# Web Service Intro


IT Learning &
Outsourcing Center

Lector: Dimitar Topuzov

www.pragmatic.bg

E-mail: dtopuzov@gmail.com
LinkedIn: http://bg.linkedin.com/pub/dimitar-topuzov/18/470/833/en

# Content




- What is Web Service
- Web Service Concepts
- Why Web Services
- Multitier Software Architectures
  - Web Services in context of Multitier Software Systems
  - The place of Web Services in such systems
- Testing Web Services
  - Why we test Web Services
  - Types of tests we can perform

# What is a Service

- In the real world a "service" is:
  - A piece of work performed by a service provider
  - Provides a client (consumer) some desired result by some input parameters
  - Has quality characteristics (price, execution time, constraints, etc.)

# Example



# What is a Web Service




- Web services are services that can be accessed over a network
- Takes some input, do some work and produces some output
- Request-Response model: Client request, server responses

# What is a Web Service




*" A software system designed to support interoperable machine-to-machine interaction over a network… "*

*W3C definition*

# ..in Bulgarian Please



*"Уеб услугата представлява софтуерна система, която предоставя комуникация между взаимносъвместими компютърни системи по компютърни мрежи"*

*W3C definition*
*Translated in Bulgarian By Wikipedia*

# IBM Definition

"Web services are self-describing, self-contained, modular applications that can be mixed and matched with other Web services to create innovative products, processes and value chains. Web services are Internet applications that fulfill a specific task or a set of tasks that work with many other Web services in an interoperable manner to carry out their part of a complex work flow or a business transaction."

# Sun Microsystems Definition



"A Web service describes specific business functionality exposed by a company, usually through an Internet connection, for the purpose of providing a way for another company or software program to use the service."

# Microsoft Definition

"A Web service is a unit of application logic providing data and services to other applications. Applications access Web services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web service is implemented. Web services combine the best aspects of component-based development and the Web."

# Conclusion

- Definitions above are different, but one thing is sure – Three of the major players in the industry are talking positively about the same technology
  - This in itself means the idea of Web Services is worth
  - All proprietary technologies of those companies (like Microsoft Common Object Model (COM) for example) are dead and Web Services are here to stay

# Web Service Concepts

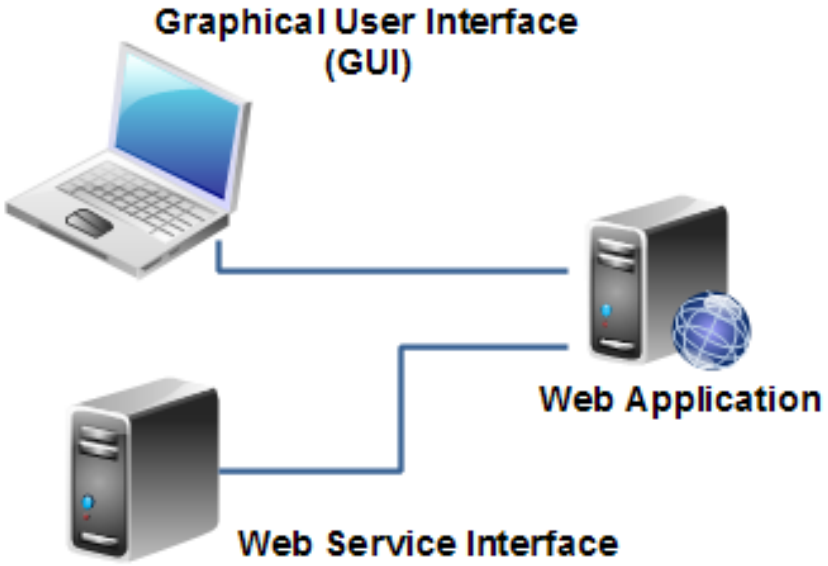


- Communication through standard protocols
  - HTTP, FTP, SMTP (for transport)
  - XML, JSON, RSS (for data)
- Autonomous
  - Each service operates autonomous without aware that other services exists
- Stateless
  - Do not remember a durable state between requests

