Microservices Basics

Instructor Notes:

Add instructor notes here.

Microservices Basics

Instructor Notes:

This lesson is to give an Introduction on Java Server Pages

Lesson Objectives

In this lesson, you will learn:

- What is Service ?
- What is Monolithic Architecture
- Introduction to Microservice
- Emergence of Microservice
- Benefits of Microservice Architecture
- Diff between Microservice and Monolithic
- Challenges in Microservices

What is Service ?
What is Monolithic Architecture
Introduction to Microservice
Emergence of Microservice
Benefits of Microservice Architecture
Diff between Microservice and Monolithic
Challenges in Microservices

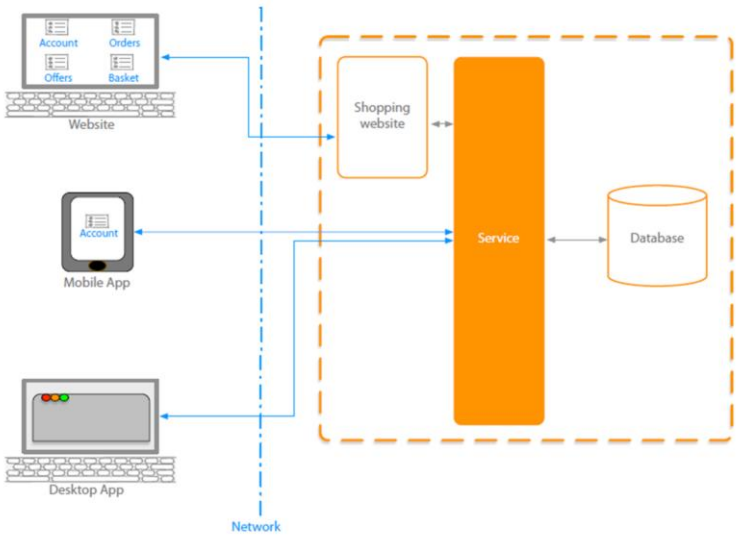
Microservices Basics

Instructor Notes:

Discuss the features of JSP. Explain how a JSP page is processed.

1.1: What is Service ?

What is Service ?



Service is a piece of SW which provides the functionalities to the other pieces of SW . It could be

- CRUD app service
- Validation service etc.

Service should be implemented in a SOA where the functionality should be reusable and can communicate to Client application and other services.

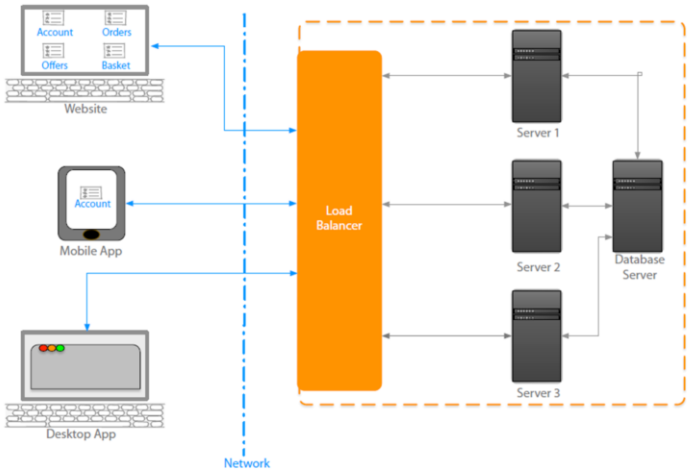
Microservices Basics

Instructor Notes:

Discuss the features of JSP. Explain how a JSP page is processed.

1.1: What is Service ?
What is Service – Big picture

- In a distributed application , SOA implemented Service can be distributed among multiple server with proper scalability and can be reuse by any type of application .



In a distributed application , SOA implemented Service can be distributed among multiple server with proper scalability and

Can be reuse by any type of application .

When the traffic come for service , then the Load balancer can redirect the traffic to the different server instance where the service is running

We can have multiple instance of the service will be running on Different web server.

