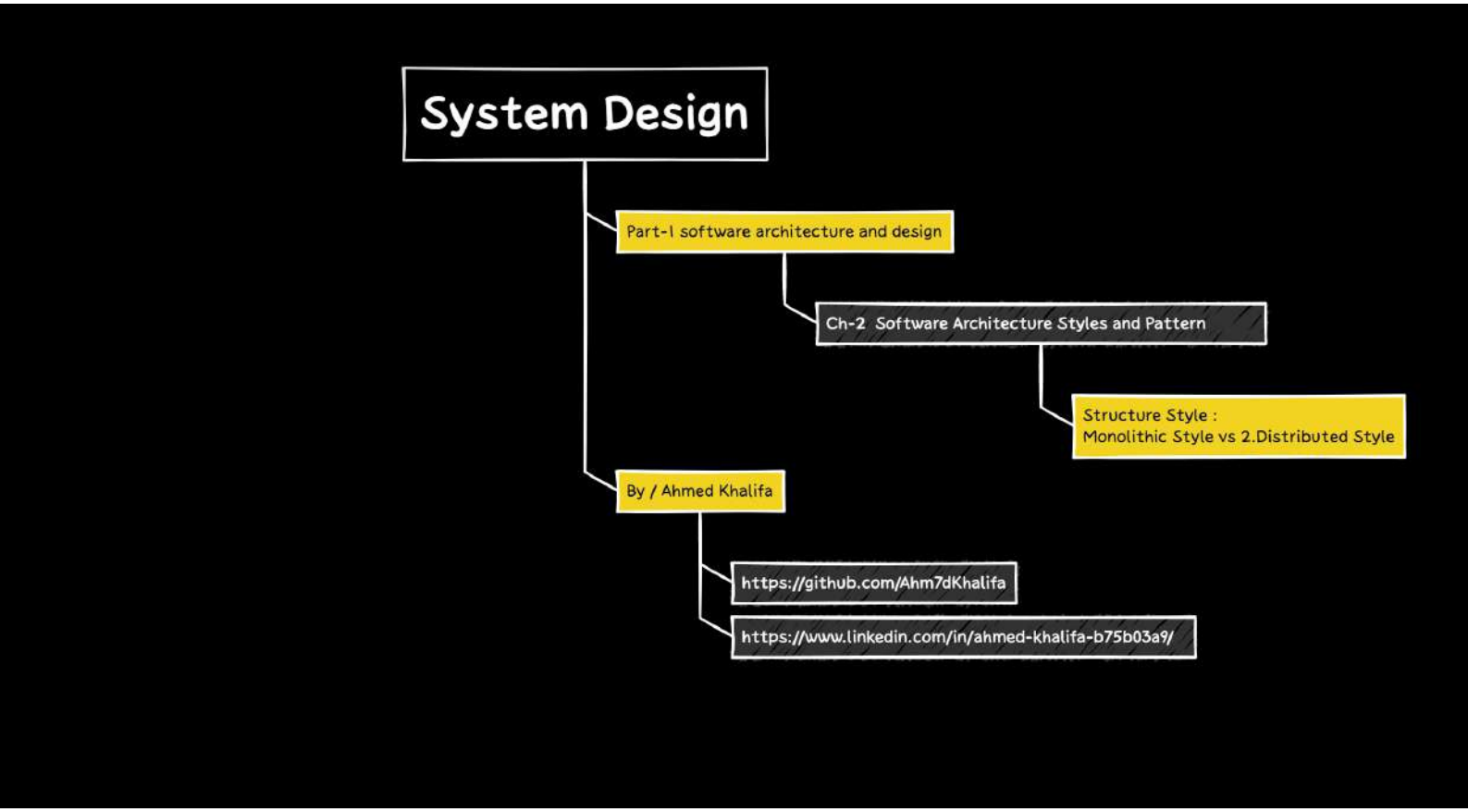
