

CCNA (200-301) Cisco Certified Network Associate

Prepared by Eng. Taghreed Elsayed

Lecture 10

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

مزيد من التفاصيل

Automation and Programmability



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

What is Network Automation and Why We Need It?

- ❑ **Network automation** is the process of automating the provisioning, orchestration, and management of network and their services. There are lots of benefits network automation can offer.
- ❑ It allows network operations to be automated, which reduces human-to-machine interaction and, therefore, greatly reduces the chance of human error such as typos.
- ❑ Network automation can also save time and manual labor because it provides Zero Touch Provisioning (ZTP) that configures network devices automatically without someone needing to configure it locally, hence reducing the operational expenditures, especially for service providers and enterprises.



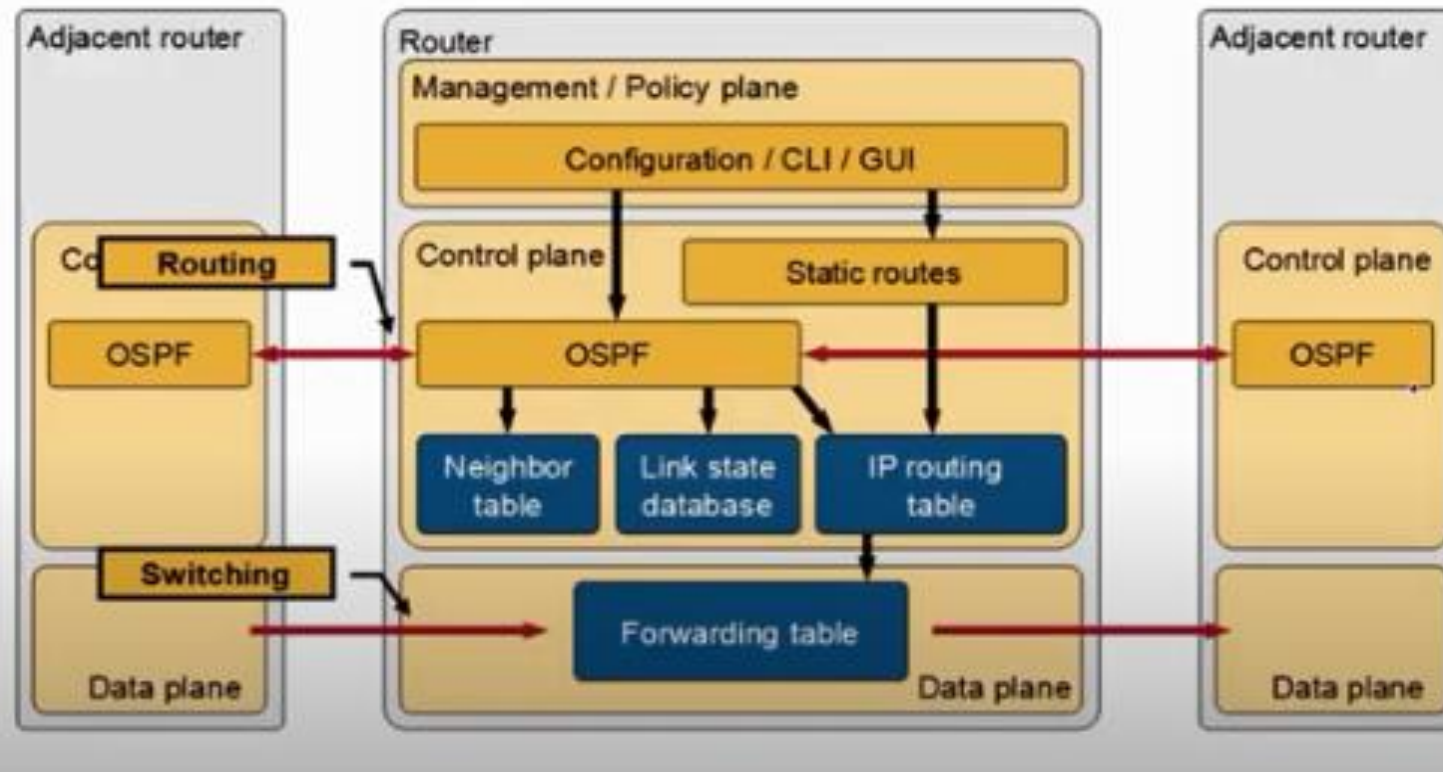
- ❑ **Finally**, one of the more important benefits of network automation is guarantee consistency in your configurations. With automation, you can create a script and then instruct the controller to deploy that to the devices wherein each device receives the same configuration.
- ❑ This ensures that you are familiar with the configuration of each device, and it also means minimal configuration errors because the administrator is not typing the same configuration commands over and over on each device. It also greatly helps in troubleshooting and in generating reports.