The few important characteristics of services are

- It is stateless : - Request come for a single service instance doesn't require to remember the previous request.
- It provides a common interface to all type of application where the functionality is declared and encapsulate the implementation from client

When the demand increases , we can increase the number of instances of the service

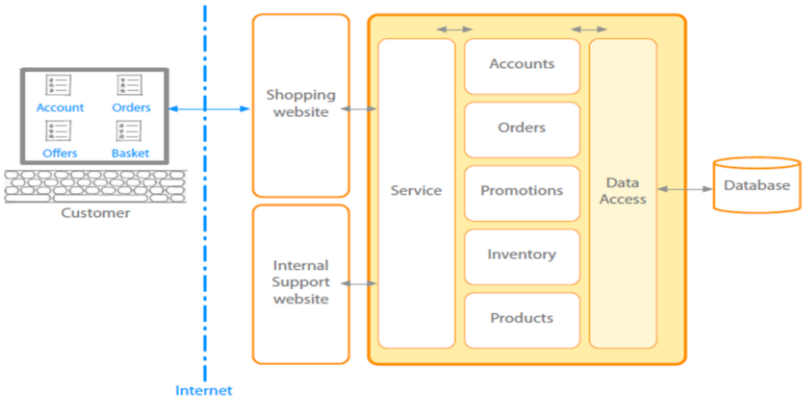
Microservices Basics

Instructor Notes:

1.2: What is Monolithic Architecture

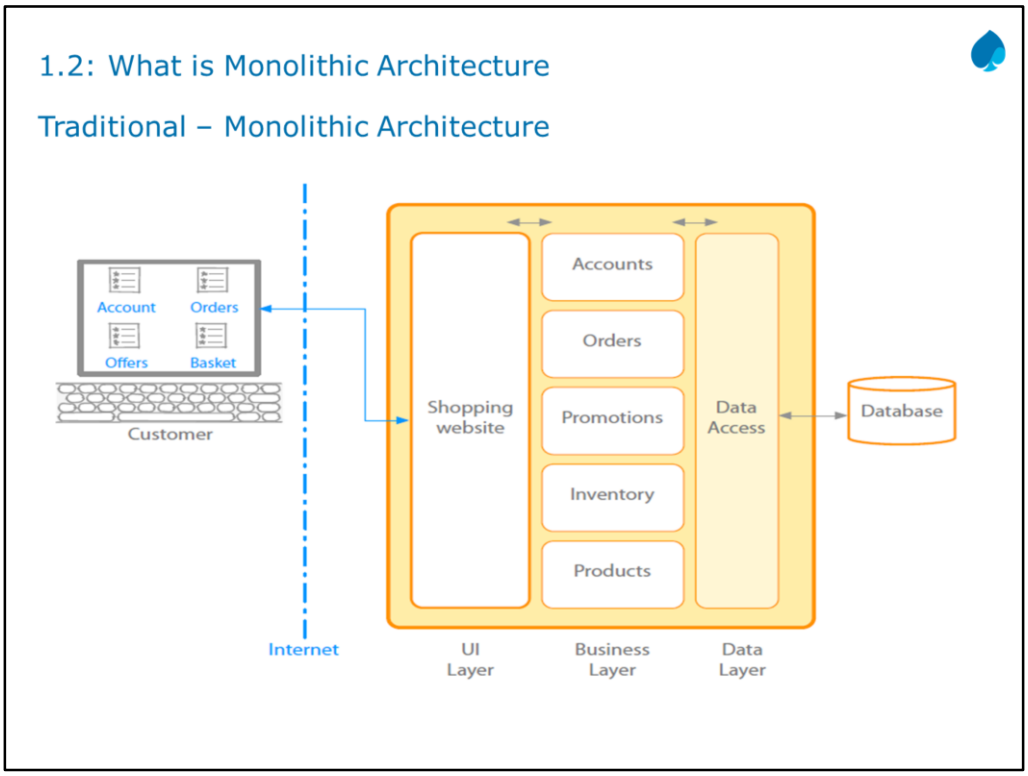
Traditional – Monolithic Architecture

The Monolithic architecture is an architectural style which structures the complete application into one Executable component.



Microservices Basics

Instructor Notes:



It is good for small Application where the below three itility can be easily maintained.

- Maintainability
- Testability
- Deployibility

Monolithic software is designed to be self-contained; components of the Program which are interconnected and interdependent rather than loosely coupled.

Microservices Basics

Instructor Notes:

1.2: What is Monolithic Architecture
Challenges – Monolithic Architecture

- A typical enterprise application
- No restriction on Size
- Large codebase
- Longer development times
- Inaccessible features
- Fixed technology stack
- High level of coupling between Modules and services
- Failure could effect whole system
- Single service on server
- Minor change could result in complete rebuild

A successful application always try to grow where the Monolithic architecture faced the problem

Monolithic Architecture is
- A typical enterprise application
- No restriction on Size
- Large codebase
- Longer development times
- Inaccessible features
- Fixed technology stack
- High level of coupling between Modules and services
- Failure could effect whole system
- Single service on server
- Minor change could result in complete rebuild

Challenges started where the application became bigger in size .
Agile development and deployment became very difficult or may be impossible.