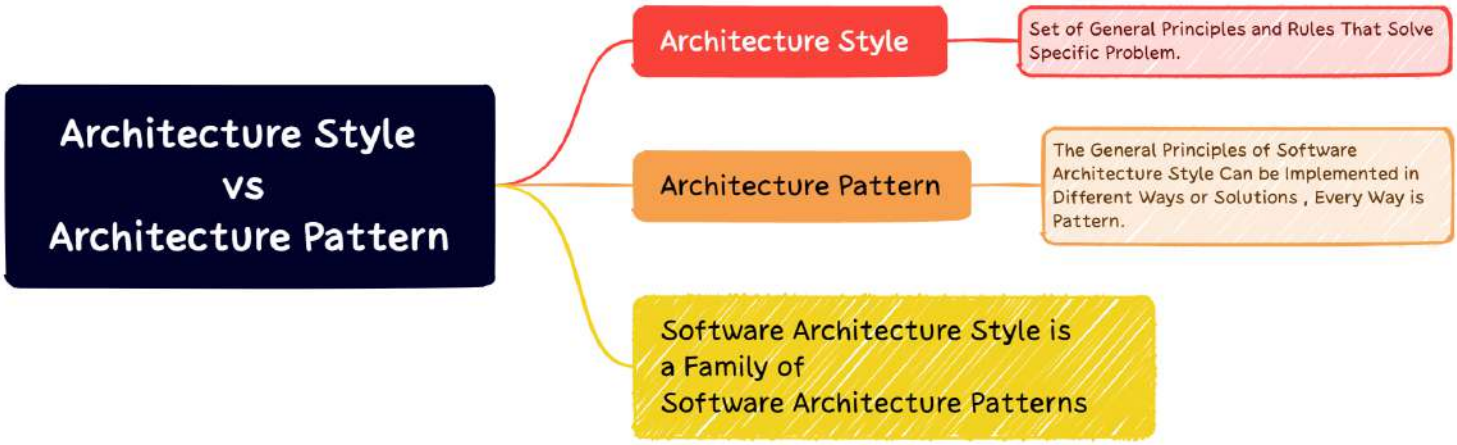
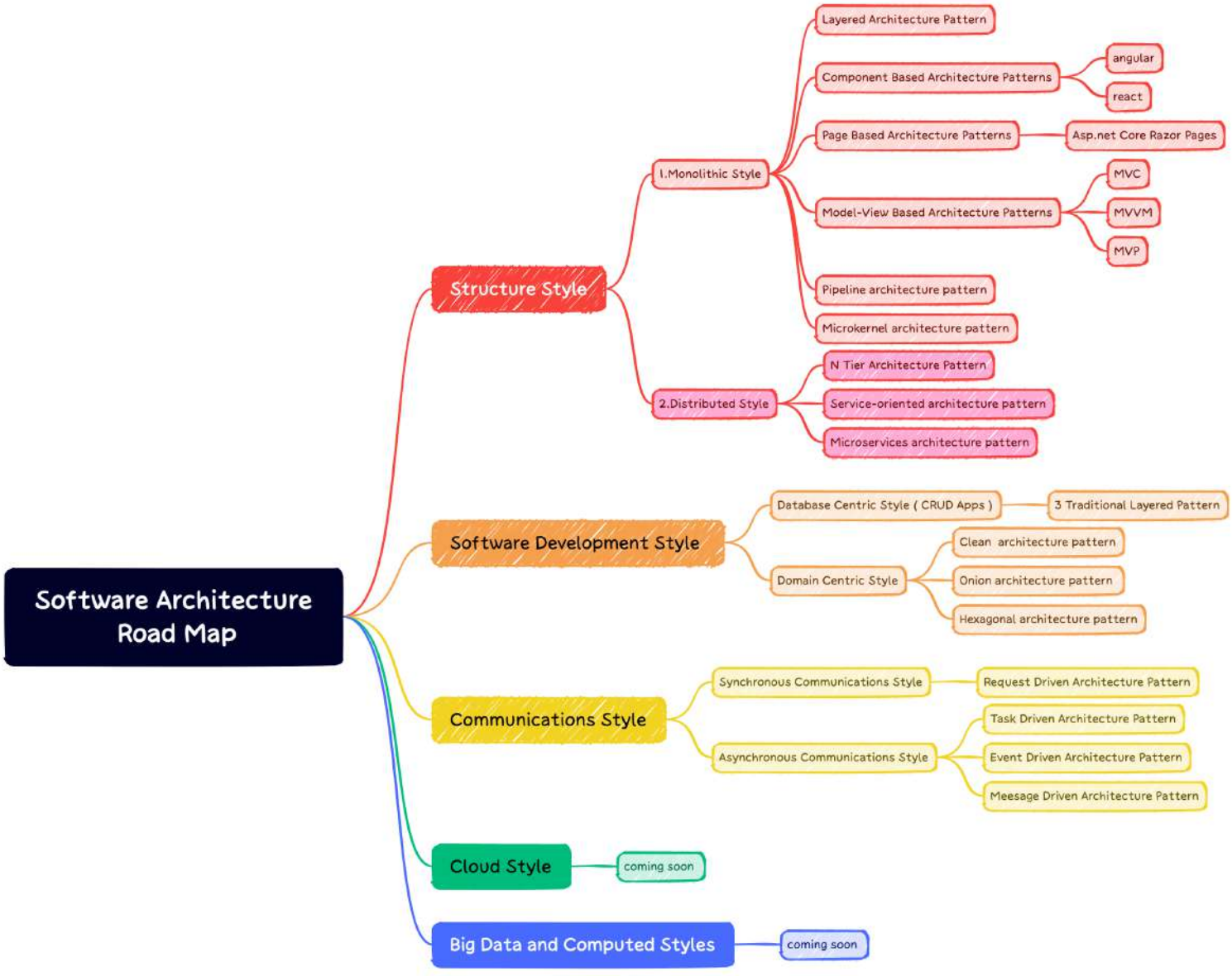
