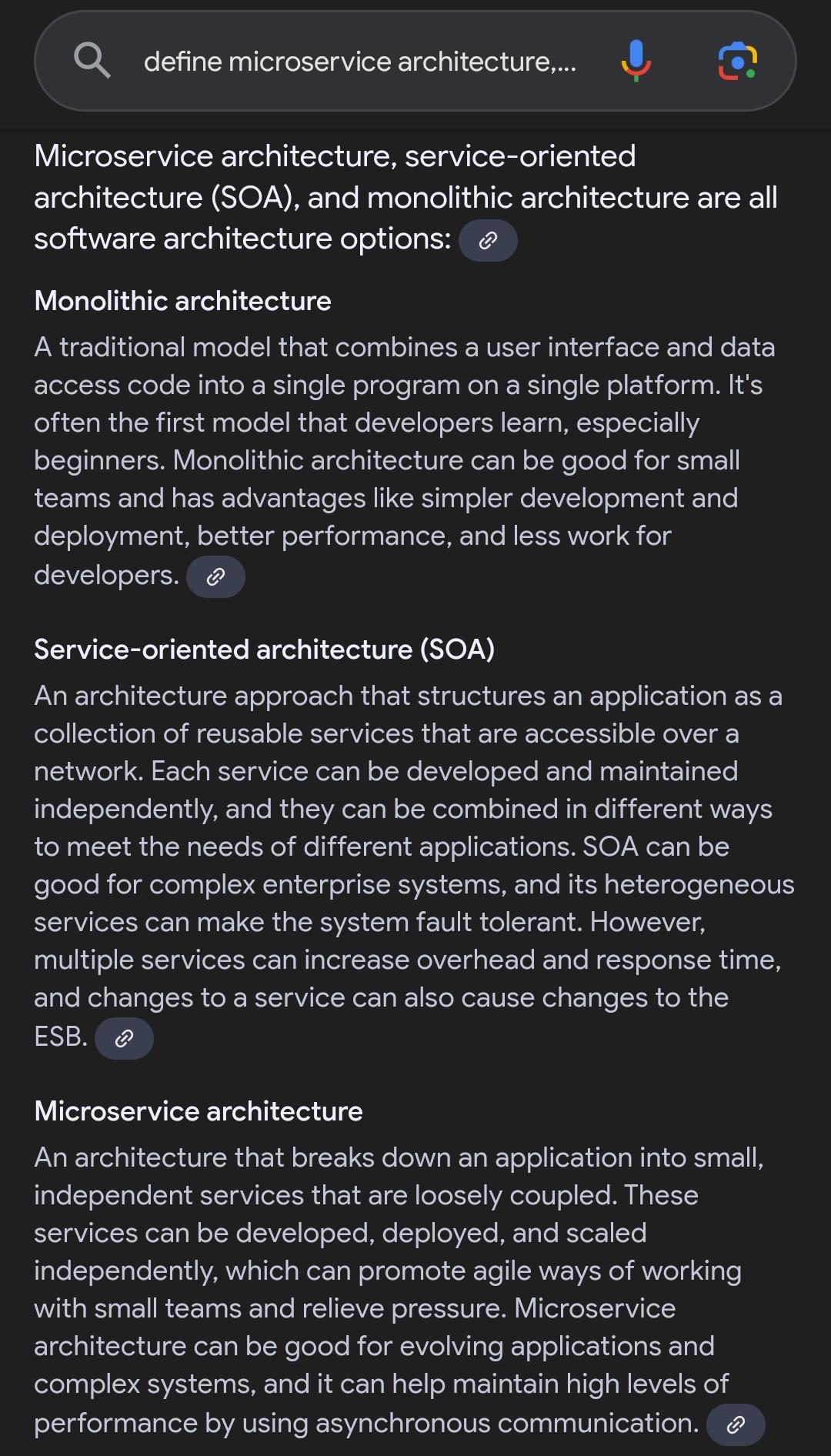
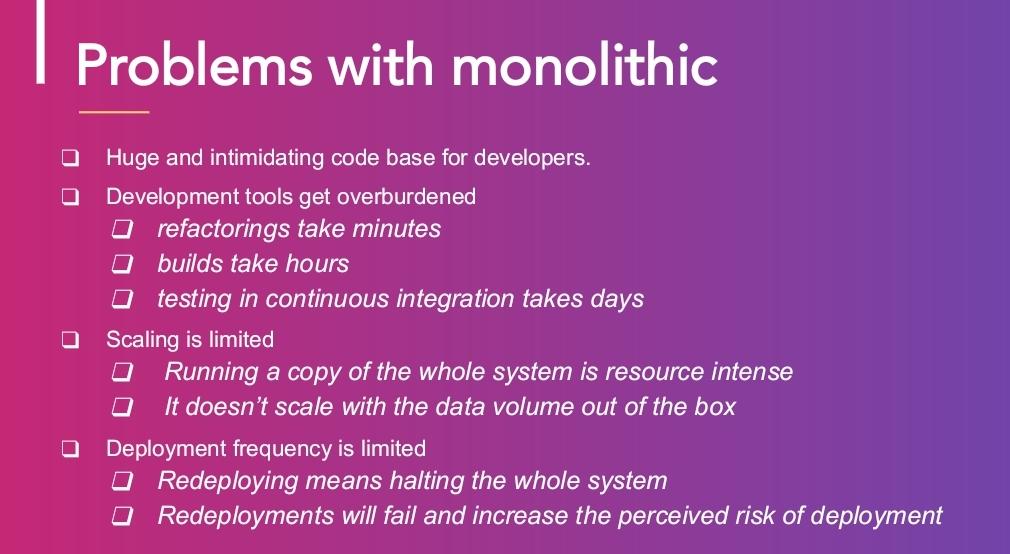
| **Unit** | **Topics** | **Hours** |
| --- | --- | --- |
| **I**  **II** | **Microservices:** Understanding Microservices, Adopting Microservices, The Microservices Way.  **Microservices Value Proposition:** Deriving Business Value, defining a Goal-Oriented, Layered Approach, Applying the Goal-Oriented, Layered Approach.  **Designing Microservice Systems:** The Systems Approach to Microservices.  **Establishing a Foundation**: Goals and Principles, Platforms, Culture.  **Service Design:** API design, Data and Microservices, Distributed Transactions and Sagas.  **System Design and Operations:** Independent Deployability, More Servers, Docker and Microservices, Role of Service Discovery, Need for an API Gateway, Monitoring and Alerting.  **Adopting Microservices in Practice:** Solution Architecture Guidance, Organizational Guidance, Culture Guidance, Tools and Process Guidance, Services Guidance. | **10**  **10** |
| **III** | **ASP.NET Core Primer:** Introduction, Installing .NET Core, Building a Console App, Building ASP.NET Core App.  **Delivering Continuously:** Introduction to Docker, Continuous integration  with Wrecker, Continuous Integration with CircleCI, Deploying to Docker Hub.  **Building Microservice with ASP.NET Core:** Microservice, Team Service, API First Development, Test First Controller, creating a CI pipeline, Integration Testing, Running the team service Docker Image. | **10** |

Chp 1. The Microservices Way

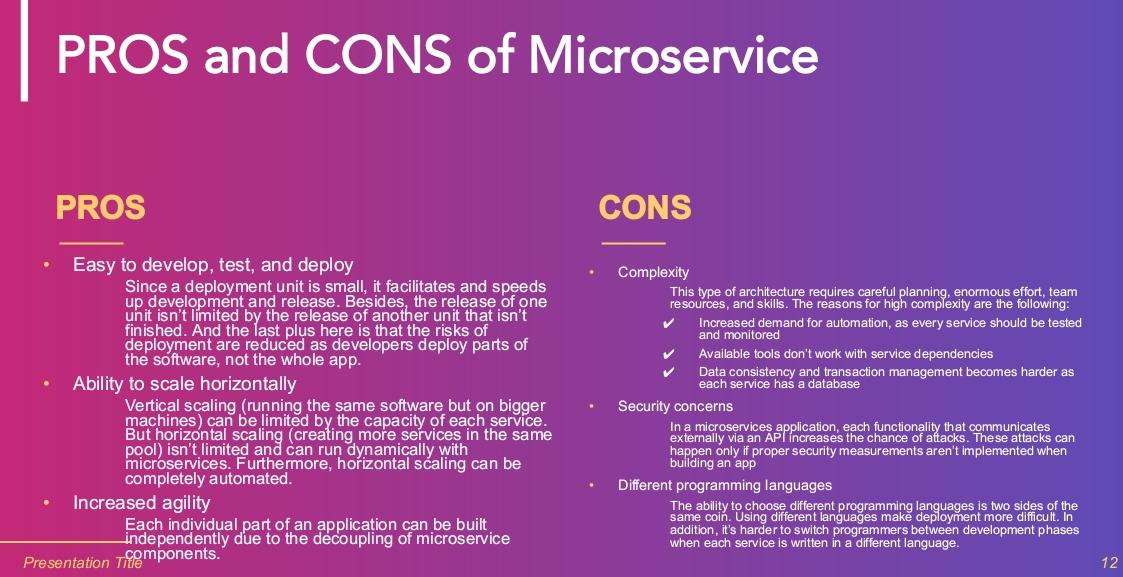
Q. Difference between MA and Monolithic architecture