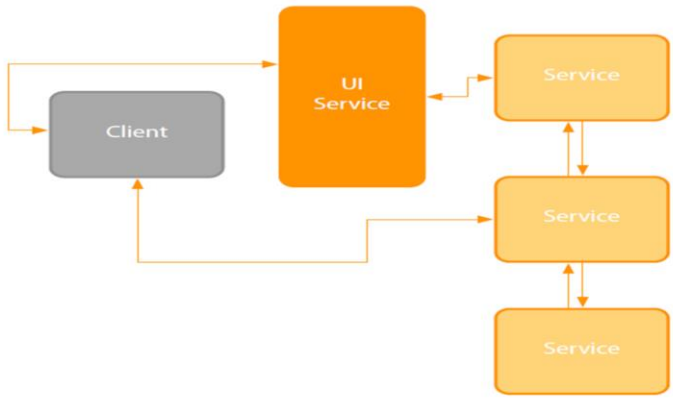
In a tightly-coupled architecture, each component and its associated components must be present in order for code to be executed or compiled.
if any program component must be updated, the whole application has to be rewritten

Microservices Basics

Instructor Notes:

1.3: Introduction to Microservice Architecture
Microservice Architecture

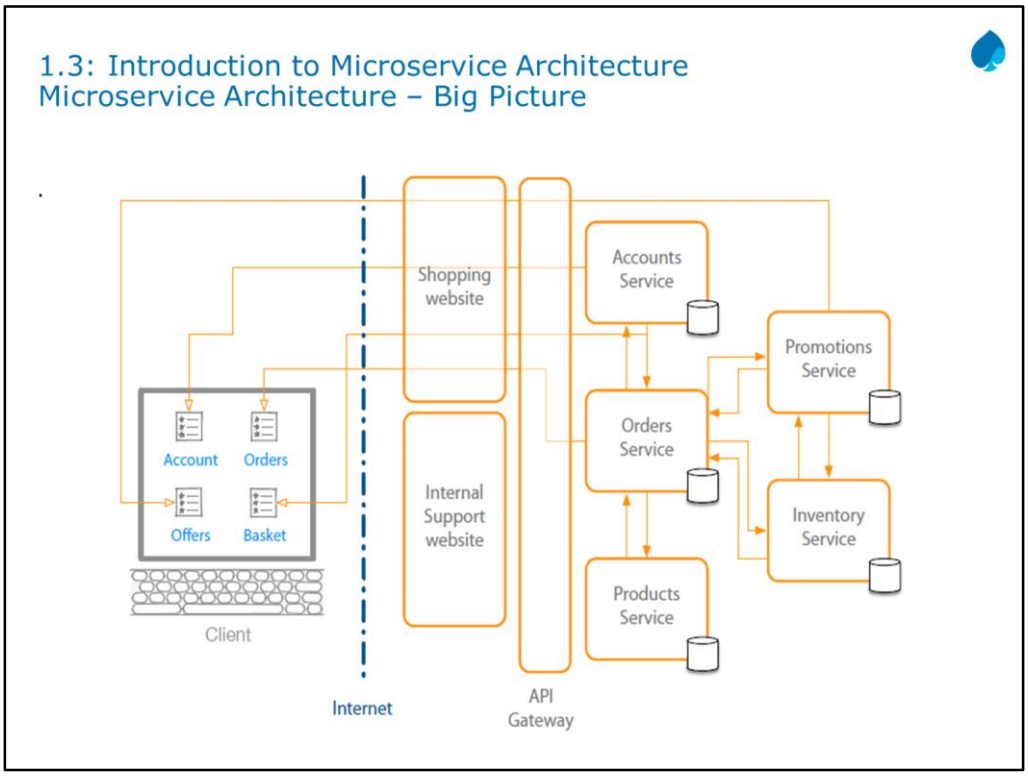
- This is a kind of architectural style which structures an application as a set of loosely coupled , services organized around the business capabilities.
- It is a improved version of Software oriented Architecture



- **SOA done well**
  - Knowing how to size a service
  - Traditional SOA resulted in monolithic services
- **Micro sized services provide**
  - Efficiently scalable applications
  - Flexible applications
  - High performance applications
- **Application(s) powered by multiple services**
- **Small service with a single focus**
- **Lightweight communication mechanism**
  - Both client to service and service to service
- **Technology agnostic API**
- **Independent data storage**
- **Independently changeable**
- **Independently deployable**
- **Distributed transactions**
- **Centralized tooling for management**