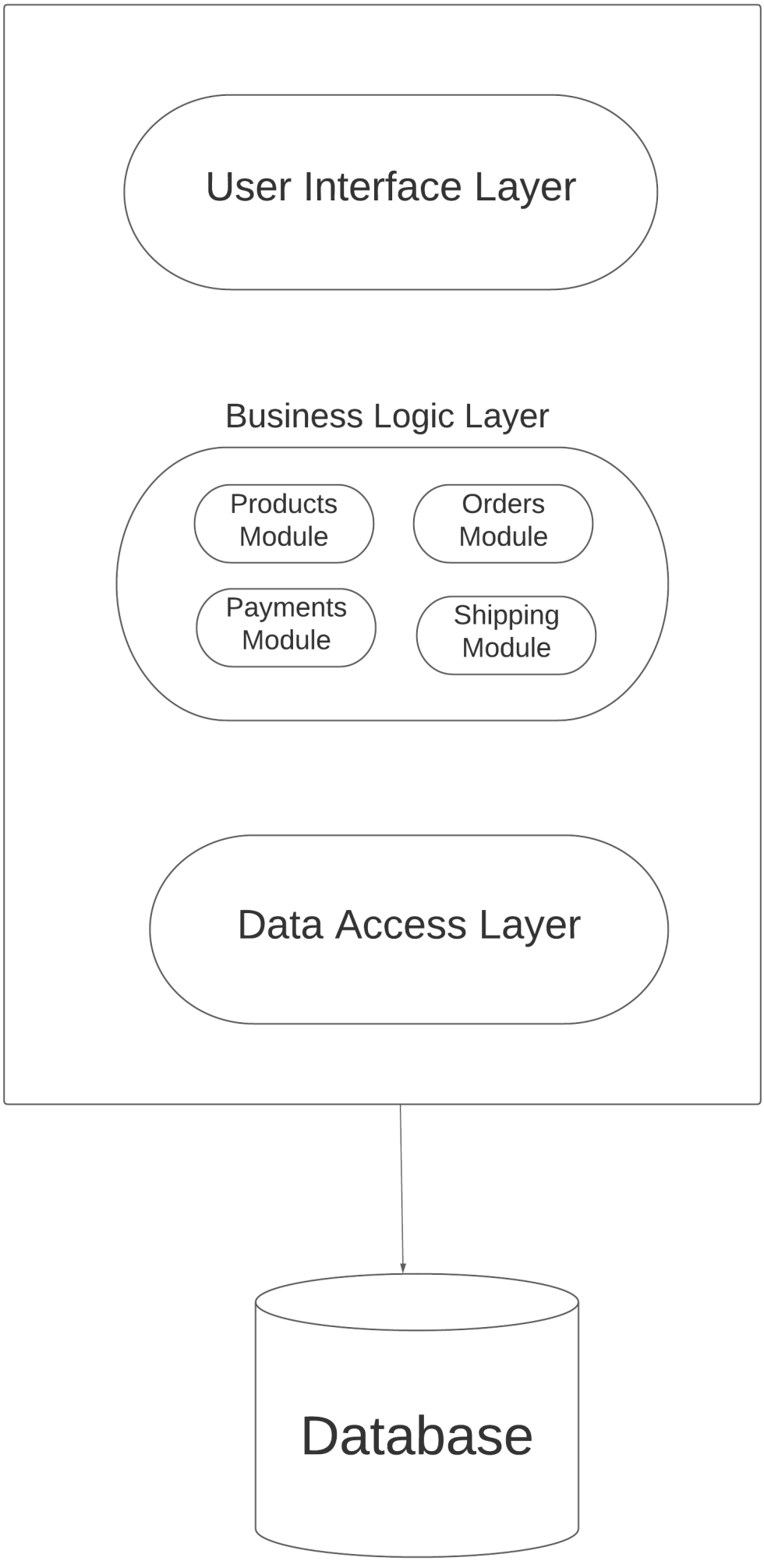
