Web Services Development

1. Introduction

Frank Walsh

Agenda

- Me
- Module Outline
- What is a Web Service?
- Brief History of Web Services
- Current trends in Web Services

Me

- Frank Walsh
- Room F21
- fxwalsh@wit.ie
- Contact/message me via Moodle

Module Outline(Tentative)

- Introduction to Web Services
- Javascript overview
- Representational State Transfer
- Web APIs
- Microservices

Technical Content(Tentative)

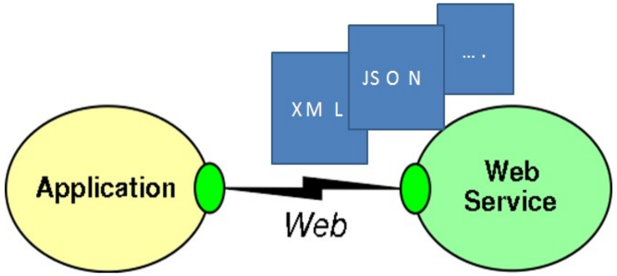
- Javascript
- Node.js
  - Express package
- NoSQL data stores
  - Mongo/Dynamo
- Cloud Platforms
  - Amazon/Azure/Bluemix

Module Assessment Structure

- 50% Continuous Assessment
  - 2 assignments
- 50% Exam
- Content administered via Moodle
- All assignment submission though Moodle
- Late assignments will be deducted 20% per day.
  - Please submit on time
  - Extensions only with proper certification.

What's a Web Service

- Apps that communicate over the World Wide Web(the www)
- Characteristics:
  - Based on open standards.
  - Interoperation between apps running on a varaiety of platforms and devices.
  - Web services can be combined to perform more complex processes.

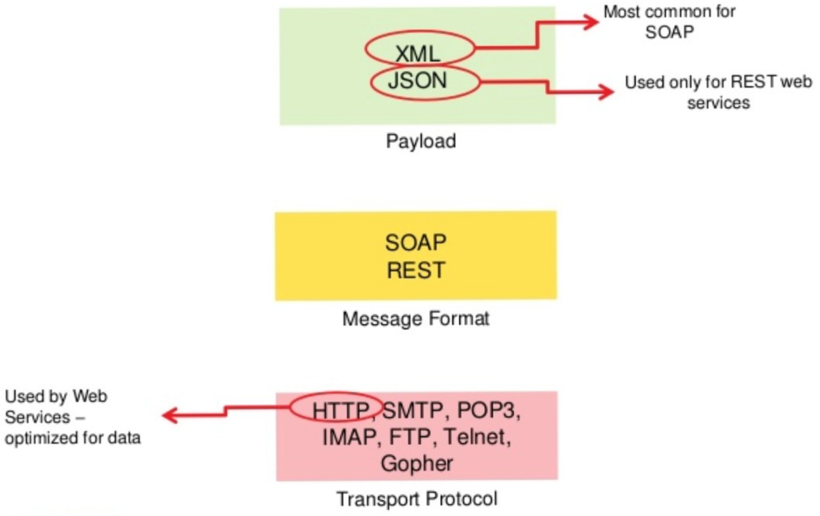


WWW

- World Wide Web resulted in simple, straightforward technologies
-
- Limiting web use to browsers restricts potential applications
- What about using web infrastructure for richer interoperability between clients and servers?
- Allows provision of services that integrate several other services
    - Even across different organisations

Open Standards

- Web Services should “support inter-operable machine-to-machine interaction over a network“ using open protocols
- Examples of typical open sandards used in Web Service Development are:
  - HTTP
  - XML, JSON
  - SOAP



XML
JSON
Most common for SOAP
Used only for REST web services
Payload
SOAP
REST
Message Format
Used by Web Services – optimized for data
HTTP, SMTP, POP3, IMAP, FTP, Telnet, Gopher
Transport Protocol

Web Service Interoperability

- Different languages on different platforms can communicate with each other in a open standards-based way.
  - Previously had several proprietry middleware solutions:
    - Microsoft Distributed Common Object Model(DCOM)
    - CORBA
    - Java Remote Method Invocation(RMI)
- Helps to address heterogeneity in DS systems.
- Component nature of web services promotes flexibility, modularity.

