USN | | | | | | | | | |

Internal Assessment Test 3 – December 2024

Sub:	**Internet of Things**	Sub Code:	**21CS735**	Branch:	**AInDS**

Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VII**	**OBE**

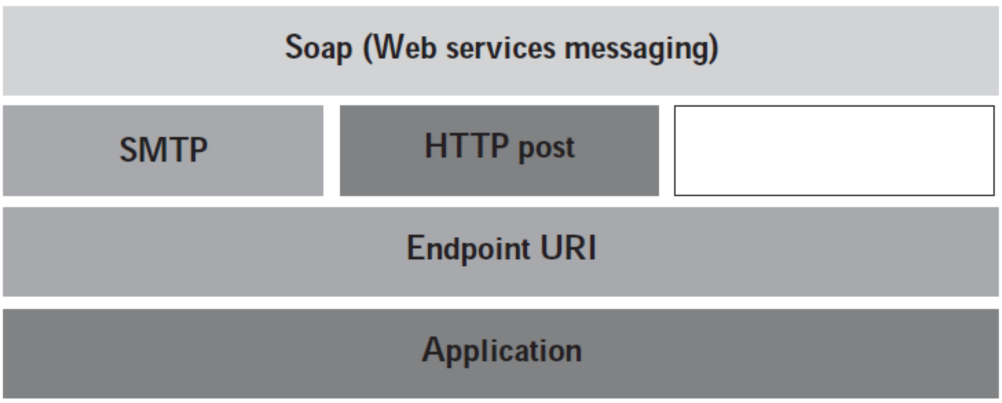
		<u>**Answer any FIVE Questions**</u>	**MARKS**	**CO**	**RBT**
Q1	a	Compare REST with SOAP	5	CO5	L2
	b	Explain in detail AMQP	5	CO5	L1
Q 2		What is need of identification protocol? Explain uCode categories	10	CO5	L2
Q 3		Explain interoperable solution that has been developed for IoT	10	CO5	L2
Q 4	a	Explain DLNA also technological components used in DLNA	5	CO5	L1
	b	Give the reason for requirement for interoperability and interoperable solutions in IoT	5	CO5	L2
Q 5		Name five of the most popular interoperability frameworks and explain HomeKit in detail	10	CO5	L2
Q 6		What is the need of Semantic protocol and explain any one in detail	10	CO5	L2

CI **CCI** **HOD**

Q 1 a) Compare REST with SOAP

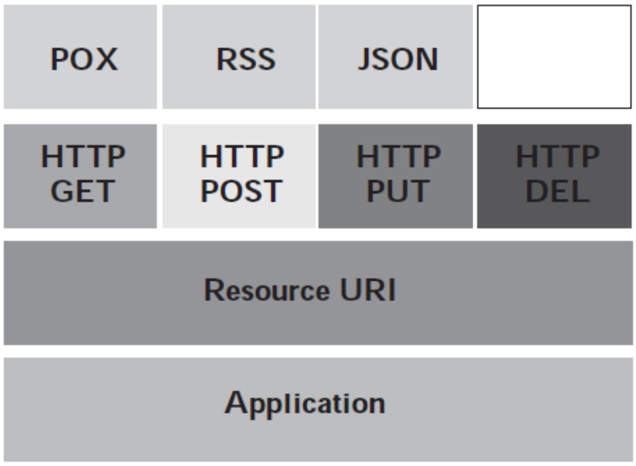
SOAP

SOAP or simple object access protocol is used for exchanging structured information in web services by making use of XML information set formatting over the application layer protocol (HTTP, SMTP) based transmission and negotiation of messages, as shown in Figure 8.21 [21]. This allows SOAP to communicate with two or more systems with different operating systems using XML, making it language and platform independent. The use of SOAP facilitates the messaging layer of the web services protocol stack.