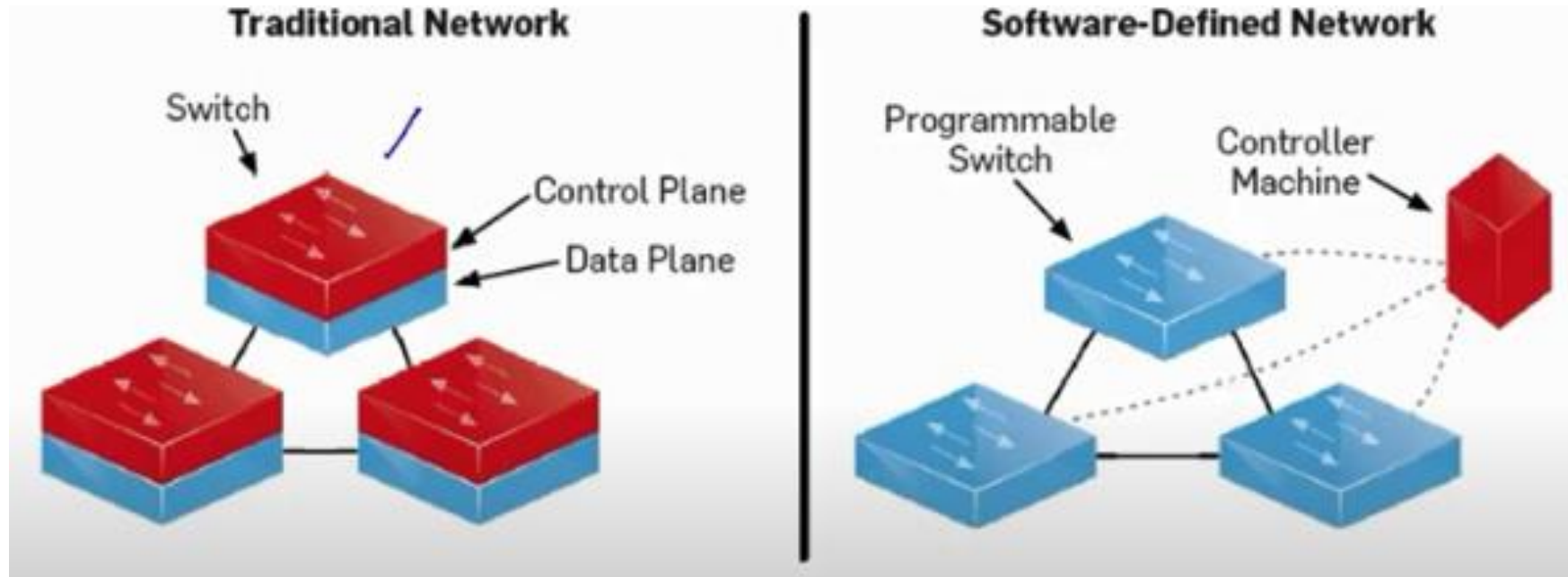
Traditional Network Management

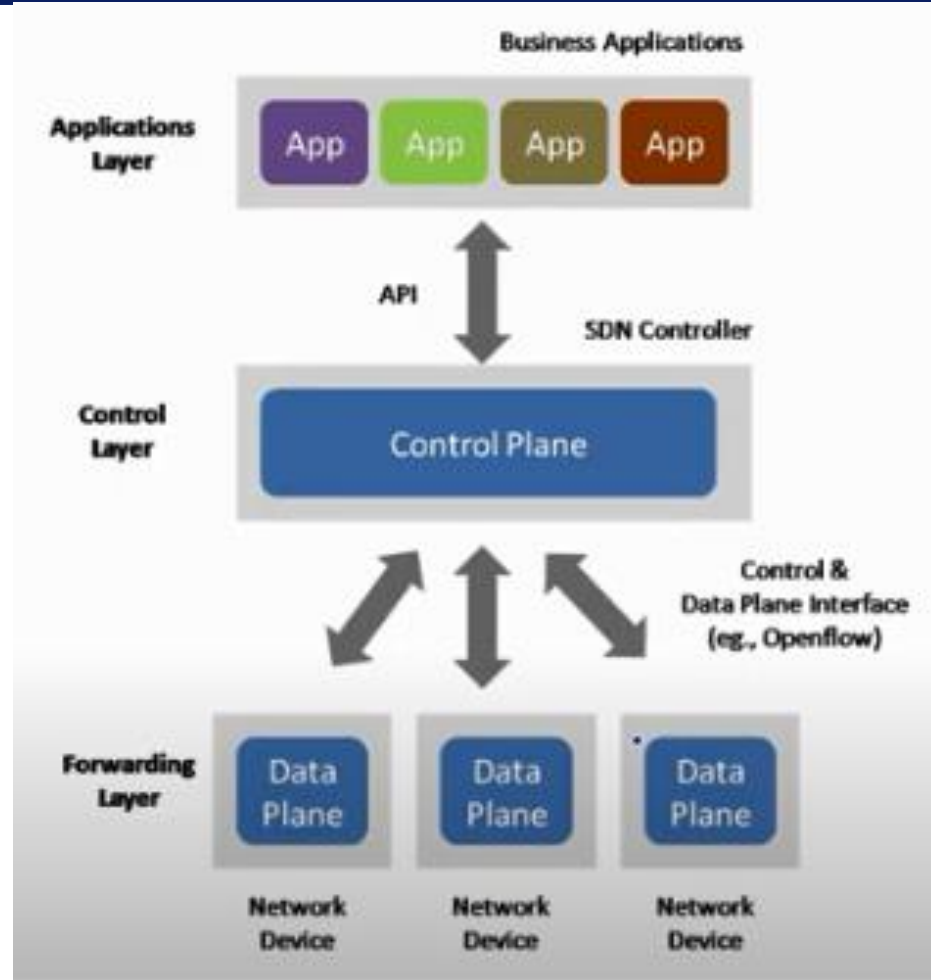
- ❑ With traditional networking, you can only manage network devices one at a time using SSH to the command line. This process is time-consuming, labor-intensive, and prone to human error. While this can work well in smaller networks, it does not work well in a large enterprise and does not also scale very well.
- ❑ Traditional Network Monitoring Systems (NMS) such as Solar Winds , Cisco Works, and Cisco Prime Infrastructure have also been available for a long time and use protocols such as SNMP and Net flow to gather information reports on the state of the network.

- ❑ While SNMP uses MIB and OID to collect useful data and can also be used to push configuration to devices, it does not build for real-time programmatic access and has limited functionality.
- ❖ SNMP also has security concerns and can be complex to implement and operate. These are the main challenges of traditional Networks:
 - Scalability Issues
 - Management Complexity
 - Slower Issue Resolution