# Stateless Example



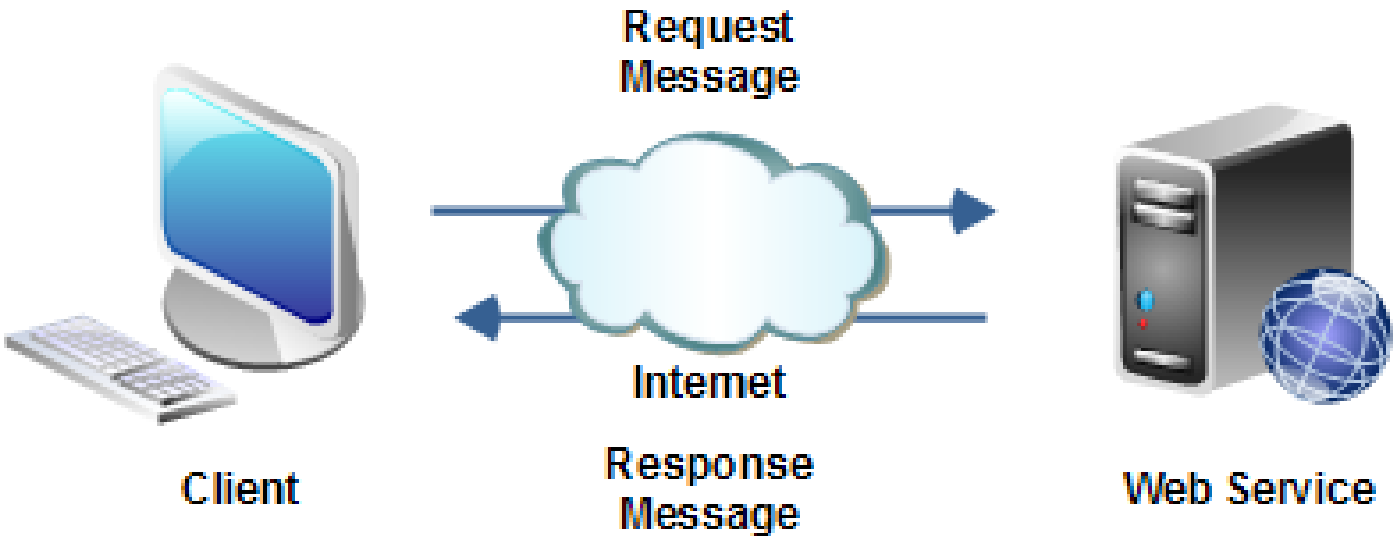
# Web Services vs. Websites




- Web application are designed to be accessed by humans via Web Client.
- Web services are intended to be used by machine (other software).