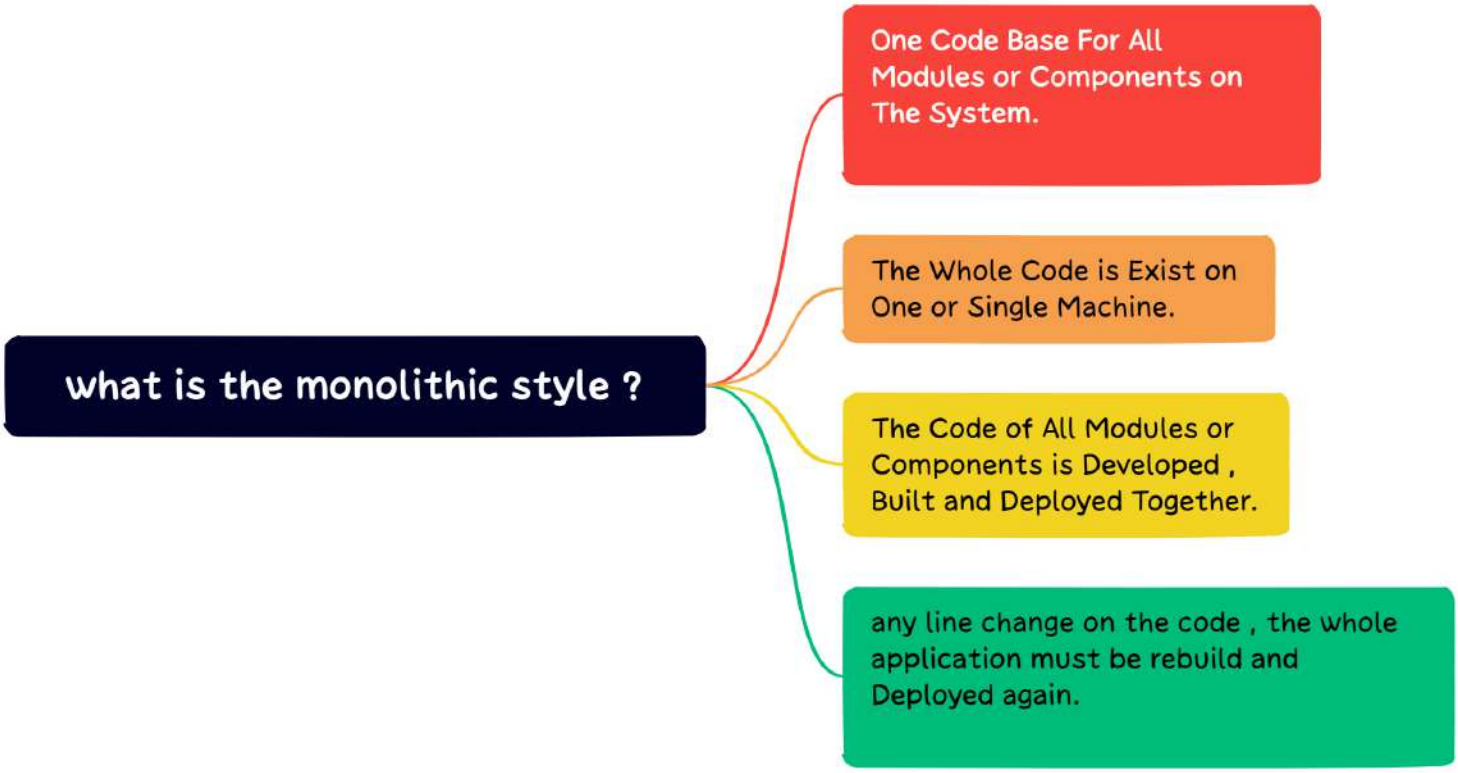