Management, Control and Data Planes







Google's Software Defined WAN



Chapter 04

Automation and Programmability



جامعة الزكية
EVERYONE'S SMART UNIVERSITY



Cisco TCL

```
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#
Router#tclsh
Router(tcl)#
Router(tcl)#
Router(tcl)#
Router(tcl)#
Router(tcl)#$ list "1.1.1.1" "2.2.2.2" "3.3.3.3" "4.4.4.4" "5.5.5.5"]
1.1.1.1 2.2.2.2 3.3.3.3 4.4.4.4 5.5.5.5
Router(tcl)#
Router(tcl)#
Router(tcl)#
Router(tcl)#
Router(tcl)#
Router(tcl)#foreach ip $trial { puts [ exec "ping $ip" ]}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
```



SDN Benefits

Traditional disadvantage

- Time consuming
- Error – Prone
- High conversions
- Limits automation
- Limits innovation

- Response to link or node failure is faster
- Loop Avoidance is Simpler
- Management is simpler
- less time consuming
- Save money
- Support Automation
- Support innovation
- Agility



S D N Disadvantage

- Single point of failure
- Single point of attack



Controller-Based Networks

- ❑ A Controller-based network uses controller software or appliance to automate network operations. Software-defined networking uses a controller for network automation. It allows you to centrally manage your network device's configuration and enable programs to automatically configure and operate networks through the power of Application Programming Interfaces (APIs).
- ❑ Intelligent network automation through a controller-based network provides ease of management to your network than traditional networking because all of the configuration settings can be easily managed in the GUI of the controller.

- ❑ Controllers can also provide central aggregation points for monitoring and data analytics for more visibility and easier troubleshooting, particularly on a large network enterprise.
- ❑ **SD-Access is Cisco's solution for network automation** that replaces traditional network management. It offers a single point of automation, orchestration, and management of networks through the Cisco DNA Center controller. SD-Access has two key components:
 - **Cisco Campus Fabric Solution**- it is a network virtualization overlay built on top of the network underlay.
 - **Cisco DNA Center – SDN** controller for centralized management and NMS.

Network Automation Tools

- ❑ Multiple methods can be used to automate network management – Python scripts, NETCONF, RESTCONF, Ansible, Puppet, SDN, Cisco DNA Center, etc. However, not all methods are supported by all devices so you should choose which is most suitable for your environment.
- ❑ Network configuration management tools provide an automated method to execute and monitor changes to our networks. Some tools even offer network automation frameworks wherein a library of commands and workflows are already provided

- ❑ If **such changes do occur, it either reconfigures the device or informs the network engineers to make the change.** There are two types of configuration management tools:
 - agent-based tools – need the installation of a code (agent) on the managed device, such as Puppet, Salt stack, and Chef.
 - agentless tools – does not need the installation of any code (agent) on them, such as Ansible , Salt stack SSH, and Puppet Bolt.

Cisco DNA Center

- ❑ The Cisco Digital Network Architecture (DNA) Center is an SDN controller that is used to support Cisco's Software-Defined Access for network automation. The DNA Center provides centralized management on your system, automates configuration changes, and performs analysis of the devices.
- ❑ Cisco DNA Center Assurance is a tool for monitoring and network management. It monitors traffic statistics, the health status of the network, and an overview of all issues in the network. It also correlates issues and provides recommendations on what may cause such network issues to help you assist in diagnosing and troubleshooting.

- ❑ Cisco DNA Center also enhances network visibility use that visibility to improve performance, It enables proactive prediction of network-related and security-related risks by using telemetry to improve the performance of the network.

SDN

Software Defined Network :-

- the direct application to automation technology
- managing network using software
- software can either building scripts (like cisco TCL or python) or using controller



SDN-controller :-

- The main software responsible for managing , controlling , implementing and administrating the network
- Controller can be separated S/W installed on server
Either server with its own controller (like cisco DNA with its APIC controller)
- DNA server can be in the same network or in a remote network (cloud SDN)
- DNA uses tools to control network (like puppet , chef and ansible)

SDN Architecture :-

A-undelay network

- the protocols & features to get reachability
- open standard protocol (OSPF , ISIS)
- like passive solution



B-Overlay network

- virtual network created on top of the overlay
- uses some protocols (VRF / MPLS-VPN / VXLAN)