REST

Representational state transfer or REST encompasses a set of constraints for the creation of web services, mainly using a software architectural style [22]. The web services adhering to REST styles are referred to as RESTful services; these services enable interoperability between various Internet-connected devices. RESTful systems are stateless: the web services on the server do not retain client states. The use of stateless protocols and standards makes RESTful systems quite fast, reliable, and scalable. The reuse of components can be easily managed without hindering the regular operations of the system as a whole. Requesting systems can manipulate textual web resource representations by making use of this stateless behavior of REST. RESTful web services, in response to requests made to a resource's URI, mainly responds with either an HTML, XML, or JSON (JavaScript Object Notation) formatted payload. As RESTful services use HTTP for transfer over the network, the following four methods are commonly used: 1) GET (read-only access to a resource), 2) POST (for creating a new resource), 3) DELETE (used for removing a resource), and 4) PUT (used for updating an existing resource or creating a new one). Figure 8.23 represents the REST style and its components.



Q 1 b) Explain in detail AMQP

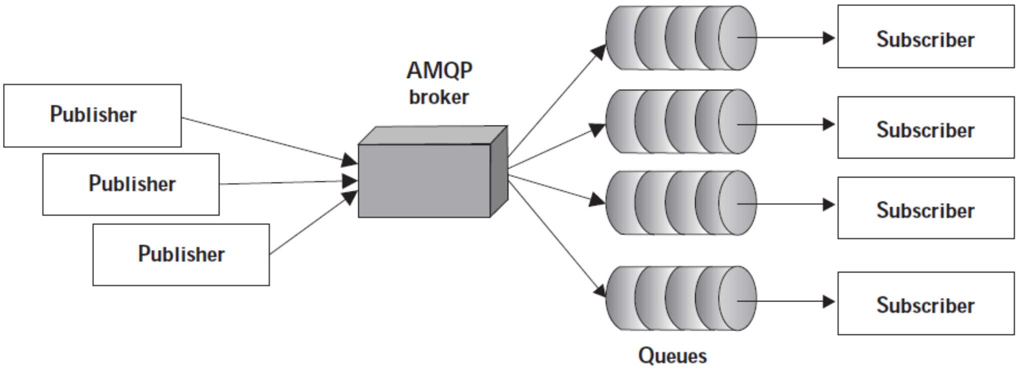
AMQP or the advanced message queuing protocol is an open standard middleware at the application layer developed for message-oriented operations [19]. It tries to bring about the concept of interoperability between clients and the server by enabling cross-vendor implementations The protocol is armed with features of message orientation, queuing, reliability, security, and routing. Both request–response and publish–subscribe methods are supported. AMQP is considered as a wire-level protocol. Here, the data format description is released on the network as a stream of bytes. This description allows AMQP to connect to anyone who can interpret and create messages in the same format. It also results in a level of interoperability where anyone with compliant or supporting means can make use of this protocol without any need for a specific programming language.

AMQP Features

AMQP is built for the underlying TCP and is designed to support a variety of messaging applications efficiently. It provides a wide variety of features such as flow-controlled communication, message-oriented communication, message delivery guarantees (at most once, at least once, and exactly once), authentication support, and an optional SSL or TLS based encryption support. The AMQP is specified across four layers: 1) type system, 2) process to process asynchronous and symmetric message transfer protocol, 3) extensible message format, and 4) set of extensible messaging capabilities. In continuation, the primary unit of data in AMQP is referred to as a frame. These frames are responsible for the initiation of connections, termination of connections, and control of messages between two peers using AMQP. There are nine frame types in AMQP:

(i) Open: responsible for opening the connection between peers.
(ii) Begin: responsible for setup and control of messaging sessions between peers.
(iii) Attach: responsible for link attachment.
(iv) Transfer: responsible for message transfer over the link.
(v) Flow: responsible for updating the flow control state.
(vi) Disposition: responsible for updating of transfer state.

(vii) Detach: responsible for detachment of link between two peers.
(viii) End: responsible for truncation of a session.
(ix) Close: responsible for closing/ending a connection.