# Web Service and Client






.NET Web Application

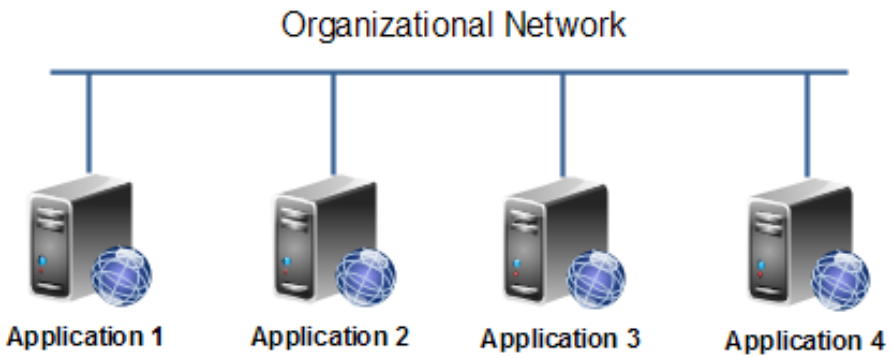
Runs on Window

Weather service implemented in Java

Runs on Linux

# Why Web Services

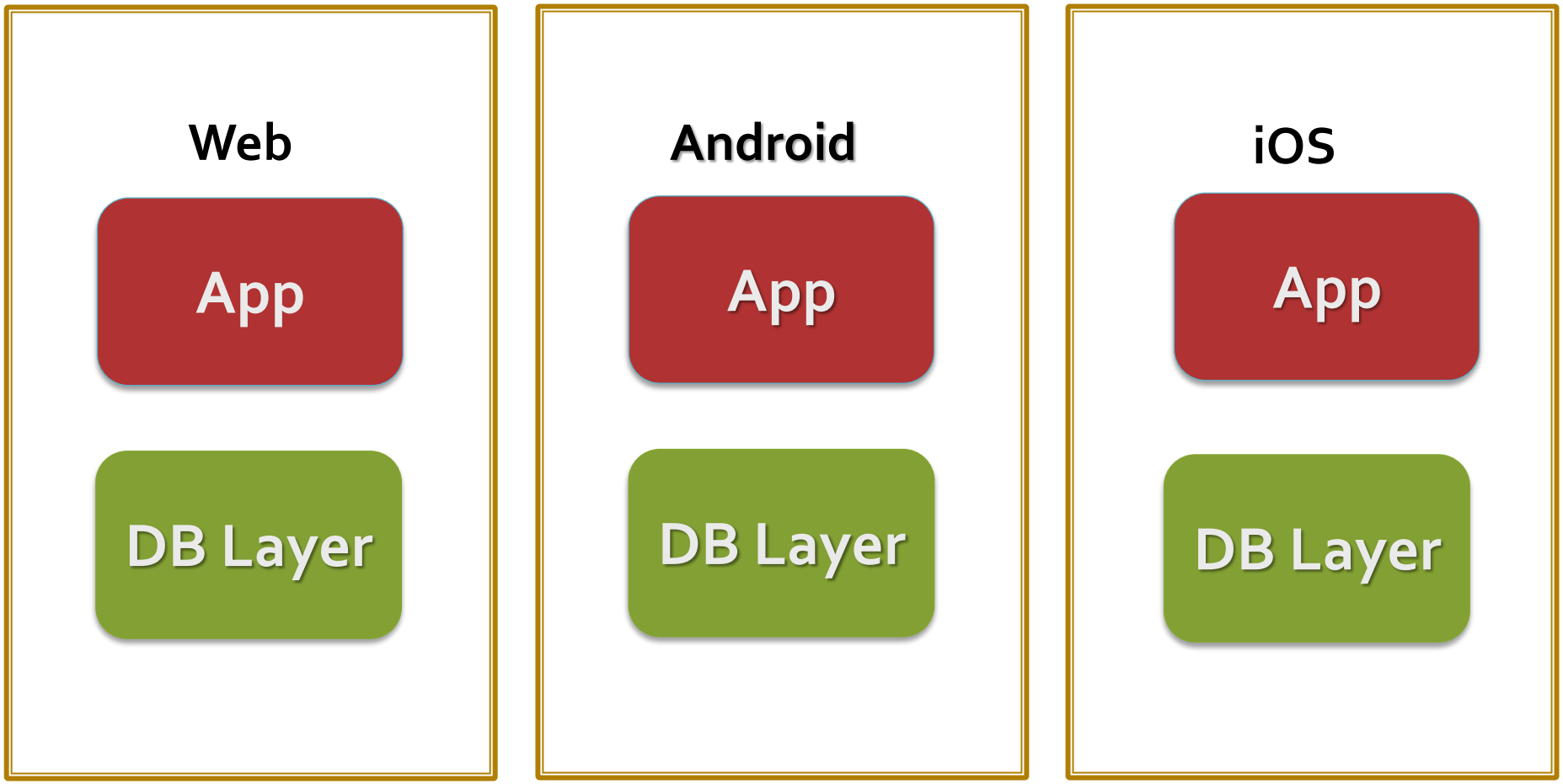
- Connect existing software
  - Web services help solve the interoperability problem
  - Giving different applications a way to link their data
  - Using Web services you can exchange data between different applications and different platforms



