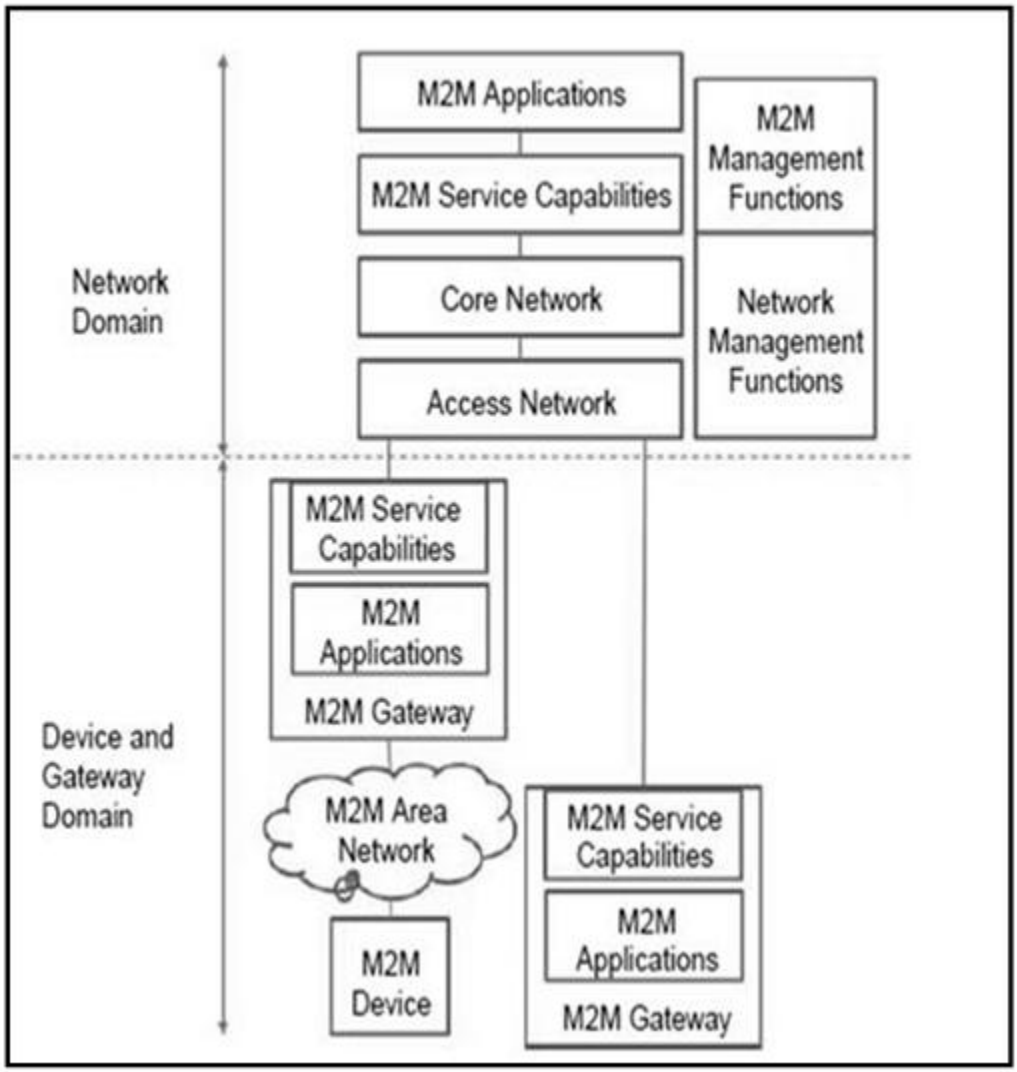
Internal Assesment Test –4, January 2022								
INTERNET OF THINGS	**Code:**	**18MCA542**						
2-02-2022	Duration:	90 mins	Max Marks:	50	Sem	V		MCA

1. **With a neat diagram, explain ETSI High level architecture**



• **M2M Device:** This is the device of interest for an M2M scenario, for example, a device with a temperature sensor. An M2M Device contains M2M Applications and M2M Service Capabilities. An M2M device connects to the Network Domain either directly or through an M2M Gateway:

• Direct connection: The M2M Device is capable of performing registration, authentication, authorization, management, and provisioning to the Network Domain. Direct connection also means that the M2M device contains the appropriate physical layer to be able to communicate with the Access Network.

Through one or more M2M Gateway: This is the case when the M2M device does not have the appropriate physical layer, compatible with the Access Network technology, and therefore it needs a network domain proxy. Moreover, a number of M2M devices may form their own local M2M Area Network that typically employs a different networking technology from the Access Network. The M2M Gateway acts as a proxy for the Network Domain and performs the procedures of authentication, authorization, management, and provisioning. An M2M Device could connect through multiple M2M Gateways.