Microservices Basics

Instructor Notes:



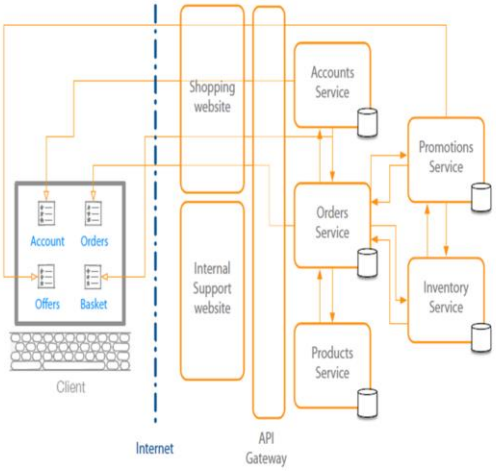
- Service is a independently Deployable component and certainly implementing some business rules.
- Each service should have its own database which is very important from loos coupling point of view.
- Microservices are as per the Business requirement .
- One can design small Micro business services independently .
- All Microservices are loosely coupled So changes in one will not effect the other Service .
- In Implementation view we can have multiple service component .war files.

Microservices Basics

Instructor Notes:

1.4: Emergence of Microservice
Emergence Of Microservices

- Need to respond to change quickly
- Need for reliability
- Business domain-driven design
- Automated test tools
- Release and deployment tools

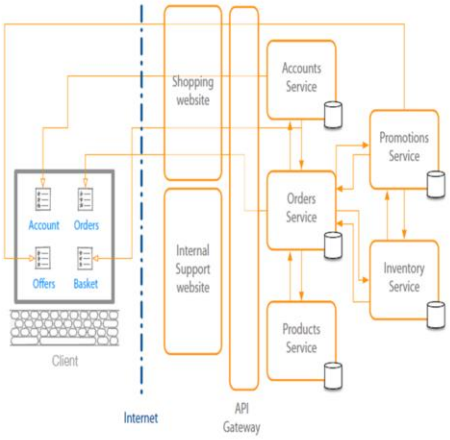


Microservices Basics

Instructor Notes:

1.4: Emergence of Microservice
Emergence Of Microservices

- On-demand hosting technology
- On-line cloud services
- Need to embrace new technology
- Asynchronous communication technology
- Simpler server side and client side techno



Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture

Benefits of Microservices

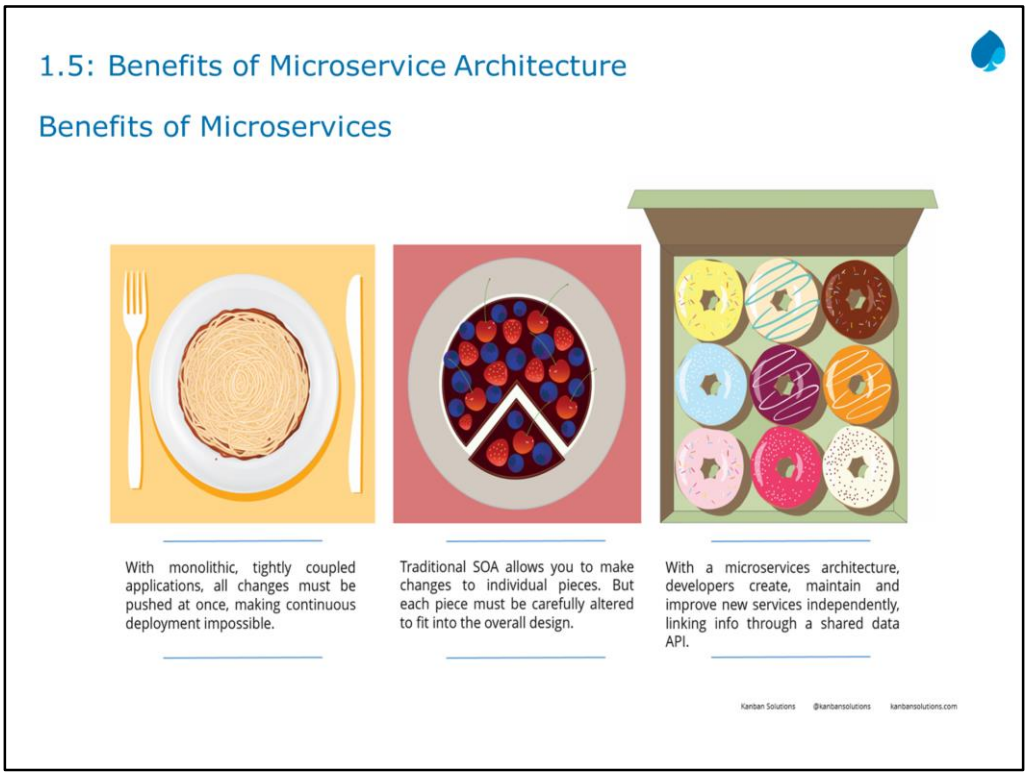
It enables the continuous delivery and deployment of large, complex applications.