System Design
Part-1 software architecture and design
Ch-2 Software Architecture Styles and Pattern
Structure Style :
Monolithic Style vs 2.Distributed Style
By / Ahmed Khalifa
https://github.com/Ahm7dKhalifa
https://www.linkedin.com/in/ahmed-khalifa-b75b03a9/





monolithic style




what is the monolithic style ?
One Code Base For All Modules or Components on The System.
The Whole Code is Exist on One or Single Machine.
The Code of All Modules or Components is Developed , Built and Deployed Together.
any line change on the code , the whole application must be rebuild and Deployed again.

Advantages and Disadvantages of monolithic Style

	Topic	Advantages	Disadvantages
1	Communications Performance and Latency	- connections between different modules is done at same the machine or code base and this more faster than distributed or microservices architecture which Need to make network call.	
2	Deployment	- easy and simple	- any line change on the code , the whole application must be rebuild and deployed again. - slower : when the size of project increase , the time of deployment increase.
3	Development	- easy to understand when : developer easy to understand the business and code when the project is small.	- complex to understand when : developer has a lot of complexity to understand the business and code when the project is large specially when there are a lot of coupled code and intercommunications between modules or components.

Advantages and Disadvantages of monolithic Style

	Topic	Advantages	Disadvantages
4	Scalability		- can't scale individual components for ex if orders module has a lot of traffic and load than other modules , we can not scale orders module only , we must copy the whole application on another server.
5	Availability		- If there's an error in any module, it could affect the entire application's availability
6	Technology Stack		- Usually Single Technology Stack (one programming language - single database - ...) - very very difficult and complex to adopt any new technology which is well suited for a particular functionality as it affects the entire application, both in terms of time and cost. - for ex :