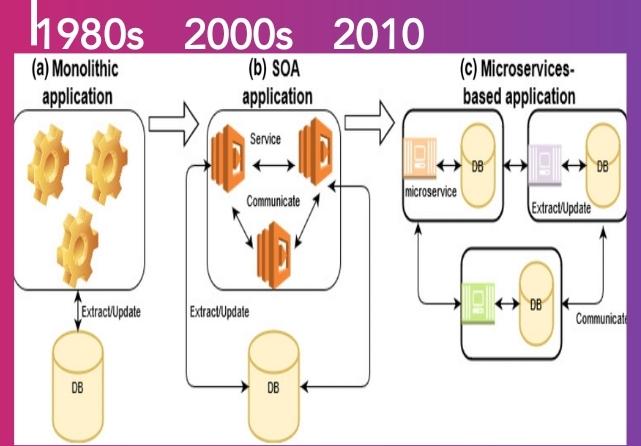




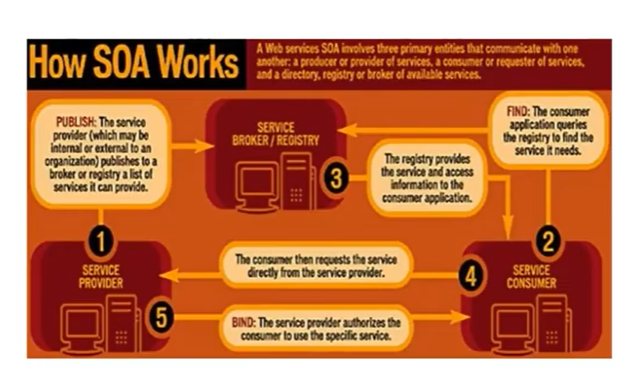




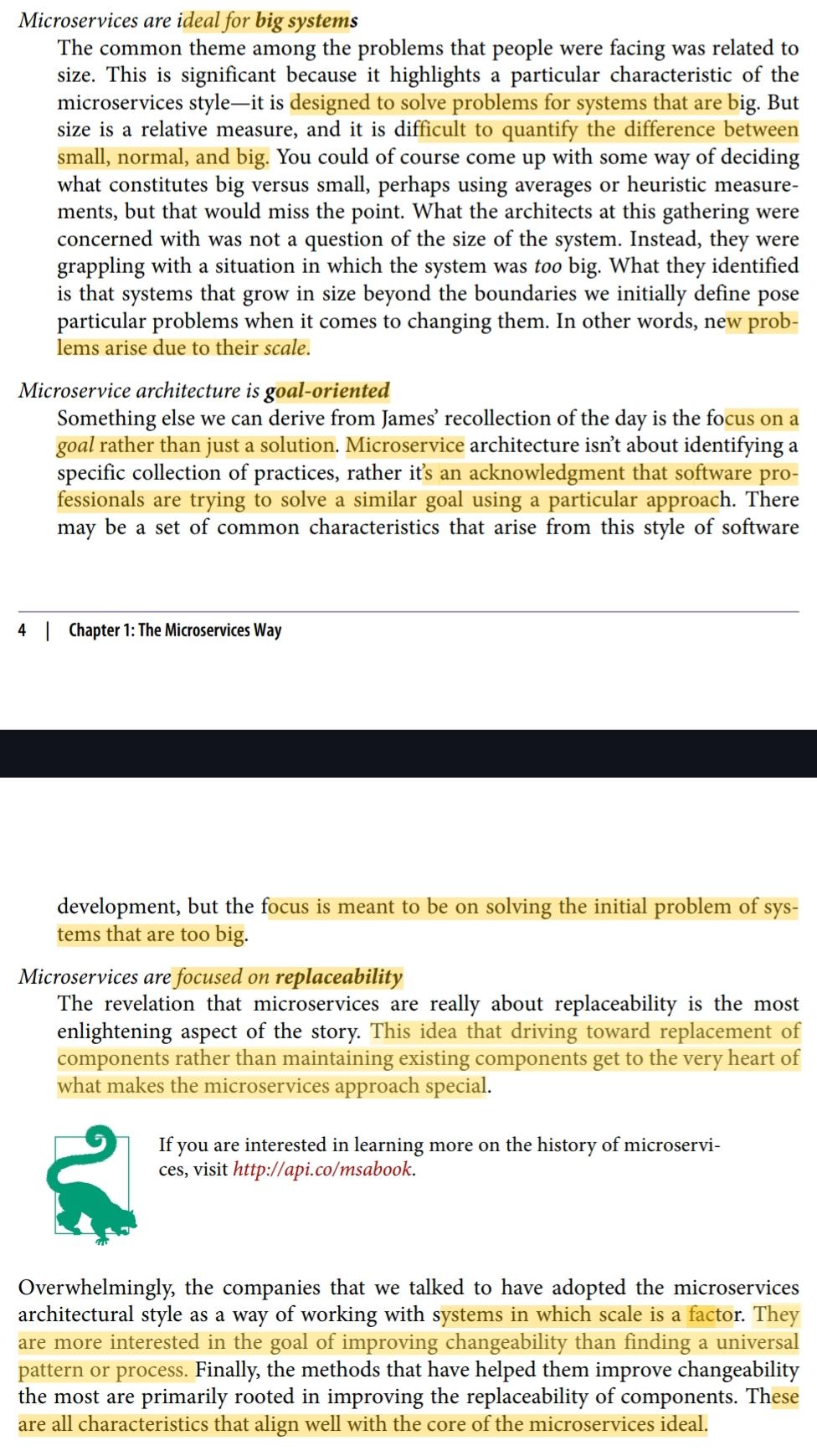




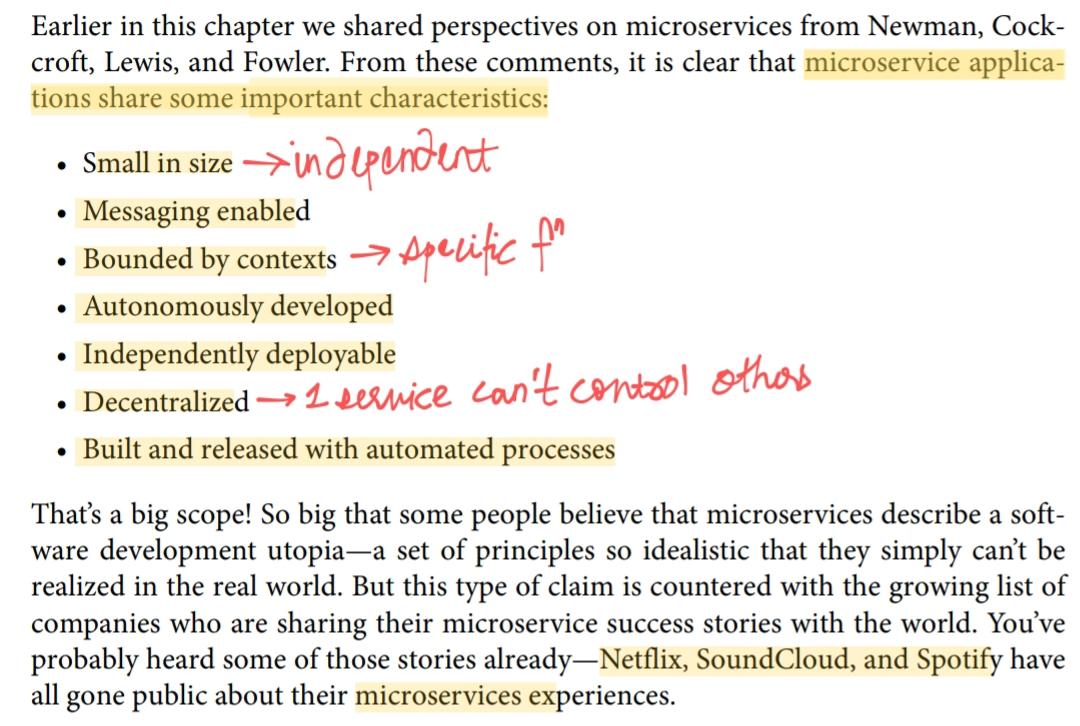
Q. SOA



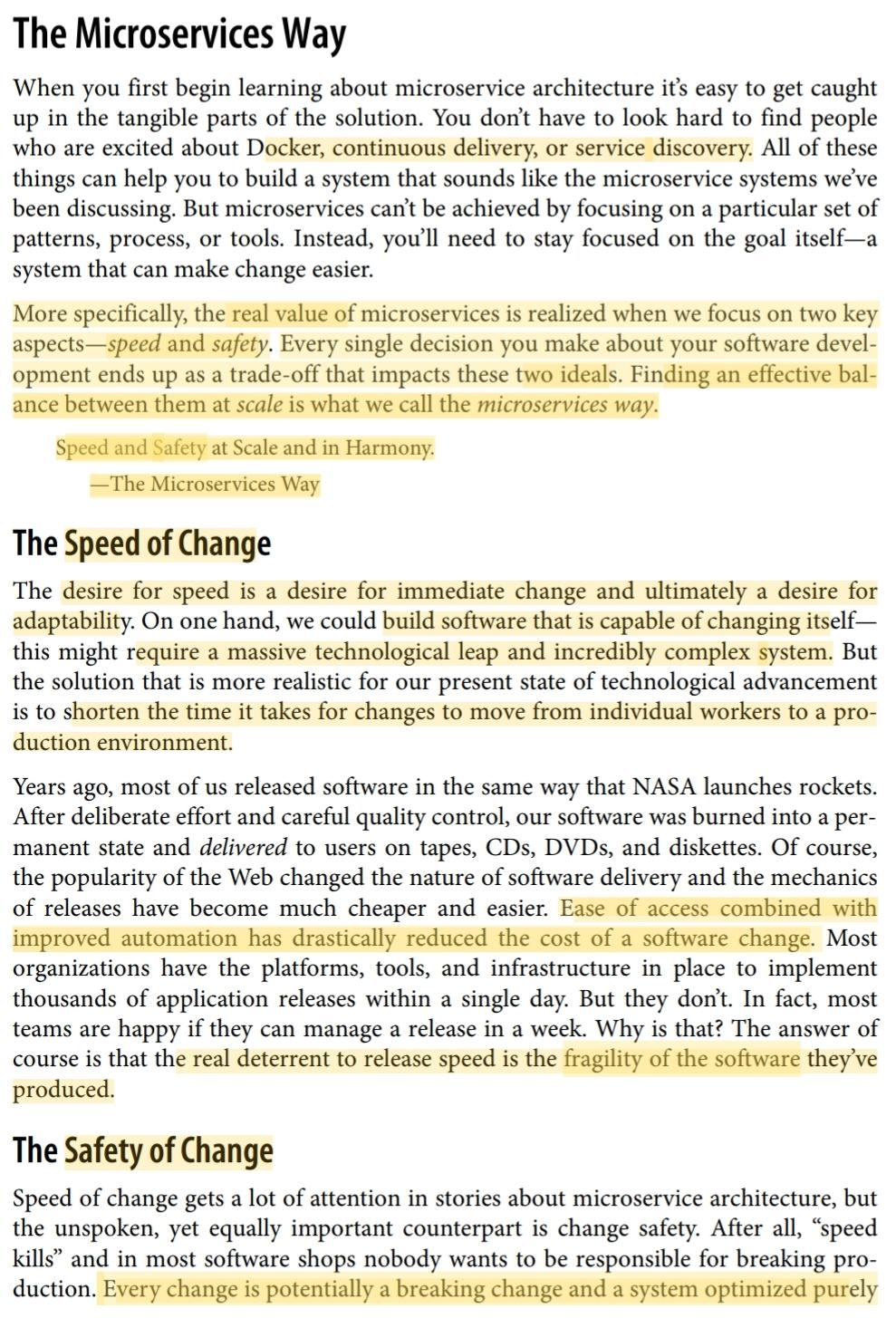
Q. Why do we need Microservice architecture?