- Better testability
- Better deployibility
- It enables you to organize the development effort around multiple teams.
- Each team(of 2 to 5 members) owns and is responsible for one or more single service.
- Each team can develop, deploy and scale their services independently of all of the other teams

Shorter development times
•Reliable and faster deployment
•Enables frequent updates
•Decouple the changeable parts
•Security
•Increased uptime
•Fast issue resolution
•Highly scalable and better performance
•Better ownership and knowledge
Right technology
•Enables distributed teams

Microservices Basics

Instructor Notes:



Shorter development times
- Reliable and faster deployment
- Enables frequent updates
- Decouple the changeable parts
- Security
- Increased uptime
- Fast issue resolution
- Highly scalable and better performance
- Better ownership and knowledge

Right technology
- Enables distributed teams

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Benefits of Microservices

- Each Microservice is relatively small.
  - For Developer , it is easy to understand
  - Because of small size ,the IDE is faster making developers more productive
  - The application starts faster which makes the developer speeds up the deployment .
- It allows the easy and flexible way to integrate automatic deployment with Continuous integration tools(Jenkin , Hudson etc.)
- It improved fault isolation.
- Eliminates any long-term commitment to a technology stack

Enables the continuous delivery and deployment of large, complex applications.

Better testability - services are smaller and faster to test

Better deployability - services can be deployed independently

It enables you to organize the development effort around multiple, auto teams. It enables you to organize the development effort around multiple teams. Each (two pizza) team is owns and is responsible for one or more single service. Each team can develop, deploy and scale their services independently of all of the other teams.

Each microservice is relatively small

Easier for a developer to understand

The IDE is faster making developers more productive

The application starts faster, which makes developers more productive, and speeds up deployments

Improved fault isolation. For example, if there is a memory leak in one service then only that service will be affected. The other services will continue to handle requests. In comparison, one misbehaving component of a monolithic architecture can bring down the entire system.

Eliminates any long-term commitment to a technology stack. When developing a new service you can pick a new technology stack. Similarly, when making major changes to an existing service you can rewrite it using a new technology stack.

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Benefits of Microservices

It has code for Business logic only ,no mixup with Html,CSS and other UI components

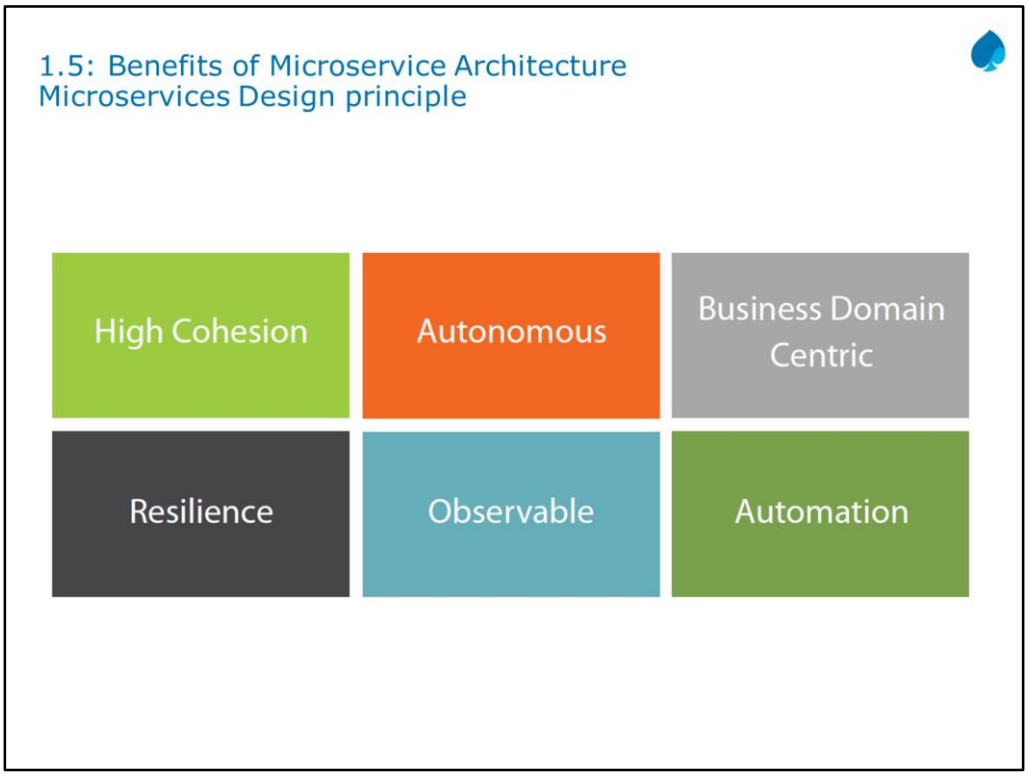
It can be deployed on commodity hardware or low/medium configuration server.

