
1.3 تفاوت‌های ساختاری:  
 - معماری میکروسرویس دارای ساختار غیرمتمرکز است، در حالی که معماری مونو لیتیک متمرکز است.  
 - در میکروسرویس‌ها هر سرویس به‌طور مستقل مقیاس‌پذیر است، اما در مونو لیتیک‌ها کل سیستم باید مقیاس‌پذیر شود.

# 2. روش‌های احراز هویت و مدیریت امنیت در هر معماری

2.1 میکروسرویس:  
احراز هویت معمولاً به‌صورت متمرکز انجام می‌شود و استفاده از ابزارهایی مانند OAuth2 و JWT برای مدیریت امنیت بسیار رایج است. هر میکروسرویس می‌تواند سرویس احراز هویت مستقل داشته باشد، ولی معمولا یک سرویس مرکزی برای این کار اختصاص داده می‌شود.  
  
2.2 مونو لیتیک:  
در این مدل، احراز هویت معمولاً به‌طور یکپارچه در داخل سیستم انجام می‌شود. یک سیستم متمرکز برای تمام احراز هویت‌ها و مجوزهای دسترسی وجود دارد که تمام قسمت‌های سیستم به آن متصل هستند.

# 3. (Scalability)مقایسه قابلیت توسعه‌پذیری

3.1 میکروسرویس:  
توسعه‌پذیری در معماری میکروسرویس بسیار انعطاف‌پذیر است زیرا می‌توان هر سرویس را به‌طور جداگانه مقیاس‌پذیر کرد. اگر یک سرویس تحت فشار باشد، تنها آن سرویس مقیاس می‌یابد، نه کل سیستم.

3.2 مونو لیتیک:  
مقیاس‌پذیری در مونو لیتیک پیچیده‌تر است چون برای مقیاس‌پذیری، کل سیستم باید مقیاس یابد که این کار منابع زیادی می‌طلبد.

# 4. بررسی هزینه‌های پیاده‌سازی، نگهداری و توسعه

4.1 میکروسرویس:  
پیاده‌سازی و نگهداری معماری میکروسرویس معمولاً گران‌تر است، زیرا هر سرویس نیاز به منابع و زمان بیشتری برای توسعه، تست، و نگهداری دارد. همچنین، پیچیدگی‌هایی مانند مدیریت سرویس‌ها و ارتباطات بین آنها وجود دارد.  
  
 4.2 مونو لیتیک:  
پیاده‌سازی مونو لیتیک معمولاً ارزان‌تر است، زیرا توسعه و نگهداری آن به‌صورت متمرکز انجام می‌شود و پیچیدگی‌های ارتباطی بین بخش‌ها کمتر است. اما مقیاس‌پذیری و انعطاف‌پذیری آن محدودتر است.

# 5. سایر نکات مهم

5.1 پیچیدگی:  
معماری میکروسرویس پیچیده‌تر است زیرا نیاز به مدیریت سرویس‌های متعدد و ارتباطات بین آنها وجود دارد. اما مونو لیتیک پیچیدگی کمتری دارد و توسعه‌دهندگان می‌توانند راحت‌تر به سیستم دسترسی داشته باشند.  
  
 5.2 تست:  
تست در معماری میکروسرویس ممکن است چالش‌برانگیز باشد زیرا باید اطمینان حاصل شود که تمام سرویس‌ها به‌طور صحیح با یکدیگر ارتباط دارند. در معماری مونو لیتیک، تست به‌راحتی انجام می‌شود زیرا تمامی کد در یک محیط متمرکز قرار دارد.  
  
5.3 استقرار:  
در میکروسرویس‌ها، استقرار هر سرویس به‌طور جداگانه انجام می‌شود، که می‌تواند منجر به مشکلات هماهنگی و استقرار متعدد شود. در مونو لیتیک، استقرار یکپارچه است و تمامی اجزای سیستم در یک بار استقرار می‌یابند.  
  
5.4 خطایابی:  
خطایابی در معماری میکروسرویس چالش‌برانگیز است زیرا باید از ابزارهای پیچیده برای ردگیری خطاهای سرویس‌های مختلف استفاده کرد. در مونو لیتیک، خطایابی ساده‌تر است زیرا تمامی بخش‌ها در یک سیستم واحد قرار دارند.

# 6. جمع‌بندی کلی و توصیه‌ها

معماری میکروسرویس برای پروژه‌هایی مناسب است که نیاز به مقیاس‌پذیری بالا و انعطاف‌پذیری دارند و می‌توانند پیچیدگی‌های مدیریتی را تحمل کنند. معماری مونو لیتیک برای پروژه‌هایی مناسب است که نیاز به سرعت توسعه و کاهش پیچیدگی دارند، به ویژه در پروژه‌های کوچک تا متوسط که مقیاس‌پذیری بالا ضروری نیست.