# Why Web Services

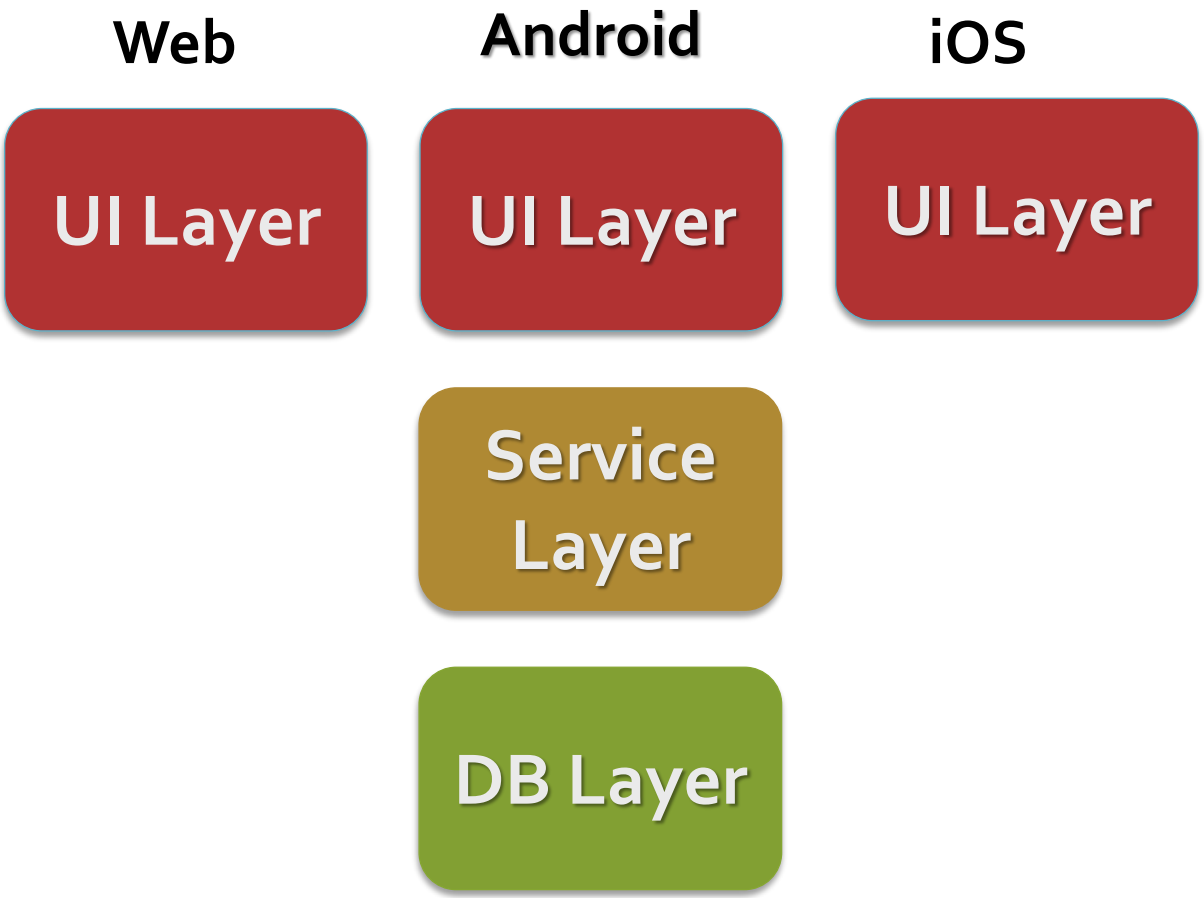
- Reuse application components
  - Ideally, there will only be one type of each application component, and anyone can use it in their application