Easy to integrate with 3rd party service.

Each microservice has it's own Database but it depends upon requirement .One can use common DB i.e. Oracle , mySql for all service.

Microservices Basics

Instructor Notes:

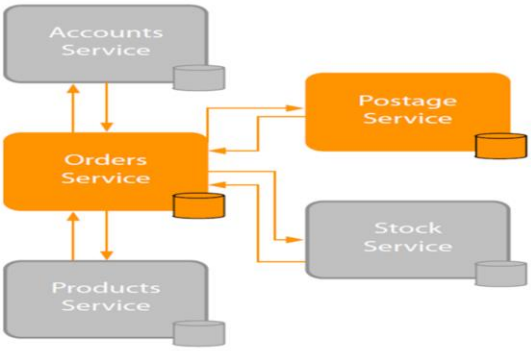


Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Microservices Design principle – High Cohesion

- It gives you scalability , flexibility and Reliability



Single focus
Single responsibility
- SOLID principle
- Only change for one reason

- Reason represents
  - A business function
  - A business domain
- Encapsulation principle
  - OOP principle
- Easily rewritable code

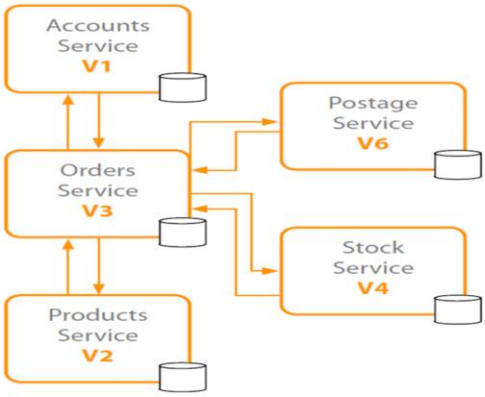
Why
- Scalability
- Flexibility
- Reliability

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Microservices Design principle - Autonomous