• **M2M Area Network:** This is typically a local area network (LAN) or a Personal Area Network (PAN) and provides connectivity between M2M Devices and M2M Gateways. Typical networking technologies are IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (ZigBee, IETF 6LoWPAN/ROLL/CoRE), MBUS, KNX (wired or wireless) PLC, etc.

• **M2M Gateway:** The device that provides connectivity for M2M Devices in an M2M Area Network towards the Network Domain. The M2M Gateway contains M2M Applications and M2M Service Capabilities. The M2M Gateway may also provide services to other legacy devices that are not visible to the Network Domain. The Network Domain contains the following functional/topological entities:

• **Access Network**: this is the network that allows the devices in the Device and Gateway Domain to communicate with the Core Network. Example Access Network Technologies are fixed (xDSL, HFC) and wireless (Satellite, GERAN, UTRAN, E-UTRAN W-LAN, WiMAX).

• **Core Network:** Examples of Core Networks are 3GPP Core Network and ETSI TISPAN Core Network. It provides the following functions:

- IP connectivity.
- Service and Network control.
- Interconnection with other networks.
- Roaming.

• **M2M Service Capabilities:** These are functions exposed to different M2M Applications through a set of open interfaces. These functions use underlying Core Network functions, and their objective is to abstract the network functions for the sake of simpler applications. More details about the specific service capabilities are provided later in the chapter.

• **M2M Applications:** These are the specific M2M applications (e.g. smart metering) that utilize the M2M Service Capabilities through the open interfaces.

• **Network Management Functions:** These are all the necessary functions to manage the Access and Core Network (e.g. Provisioning, Fault Management, etc.).

• **M2M Management Functions:** These are the necessary functions required to manage the M2M Service Capabilities on the Network Domain while the management of an M2M Device or Gateway is performed by specific M2M Service Capabilities. There are two M2M Management functions:

• **M2M Service Bootstrap Function (MSBF):** The MSBF facilitates the bootstrapping of permanent M2M service layer security credentials in the M2M Device or Gateway and the M2M Service Capabilities in the Network Domain. In the Network Service Capabilities Layer, the Bootstrap procedures perform, among other procedures, provisioning of an M2M Root Key (secret key) to the M2M Device or Gateway and the M2M Authentication Server (MAS).

• **M2M Authentication Server (MAS):** This is the safe execution environment where permanent security credentials such as the M2M Root Key are stored. Any security credentials established on the M2M Device or Gateway are stored in a secure environment such as a trusted platform module.

2) Design the and explain CRISP-DM process framework in detail with example

The phases in the CRISP-DM process model are described in Figure 5.15, which is followed by descriptions of each of the phases. These are illustrated using an example from Predictive Maintenance (PdM) for pump stations in a water distribution network. Although the figure indicates a certain order between the phases, analytics is an iterative process, and it's expected that you will have to move back and forth between the phases to a certain extent.

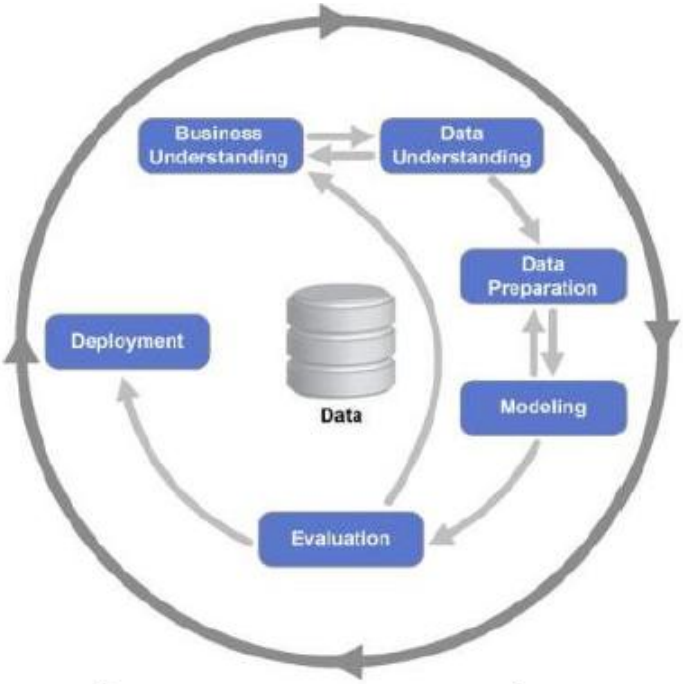


Figure 5. 15 CRISP-DM Process Diagram.

Business Understanding

 The first phase in the process is to understand the business objectives and requirements, as well as success criteria. This forms the basis for formulating the goals and plan for the data mining process.