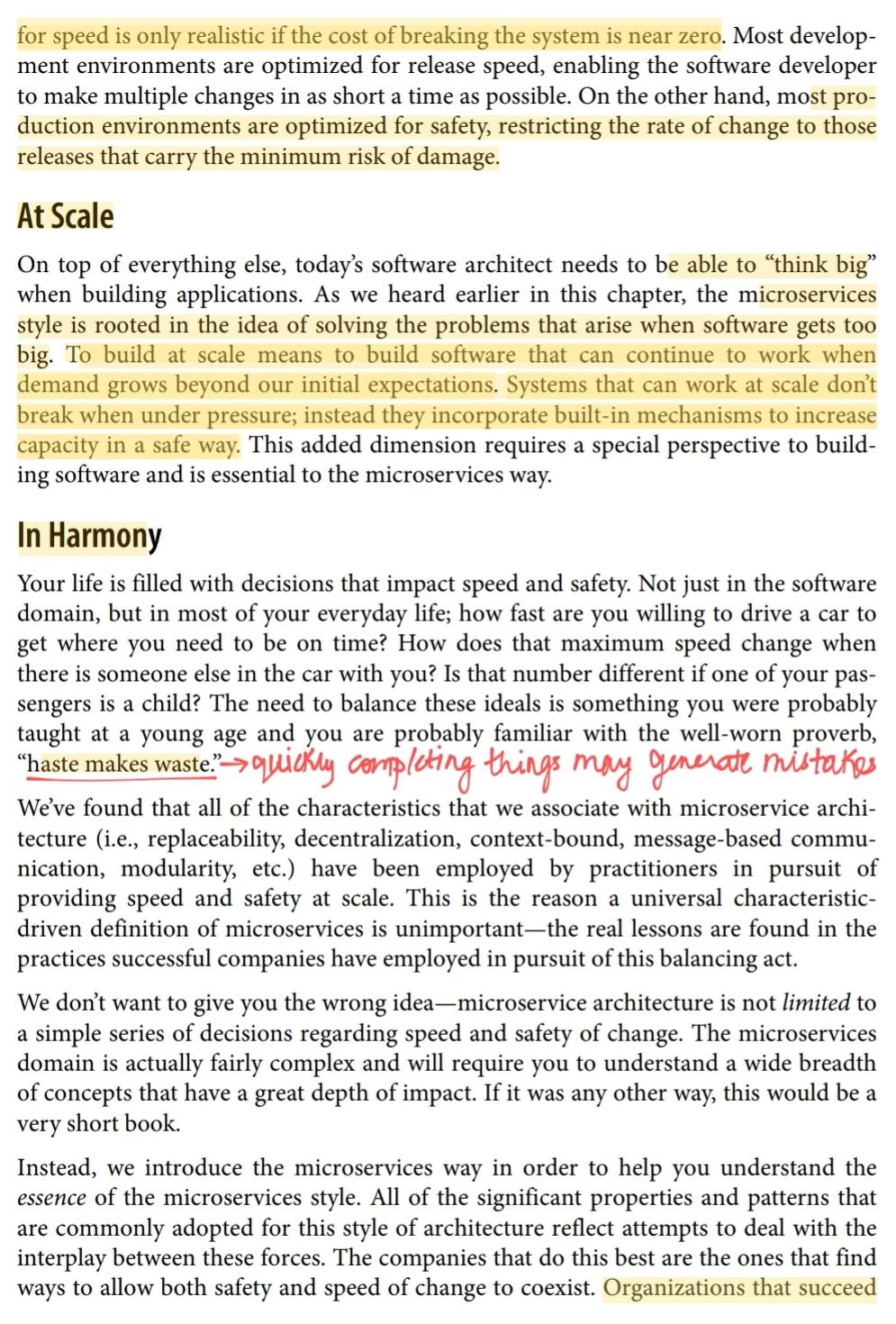


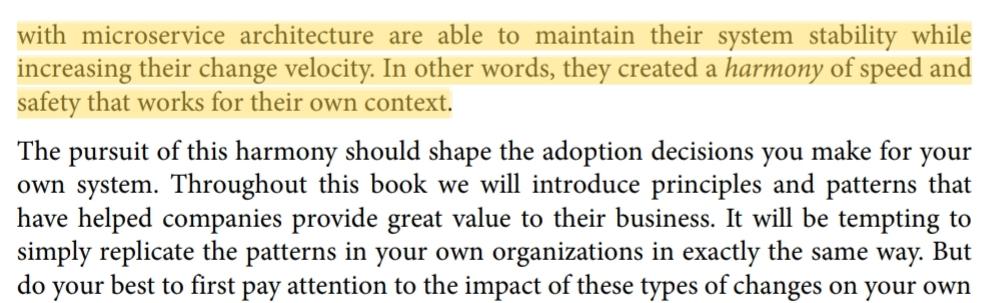
Q. Characteristics of MS application



Q. What is Microservice way or Business value?

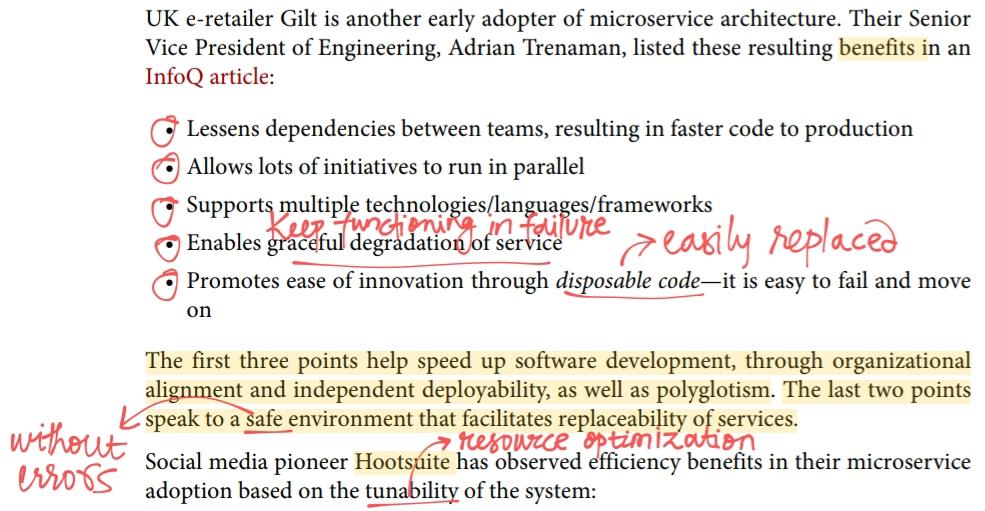




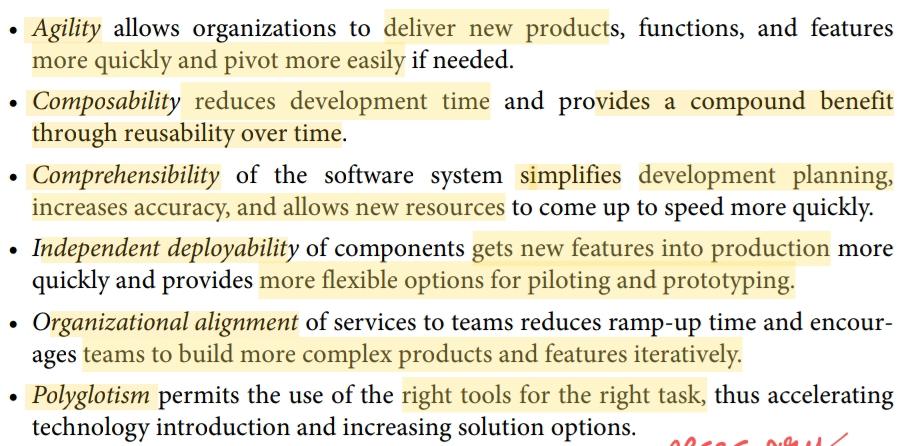


Chp 2. The Microservices Value Proposition

Q. Benefits of MS



Q. Business value - speed properties

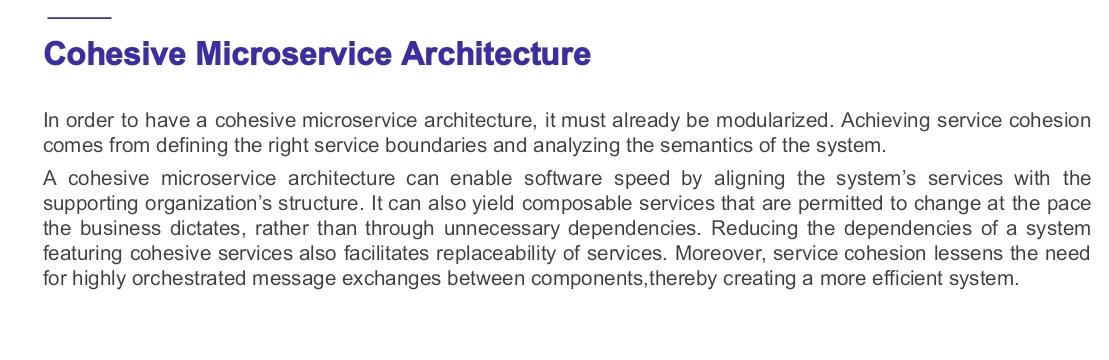


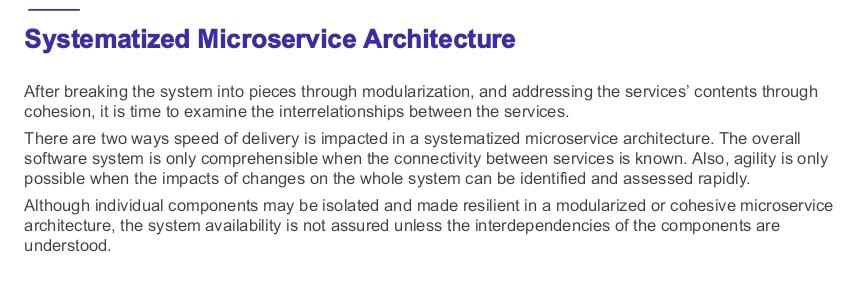
Q. Business value - safety properties



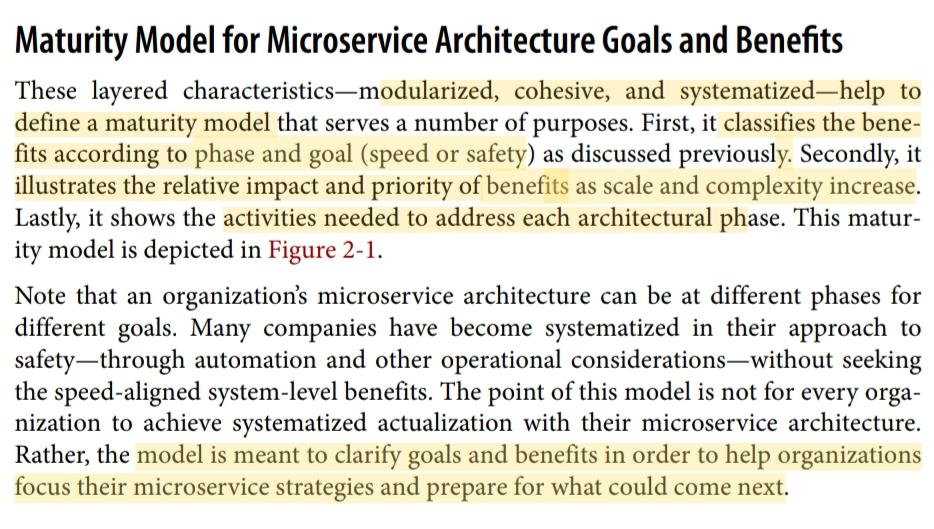
Q. What is a goal oriented, layered approach?

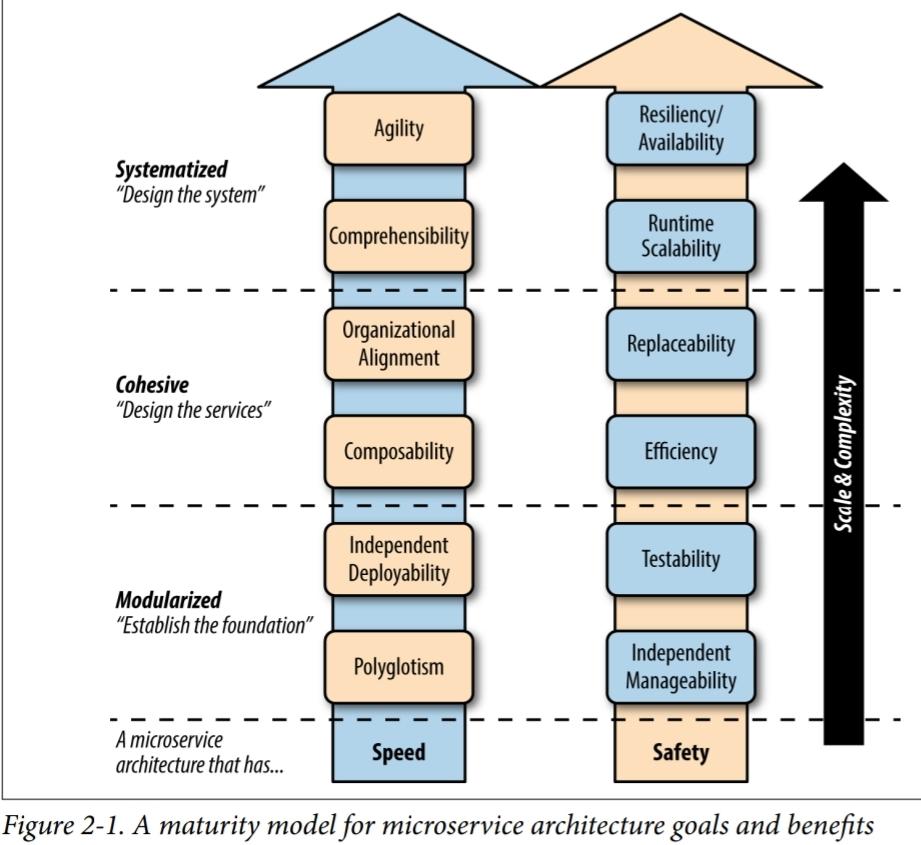






Q. Explain maturity model with diagram



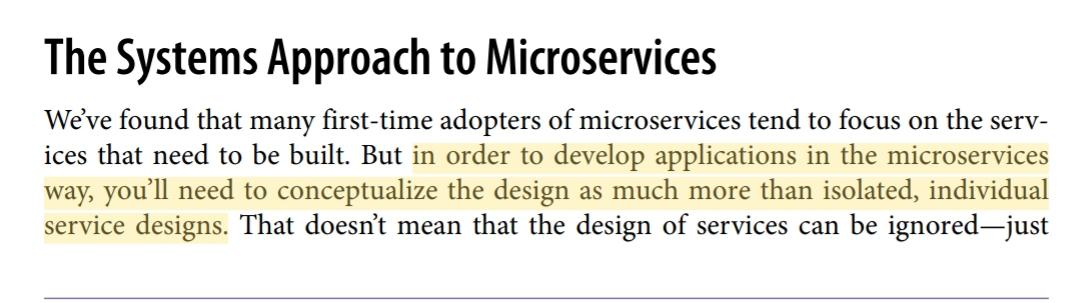


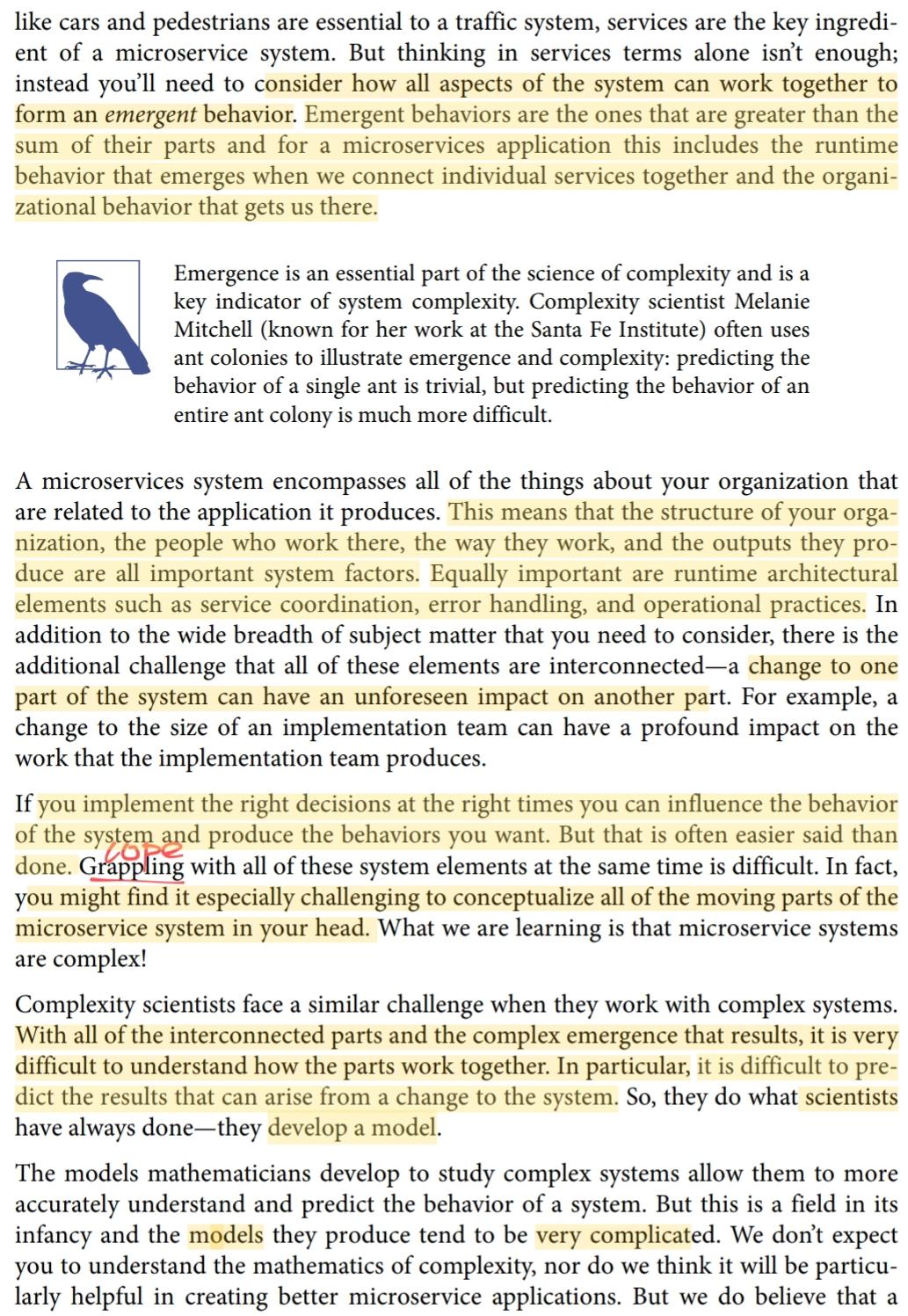
To begin with, define the high-level business objectives you want to accomplish, and