- Loose Coupling
- Stateless
- Concurrent development



Loose coupling
- Honor contracts and interfaces

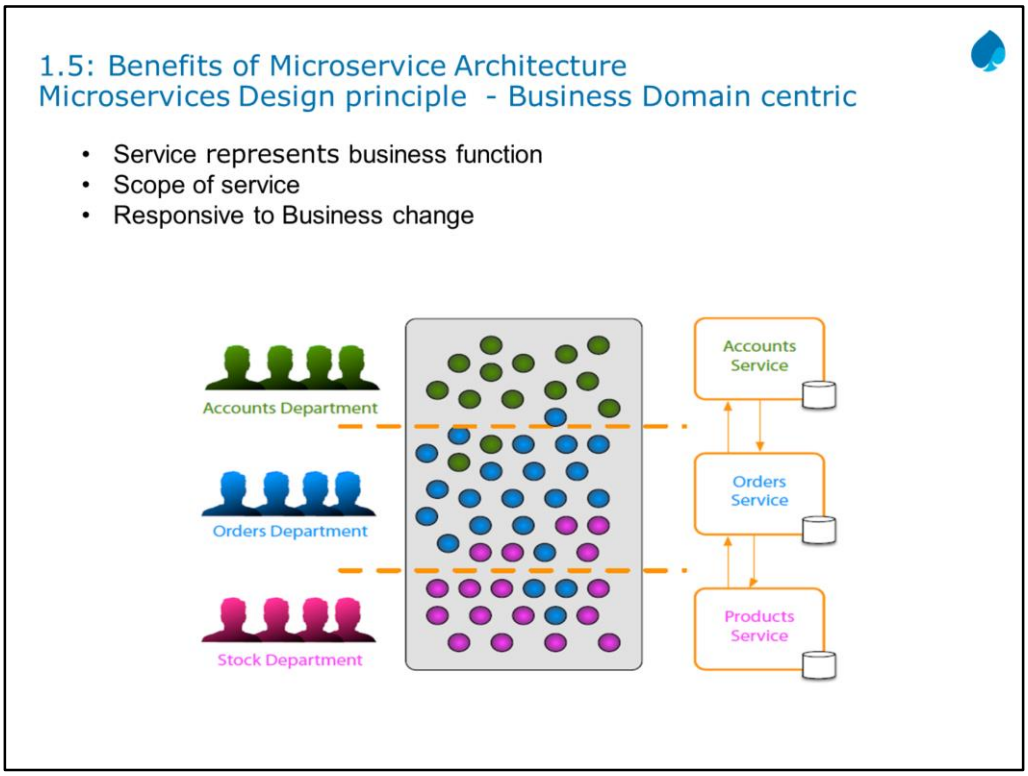
Stateless
- Independently changeable

Independently deployable
- Backwards compatible

Concurrent development

Microservices Basics

Instructor Notes:



Service represents business function
•Accounts Department(Specific SW code for Accounting related)
•Postage calculator (Business functionality)

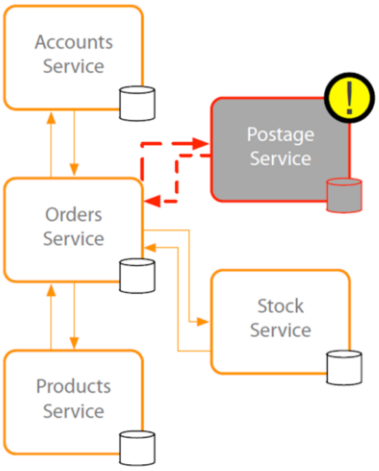
Scope of service
Bounded context from DDD(Domain Driven Design)
Identify boundaries\seams
Shuffle code if required
•Group related code into a service
•Aim for high cohesion
Responsive to business change(Based 0n Business change , microservice also change accordingly)

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Microservices Design principle - Resilience

- If one service fails , It didn't disturb the process of other system
- Design for known failure
- Degrade functionality on failure detection
- Default functionality on failure detection



Embrace failure
- Another service
- Specific connection
- Third-party system

Degrade functionality
Default functionality
Multiple instances
- Register on startup
- Deregister on failure

Types of failure
- Exceptions\Errors
- Delays
- Unavailability

Network issues
- Delay
- Unavailability

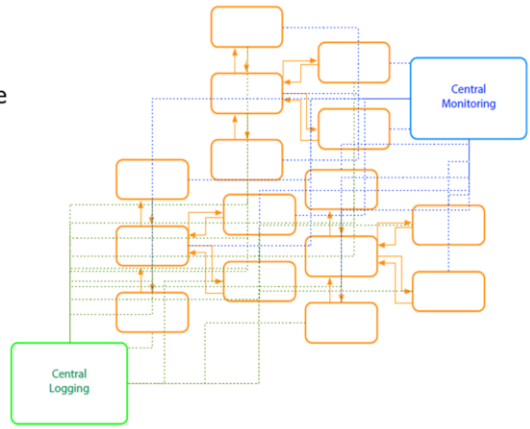
Validate input
- Service to service
- Client to service

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Microservices Design principle - Observable

- Centralized Monitoring
  - Real time monitoring
  - Monitor The host
    - CPU , Memory , Disk usage etc
  - Expose Metrics within the service



System Health
- Status
- Logs
- Errors

Centralized monitoring

Centralized logging

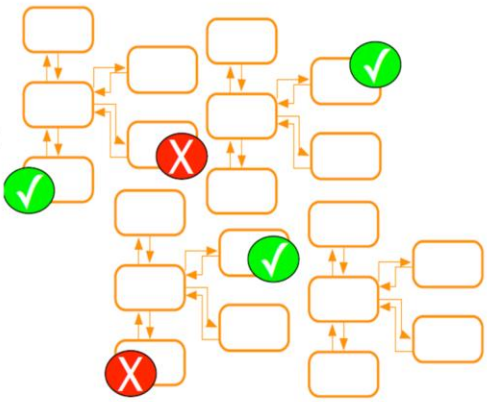
Why
- Distributed transactions
- Quick problem solving
- Quick deployment requires feedback
- Data used for capacity planning
- Data used for scaling
- What's actually used
- Monitor business data

Microservices Basics

Instructor Notes:

1.5: Benefits of Microservice Architecture
Microservices Design principle - Automation

- It use tools to do testing and provide feedback automatically
- CI tools
  - Work with Source control system
  - Automatic after check in
  - Unit testing and Integration testing required for production code.



