Common Libraries vs. Microservices in Spring Boot Applications

Performance

Libraries avoid network calls and improve latency, while microservices introduce network overhead.

Simplicity & Development Ease

Common libraries reduce architectural complexity and allow faster development.

Transaction Management

Libraries handle transactions within the same process, avoiding distributed transaction challenges.

Debugging & Monitoring

Easier to debug and monitor a monolithic application with libraries than a distributed microservices system.

Code Reuse

Libraries encapsulate reusable logic efficiently.

Cost-Effectiveness

Libraries require fewer resources compared to maintaining multiple microservices.

When to Use

Ideal for performance-critical, tightly coupled functionality without independent scaling needs.

Example Use Case

A banking system using a shared authentication library instead of a separate service.

Conclusion

A hybrid approach - leveraging both common libraries and microservices - is often the best solution.