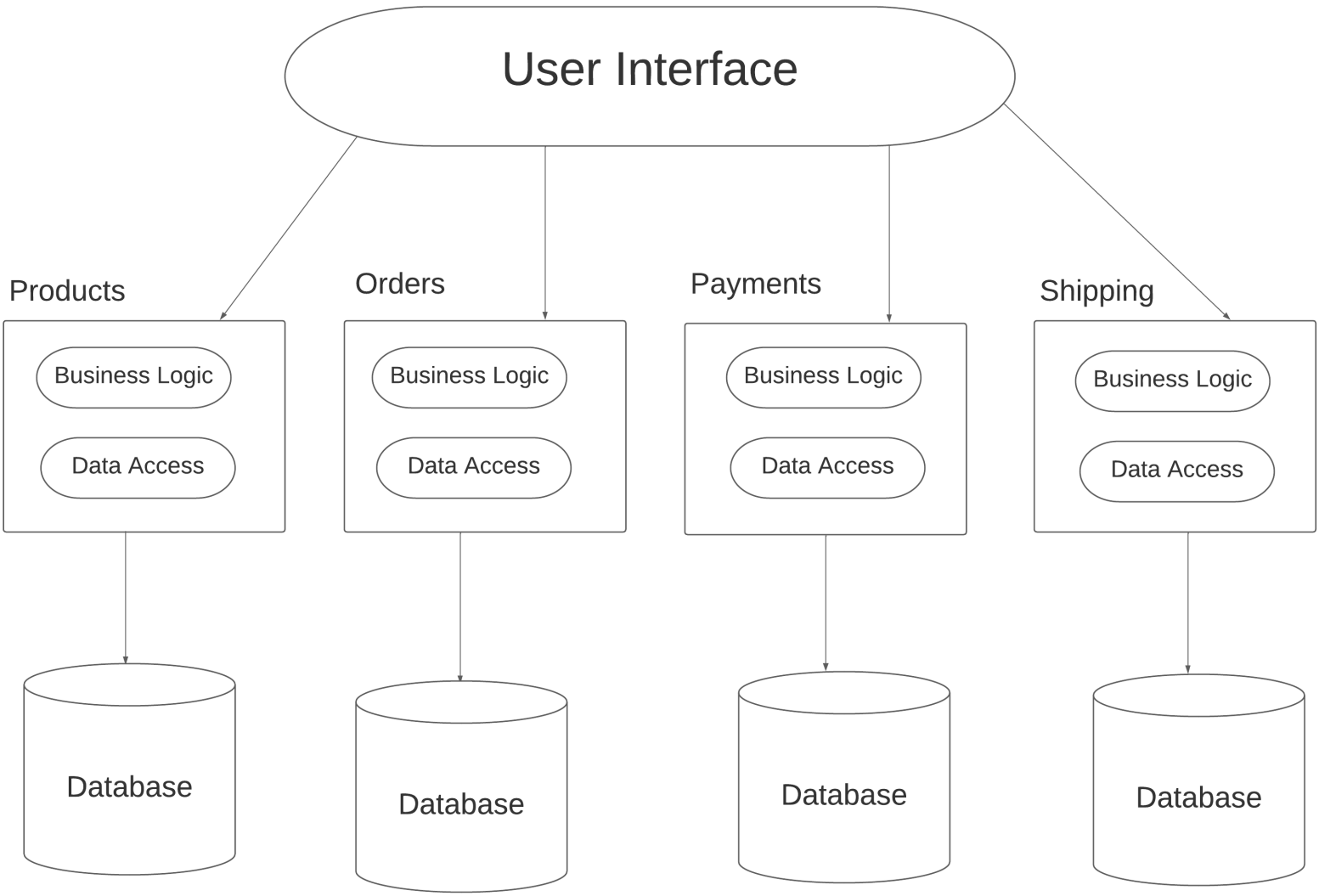
Advantages and Disadvantages of monolithic Style