# Monolithic Apps



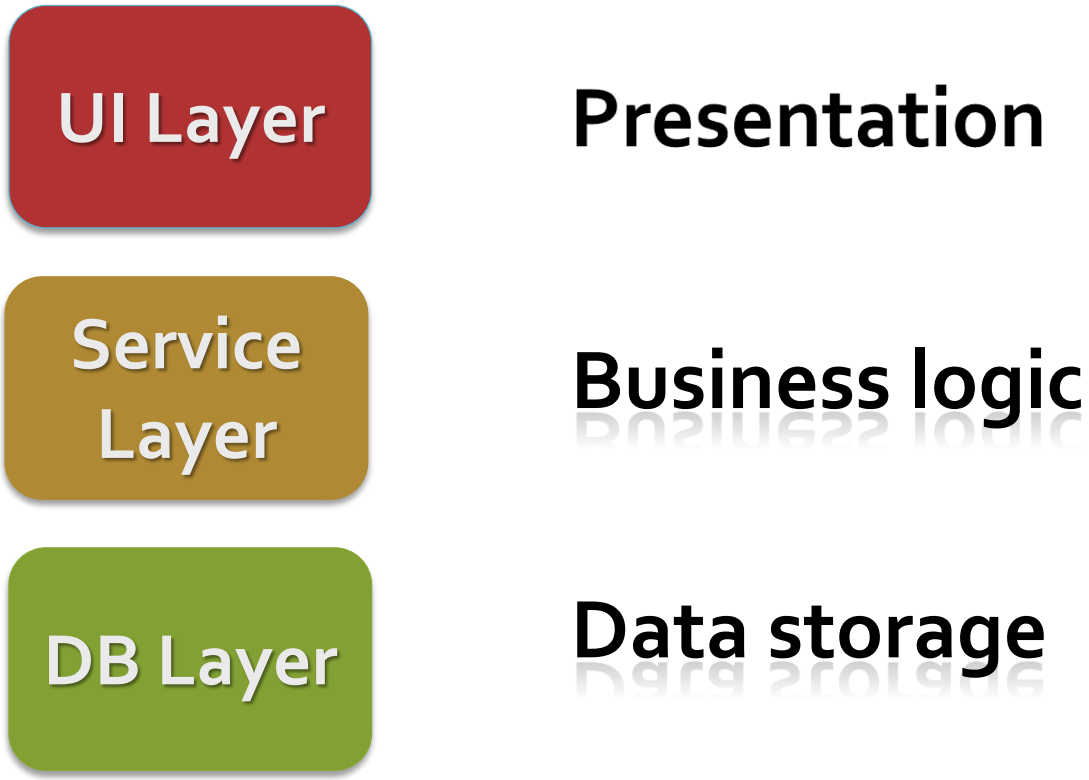
# Multitier Architecture





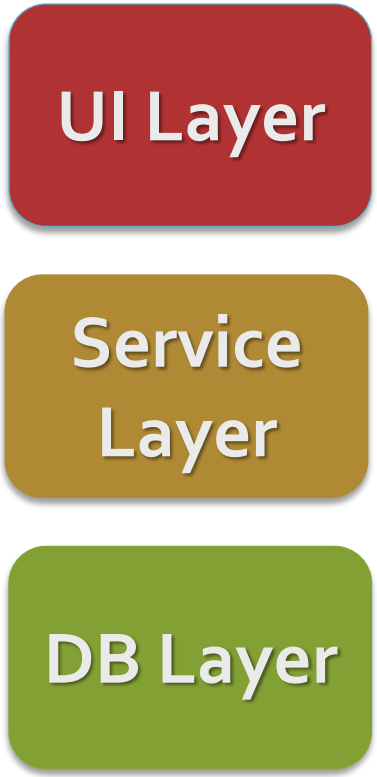
# Multitier Architecture





# Testing Service Layer

- Benefits:
  - Good way to test business logic
  - Faster tests
  - Less fragile
  - Lower Maintenance
- Disadvantages:
  - Not testing what user actually use and see



# Functional Testing

- Based on tools
  - Due to the fact that web services are intended to be consumed by machines it is hard a human to consume and test them directly
- We can test
  - If response message is in correct format
  - If response message has correct data
  - If status code of response message is correct

# Performance Testing



- Based on tools
  - Same as testing Web or other software performance can't be measured with out tools
- We can test performance in terms of
  - Time web service need to do the actual work and return a response
  - Time between sending a request and receiving a response (including time for transfer over the network)

# Load Testing

- Based on tools
  - It is hard to simulate a lot of simultaneous users without tools
- Test the performance under load of X simultaneous clients
  - We can measure the same metric as those we measure during performance testing
- Monitor the servers during load testing is must!
  - Memory usage
  - CPU usage
  - Disk I/O operations

# Security Testing

- In order to test Security we can test
  - Authentication
  - Message Confidentiality
    - Data elements meant to be kept confidential must be encrypted
  - Content Validation
    - Web services need to validate input before consuming it
  - XML Denial of Service Protection
    - Validation against recursive payloads
    - Validation against oversized payloads

# Usability Testing

We can not test usability in terms of “how it looks”, but we still can test if it is easy to perform some operation or get data via Web Service.

# Questions