 Many organizations may have a feeling that they are sitting on valuable data, but are unsure how to capitalize on this. In these cases, it's not unusual to bring in the help of an analytics team to identify potential business cases that can benefit from the data.

Data Understanding

 The next phase consists of collecting data and gaining an understanding of the data properties, such as amount of data and quality in terms of inconsistencies, missing data, and measurement errors. The tasks in this phase also include gaining some understanding of actionable insights contained in the data, as well as to form some basic hypotheses.

Data Preparation

 Before it's possible to start modeling the data to achieve our goals, it's necessary to prepare the data in terms of selection, transformation, and cleaning. In this phase, it's frequently the case that new data is necessary to construct, both in terms of entirely new attributes as well as imputing new data into records where data is missing.

 It's quite common for this phase to consume more than half the time of a project.

Modeling

 At the modeling phase, it's finally time to use the data to gain an understanding of the actual business problems that were stated in the beginning of the project. Various modeling techniques are usually applied and evaluated before selecting which ones are best suited for the particular problem at hand. As some modeling techniques require data in a specific form, it's quite common to go back to the data preparation phase at this stage. This is an example of the iterativeness of CRISP-DM and analytics in
general.

 After evaluating a number of models, it's time to select a set of candidate models to be methodically assessed. The assessment should estimate the effectiveness of the results in terms of accuracy, as well as ease of use in terms of interpretation of the results. If the assessment shows that we have found models that meet the necessary criteria, it's time for a more thorough evaluation, otherwise the work on finding suitable models has to continue.

Evaluation

 Now the project is nearing its end and it's time to evaluate the models from a business perspective using the success criteria that were defined at the beginning of the project. It is also customary to spend some time reviewing the project and draw conclusions about what was good and bad. This will be valuable input for future projects. At the end of the evaluation phase, a decision whether to deploy the results or not should be made.

Deployment

 At this last phase in the project, the models are deployed and integrated into the organization. This can mean several things, such as writing a report to disseminate the results, or integrating the model into an automated system. This part of the project involves the customer directly, who has to provide the resources needed for an effective deployment. The deployment phase also includes planning for how to monitor the models and evaluate when they have played out their role or need to be maintained. As last steps, a final report and project review should be performed.

3) With a neat diagram, Explain the IoT Reference Model

An ARM consists of two main parts: a Reference model and Reference architecture.

The foundation of an IoT Reference Architecture description is an IoT reference model. A reference model describes the domain using a number of sub-models (**Figure 4.11**). The domain model of an architecture model captures the main concepts or entities in the domain in question, in this case M2M and IoT. When these common language references are established , the domain model adds descriptions about the relationship between the concepts.

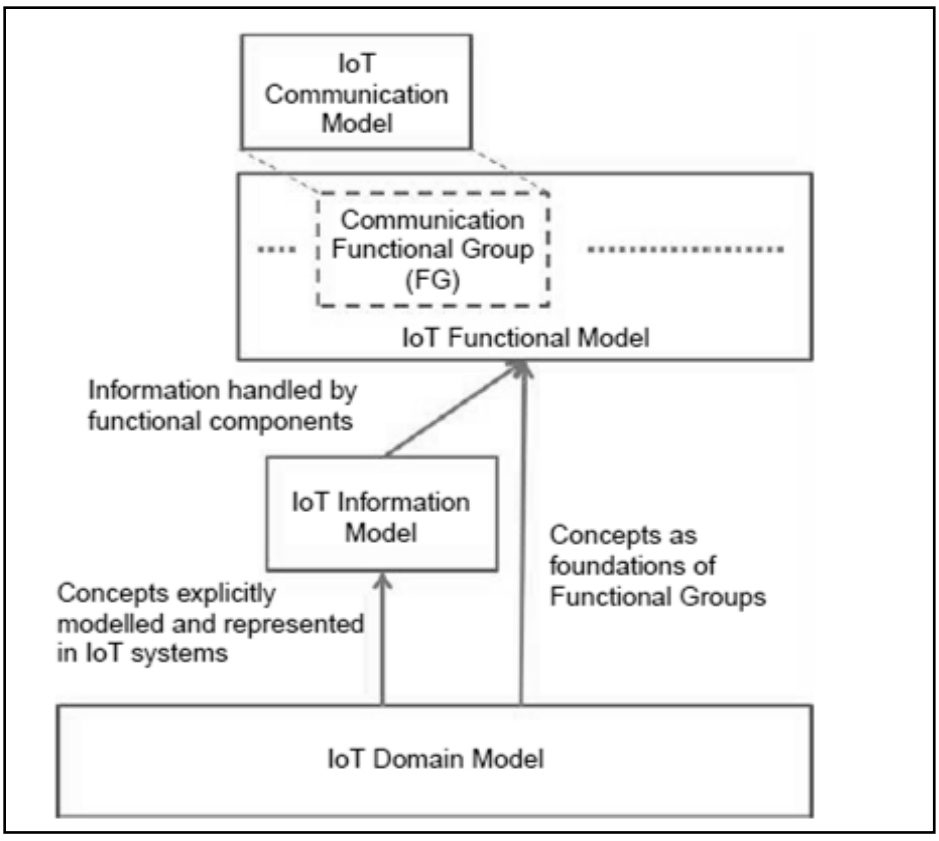


Figure 4.11: IoT Reference Model.