Brief History

- Web services evolved from older, legacy technologies used to develop Distributed Systems
  - RPC, ORPC (DCOM, CORBA and JAVA RMI)
- These old solutions had the following issues:
  - Interoperability
  - Firewalls
  - Complexity

Brief History - Interoperability

- Old Distributed systems had interoperability issues because vendor implemented its own proproetry technology for interprocess communciation.
  - DCOM apps strictly bound to Windows Operating system.
  - RMI bound to Java programming language.

Brief History - Firewalls

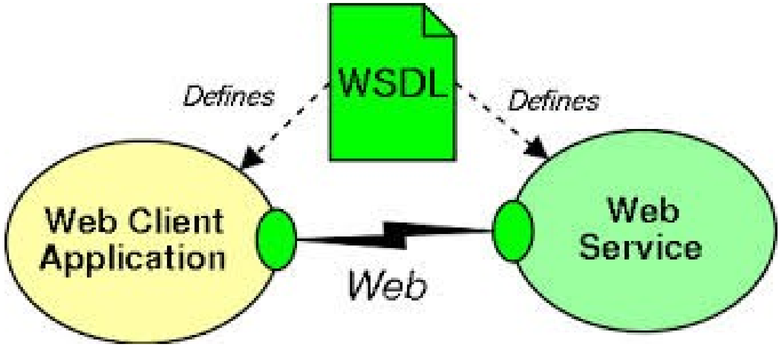
- Collaboration across organisations was an issue because distributed system middleware such as CORBA and DCOM used non-standard ports.
    - Firewalls usually don't like this and will block.
- Web Services use HTTP as a transport protocol
    - firewalls tend to allow access though port 80 (HTTP), leading to easier cross-organisation collaboration.

Brief History - Middleware

- Old middleware technologies such as RMI, COM, and CORBA are all different.
- New technologies and languages have to be learnt to implement these services.
  - Usually requires specialised “integrators” to deign/implement.
- Adds complexity and expense to

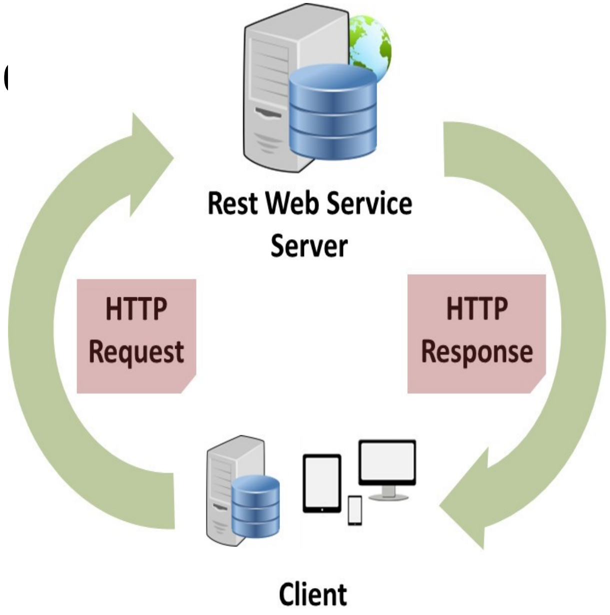
Brief History – XML/Big Web Services

- Sometimes known as SOAP-based or XML or WSDL web services (or any combination of).
  - Standards based
- Typically accessed by software
- Access is programmatic unlike access to web resource by browser(if it is in browser it's via Javascript/AJAX client).
- The reason it's programmatic is that each client of a service must be "coded" to the interface.
- Allows description of complex data structuress in request and response
- Using XML Schema standard http://www.w3.org/XML/Schema



Brief History – RESTful Web Services

- Short for REpresentational State Transfer
  - A software architecture style for distributed hypermedia systems(WWW)
- A set of principles that define how Web standards(HTTP and URIs) can be used.
- One “incarnation” of the REST style is HTTP (and a set of related set of standards, such as URI).
  - The way the Web’s architecture “should” be use
- Coined by Roy Fielding in his PhD thesis
- The “right” way to implement heterogeneous application-to-application communication?...



