		✅	❌
Single node		Consistent data Operational simplicity	Failure-prone Vertical scalability only
Distributed	Strong consistency (Real-time consensus)	Consistent data High-availability	Higher latency Relatively low throughput Harder to implement (assuming you want to roll your own system)
	Eventual Consistency (Deferred consensus)	High-availability Highest throughput Lowest latency Suitable for offline-first apps	State isn’t immediately consistent Possible data losses due to conflicts

		✅	❌
Single node		Consistent data Operational simplicity	Failure-prone Vertical scalability only
Distributed	Strong consistency (Real-time consensus)	Consistent data High-availability	Higher latency Relatively low throughput Harder to implement (assuming you want to roll your own system)
	Eventual Consistency (Deferred consensus)	High-availability Highest throughput Lowest latency Suitable for offline-first apps	State isn't immediately consistent Possible data losses due to conflicts
	Something else?		