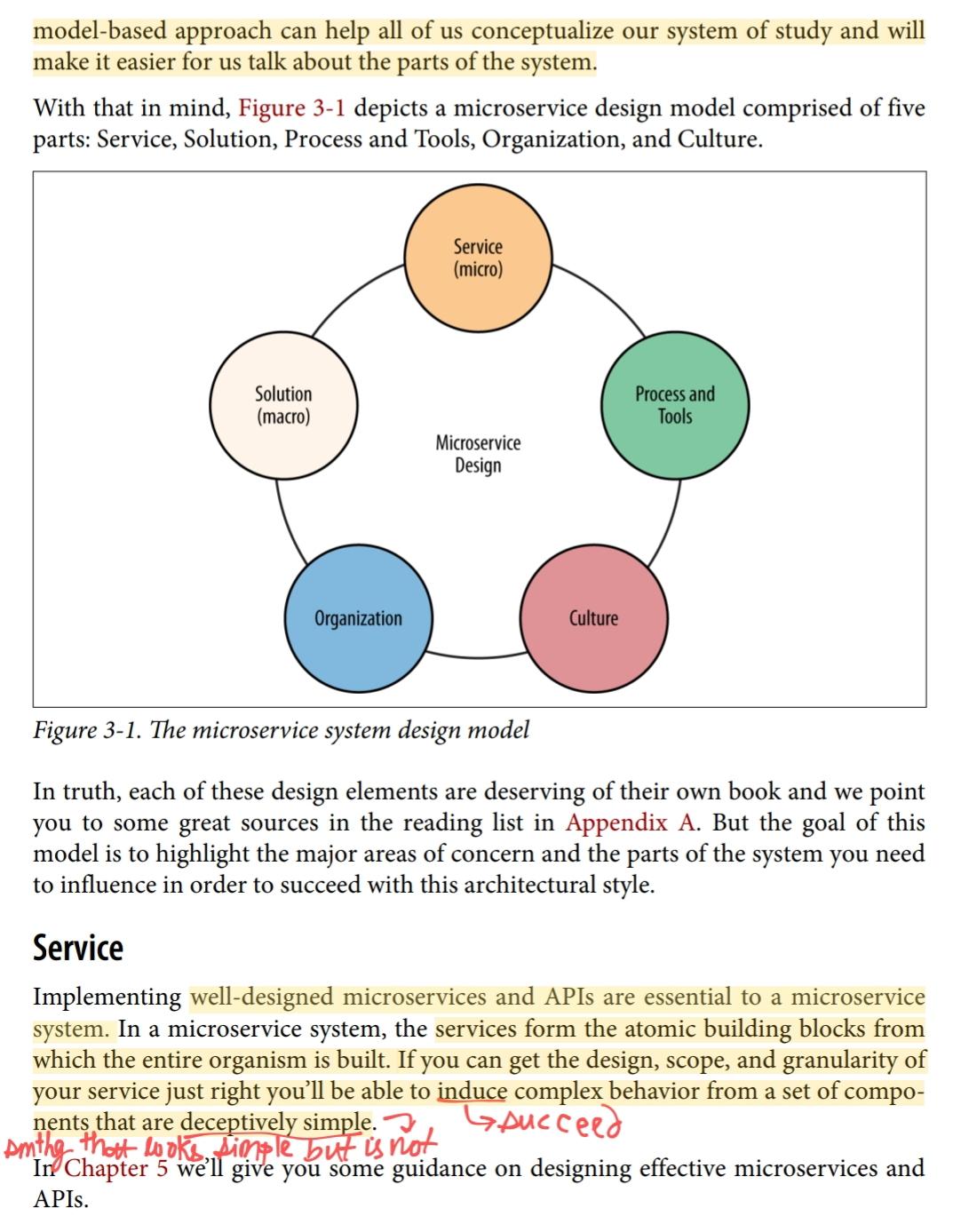
then weigh these against the dual goals of speed and safety. You can then use the maturity model to determine the complexity of the goal, and identify the best approach to achieve it.

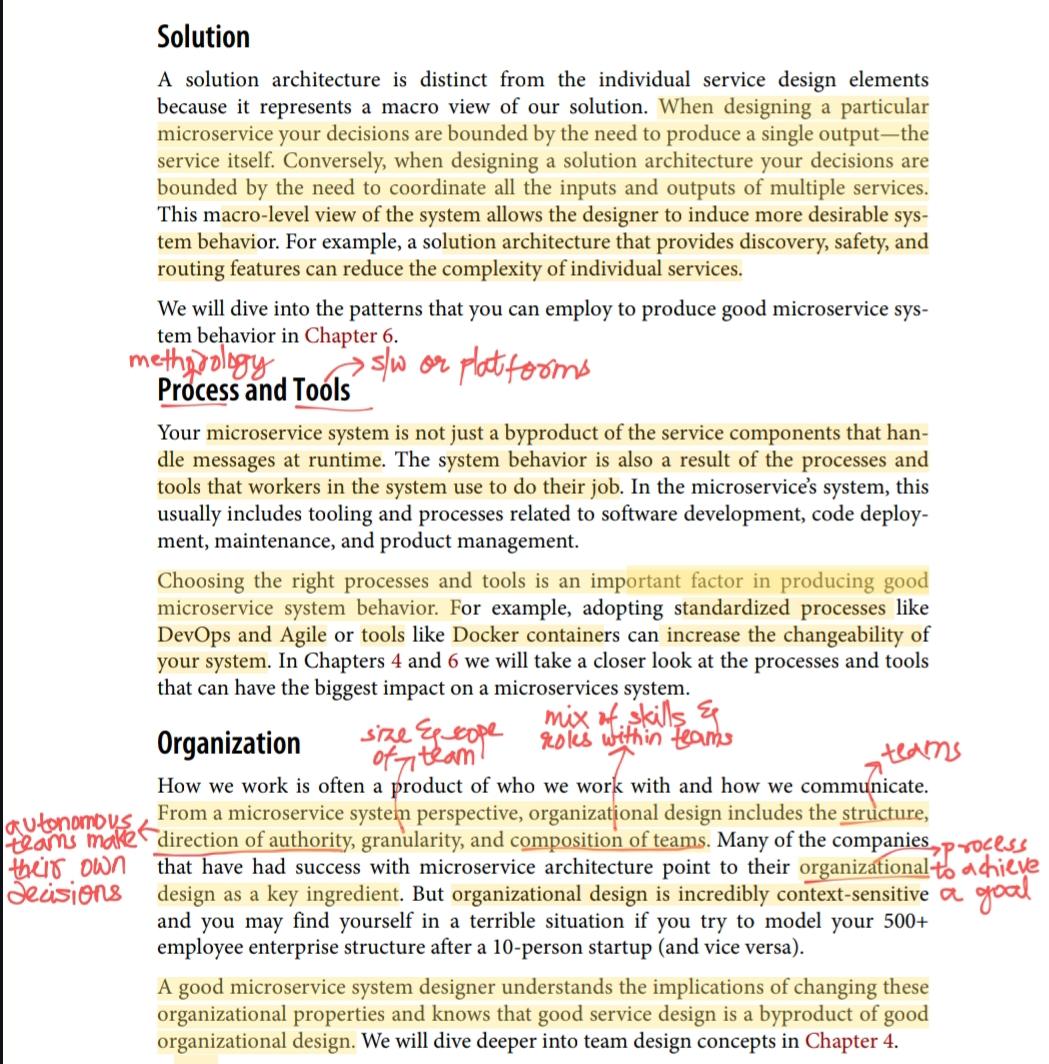
Chp 3. Designing Microservice Systems

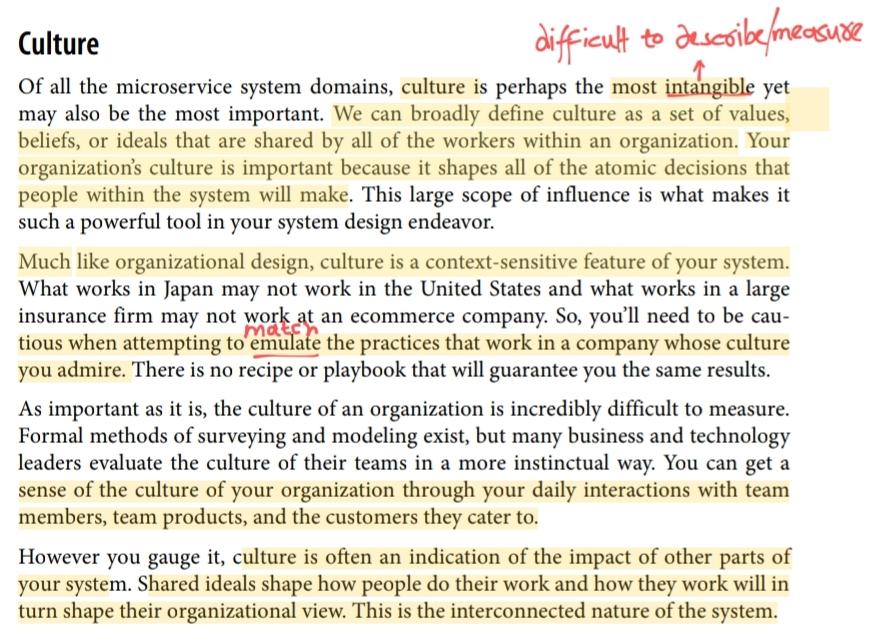
Q. Explain systems approach to MS along with MS system design model diagram