C-SDN fabric

- the physical devices used to build the underlay
- the devices controlled by controller



SDN Implementation :-

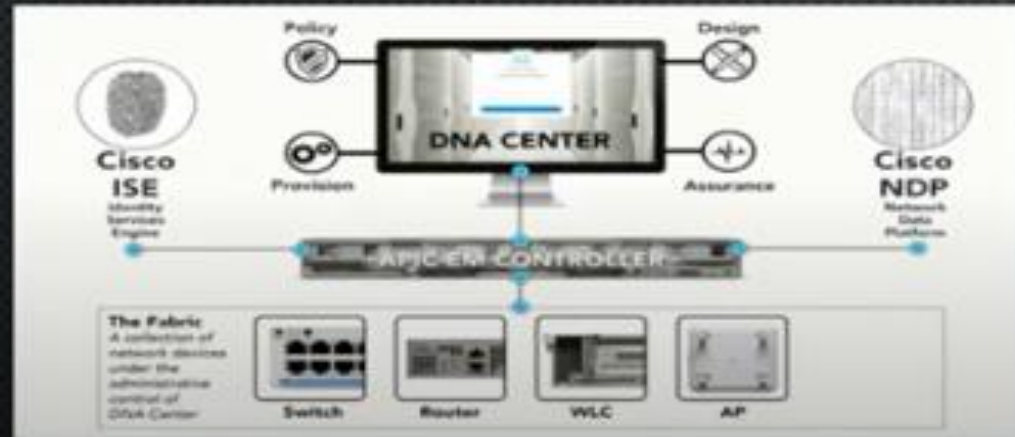
Imperative approach	Declarative approach
The control plane resides completely in the controller	The control plane resides within network devices
The controller has complete control over programming and forwarding decisions	The controller will declare the requirements of F/W and routing decisions
Devices asks the controller before any F/W or routing action	Devices take decisions based on the pre-instructions from the controller
Stateful (dependent)	Stateless (independent)



DNA

Digital Network Architecture :-

- central automation appliance
- creates intent-based network
- design WLAN SSID
- managed through GUI
- Built in APIC (application policy infrastructure controller)
- built in NDP (network data platform)