Q 2) What is need of identification protocol? Explain uCode categories

The surge of IoT devices and Things which are connected over the Internet, makes it significantly hard to identify each device securely. The number of connected things is rising exponentially; with this rise the need to design and develop protocols that can provide unique and distinguishable identifiers to so many Things becomes overwhelming. However, unified global efforts have come up with certain solutions to address the challenges regarding identification of Things, which keep on updating from time to time. Some of the commonly encountered ones are EPC, uCode, and URIs.

uCode

Another identification number system, the uCode is designed to uniquely identify real-world things and objects whose information is digitally associated with the uCode system [25]. The uID center in Japan provides support for the uCode system. The uCode system can be used with any application, business processes, and technology (RFID, barcodes, matrix codes). uCode is application and technology independent and uses 128 bit codes for uniquely tagging/naming physical objects. The uCode provides 3:4 _ 1038 unique codes for individually tagging objects. These features make uCode a crucial enabling technology for IoT. Figure 8.26 represents the working of uCode tags and its various stakeholders. The uCode tags are generally grouped into five categories: 1) print tags, 2) acoustic tags, 3) active RF tags, 4) active infrared tags, and 5) passive RFID tags. In contrast to other identification systems, the uCode system has the following distinct features: (i) It does not display product types, albeit it identifies individual objects. Existing codes identify products by individual vendors, making the possibility of identifier tag reuse a possibility, which is avoided in the uCode system.

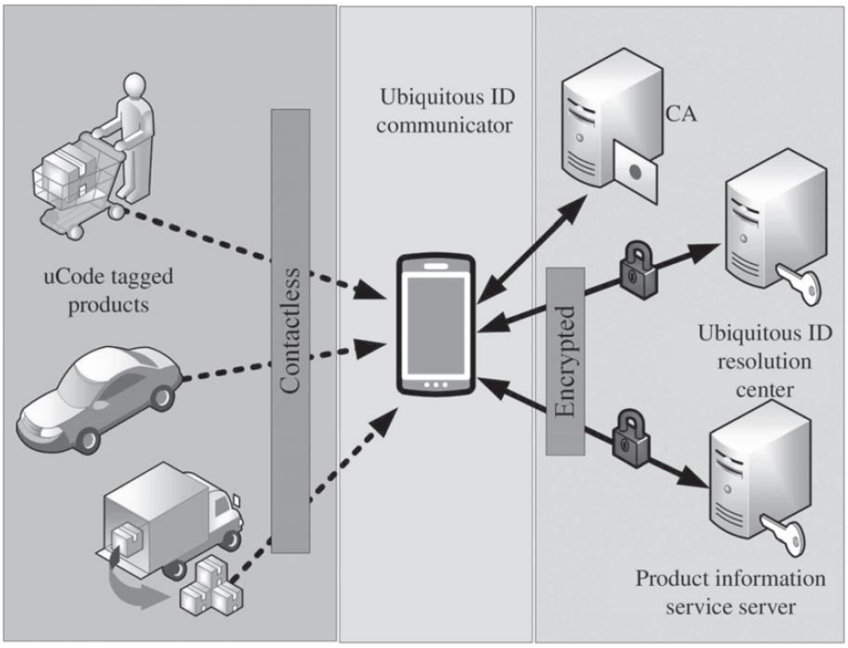
(ii) In addition to physical objects, the uCode can be associated with places, concepts, and contents, enabling this system to identify such items universally.

(iii) Being application and business agnostic, the uCode system can be used across industries and organizations. The system provides a unique identification number, which does not carry any meaning or information about the tagged object/item. This enables the same system to be used seamlessly across organizations, industries, and product types.

(iv) uTRON, a ubiquitous security framework, which is incorporated with the ubiquitous ID architecture of the uCode system, makes it entirely secure and enables information privacy protection.

(v) The tag agnostic nature of the uCode system makes it possible for various systems such as RFIDs, and barcodes to store uCode information. This makes uCode highly ubiquitous and pervasive.