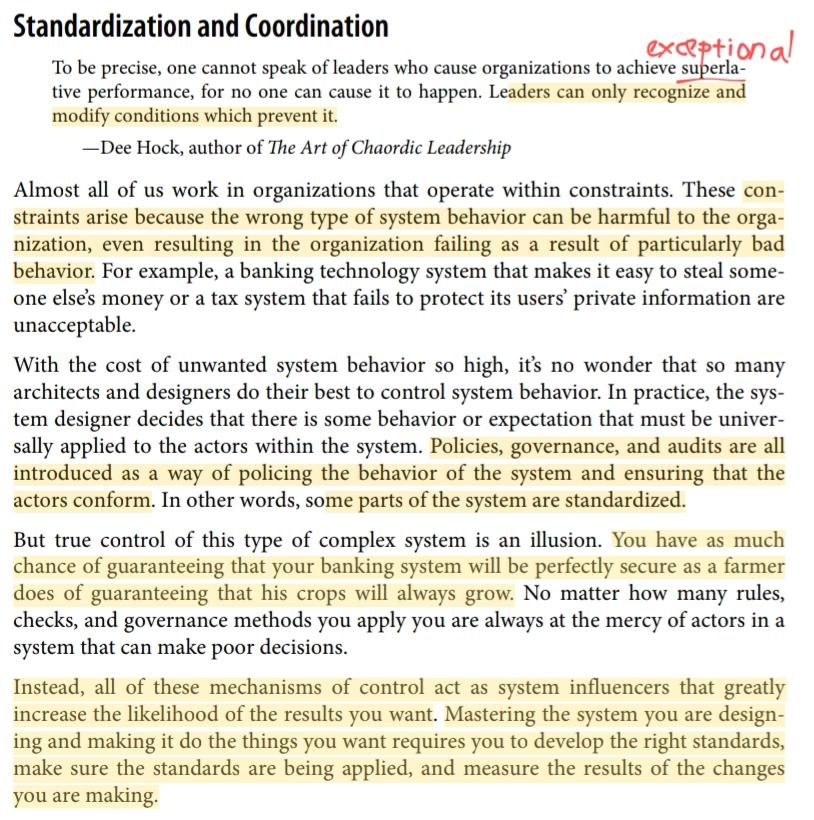


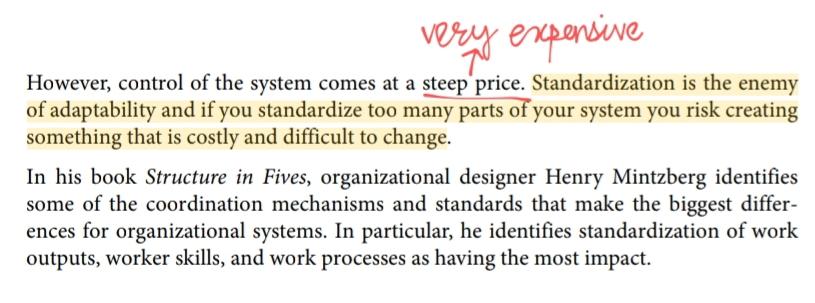


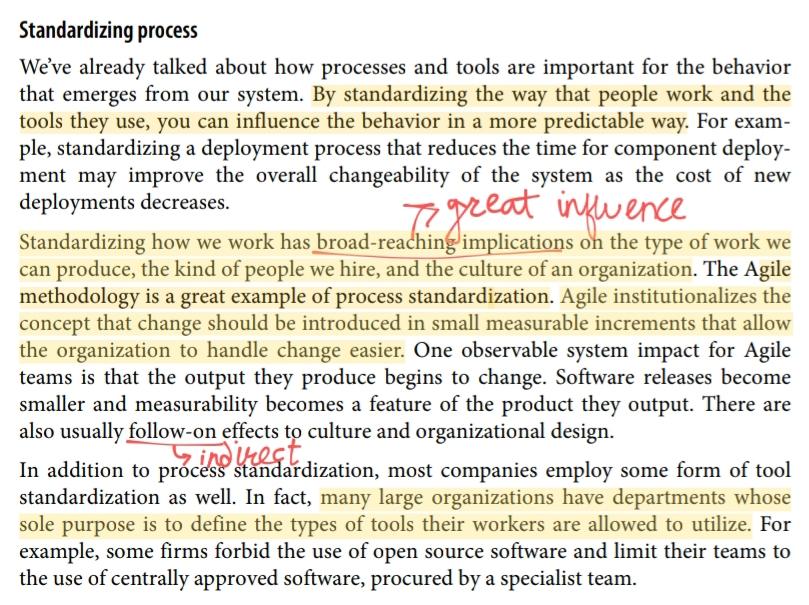


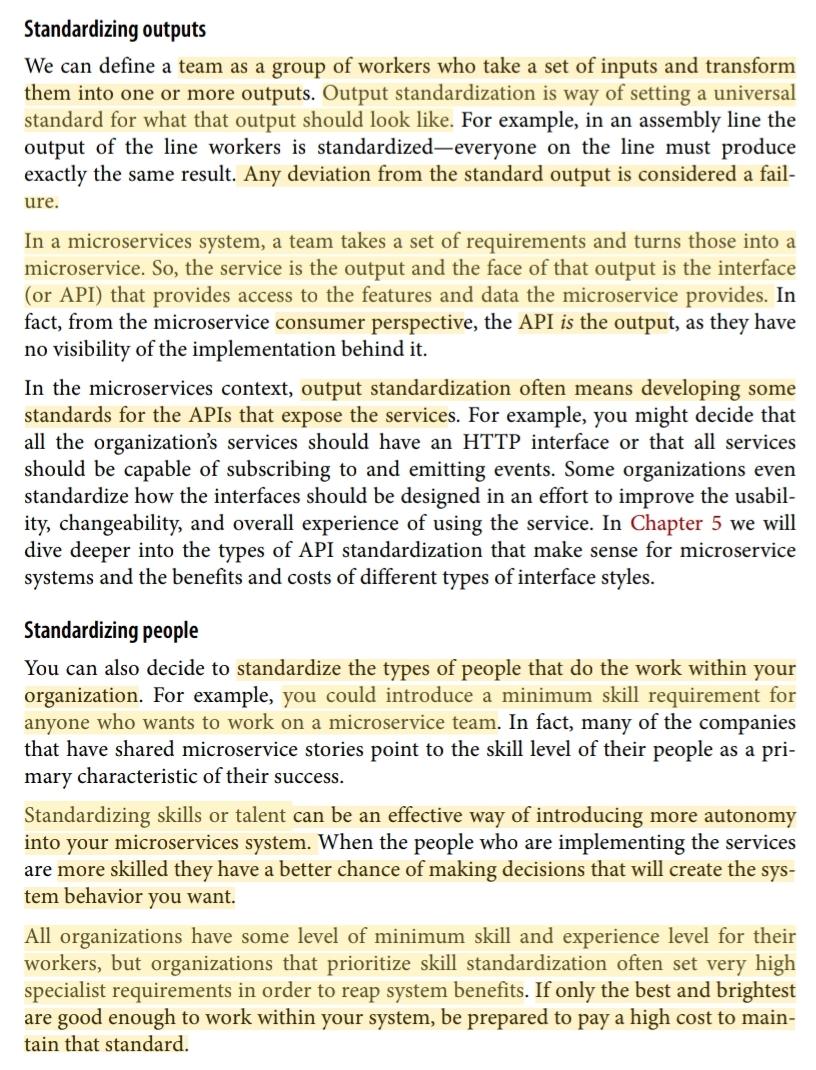


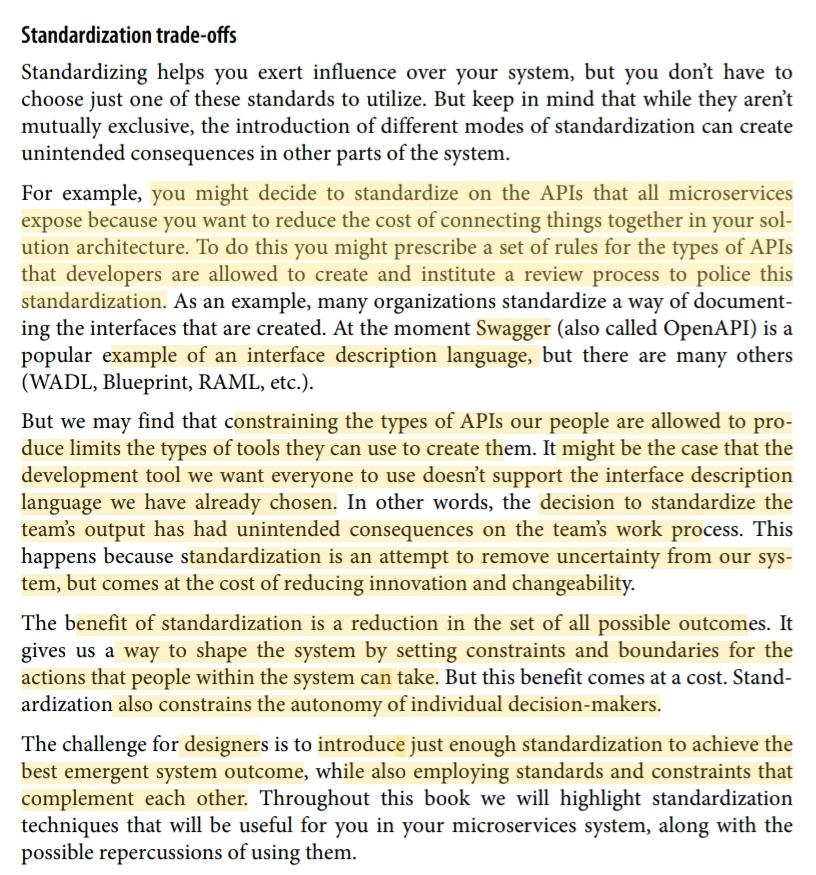
Q. Explain standardization and coordination in detail



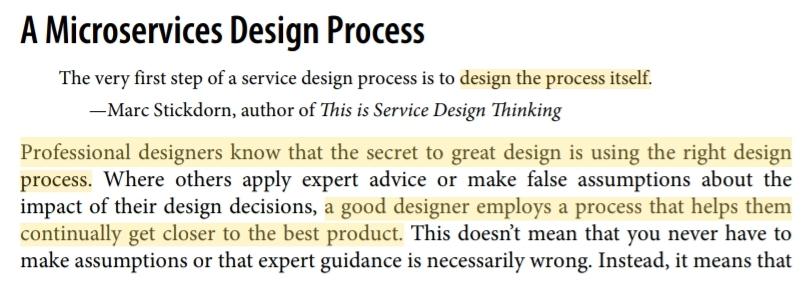


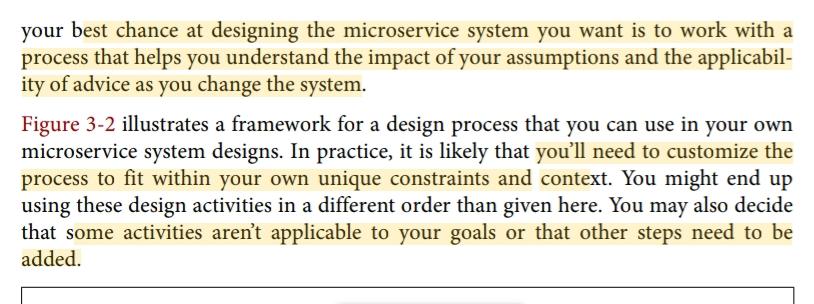


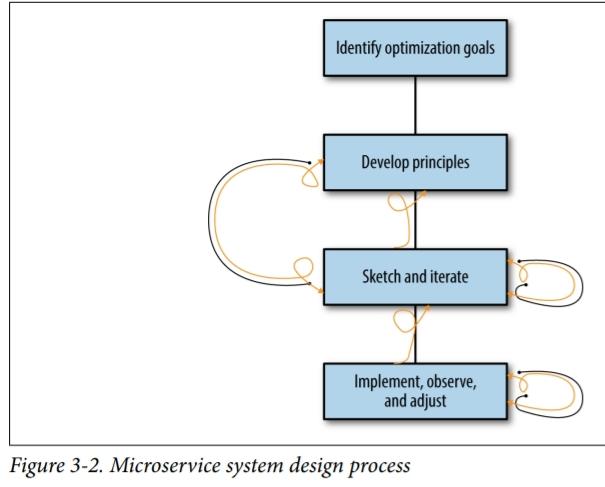


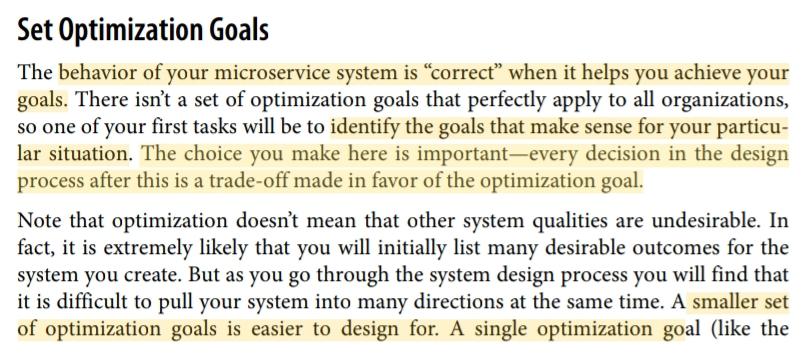


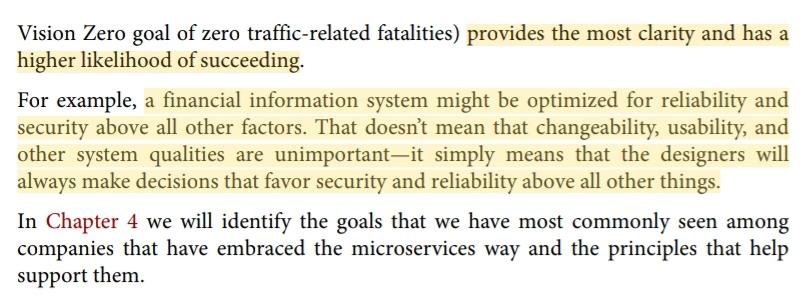
Q. Explain processes in MS Design with diagram

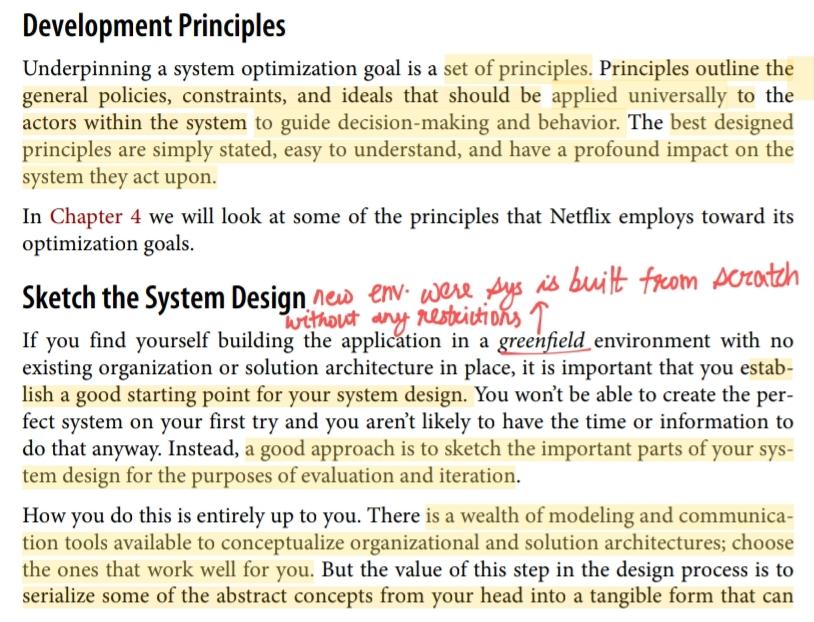


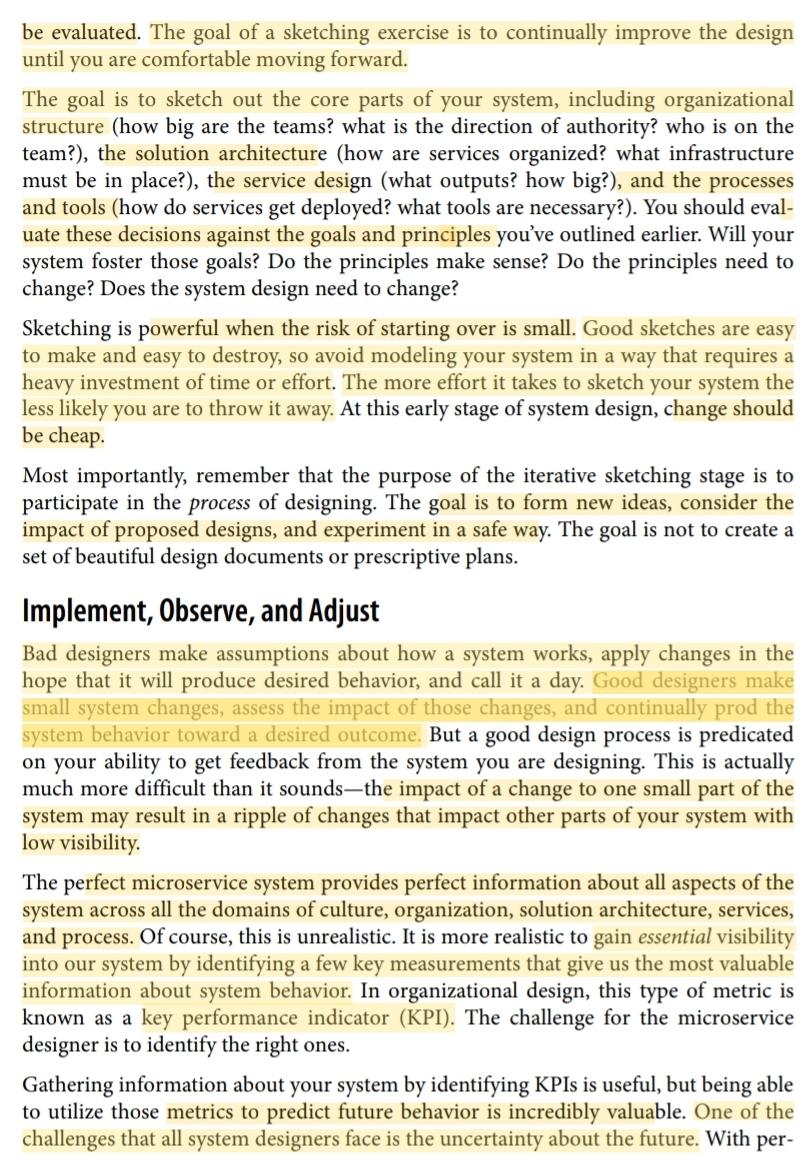


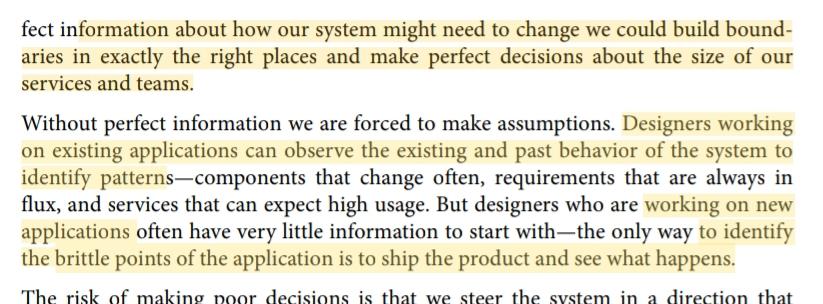














U3

Chp 1.