These concepts and relationships serve the basis for the development of an information model because a working system needs to capture and process information about its main entities and their interactions.

A working system that captures and operates on the domain and information model contains concepts and entities of its own, and these needs to be described in a separate model, the functional model. An M2M and IoT system contain communicating entities, and therefore the corresponding communication model needs to capture the communication interactions of these entities.

Apart from the reference model, the other main component of an ARM is the Reference Architecture. A System Architecture is a communication tool for different stakeholders of the system. Developers, component and system managers, partners, suppliers, and customers have different views of a single system based on their requirements and their specific interactions with the system.

The task becomes more complex when the architecture to be described is on a higher level of abstraction compared with the architecture of real functioning systems. The high- level abstraction is called Reference Architecture as it serves as a reference for generating concrete architectures and actual systems, as shown in the Figure 4.12

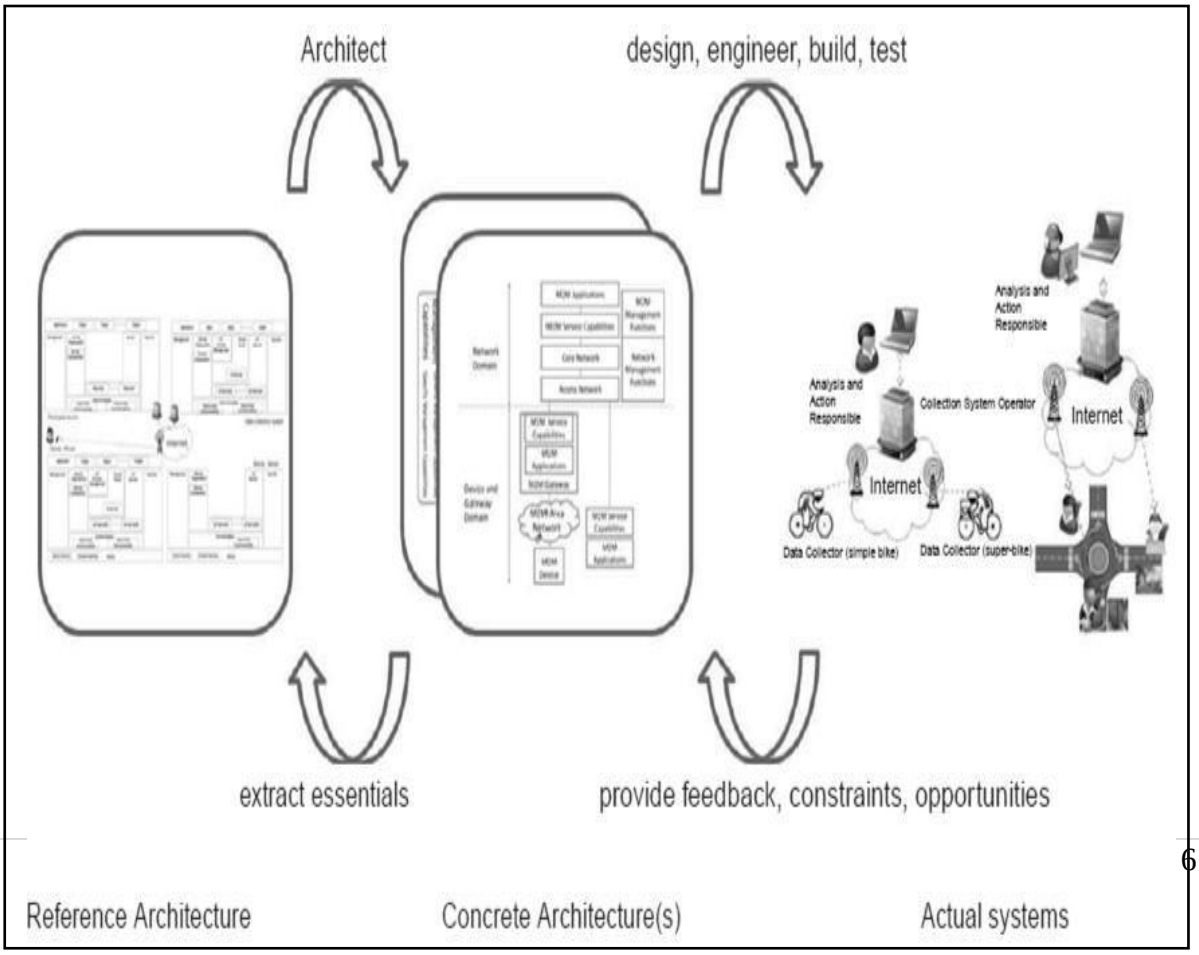


Figure 4.12: From reference to concrete architectures and actual systems.