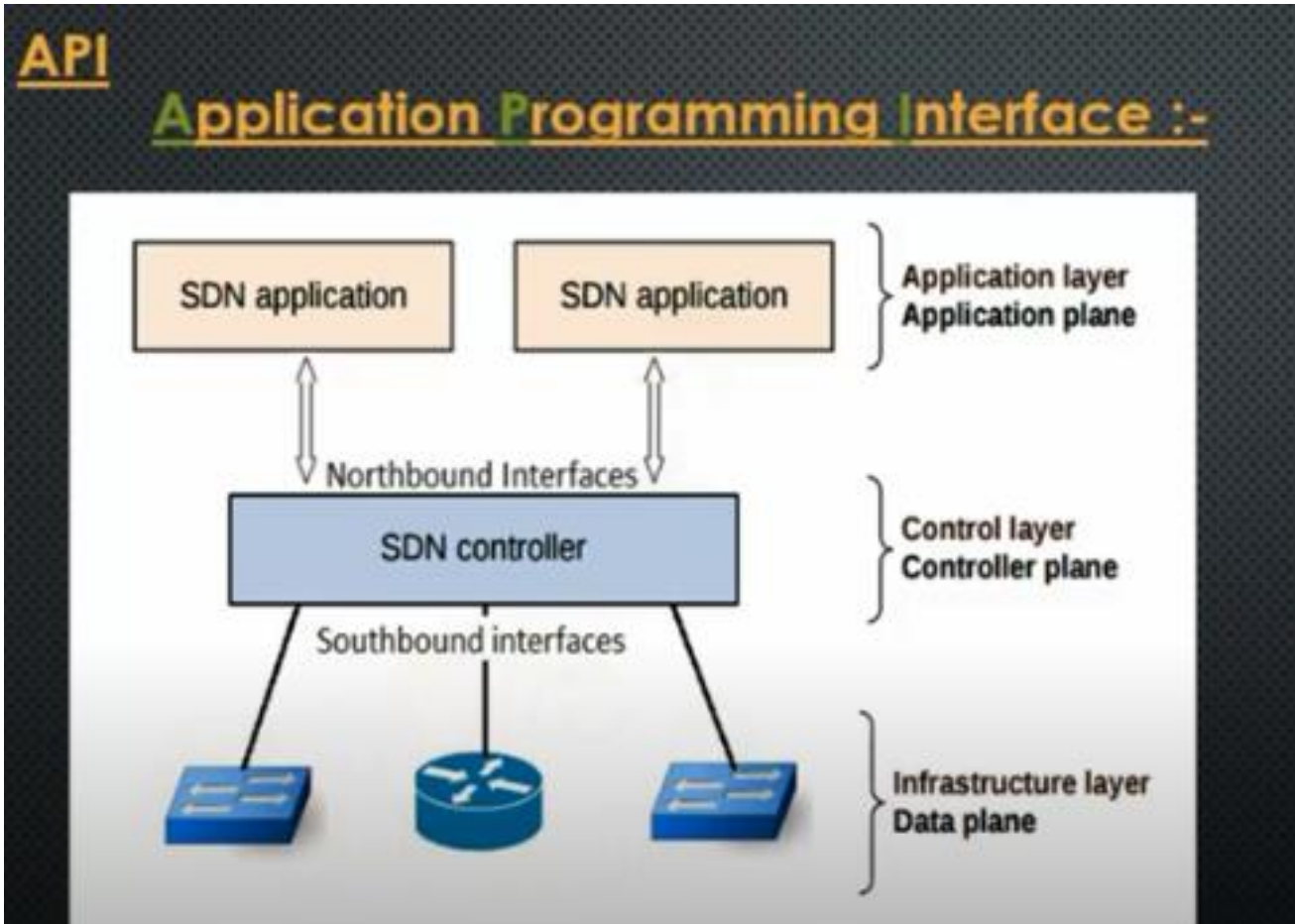


Application Programming Interface :-

Software that allows two applications to talk to each other

API types





API

Application Programming Interface :-

- API written by programming language (XML / JSON)
- API uses server / client relation
- In the northbound API , controller = sever , application = client
- in the southbound API , network devices = server , controller = client
- if we have cluster DNA , communication between different DNA using API



REST-based API

REpresentational State Transfer API :-

- standard data structure between applications , controller and nodes
- mostly found in the northbound
- utilize http verbs (post , get , put , delete)
- developed using CRUD (create , read , update , delete)



DevOps tools

Development Operations:-

- configuration management tools used to facilitate automation of complex IT application environment
- Uses CLI / scripting
- like (puppet / chef /ansible)



DevOps tools

Development Operations:-

	Puppet	Chef	Ansible
Configuration language	Ruby	ruby	YAML
architecture	Master – agent	Master – agent	agentless
Configuration management	Pull	Pull	Push
Installation process	Time intensive	Time Intensive	Fast
implementation	declarative	Imperative	declarative



SOAP vs REST: Comparing the Web API Services

- ❑ Application Programming Interface (API) serves as an interface that enables applications or computer programs to communicate with each other directly. It is used to expose functionalities and procedures to the outside world through a service path or URL.
- ❑ API plays a big part in our network by configuring and monitoring computer systems, network devices, or applications that involve programmatically interacting through software.
- ❑ APIs are built for real-time programmatic access (built for computers to interact with each other without human intervention) and can be used for connecting to an individual or multiple devices on our network simultaneously.



Web Services

- ❑ Web services are the type of API that typically runs over HTTP/HTTPS. However, some web services, such as the Simple Object Access Protocol (SOAP), can use different transports such as Simple Mail Transfer Protocol (SMTP). Web services don't necessarily use HTTP, but this is almost always the case and is usually assumed unless mentioned otherwise.
- ❑ All web services can be considered APIs, but not all can be regarded as web services. APIs can utilize any communication protocol and aren't limited to HTTP like web services are.
- ❑ **The two main types of APIs used for web service that runs** over the Internet and typically use HTTP/HTTPS are SOAP and REST.