Q. What are the .NET Core Terminology

**Core CLR**

The CoreCLR is a lightweight, cross-platform runtime that provides many of the same features that the Common Language Runtime (CLR) provides on the Windows desktop or server including:

Garbage Collection: It is responsible for cleanup unused object references in an managed application

JUST-IN-TIME Compilation:It is responsible for Compiling Intermediate Code in the .NET assemblies into native on demand

Exception Handling:It is a part of runtime and not base class library

With .NET Core, the CoreCLR is now the smallest possible thing that can provide runtime services to .NET Core applications.

**Core FX**

CoreFX is a set of modular assemblies (available as NuGet packages and completely open source, available on GitHub) from which you can pick and choose.

Your application no longer needs to have every single class library assembly installed on the target server.

With CoreFX, you can use only what you need, and in true cloud-native fashion you should vendor (bundle) those dependencies with your application and expect nothing of your target deployment environment.

**.NET Platform Standard**

It aims to simplify this process and allow for a more manageable architecture to support .NET Core’s cross-platform goals for binary portability.

.NET Standard as a collection of interfaces that can either be implemented by the traditional .NET Framework

Latest version of .NET framework is 5.0 and 6.0

**ASP.NET Core**

ASP.NET Core is a collection of small, modular components that can be plugged into your application to let you build web applications and microservices.

Within ASP.NET Core you will find APIs for routing, JSON serialization, and rigging up MVC controllers and views.

ASP.NET came with the .NET Framework—you could not separate the two.

After the split between lightweight and heavyweight frameworks, you could install versions of the .NET Framework that did not include ASP.NET.

Q. Write down the steps for building a console application.

**Step 1**- To create the project in the current directory write the command dotnet new console.

**Step 2**-Once the project is created you can type dotnet restore,which analyzes the project dependencies and downloads whatever packages are necessary.The steps are required everytime you modify the project file.

**Step 3**-dotnet build to compile the program/or build the program/check the errors

**Step 4**-dotnet run

Hello World!

Our project consists of two files .csproj and program.cs.

Q. Write a code for program.cs or .csproj file for a simple hello world program.

**Program.cs :**

using System;

namespace ConsoleApplication

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello World!");

}

}

}

**.csproj :**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>netcoreapp1.1</TargetFramework>

</PropertyGroup>

</Project>

Explanation

NET Core version - 1.1

(dependency) Microsoft.NETCore.App version - 1.1.0.

Though the <Project> file provided doesn't directly show dependency changes, the mention of "changing the dependency on Microsoft.NETCore.App to version 1.1.0" implies that the underlying runtime library package version is also updated to align with the .NET Core 1.1 runtime.

Q. Write down the steps for building a asp.net core application

First, we’re going to want to add a few package references to our project:

Microsoft.AspNetCore.Mvc

Microsoft.AspNetCore.Server.Kestrel

Microsoft.Extensions.Logging (three different packages)

Microsoft.Extensions.Configuration.CommandLine

**.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>netcoreapp1.1</TargetFramework>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.Mvc”

Version="1.1.1" />

<PackageReference Include="Microsoft.AspNetCore.Server.Kestrel"

Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Logging"

Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Logging.Console"

Version="1.1.1"/>

<PackageReference Include="Microsoft.Extensions.Logging.Debug"

Version="1.1.1"/>

<PackageReference

Include="Microsoft.Extensions.Configuration.CommandLine"

Version="1.1.1"/>

</ItemGroup>

</Project>

**Program.cs**

using System;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Builder;

using Microsoft.Extensions.Configuration;

namespace HelloWorld

{

class Program

{

static void Main(string[] args)

{

var config = new ConfigurationBuilder()

.AddCommandLine(args)

.Build();

var host = new WebHostBuilder()

.UseKestrel()

.UseStartup<Startup>()

.UseConfiguration(config)

.Build();

host.Run();

}

}

}

UseStartup<> generic method to define a startup class that handles the new startup hooks.

The startup class is expected to be able to support the following methods:

* A constructor that takes an IHostingEnvironment variable
* The Configure method, used to configure the HTTP request pipeline and the application
* The ConfigureServices method, used to add scoped services to the system to be made available via dependency injection

**Startup.cs**

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Logging;

using Microsoft.AspNetCore.Http;

namespace HelloWorld {

public class Startup

{

public Startup(IHostingEnvironment env)

{

}

public void Configure(IApplicationBuilder app,

IHostingEnvironment env, ILoggerFactory loggerFactory)

{

app.Run(async (context) =>

{

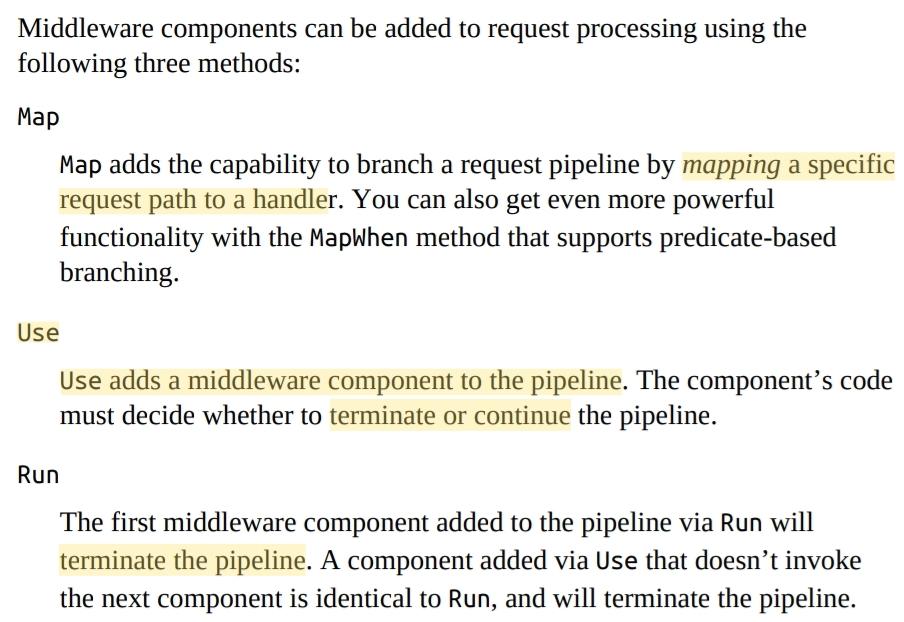
await context.Response.WriteAsync("Hello, world!");

});

}

}

}



$ dotnet run

curl will invoke the middleware and give you a response:

$ curl localhost:5000

O/p -> Hello, world!

**Q. Write and explain the commands for Docker or Docker image**

Docker lets you manually pull images into your local cache from a remote repository like docker hub

$ docker run -p 8080:8080 dotnetcoreservices/hello-world

Images will be cached locally. This command maps port 8080 inside the Docker image to port 8080 outside the Docker image.

Docker provides network isolation, so unless you explicitly allow traffic from outside a container to be routed inside the container, the isolation will function just like a firewall. Since we’ve mapped the inside and outside ports, we can now hit port 8080 on localhost.

We can see that this application is running with the following Docker command:

$ docker ps

O/p -> CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

61a68ffc3851 dotnetcoreservices/hello-world "/pipeline/source/..." 3 minutes ago Up 2 minutes

0.0.0.0:8080->8080/tcp priceless\_archimedes

$ curl http://localhost:8080/will/it/blend?

O/p - > Hello, world!

This shows that we can download a fully functioning piece of software from docker hub, cache the image locally, and execute the Docker image’s run command.

**To kill a running Docker process**

$ docker kill 61a68ffc3851

**Q. What is a CI tool with Wrecker? Explain with examples.**

Wercker provides a software package that helps developers and operations people embrace CI best practices.

Wercker is absurdly easy to use, the interface is intuitive, and its tight integration with Docker and support for spinning up multiple attached Docker images for integration testing

Wercker, there are three basic steps to get going for CI:

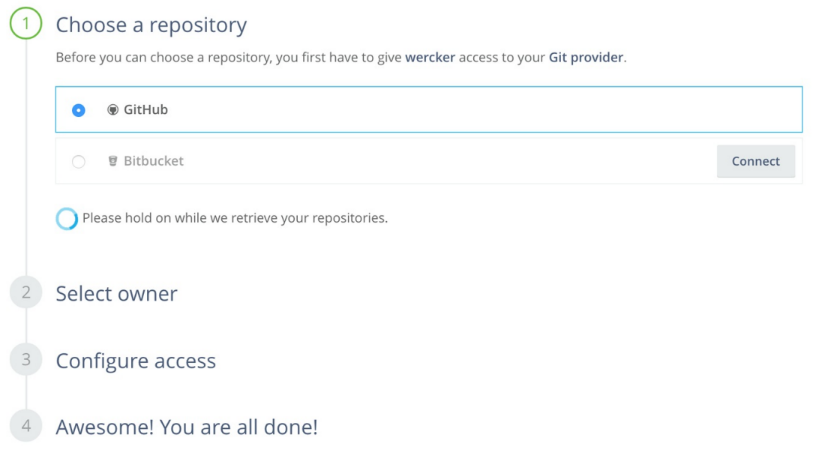
Create an application in Wercker using the website.

Add a wercker.yml file to your application’s codebase.

Choose how to package and where to deploy successful builds.

First thing you should login to your existing github account or create it.

Need to create repo.Like

****

The wizard will prompt you to choose a GitHub repository as the source for your build.

Once you’ve created the application, you need to add a wercker.yml file to the repository

Wercker.yaml file contains most of the metadata used to describe and configure your automatic build.

When starting service locally in interactive mode while still operating inside the wercker-generated docker image.

Your code is added in the Docker image specified in your wercker.yml file.

To run Wercker build locally -> need Wercker CLI.

Example :

no-response-timeout: 10

build:

steps:

- script:

name: restore

code: |

dotnet restore

- script:

name: build

code: |

dotnet build

- script:

name: publish

code: |

dotnet publish -o publish

- script:

name: copy binary

code: |

cp -r . $WERCKER\_OUTPUT\_DIR/app

cd $WERCKER\_OUTPUT\_DIR/app

1. dotnet restore to restore or download dependencies for the .NET

Application

2. dotnet build to compile the application.

3. dotnet publish to compile and then create a published, “ready to

execute” output directory.

Q. What is a CI tool with CircleCI? Explain with example.

Where Wercker runs your builds inside a Docker image and produces a Docker image as an artifact output, CircleCI offers control at a slightly lower level.

Go to http://circleci.com you can sign up for free with a new account or log in using your GitHub account

**main.py**

def to\_upper(name):

return name.upper()

def say\_hello(name):

print(f"Hello, {name}.")

if \_\_name\_\_=='\_\_main\_\_':

name="Ariba"

say\_hello(name)

print(to\_upper(name))

**test.py**:

import unittest

from main import to\_upper

class MyTestCase(unittest.TestCase):

def test\_to\_upper(self):

name="AribaKhan"

up=to\_upper(name)

self.assertEqual(up, "ARIBAKHAN")

if \_\_name\_\_=='\_\_main\_\_':

unittest.main()

Configuration file to tell CircleCI how to build the app.

**config.yml**:

version: 2.1

jobs:

build:

docker:

- image: cimg/python:3.11

steps:

- checkout

- run: python main.py

test:

docker:

- image: cimg/python:3.11

steps:

- checkout

- run: python test.py

deploy:

docker:

- image: cimg/python:3.11

steps:

- run: echo "Ariba practical is done!"

workflows:

build\_and\_test\_deploy:

jobs:

- build

- test:

requires:

- build

- deploy:

requires:

- test

filters:

branches:

only: main

Stage all changes in your working directory with git add ..

Commit these staged changes to your local repository with a message using git commit -m "update".

Push the commit to the remote repository's main branch using git push origin main.

Now you can set up your project in CircleCI and see the output.

**MCQ on SOA (Service Oriented Architecture)**

\*\*1. What is a primary characteristic of Service-Oriented Architecture (SOA)?\*\*

A. Centralized data storage

B. Monolithic design

C. Loose coupling between services

D. Single point of failure

\*Answer:-**C. Loose coupling between services**

\*\*2. In a microservices architecture, what is typically used to handle communication between services?\*\*

A. Direct method calls

B. Shared memory

C. RESTful APIs or messaging queues

D. Local function calls

Answer:- **C. RESTful APIs or messaging queues**

\*\*3. Which of the following is NOT a typical benefit of adopting a microservices architecture?\*\*

A. Improved scalability

B. Increased deployment speed

C. Reduced service isolation

D. Enhanced fault tolerance

Answer:-**C. Reduced service isolation**

\*\*4. What role does an API Gateway play in a microservices architecture?\*\*

A. It handles service discovery and routing

B. It maintains the central database

C. It aggregates microservices into a monolith

D. It performs data replication

Answer:**-A. It handles service discovery and routing**

\*\*5. Which principle is a key advantage of microservices compared to a traditional monolithic architecture?\*\*

A. All services share a single database

B. Tight integration between services

C. Independent deployment and scaling of services

D. Single codebase for all functionalities

Answer:- C**. Independent deployment and scaling of services**

\*\*6. In microservices architecture, how are services typically deployed?\*\*

A. As a single large application

B. As individual containers or processes

C. As a shared library

D. As a monolithic application with service wrappers

Answer:-**B. As individual containers or processe**s

\*\*7. Which of the following is a common challenge associated with microservices architecture?\*\*

A. Reduced complexity in service management

B. Simplified data consistency management

C. Increased inter-service communication overhead

D. Centralized logging and monitoring

Answer:-**C. Increased inter-service communication overhead**

\*\*8. What is the main goal of service decomposition in a microservices architecture?\*\*

A. To create large, monolithic services

B. To simplify service integration

C. To break down applications into smaller, manageable services

D. To enforce service dependencies

Answer:- **C. To break down applications into smaller, manageable services**

\*\*9. What does the term "service discovery" refer to in a microservices context?\*\*

A. The process of detecting which services are running and their locations

B. The process of discovering new microservices on the network

C. The manual process of configuring service endpoints

D. The process of identifying outdated microservices

Answer:- **A. The process of detecting which services are running and their locations**

\*\*10. Which approach is commonly used to ensure data consistency across microservices?\*\*

A. Using a centralized database

B. Applying eventual consistency and distributed transactions

C. Relying on synchronous method calls

D. Using a single large monolith

Answer:- **B. Applying eventual consistency and distributed transactions**

What is a defining characteristic of microservices architecture?