(vi) The uCode represents pure numbers and is devoid of any meaning or information related to the tagged item/object. This makes the reassignment of uCode tags quite robust and straightforward.



The operational process of reading a uCode is as follows:

(i) uCode tags are read using mobile phone cameras to identify the ucode.

(ii) An inquiry about the uCode is sent to the uCode resolution server from the mobile phone over the Internet.

(iii) The uCode resolution server returns information about the uCode to the mobile phone. The returned information contains the source of the read uCode information.

(iv) The ubiquitous communicator then acquires the contents and service information from the information providing source of the read uCode.

Q 3) Explain interoperable solution that has been developed for IoT

The significant range of interoperable solutions that has been developed for IoT can be broadly categorized into the following groups:

(i) **Device:** The existence of a vast plethora of devices and device types in an IoT ecosystem necessitates device interoperability. Devices can be loosely categorized as low-end, mid-end, and high-end devices based on their processing power, energy, and communication requirements. Low-end devices are supposed to be deployed in bulk, with little or no chance of getting their energy supplies replenished, depending on the application scenario. These devices rely on low-power communication schemes and radios, typically accompanied by low-data rates. The interface of such devices with high-end devices (e.g., smartphones, tablets) requires device-level interoperability [2].

(ii) **Platform:** The variations in the platform may be due to variations in operating systems (Contiki, RIOT, TinyOS, OpenWSN), data structures, programming languages (Python, Java, Android, C++), or/and application development environment. For example, the Android platform is quite different from the iOS one, and devices running these are not compatible with one another [3].

(iii) **Semantic:** Semantic conflicts arise during IoT operations, mainly due to the presence of various data models (XML, CSV, JSON), information models (_C, _F, K, or different representations of the same physical quantity), and ontologies [4]. There is a need for semantic interoperability, especially in a WoT environment, which can enable various agents, applications, and services to share data or knowledge in a meaningful manner.

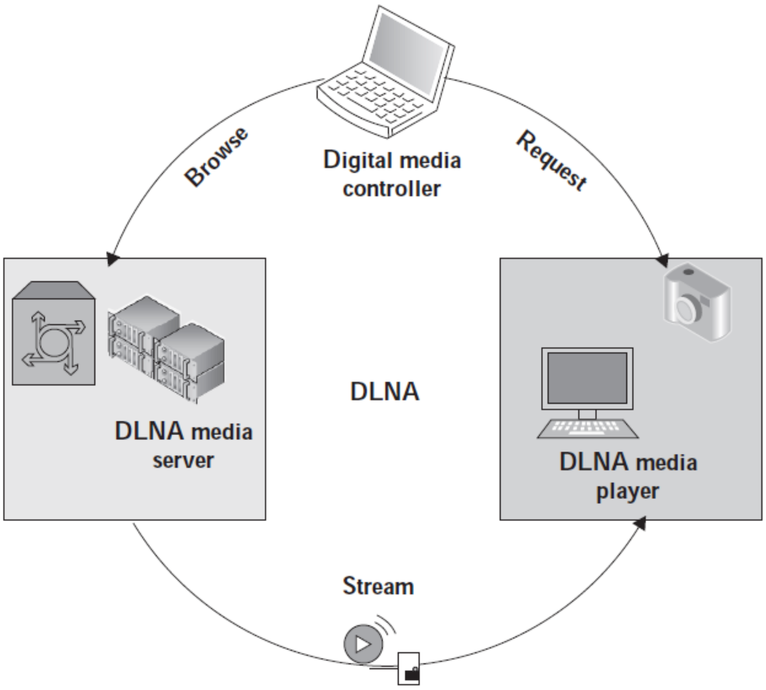
(iv) **Syntactic:** Syntactic interoperability is a necessity due to the presence of conflicts between data formats, interfaces, and schemas. The variation in the syntactical grammar between a sender and a receiver of information results in massive stability issues, redundancies, and unnecessary data handling efforts [5]. For example, a packet from a device has a format as Header-Identifier-SensorASensorB-

Footer, whereas another device from a different manufacturer, but deployed for the same application has the data format as Header-Identifier-SensorB-SensorA-Footer. This change in position of sensor A and sensor B in the two packets creates syntactic errors, although they contain the same information.