Simple Object Access Protocol (SOAP) API

- ❑ SOAP APIs are standard communication protocol systems that permit different operating systems like Linux and Windows to communicate. It was developed by Microsoft and is commonly used with HTTP transport with XML data format to access web services. It has three well-known characteristics.
 - Independent
 - Extensible
 - Neutral

❑ A more reliable way to communicate between applications is over HTTP, as it is supported by all Internet browsers and servers for web service. **SOAP was created to accomplish this.** SOAP web services provide a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP messages are nothing but **XML documents and can contain four elements.**

- Envelope
- Header
- Body
- Fault

❑ Representational State Transfer (REST) API

- ❑ REST API or Representational State Transfer is an architecture style for designing a distributed system that gives guidelines for the structure and organization of an API for web service. REST is protocol-independent. It supports any transport and data format like HTTP/S transport and JavaScript Object Notation (JSON) or XML data format.
- ❑ Restful web services use HTTP methods to collect and manipulate data. A URL identifies every resource, and every resource should support the HTTP common operations. It offers a consistent way to interact with APIs from multiple vendors.

Below are the most common HTTP “CRUD” functions and their associated use cases:

Operation	SQL	HTTP
Create	INSERT	PUT / POST
Read (Retrieve)	SELECT	GET
Update (Modify)	UPDATE	PUT / POST / PATCH
Delete (Destroy)	DELETE	DELETE

- ❑ Unlike SOAP protocol, REST APIs do not require a new message format. The HTTP API is CRUD (Create, Retrieve, Update, and Delete).

SOAP vs REST

- ❑ Use SOAP API if you are dealing with transactional operations and you already have an audience that is satisfied with this technology. Use REST web services if you're focused on wide-scale API adoption or if your API is targeted at mobile apps.

❖ SOAP Advantages:

- Follows a formal enterprise approach
- It is platform- and language-independent.
- Information about objects is communicated to clients.
- SOAP bypasses all firewalls.

SOAP Disadvantages:

- Uses high bandwidth to communicate metadata
- Hard to implement
- Unpopular among web and mobile developers

SOAP Use Cases:

- Financial services
- Payment gateways
- Telecommunication services

REST Advantages:

- Follows the philosophy of the Open Web
- Relatively easy to implement and maintain
- Separates client and server implementations
- A single entity doesn't control communication
- The client can store information to prevent multiple calls
- Able to return data in multiple formats (JSON, XML, etc.)

REST Disadvantages:

- Works on top of the HTTP protocol only
- Must enforce security and authorization on top of it

REST Use Cases:

- Social media services
- Social networks
- Webchat services
- Mobile services

Data Serialization Formats – JSON, YAML, and XML

- ❑ Data serialization is a technique that translates structured data into a format that enables storage or sharing of the digital information in a state that permits recovery of its original structure. Various computer systems have different hardware architecture, operating systems, and/or addressing mechanisms, which can be a challenge that is why serializing the data enables different users to have a standardized data serialization formats that can be easily shared across various applications.

❑ Data Serialization Formats and its Functionalities

Choosing the data serialization format for an application should depend on factors such as human readability, the complexity of data, and storage and speed limitations. Here are some of the common data serialization formats used to ease the storage and sharing of data across the Internet.

JSON – JavaScript Object Notation

This serialization format was standardized in 2013 and is considered a lightweight data-interchange format that translates data structures easier to be read and written by humans. This human-readable format has a short syntax textual format with limited data types derived from JavaScript data formats

- ❑ JSON data does not need a separate parser since they are referenced to JavaScript objects compared to XML.

The following are the benefits of using JSON files:

1. Self-describing
2. Simplicity and compact
3. Ease in learning and understanding



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