- Concrete architectures are instantiations of rather abstract and high-level Reference Architectures.
- A Reference Architecture captures the essential parts of an architecture, such as design principles, guidelines, and required parts (such as entities), to monitor and interact with the physical world for the case of an IoT Reference Architecture.
- A concrete architecture can be further elaborated and mapped into real world components by designing, building, engineering, and testing the different components of the actual system. As the figure implies, the whole process is iterative, which means that the actual deployed system in the field provides invaluable feedback with respect to the design and engineering choices, current constraints of the system, and potential future opportunities that are fed back to the concrete architectures. The general essentials out of multiple concrete architectures can then be aggregated, and contribute to the evolution of the Reference Architecture.
- The IoT architecture model is related to the IoT Reference Architecture as shown in Figure 4.13. This figure shows two facets of the IoT ARM: (a)how to actually create an IoT ARM, and (b) how to use it with respect to building actual systems.

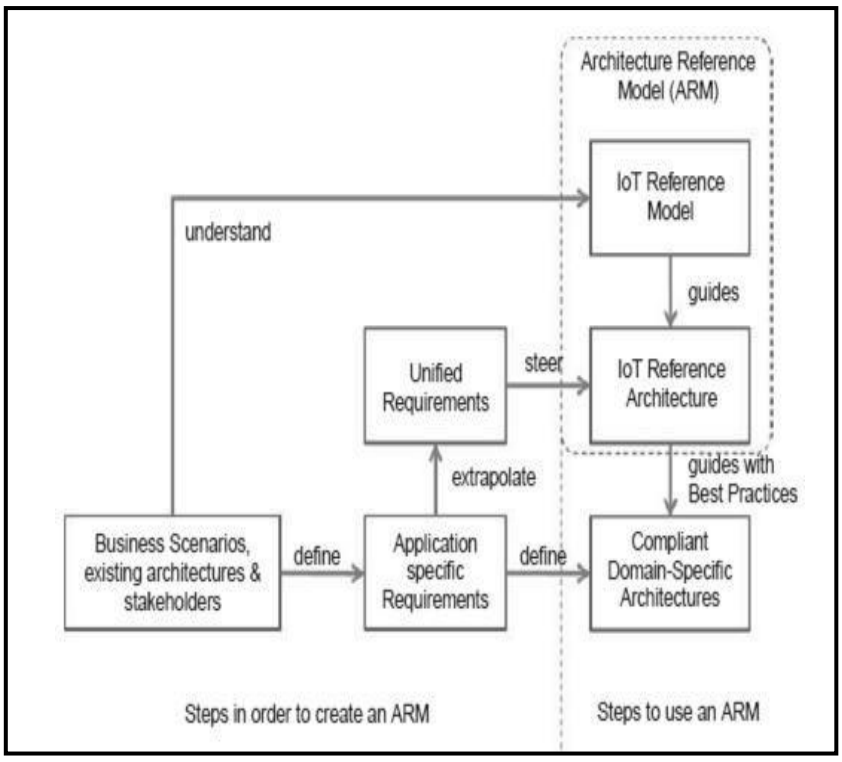


Figure 4.13: IoT Reference Model and Reference Architecture dependencies

4) Illustrate everything as a service (XaaS)

There is a general trend away from locally managing dedicated hardware toward cloud infrastructures that drives down the overall cost for computational capacity and storage. This is commonly referred to as "cloud computing."

- Cloud computing is a model for enabling ubiquitous, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be provisioned, configured, and made available with minimal management effort or service provider interaction.
- Cloud computing, however, does not change the fundamentals of software engineering. All applications need access to three things: compute, storage, and data processing capacities. With cloud computing, a fourth element is added _ distribution services _ i.e. the manner in which the data and computational capacity are linked together and coordinated.
- A cloud-computing platform may therefore be viewed conceptually (*Figure 5.11*). Several essential characteristics of cloud computing have been defined as follows:

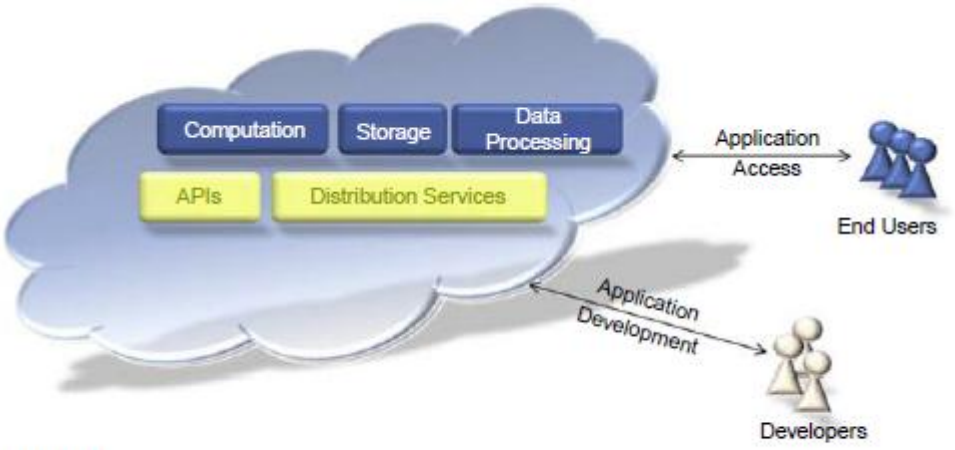


FIGURE 5.11

Conceptual Overview of Cloud Computing.

- **On-Demand Self-Service**. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed, or automatically, without requiring human interaction with each service provider.
- **Broad Network Access.** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g. mobile phones,tablets, laptops, and workstations).
- **Resource Pooling.** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer

generally has no control or knowledge over the exact location of the provided resources, but may be able to specify location at a higher level of abstraction (e.g. country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

o **Rapid Elasticity**. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited, and can be appropriated in any quantity at any time.

o **Measured Service**. Cloud systems automatically control and optimize resource use by leveraging a metering capability, at some level of abstraction, appropriate to the type of service (e.g. storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

· Once such infrastructures are available, however, it is easier to deploy applications in software. For M2M and IoT, these infrastructures provide the following:

1. Storage of the massive amounts of data that sensors, tags, and other "things" will produce.
2. Computational capacity in order to analyze data rapidly and cheaply.
3. Over time, cloud infrastructure will allow enterprises and developers to share datasets, allowing for rapid creation of information value chains.

Cloud computing comes in several different service models and deployment options for enterprises wishing to use it. The three main service models may be defined as

o **Software as a Service (SaaS):** Refers to software that is provided to consumers on demand, typically via a thin client. The end-users do not manage the cloud infrastructure in any way. This is handled by an Application Service Provider (ASP) or Independent Software Vendor (ISV). Examples include office and messaging software, email, or CRM tools housed in the cloud. The end-user has limited ability to change
anything beyond user-specific application configuration settings.

o **Platform as a Service (PaaS):** Refers to cloud solutions that provide both a computing platform and a solution stack as a service via the Internet. The customers themselves develop the necessary software using tools provided by the provider, who also provides the networks, the storage, and the other distribution services required. Again, the provider manages the underlying cloud infrastructure, while the customer has control over the deployed applications and possible settings for the application-hosting environment o **Infrastructure as a Service (IaaS):** In this model, the provider offers virtual machines and other resources such as hypervisors (e.g. Xen, KVM) to customers. Pools of hypervisors support the virtual machines and allow users to scale resource usage up and down in accordance with their computational requirements. Users install an OS image and application software on the cloud infrastructure. The provider manages the underlying cloud infrastructure, while the customer has control over OS, storage, deployed applications, and possibly some networking components.

o **Deployment Models:**

· ***Private Cloud***: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g. business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

· ***Community Cloud****:* The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g. mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

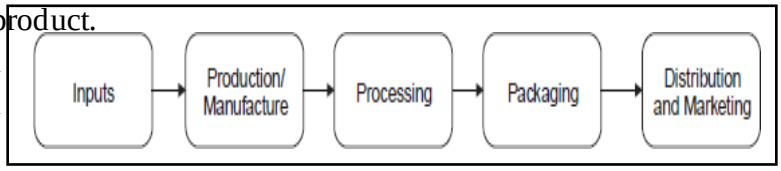
· ***Public Cloud:*** The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination thereof. It exists on the premises of the cloud provider.

· ***Hybrid Cloud:*** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g. cloud bursting for load balancing between clouds).

5) Summarize the steps involved in IoT Value chains..

A value chain describes the full range of activities that firms and workers perform to bring a product from its conception to end use and beyond, including design, production, marketing, distribution, and support to the final consumer.

A simplified value chain is illustrated in Figure 2.2; it is comprised of five separate activities that work together to create a finalized product. These activities may be contained within a single firm or divided among different firms.



6) List and explain the fundamental roles of information-driven global value chain (I-GVC)

IoT Value Chains, meanwhile, are about the use and reuse of data across value chains and across solutions. creating the possibility for many different economic entities to combine and share their data as long as they have a well-defined interface and description of how the data is formatted.