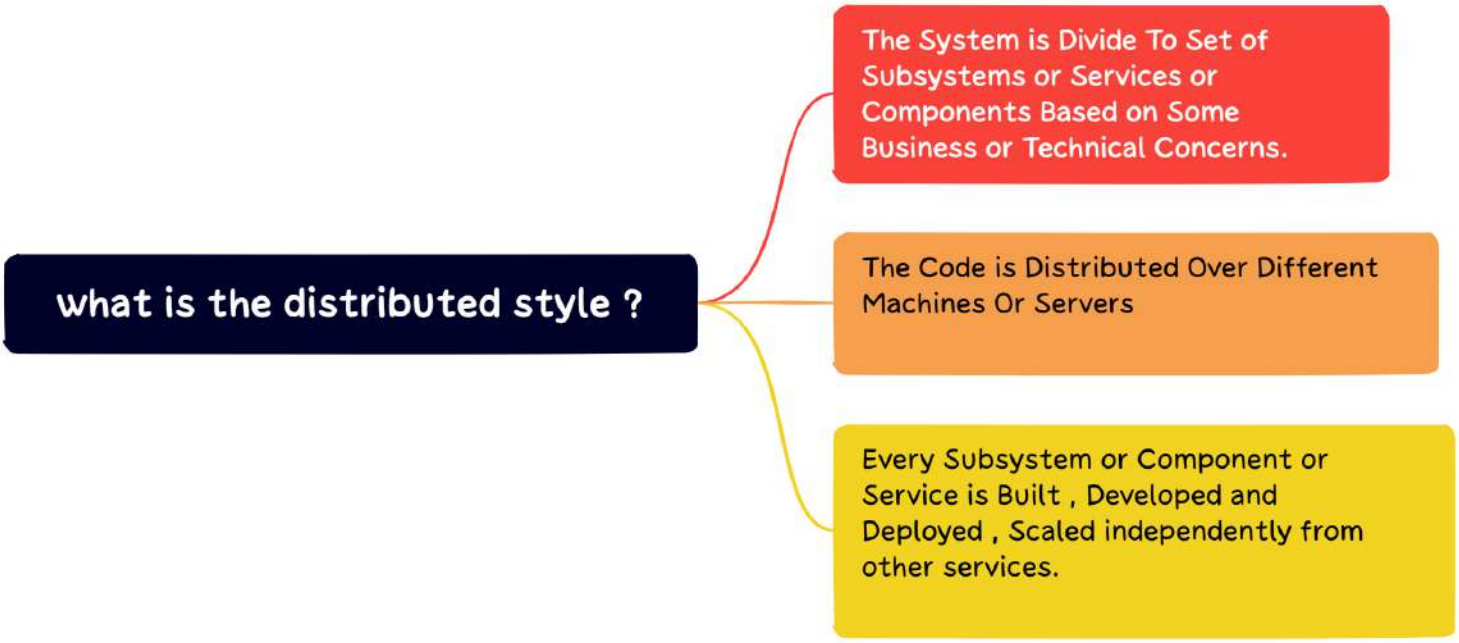
	Topic	Advantages	Disadvantages
7	Simplified testing	Since a monolithic application is a single, centralized unit, end-to-end testing can be performed faster than with a distributed application	
8	Easy debugging	With all code located in one place, it's easier to follow a request and find an issue.	

Type something

Distributed Style

Type something





Advantages and Disadvantages of distributed style

	Topic	Advantages	Disadvantages
1	Communications Performance and Latency		- More Slower Than monolithic : Because There are a networking Call Between Different Services Because Every Service in Different Machine.
2	Deployment	- more complex than monolithic	- faster : when any line change on some service , the system rebuild and deploy only this service , no need to rebuild or redeploy the whole system as monolithic.
3	Development	- Every Team Will Focus Only on Specific Business and Subdomain.	Developers Will Face Challenges Like : - Integration Between Different Services. - Distributed Transactions. - Consistency.

Advantages and Disadvantages of distributed style

	Topic	Advantages	Disadvantages
4	Scalability	Every Service Can Scaled Independently Based on its Traffic Without Needing To Replication The Whole System as Monolithic Architecture	
5	Availability	if one service is down , the others services is still working without any problems	
6	Technology Stack	Every Service Can Have Different Programming Language, Frameworks , Database , Architecture Pattern	

Advantages and Disadvantages of distributed style

	Topic	Advantages	Disadvantages
7	Debugging		- More Complex single business process can run across multiple machines or services , further complicating testing.
8	Testing		- More Complex single business process can run across multiple machines or services , further complicating testing.