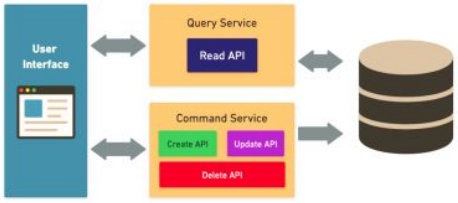
Software Architecture Styles



CQRS Architecture

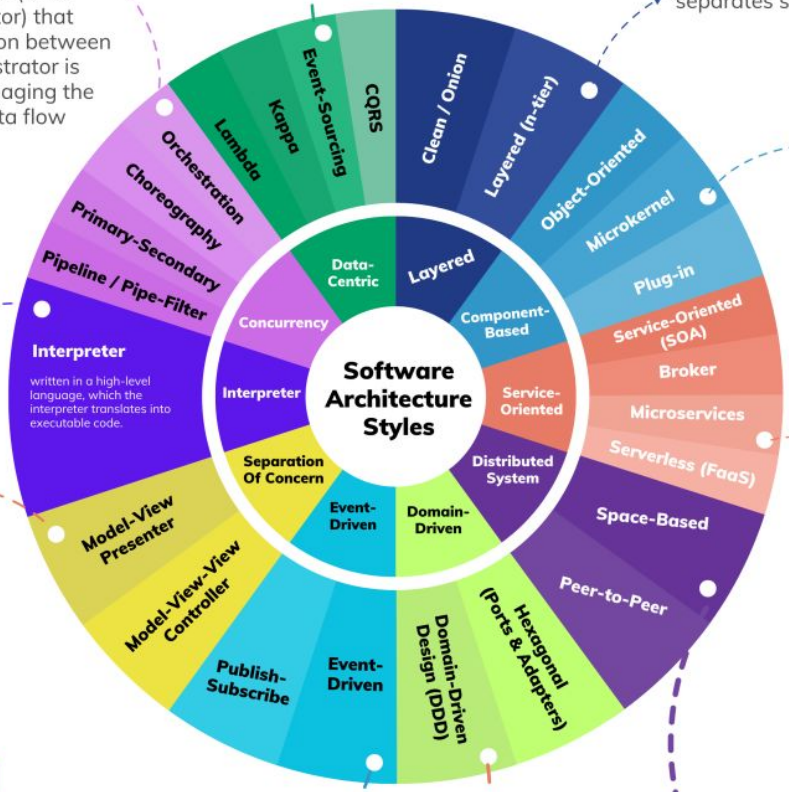
separates read and write operations for a data store. It enables independent scaling of read and write workloads and optimizes them separately.

Orchestration Architecture

a central coordinator (often called an orchestrator) that directs the interaction between services. The orchestrator is responsible for managing the control flow and data flow between services.

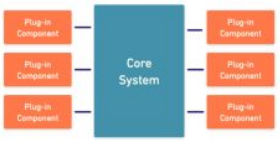
Layered (n-tier) Architecture

separates software into logical layers



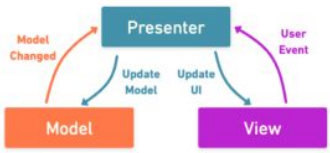
Microkernel Architecture

separates a minimal functional core from extended functionality and customer-specific parts.



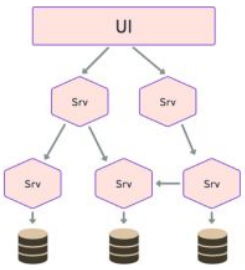
MVP Architecture

derivative of the Model-View-Controller (MVC) pattern, which aims to separate the concerns of data management, user interface, and control flow.



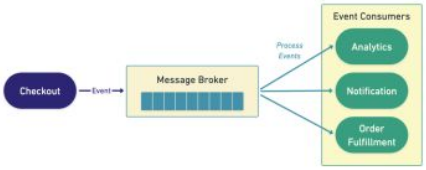
Microservice Architecture

This architecture designs a software application as a suite of independently deployable, small, modular services.



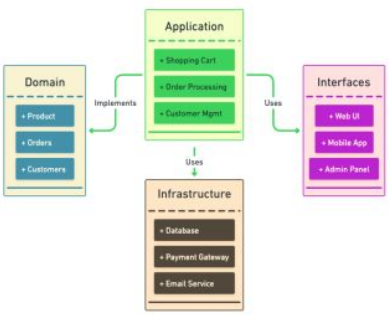
Event-Driven Architecture

promotes the production, detection, consumption of, and reaction to events.



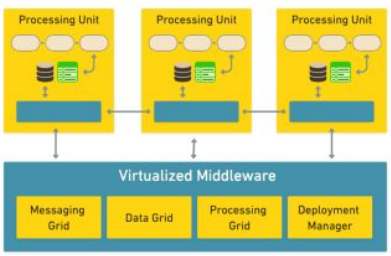
DDD Architecture

focuses on the domain logic and complexity rather than the technology used.



Space-Based Architecture

This resolves the issues of data consistency, reliable performance, and scalability for large-scale distributed systems.



ByteByteGo