1. Inputs: The first thing that is apparent for an IoT value chain is that there are significantly more inputs than for an M2M solution:

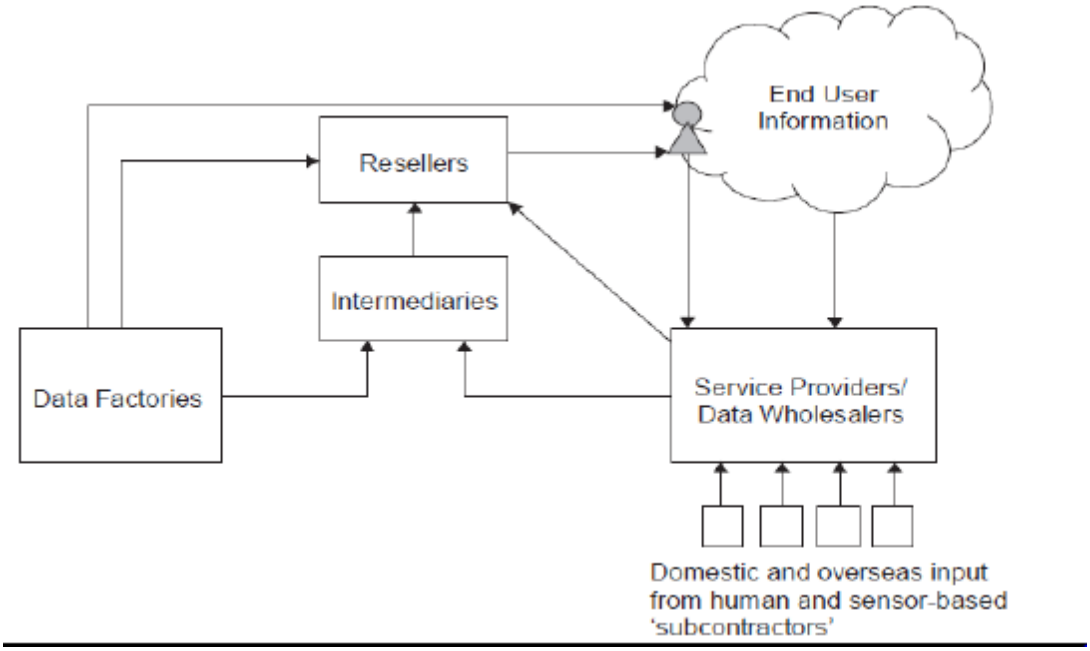
Devices/Sensors: these are very similar to the M2M solution devices and sensors, and may in fact be built on the same technology. As we will see later, however, the manner in which the data from these devices and sensors is used provides a different and much broader marketplace than M2M does.

Open Data: Open data is an increasingly important input to Information Value Chains. A broad definition of open data defines it as: "A piece of data is open if anyone is free to use, reuse, and redistribute it _subject only, at most, to the requirement to attribute and/or share-alike Open data requires a license stating that it is open data.

OSS/BSS: The Operational Support Systems and Business Support Systems of mobile operator networks are also important inputs to information value chains, and are being used increasingly in tightly closed information marketplaces that allow operators to deliver services to enterprises for example, where phone usage data is already owned by the company in question.

Corporate Databases: Companies of a certain size generally have multiple corporate databases covering various functions, including supply chain management, payroll, accounting, etc. Over the last decades, many of these databases within corporations have been increasingly interconnected using Internet Protocol (IP) technologies.

Production/Manufacture: In the production and manufacturing processes for data in an IoT solution, the raw inputs described above will undergo initial development into information components and products. Irrespective of input type described above, this process will need to
include tagging and linking of relevant data items in order to provide provenance and traceability across the information value chain. Some examples, as illustrated in Figure 1.3, are as follows:


End User
Information
Resellers
Intermediaries
Data Factories
Service Providers/
Data Wholesalers
Domestic and overseas input
from human and sensor-based
'subcontractors'