(v) **Network:** The large range of connectivity solutions, both wired and wireless, at the disposal of developers and manufacturers of IoT devices and components, further necessitates network interoperability. Starting from the networks and sub-networks on the ground, to the uplink connectivity solutions, there is a need for uniformity or means of integrating to devices enable seamless and interoperable operations.

Q4 a) Explain DLNA also technological components used in DLNA

The Digital Living Network Alliance (DLNA), previously known as the Digital Home Working Group (DHWG), was proposed by a consortium of consumer electronics companies in 2003 to incorporate interoperability guidelines for digital media sharing among multimedia devices such as smartphones, smart TVs, tablets, multimedia servers, and storage servers. Primarily designed for home networking, this standard relies majorly on WLAN for communicating with other devices in its domain and can easily incorporate cable, satellite, and telecom service providers to ensure data transfer link protection at either end. The inclusion of a digital rights management layer allows for multimedia data sharing among users while avoiding piracy of data. The consumers in DLNA, which may consist of a variety of devices such as TVs, phones, tablets, media players, PCs, and others, can view subscribable content without any additional add-ons or devices through VidiPath. Figure 9.3 shows the steps involved in a typical DLNA-based multimedia streaming application. As of 2019, DLNA has over a billion devices following its guidelines globally [7].



DLNA outlines the following key technological components, which enable interoperability guidelines for manufacturers [7].

(i) Network and Connectivity
(ii) Device and Service Discovery and Control
(iii) Media Format and Transport Model
(iv) Media Management, Distribution, and Control
(v) Digital Rights Management and Content Protection
(vi) Manageability

Q 4 b) Give the reason for requirement for interoperability and interoperable solutions in IoT

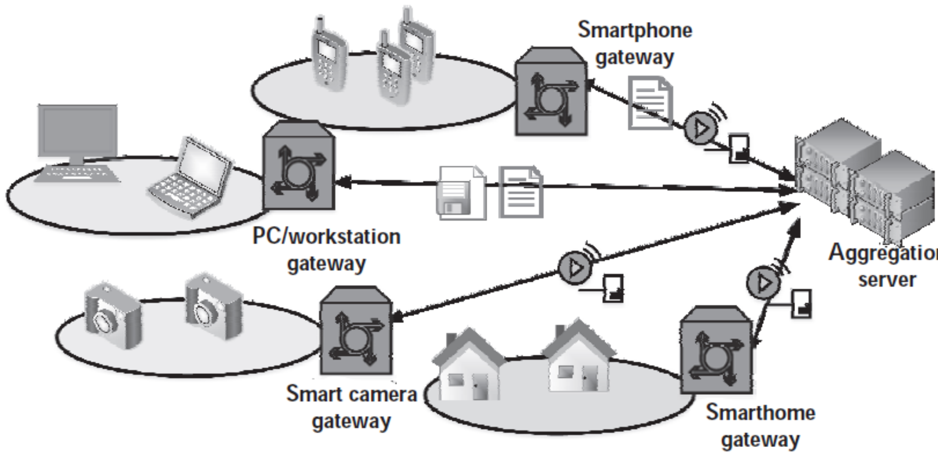
The urgency in the requirement for interoperability and interoperable solutions in IoT arose mainly due to the following reasons:

Large-scale Cooperation: There is a need for cooperation and coordination among the huge number of IoT devices, systems, standards, and platforms; this is a long-standing problem. Proprietary solutions are seldom reusable and economical in the long run, which is yet another reason for the demand for interoperability.

(ii) **Global Heterogeneity:** The network of devices within and outside the purview of gateways and their subnets are quite large considering the spread of IoT and the applications it is being adapted to daily. Device heterogeneity spans the globe when connected through the Internet. A common syntax, platform, or standard is required for unifying these heterogeneous devices.