Tools to reduce testing
- Manual regression testing
- Time taken on testing integration
- Environment setup for testing

Tools to provide quick feedback
- Integration feedback on check in
- Continuous Integration

Tools to provide quick deployment
- Pipeline to deployment
- Deployment ready status
- Automated deployment
- Reliable deployment
- Continuous Deployment

Why
- Distributed system
- Multiple instances of services
- Manual integration testing too time consuming
- Manual deployment time consuming and unreliable

Microservices Basics

Instructor Notes:

1.6: Diff between Microservice and Monolithic
Microservice Vs Monolithic

Monolithic	Microservices
Single deployment	Every microservice is a separate deployment. Many more modules with specific functionality and it ensures Module independency.
This type of application puts all its functionality into a single process.	This architecture puts each element of its functionality into a separate service.
Single codebase	More than one code base
Interaction between classes are synchronous	

Microservices Basics

Instructor Notes:

1.6: Diff between Microservice and Monolithic
Microservice Vs Monolithic

Monolithic	Microservices
The whole app has one DB.	Every microservice has its own DB.
It scales by replicating the monolith on multiple servers	It scales by distributing these services across servers and replicating as needed.
Any changes in application requires a full redeploy	It is loosely coupled, any changes in one service will not effect another service.
Hard to work with multiple teams	It is easy , as the size of the service is small and independent.
If the application is big then Hard to manage	It separate deployment ,separate monitoring

Microservices Basics

Instructor Notes:

1.6: Diff between Microservice and Monolithic
Microservice Vs Monolithic

Monolithic	Microservices
Continuous deployment is difficult .	Continuous deployment is easy.
Bug in any module (e.g. memory leak) can potentially bring down the entire process. Here all instances of the application are identical , so that bug will impact the availability of the entire application.	Bug in one service can only effect that service. It can't effect other service.
Monolithic applications has a barrier to adopting new technologies.	Microservice supports to adopt the new technology.

Microservices Basics

Instructor Notes:

1.6: Challenges in Microservices
Challenges / Drawbacks - Microservices

- **Challenges**
  - It brings a lot of operation overhead.
  - Because of distributed deployment , default to trace problem.
  - Complicated to manage whole products where number of servi increase.
- **Drawbacks**
  - Developer tools/IDEs are oriented on building monolithic applications and don't provide explicit support for developing distributed applications.
  - The microservice architecture replaces N monolithic application instances with NxM services instances. If each service runs in its own JVM (or equivalent), which is usually necessary to isolate the instances, then there is the overhead of M times as many JVM runtimes.

Microservices Basics

Instructor Notes:

1.3: Introduction to Microservice Architecture
Microservices implementation stack

- Spring Boot and Spring cloud
- Microservice with Python
- Microservice with PHP
- Prometheus Microservices
- Netflix Microservices Monitoring
- Fabric 8
- Dropwizard
- Netty
- jackson

Microservices Basics

Instructor Notes:

Summary

In this lesson, you have learnt:

- What is Service ?
- What is Monolithic Architecture
- Introduction to Microservice
- Emergence of Microservice
- Benefits of Microservice Architecture
- Diff between Microservice and Monolithic
- Challenges in Microservices

Microservices Basics

Instructor Notes:

Answers for the Review Questions:

Answer 1: Service

Answer 2: Monolithic service

Answer 3: True

Review – Questions

Question 1: _____ is a piece of SW which provides the functionalities to the other pieces of SW

Question 2: _____ architecture is an architectural style which structures the complete application into one Executable component .

Question 3: Service is Stateless .

- True
- False

Microservices Basics

Instructor Notes:

Answers for the Review Questions:

Answer 4: Microservice

Answer 5: All of the above

Answer 6: All of the above

Review – Questions

Question 4: _____ architecture is a Service-oriented architecture, that is composed of loosely coupled elements that have bounded contexts.

Question 5: Which of the followings important ilities which directly impact on development velocity

- scalability
- Deployibility
- Maintainability
- All of the above

Question 6: Which of the followings are the microservice implementation stack

- Spring boot
- Netflix Microservices Monitoring
- All of the above