A. Monolithic structure with tightly coupled components  
B. Single, large database shared among all services  
C. Decomposition of an application into small, independently deployable services  
D. Centralized logging and monitoring

Answer: C. Decomposition of an application into small, independently deployable services

. When adopting microservices, which of the following is a common first step?

A. Rewriting the entire application in a new programming language  
B. Creating a detailed plan for inter-service communication  
C. Decomposing the monolithic application into smaller services  
D. Merging all services into a single repository

**Answer: C. Decomposing the monolithic application into smaller services**

**What is a key benefit of adopting microservices over a traditional monolithic architecture?**

A. Increased deployment complexity  
B. Centralized service management  
C. Enhanced flexibility and scalability  
D. Reduced service isolation

**Answer:** C. Enhanced flexibility and scalability

**In the microservices approach, what is the purpose of the API Gateway?**

A. To manage service dependencies and enforce service contracts  
B. To provide a unified interface for routing requests to various microservices  
C. To store all application data in a single location  
D. To perform database transactions across multiple services

**Answer:** B. To provide a unified interface for routing requests to various microservices

**5. Which of the following is NOT a typical consideration when adopting a microservices architecture?**

A. Service discovery  
B. API management  
C. Monolithic service deployment  
D. Inter-service communication

**Answer:** C. Monolithic service deployment

**6. How do microservices typically handle data consistency?**

A. By using a single, centralized database  
B. Through distributed transactions and eventual consistency  
C. By maintaining a shared in-memory cache  
D. By coupling all services to the same data schema

**Answer:** B. Through distributed transactions and eventual consistency

**7. What is a common challenge when working with microservices?**

A. Simplified inter-service communication  
B. Increased management overhead and complexity  
C. Reduced need for service monitoring  
D. Single point of failure

**Answer:** B. Increased management overhead and complexity

**8. What is a "bounded context" in the context of microservices?**

A. A specific module or library that handles service orchestration  
B. A well-defined boundary within which a microservice operates and maintains its own data  
C. A service that handles inter-service communication  
D. A global context for all microservices within an application

**Answer:** B. A well-defined boundary within which a microservice operates and maintains its own data

**9. What does the "Microservices Way" emphasize for service development and deployment?**

A. Large, tightly coupled services with centralized management  
B. Frequent, independent deployments and failure isolation  
C. Shared codebases and monolithic data storage  
D. Synchronous communication and large-scale monoliths

**Answer:** B. Frequent, independent deployments and failure isolation

**10. In microservices, how is service communication typically managed?**

A. Through direct database access  
B. Using RESTful APIs, gRPC, or messaging queues  
C. By shared libraries  
D. Through synchronous method calls within a single service

**Answer:** B. Using RESTful APIs, gRPC, or messaging queues

**Unit 1 Chapter 1**

\*\*Understanding Microservices\*\*

1. \*\*What is a key characteristic of a microservice architecture?\*\*

- A) Centralized data storage

- B) Single large codebase

- C) Small, independent services

- D) Monolithic deployment

\*\*Answer:\*\* C) Small, independent services

2. \*\*Which of the following best describes the primary advantage of using microservices?\*\*

- A) Easier management of a single, large application

- B) Improved scalability and flexibility

- C) Simplified monolithic design

- D) Decreased need for automated testing

\*\*Answer:\*\* B) Improved scalability and flexibility

3. \*\*In a microservices architecture, how is data typically managed?\*\*

- A) Through a single, centralized database

- B) Each microservice manages its own database

- C) Data is managed through a shared file system

- D) By using a global state management service

\*\*Answer:\*\* B) Each microservice manages its own database

### \*\*Adopting Microservices\*\*

1. \*\*What is a common challenge when adopting microservices?\*\*

- A) Reduced need for network communication

- B) Increased complexity in data consistency

- C) Simplified deployment process

- D) Decreased overhead in service management

\*\*Answer:\*\* B) Increased complexity in data consistency

2. \*\*Which of the following strategies is useful when transitioning from a monolithic architecture to microservices?\*\*

- A) Rewriting the entire application at once

- B) Gradually refactoring parts of the monolith into microservices

- C) Removing all existing tests

- D) Keeping the monolithic architecture and starting new projects as microservices

\*\*Answer:\*\* B) Gradually refactoring parts of the monolith into microservices

3. \*\*What role does API gateway play in a microservices architecture?\*\*

- A) It combines multiple microservices into a single service

- B) It provides a unified entry point and handles routing and request management

- C) It manages internal communication between microservices

- D) It is responsible for data storage

\*\*Answer:\*\* B) It provides a unified entry point and handles routing and request management

### \*\*The Microservices Way\*\*

1. \*\*Which of the following practices is commonly associated with the "Microservices Way"?\*\*

- A) Using a single technology stack across all services

- B) Implementing continuous integration and continuous deployment (CI/CD) pipelines

- C) Centralizing business logic in one service

- D) Avoiding the use of automated testing

\*\*Answer:\*\* B) Implementing continuous integration and continuous deployment (CI/CD) pipelines

2. \*\*In the context of microservices, what is the purpose of a service registry?\*\*

- A) To store the configuration of each microservice

- B) To manage the deployment of microservices

- C) To keep track of the instances and locations of services

- D) To handle data storage for microservices

\*\*Answer:\*\* C) To keep track of the instances and locations of services

3. \*\*What is a common pattern for managing inter-service communication in microservices?\*\*

- A) Synchronous communication via REST or gRPC

- B) Direct database queries between services

- C) Synchronous communication using shared file systems

- D) Asynchronous communication via message queues

\*\*Answer:\*\* D) Asynchronous communication via message queues

**Chapter 2**

\*\*Deriving Business Value\*\*

1. \*\*What is a primary business benefit of adopting a microservices architecture?\*\*

- A) Increased hardware requirements

- B) Enhanced ability to quickly adapt to changes and innovate

- C) Decreased need for monitoring and logging

- D) Simplified user interface design

\*\*Answer:\*\* B) Enhanced ability to quickly adapt to changes and innovate

2. \*\*How does microservices architecture contribute to cost efficiency?\*\*

- A) By consolidating all services into one large deployment

- B) By reducing the need for separate development teams

- C) By enabling scaling of only the services that require additional resources

- D) By eliminating the need for version control

\*\*Answer:\*\* C) By enabling scaling of only the services that require additional resources

3. \*\*Which metric is most useful for measuring the business impact of microservices?\*\*

- A) Number of microservices deployed

- B) Average time to deploy new features

- C) Number of code lines written

- D) Size of the codebase

\*\*Answer:\*\* B) Average time to deploy new features

### \*\*Defining a Goal-Oriented, Layered Approach\*\*

1. \*\*In a goal-oriented, layered approach to microservices, what is the primary focus of the "goal-oriented" aspect?\*\*

- A) Organizing services by technology stack

- B) Aligning service design with specific business objectives and outcomes

- C) Standardizing all services to use the same programming language

- D) Combining multiple business functions into a single service

\*\*Answer:\*\* B) Aligning service design with specific business objectives and outcomes

2. \*\*What is a key benefit of using a layered approach in microservices?\*\*

- A) It mandates a single database schema for all services

- B) It allows separation of concerns and promotes modular design

- C) It restricts the use of different technologies across services

- D) It eliminates the need for service discovery mechanisms

\*\*Answer:\*\* B) It allows separation of concerns and promotes modular design

3. \*\*Which layer in a layered microservices architecture is typically responsible for handling communication between services?\*\*

- A) Data layer

- B) Business logic layer

- C) Application layer

- D) Integration layer

\*\*Answer:\*\* D) Integration layer

### \*\*Applying the Goal-Oriented, Layered Approach\*\*

1. \*\*When applying a goal-oriented, layered approach, how should you prioritize the design of microservices?\*\*

- A) By focusing on technology stack first

- B) By defining business goals and then designing services to meet those goals

- C) By minimizing the number of services to reduce complexity

- D) By integrating all services into a monolithic application

\*\*Answer:\*\* B) By defining business goals and then designing services to meet those goals

2. \*\*What is a recommended practice when implementing the layered approach in a microservices environment?\*\*

- A) Create tightly coupled services to avoid complex interactions

- B) Ensure each layer has a clear responsibility and does not overlap with others

- C) Limit communication between layers to improve performance

- D) Use a single service for all layers to streamline development

\*\*Answer:\*\* B) Ensure each layer has a clear responsibility and does not overlap with others

3. \*\*How does the goal-oriented, layered approach help in achieving business agility?\*\*

- A) By enforcing a strict hierarchy of services

- B) By enabling independent development and deployment of services aligned with business objectives

- C) By reducing the number of services required for the application

- D) By standardizing business processes across all services

\*\*Answer:\*\* B) By enabling independent development and deployment of services aligned with business objectives

**Chapter 3**

\*\*The Systems Approach to Microservices\*\*

1. \*\*What is a key characteristic of the systems approach to designing microservices?\*\*

- A) Focus on individual service performance over system performance

- B) Emphasis on understanding the interactions and dependencies between services

- C) Centralized control of all service deployments

- D) Combining all services into a single deployment unit

\*\*Answer:\*\* B) Emphasis on understanding the interactions and dependencies between services

2. \*\*In the systems approach, what is the primary benefit of designing microservices with clear boundaries and responsibilities?\*\*

- A) Increased complexity in service interactions

- B) Improved maintainability and scalability

- C) Reduced need for automated testing

- D) Simplified data management

\*\*Answer:\*\* B) Improved maintainability and scalability

3. \*\*Which of the following best describes the concept of "service cohesion" in the systems approach to microservices?\*\*

- A) Combining multiple business functions into a single service

- B) Designing each service to handle multiple unrelated tasks

- C) Ensuring that each service has a single, well-defined purpose and responsibility

- D) Integrating all services into a single application

\*\*Answer:\*\* C) Ensuring that each service has a single, well-defined purpose and responsibility

4. \*\*What is the primary challenge addressed by the systems approach when it comes to service integration?\*\*

- A) Minimizing the number of services

- B) Managing dependencies and ensuring reliable communication between services

- C) Reducing the complexity of the user interface

- D) Centralizing data storage

\*\*Answer:\*\* B) Managing dependencies and ensuring reliable communication between services

5. \*\*Which design principle is crucial for ensuring that microservices are loosely coupled in a systems approach?\*\*

- A) Services should share the same database schema

- B) Each service should expose well-defined APIs and communicate via standardized protocols

- C) Services should be deployed as a single monolithic application

- D) Services should be tightly integrated to avoid performance overhead

\*\*Answer:\*\* B) Each service should expose well-defined APIs and communicate via standardized protocols

6. \*\*In the systems approach, what role does "service discovery" play in managing microservices?\*\*

- A) It helps services find and communicate with each other dynamically

- B) It centralizes all service logs in one location

- C) It enforces strict security policies for each service

- D) It manages data consistency across all services

\*\*Answer:\*\* A) It helps services find and communicate with each other dynamically

7. \*\*How does the systems approach address the challenge of data consistency across microservices?\*\*

- A) By centralizing data management in a single service

- B) By using distributed transactions and eventual consistency models

- C) By eliminating the need for data synchronization

- D) By storing all data in a global state management system

\*\*Answer:\*\* B) By using distributed transactions and eventual consistency models

8. \*\*What is a key benefit of applying domain-driven design principles in the systems approach to microservices?\*\*

- A) It ensures that services are tightly coupled for efficiency

- B) It helps in modeling services around business domains, improving alignment with business needs

- C) It standardizes the technology stack used across all services

- D) It reduces the number of APIs needed for service communication

\*\*Answer:\*\* B) It helps in modeling services around business domains, improving alignment with business needs

**Chapter 4**

\*\*Goals and Principles\*\*

1. \*\*What is a primary goal when establishing a microservices architecture?\*\*

- A) To centralize all business logic in one service

- B) To enhance the scalability and flexibility of the system

- C) To minimize the number of services used

- D) To reduce the need for automated testing

\*\*Answer:\*\* B) To enhance the scalability and flexibility of the system

2. \*\*Which principle emphasizes that each microservice should have a single responsibility?\*\*

- A) Scalability

- B) Single Responsibility Principle (SRP)

- C) High Cohesion

- D) Loose Coupling

\*\*Answer:\*\* B) Single Responsibility Principle (SRP)

3. \*\*What does the principle of "Loose Coupling" refer to in microservices architecture?\*\*

- A) Services should be able to operate independently without affecting each other

- B) Services should share the same database schema

- C) Services should be tightly integrated to improve performance

- D) Services should be implemented in the same programming language

\*\*Answer:\*\* A) Services should be able to operate independently without affecting each other

4. \*\*When setting goals for a microservices architecture, what is an important consideration to ensure alignment with business objectives?\*\*

- A) Choosing the latest technology stack

- B) Defining clear, measurable objectives that address specific business needs

- C) Minimizing the number of services to avoid complexity

- D) Standardizing the user interface across all services

\*\*Answer:\*\* B) Defining clear, measurable objectives that address specific business needs

### \*\*Platforms\*\*

1. \*\*Which platform component is essential for managing the deployment and orchestration of microservices?\*\*

- A) Version control system

- B) Container orchestration platform (e.g., Kubernetes)

- C) Traditional relational database

- D) Monolithic application server

\*\*Answer:\*\* B) Container orchestration platform (e.g., Kubernetes)

2. \*\*What is the purpose of a service registry in the context of microservices platforms?\*\*

- A) To store configuration files for each microservice

- B) To enable dynamic discovery and communication between services

- C) To manage user authentication and authorization

- D) To handle all data storage requirements

\*\*Answer:\*\* B) To enable dynamic discovery and communication between services