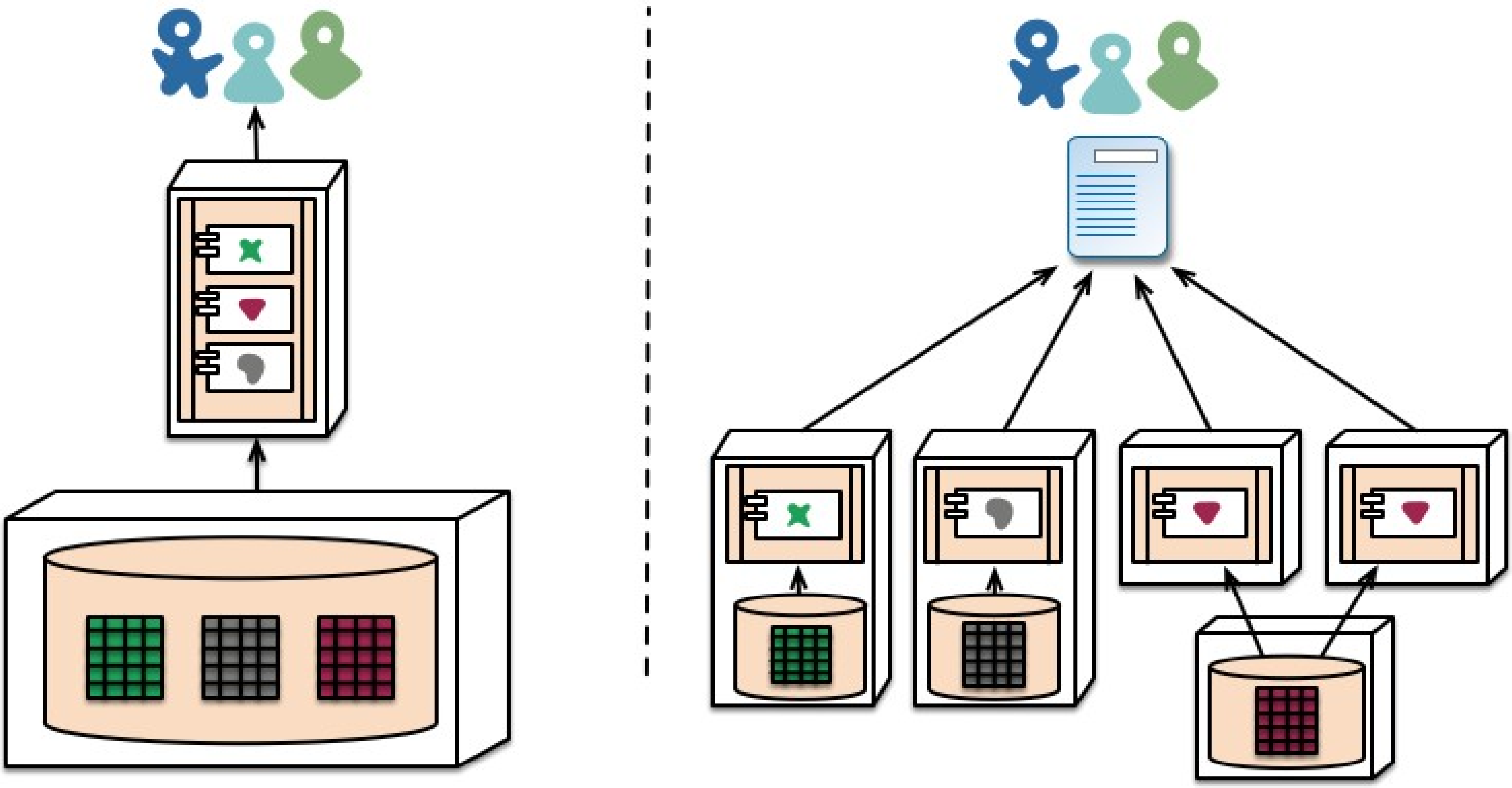
Web APIs

- Web API term used alot to refer to REST-like web services
  - Communication done via HTTP
    - Typically using HTTP resource api
  - Nothing much else defined
- Trends Web API / REST in tech world.
  - “Big Web” Services are seen as inferior to Web APIs for some.
  - More complicated to develop and tend to use more resources(Memory/bandwidth.

Microservice

- Architecural concept for developing web systems
- Approach to developing a single application as a suite of small services
  - each running in its own process and communicating via HTTP-based API
- Each service is independent, replacable and decoupled.
  - Can use different tech and data storage.
- “Do one thing and do it well”

Microservice



monolith - single database

microservices - application databases

Bibliography

- Java EE tutorial: https://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html
-