(iii) **Unknown IoT Device Configuration:** Device heterogeneity is often accompanied by further heterogeneity in device configurations. Especially considering the global-scale network of devices, the vast combinations of device configurations such as data rate, frequencies, protocols, language, syntax, and others, which are often unknown beforehand, further raise the requirement of interoperable solutions.

(iv) **Semantic Conflicts:** The variations in processing logic and the way data is handled by the numerous sensors and devices making up a typical IoT implementation, makes it impossible for rapid and robust deployment. Additionally, the variations in the end applications and their supported platform configurations further add to the challenges.



Q 5) Name five of the most popular interoperability frameworks and explain HomeKit in detail

Similar to the standards, there has been a rise in universal interoperability frameworks. These frameworks span across platforms, devices, technologies, and application areas.

HomeKit

The HomeKit software framework is designed by Apple to work with its iOS mobile operating system for achieving a centralized device integrating and control framework [17]. It enables device configuration, communication, and control of smart home appliances. Home automation is achieved by incorporating room designs, items, and their actions within the HomeKit service. Users can interact with the framework using speech-based voice commands through Apple's voice assistant, Siri, or through external apps. Smart home devices such as thermostats, lights, locks, cameras, plugs, and others, spread over a house can be controlled by a single HomeKit interface through smartphones. HomeKit-enabled device manufacturers need to have an MFi program, and all devices were initially required to have an encryption coprocessor. Later, the processor-based encryption was changed to a software-based one. Non-HomeKit devices can have the benefits of HomeKit through the use of HomeKit gateways and hubs.

HomeKit devices within a smart home securely connect to a hub either through Wi-Fi or Bluetooth. However, as the range of Bluetooth is severely limited, the full potential of the HomeKit may not be adequately exploited. This framework allows for individual as well as grouped control of connected devices based on scenarios.

Features such as preconfigured devices settings can be collectively commanded using voice commands to Siri.

Q 6) What is the need of Semantic protocol and explain any one in detail

The semantic protocols for IoT, which is a rapidly upcoming domain, focus on the meaning and logic behind data connectivity and formats. Examples include JSONLD and the Web Thing model. Primarily designed to be cross-operable and modular, these protocols enhance the robustness and utility of IoT by incorporating the reach of the Web. As an example, the integration of semantic protocols such as JSON-LD with theWeb Things model gives rise to the SemanticWeb.

JSON-LD

JavaScript object notation for linked data or JSON-LD is a lightweight protocol, which is designed for JSON-based encoding of linked data by seamlessly converting older JSON-based representations of data. The representations of the data are highly human-understandable and highly suitable for RESTful environments and unstructured data over theWeb [29]. JSON-LD has an additional resource description framework (RDF) over and above the typical JSON model and is built to be contextual. This feature allows for the interoperability of JSON data over theWeb. The contextual linking of the object properties of a JSON document follows a fixed ontology in JSON-LD through strategies such as tagging with a language by or forcibly assigning values to pre-defined groups/bins. Context embeddings in JSON-LD documents can be either direct or through the use of separate file references using HTTP link headers. Linked data allows for the existence of a network of machine-readable and standardized data over the Web, which can be parsed by starting at a singular piece of data and subsequently traversing the embedded links within it; this may lead to different locations across the Web.

A sample JSON-LD schema

```
<script type="application/ld+json">
{
  "@context": "https :// schema.org",
  "@type": "BlogPosting",
  "mainEntityOfPage": {
    "@type": "WebPage",
    "@id": "www.xyz.com"
  },
  "headline": "Hello Readers",
  "description": "This is a test",
  "image": {
    "@type": "ImageObject",
    "url": "www.img1234.com",
    "width": 696,
    "height": 14
  },
  "author": {
    "@type": "Person",
    "name": "abc"
  },
  "publisher": {
    "@type": "Organization",
    "name": "CUP",
     },
  "datePublished": ""
}
</script>
```