3. \*\*Which platform feature is important for ensuring that microservices can scale independently?\*\*

- A) Centralized logging

- B) Load balancing and autoscaling capabilities

- C) Shared file system

- D) Synchronous service calls

\*\*Answer:\*\* B) Load balancing and autoscaling capabilities

4. \*\*In the context of microservices platforms, what role does API gateway play?\*\*

- A) It centralizes data storage for all microservices

- B) It routes requests to the appropriate microservices and handles cross-cutting concerns

- C) It replaces the need for service orchestration

- D) It manages the deployment of microservices

\*\*Answer:\*\* B) It routes requests to the appropriate microservices and handles cross-cutting concerns

### \*\*Culture\*\*

1. \*\*What cultural shift is often necessary for successfully adopting microservices?\*\*

- A) Moving from a siloed to a collaborative and cross-functional team structure

- B) Emphasizing a single technology stack across the organization

- C) Centralizing decision-making processes to a single team

- D) Minimizing communication between development and operations teams

\*\*Answer:\*\* A) Moving from a siloed to a collaborative and cross-functional team structure

2. \*\*How does adopting a DevOps culture support the implementation of microservices?\*\*

- A) By segregating development and operations teams

- B) By promoting continuous integration, continuous delivery, and automated testing

- C) By limiting the use of automation tools

- D) By focusing solely on manual deployment processes

\*\*Answer:\*\* B) By promoting continuous integration, continuous delivery, and automated testing

3. \*\*Why is it important for teams to embrace a culture of experimentation and learning when working with microservices?\*\*

- A) To ensure that all services are implemented in the same way

- B) To foster innovation and improve the ability to quickly adapt to changing requirements

- C) To minimize the number of services and reduce complexity

- D) To enforce strict adherence to traditional software development practices

\*\*Answer:\*\* B) To foster innovation and improve the ability to quickly adapt to changing requirements

4. \*\*What role does communication play in a successful microservices culture?\*\*

- A) It is less important than technology choices

- B) It ensures alignment across teams and promotes effective collaboration

- C) It is replaced by automated tools and processes

- D) It focuses only on internal team interactions, not cross-team communications

\*\*Answer:\*\* B) It ensures alignment across teams and promotes effective collaboration

**Unit 3 Chapter 1**

\*\*Introduction\*\*

1. \*\*What is ASP.NET Core primarily used for?\*\*

- A) Developing mobile applications

- B) Building cross-platform web applications and APIs

- C) Creating desktop applications only for Windows

- D) Developing only static websites

\*\*Answer:\*\* B) Building cross-platform web applications and APIs

2. \*\*Which of the following describes the primary advantage of ASP.NET Core over traditional ASP.NET?\*\*

- A) It is only available for Windows operating systems

- B) It offers improved performance and is cross-platform, running on Windows, macOS, and Linux

- C) It is designed for a single web application deployment

- D) It lacks support for modern web technologies

\*\*Answer:\*\* B) It offers improved performance and is cross-platform, running on Windows, macOS, and Linux

### \*\*Installing .NET Core\*\*

1. \*\*What is the primary tool used to install .NET Core SDK on your development machine?\*\*

- A) Visual Studio Code

- B) .NET Core Installer from the official .NET website

- C) Node Package Manager (NPM)

- D) Java Development Kit (JDK)

\*\*Answer:\*\* B) .NET Core Installer from the official .NET website

2. \*\*Which command would you use to verify that .NET Core SDK is installed correctly on your system?\*\*

- A) `dotnet --version`

- B) `dotnet check`

- C) `dotnet status`

- D) `dotnet info`

\*\*Answer:\*\* A) `dotnet --version`

### \*\*Building a Console App\*\*

1. \*\*Which command creates a new .NET Core console application?\*\*

- A) `dotnet new console`

- B) `dotnet create console`

- C) `dotnet run console`

- D) `dotnet init console`

\*\*Answer:\*\* A) `dotnet new console`

2. \*\*In a .NET Core console application, which method serves as the entry point of the application?\*\*

- A) `Main()`

- B) `Start()`

- C) `Run()`

- D) `Execute()`

\*\*Answer:\*\* A) `Main()`

### \*\*Building an ASP.NET Core App\*\*

1. \*\*Which command is used to create a new ASP.NET Core web application?\*\*

- A) `dotnet new webapp`

- B) `dotnet new mvc`

- C) `dotnet new web`

- D) `dotnet new api`

\*\*Answer:\*\* A) `dotnet new webapp`

2. \*\*What is the purpose of the `Startup.cs` file in an ASP.NET Core application?\*\*

- A) To define the data models for the application

- B) To configure services and middleware used by the application

- C) To handle routing of HTTP requests

- D) To manage database connections

\*\*Answer:\*\* B) To configure services and middleware used by the application

3. \*\*In ASP.NET Core, which method is used to configure the HTTP request pipeline?\*\*

- A) `ConfigureServices()`

- B) `Configure()`

- C) `Startup()`

- D) `Initialize()`

\*\*Answer:\*\* B) `Configure()`

4. \*\*What is the default web server used by ASP.NET Core applications when running locally?\*\*

- A) IIS

- B) Apache

- C) Kestrel

- D) Nginx

\*\*Answer:\*\* C) Kestrel

**Chapter 2**

\*\*Introduction to Docker\*\*

1. \*\*What is Docker primarily used for in software development?\*\*

- A) Version control

- B) Containerization of applications and services

- C) Code review

- D) Performance monitoring

\*\*Answer:\*\* B) Containerization of applications and services

2. \*\*Which Docker component is used to build container images?\*\*

- A) Docker Engine

- B) Docker Hub

- C) Docker Compose

- D) Dockerfile

\*\*Answer:\*\* D) Dockerfile

3. \*\*What is the purpose of a Docker container?\*\*

- A) To manage database connections

- B) To provide a lightweight, isolated environment for running applications

- C) To handle web server configurations

- D) To centralize logging and monitoring

\*\*Answer:\*\* B) To provide a lightweight, isolated environment for running applications

4. \*\*Which command is used to run a Docker container from an image?\*\*

- A) `docker build`

- B) `docker create`

- C) `docker run`

- D) `docker push`

\*\*Answer:\*\* C) `docker run`

### \*\*Continuous Integration with Wrecker\*\*

1. \*\*What is Wrecker used for in a continuous integration pipeline?\*\*

- A) Container orchestration

- B) Code quality analysis

- C) Automating the build and testing of applications

- D) Managing deployment to production

\*\*Answer:\*\* C) Automating the build and testing of applications

2. \*\*Which of the following best describes the function of a CI/CD pipeline in Wrecker?\*\*

- A) It provides real-time analytics for application performance

- B) It automates the process of building, testing, and deploying code changes

- C) It handles user authentication and authorization

- D) It facilitates database schema migrations

\*\*Answer:\*\* B) It automates the process of building, testing, and deploying code changes

3. \*\*What is a common feature of Wrecker’s continuous integration process?\*\*

- A) Manual code reviews

- B) Automated tests and build processes triggered by code commits

- C) Centralized code storage

- D) Single server deployment

\*\*Answer:\*\* B) Automated tests and build processes triggered by code commits

### \*\*Continuous Integration with CircleCI\*\*

1. \*\*Which tool is used by CircleCI to define build, test, and deployment workflows?\*\*

- A) Dockerfile

- B) CircleCI Configuration File (.circleci/config.yml)

- C) Makefile

- D) Jenkinsfile

\*\*Answer:\*\* B) CircleCI Configuration File (.circleci/config.yml)

2. \*\*What is a key benefit of using CircleCI for continuous integration?\*\*

- A) It provides a user interface for managing database connections

- B) It integrates seamlessly with various version control systems and automates build and deployment processes

- C) It centralizes application logging

- D) It manages container orchestration

\*\*Answer:\*\* B) It integrates seamlessly with various version control systems and automates build and deployment processes

3. \*\*What is the primary purpose of the `workflows` section in a CircleCI configuration file?\*\*

- A) To define environment variables

- B) To specify the order of jobs and their dependencies

- C) To configure Docker images

- D) To set up build triggers

\*\*Answer:\*\* B) To specify the order of jobs and their dependencies

### \*\*Deploying to Docker Hub\*\*

1. \*\*What is Docker Hub used for?\*\*

- A) Building Docker containers locally

- B) Hosting and sharing Docker images

- C) Monitoring container performance

- D) Managing application deployments

\*\*Answer:\*\* B) Hosting and sharing Docker images

2. \*\*Which command is used to push a Docker image to Docker Hub?\*\*

- A) `docker pull`

- B) `docker push`

- C) `docker build`

- D) `docker run`

\*\*Answer:\*\* B) `docker push`

3. \*\*Before pushing a Docker image to Docker Hub, what must be done first?\*\*

- A) Log in to Docker Hub using `docker login`

- B) Create a Dockerfile

- C) Build the Docker image using `docker build`

- D) Run the Docker container locally

\*\*Answer:\*\* A) Log in to Docker Hub using `docker login`

4. \*\*What is required in the Dockerfile to ensure the image is tagged correctly before pushing to Docker Hub?\*\*

- A) `docker tag <image> <repository>:<tag>`

- B) `docker build <tag> <image>`

- C) `docker push <repository>/<image>`

- D) `docker run <tag> <image>`

\*\*Answer:\*\* A) `docker tag <image> <repository>:<tag>`

**Chapter 3**

\*\*Microservice\*\*

1. \*\*What is a primary characteristic of a microservice architecture?\*\*

- A) Monolithic design with tightly coupled components

- B) Decoupled services with a single codebase

- C) Independent, self-contained services that interact through APIs

- D) Centralized database management

\*\*Answer:\*\* C) Independent, self-contained services that interact through APIs

2. \*\*Which of the following is NOT a benefit of using microservices?\*\*

- A) Scalability of individual services

- B) Independent deployment of services

- C) Complex inter-service communication

- D) Technology and language heterogeneity

\*\*Answer:\*\* C) Complex inter-service communication

### \*\*Team Service\*\*

1. \*\*When designing a microservice for a specific team, what is a key consideration?\*\*

- A) The team's preference for a single database schema

- B) The service should encapsulate the team’s domain and have well-defined boundaries

- C) The team should use a monolithic architecture for simplicity

- D) The service should share its codebase with other teams

\*\*Answer:\*\* B) The service should encapsulate the team’s domain and have well-defined boundaries

2. \*\*How should a team service be structured for effective development and maintenance?\*\*

- A) With tightly coupled components and centralized configuration

- B) With clearly defined APIs and isolated service logic

- C) By sharing code and database with other services

- D) By using a single service for all team responsibilities

\*\*Answer:\*\* B) With clearly defined APIs and isolated service logic

### \*\*API First Development\*\*

1. \*\*What does API First Development emphasize?\*\*

- A) Building user interfaces before defining APIs

- B) Designing and defining APIs before implementing the backend logic

- C) Developing the backend logic before designing APIs

- D) Using a single API endpoint for all services

\*\*Answer:\*\* B) Designing and defining APIs before implementing the backend logic

2. \*\*Which tool can be used to define and document APIs in an API First Development approach?\*\*

- A) Swagger (OpenAPI)

- B) Docker Compose

- C) Kubernetes

- D) GitLab CI

\*\*Answer:\*\* A) Swagger (OpenAPI)

### \*\*Test First Controller\*\*

1. \*\*What is the main idea behind the Test First approach in controller development?\*\*

- A) Writing tests after the controller code is implemented

- B) Designing and writing tests before implementing the controller code

- C) Implementing the controller without testing

- D) Using mock data instead of real tests

\*\*Answer:\*\* B) Designing and writing tests before implementing the controller code

2. \*\*Which testing method is commonly used to test ASP.NET Core controllers in isolation?\*\*

- A) Integration Testing

- B) End-to-End Testing

- C) Unit Testing

- D) Load Testing

\*\*Answer:\*\* C) Unit Testing

### \*\*Creating a CI Pipeline\*\*

1. \*\*What is the primary purpose of a CI (Continuous Integration) pipeline?\*\*

- A) To deploy code changes directly to production

- B) To automate the process of building, testing, and integrating code changes

- C) To manually review code changes before integration

- D) To handle user authentication and authorization

\*\*Answer:\*\* B) To automate the process of building, testing, and integrating code changes

2. \*\*Which of the following tools is commonly used to set up a CI pipeline for ASP.NET Core projects?\*\*

- A) Jenkins

- B) Visual Studio Code

- C) Docker Desktop

- D) Postman

\*\*Answer:\*\* A) Jenkins

### \*\*Integration Testing\*\*

1. \*\*What is the goal of integration testing in the context of microservices?\*\*

- A) To test individual components in isolation

- B) To verify that different microservices and components work together as expected

- C) To perform end-to-end user interface testing

- D) To check performance under load

\*\*Answer:\*\* B) To verify that different microservices and components work together as expected

2. \*\*Which framework is often used for integration testing in ASP.NET Core applications?\*\*

- A) xUnit

- B) NUnit

- C) MSTest

- D) Xunit combined with TestServer for ASP.NET Core

\*\*Answer:\*\* D) Xunit combined with TestServer for ASP.NET Core

### \*\*Running the Team Service Docker Image\*\*

1. \*\*What is the purpose of creating a Docker image for a microservice?\*\*

- A) To define the service's data model

- B) To package the microservice with its dependencies into a portable container

- C) To manage source code versioning

- D) To create a user interface for the service

\*\*Answer:\*\* B) To package the microservice with its dependencies into a portable container

2. \*\*Which command is used to build a Docker image from a Dockerfile?\*\*

- A) `docker run`

- B) `docker push`

- C) `docker build`

- D) `docker pull`

\*\*Answer:\*\* C) `docker build`

3. \*\*How do you run a Docker container from a built image?\*\*

- A) `docker build <image>`

- B) `docker push <image>`

- C) `docker run <image>`

- D) `docker pull <image>`

\*\*Answer:\*\* C) `docker run <image>`