• **Asset Information**: Asset information may include data such as temperature over time of container during transit or air quality during a particular month. Essentially, this relates to whatever the sensor/device 9*/has been developed to monitor.
• **Open Data Sets**: Open data sets may include maps, rail timetables, or demographics about a certain area in a country or city.
• **Network Information:** Network information relates to information such as GPS data, services accessed via the mobile network, etc.
• **Corporate Information:** Corporate information may be, for example, the current state of demand for a particular product in the supply chain at a particular moment in time.
Processing: During the processing stage, data from various sources is mixed together. At this point, the data from the various inputs from the production and manufacture stage are combined together to create information.
4. Packaging: After the data from various inputs has been combined together, the packaging section of the information value chain creates information components. These components could be produced as charts or other traditional methods of communicating information to end-users.
5. Distribution/Marketing: The final stage of the Information Value Chain is the creation of an Information Product. A broad variety of such products may exist, but they fall into two main categories:
• Information products for improving internal decision-making: These information products are the result of either detailed information analysis that allows better decisions to be made during various internal corporate processes, or they enable the creation of previously unavailable knowledge about a company's products, strategy, or internal processes.
• Information products for resale to other economic actors: These information products have high value for other economic actors and can be sold to them. For example, through an IoT solution, a company may have market information about a certain area of town that another entity might pay for (e.g. a real-estate company).
4a) With a neat diagram, explain IoT architecture outline (10)
Asset Layer.
The assets of interest are the real-world objects and entities that are subject to being monitored and controlled, as well as having digital representations and identities.
o The typical examples include vehicles and machinery, fixed infrastructures such as buildings and utility systems, homes, and people themselves. Assets can also be of a more virtual character, being subjective representations of parts of the real world that are of interest to a person or an organization
Assets are instrumented with embedded technologies that bridge the digital realm with the physical world, and that provide the capabilities to monitor and control the assets as well as providing identities to the assets.

The Resource Layer
provides the main functional capabilities of sensing, actuation, and embedded identities. Sensors and actuators in various devices that may be smartphones or Wireless Sensor Actuator Networks (WSANs), M2M devices like smart meters, or other sensor/actuator nodes, deliver these functions.
o This is also where gateways of different types are placed that can provide aggregation or other capabilities that are closely related to these basic resources. Identification of assets can be provided by different types of tags; for instance, Radio Frequency Identification Or optical codes like bar codes or Quick Response(QR) codes.
Communication layer
Provide the means for connectivity between the resources on one end and the different computing instances that host and execute the service support and application logic on the other hand. It can use LAN or WAN.
WAN
o WANs can be realized by different wired or wireless technologies, for instance, fiber or Digital Subscriber Line (DSL) for the former, and cellular mobile networks, satellite, or microwave links for the latter.
o WANs can also be provided by different actors, where some networks can be regarded as public or as private.
o Particularly in the mobile network industry, there are different models for how the communications services are provided that include wholesale of access, and dedicated virtual network operators that focus on managed M2M connectivity offerings without owning licensed mobile spectrum or actual network resources.
LAN
Prime examples of LANs include Wireless Personal Area Networks (WPANs; also known as Body Area Networks, BANs) for fitness or healthcare applications, Home or Building Area Networks (HANs and BANs, respectively)used in automation and control applications, and Neighborhood Area Networks (NANs), which are used in the Distribution Grid of a Smart Electricity Grid. Communication can also be used in more ad hoc scenarios.Vehicle-to-Vehicle (V2V) is one example that can target safety applications like collision avoidance or car platooning. LANs use both wired and wireless technologies. General examples of wired LANs include Ethernet and Power Line Communication (PLC), whereas twisted pair (KNX 2013) and (BAC net 2013) over RS-232 are two detailed examples from the building automation industry. The ZigBee specifications (Zigbee 2013a), the proprietary protocol stack (Z-Wave 2013) for home automation, and ISA100.11a (ISA1002013) for industry automation.
Service Support Layer
 Executing in data centers or server farms inside organizations or in a cloud environment.
 These support services can provide uniform handling of the underlying devices and networks, thus hiding complexities in the communications and resource layers.
 Examples include remote device management that can do remote software upgrades, remote diagnostics or recovery, and dynamically reconfigure application processing such as setting event filters.
Communication-related functions include selection of communication channels if different networks can be used in parallel, for example, for reliability purposes, and publish_subscribe and message queue mechanisms. Location Based Service (LBS) capabilities and various Geographic Information System (GIS) services are also important for many IoT applications.

Data and Information Layer
 provides a more abstract set of functions as its main purposes are to capture knowledge and provide advanced control logic support Key concepts here include data and information models and knowledge representation in general,and the focus is on the organization of information.
 Knowledge Management Framework (KMF) as a collective term to include data, information, domain-specific knowledge, actionable services descriptions as,
 For example, represented by single actuators or more complex composite sensing and actuation services, service descriptors, rules,process or workflow descriptions, etc.

 The KMF needs to integrate anything from single pieces of data from individual sensors to highly domain-specific expert knowledge into a common knowledge fabric.
 Key concepts to construct the KMF include semantic annotation, Linked Data, and building different ontologies.

Application Layer
 Provides the specific IoT applications.
 There is an open-ended array of different applications, and typical examples include smart metering in the Smart Grid, vehicle tracking, building automation, or participatory sensing (PS).

Business Layer
 Focuses on supporting the core business or operations of any enterprise, organization, or individual that is interested in IoT applications.
 This is where any integration of the IoT applications into business processes and enterprise systems takes place.
 The enterprise systems can, for example, be Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), or other Business Support Systems (BSS).
 The business layer also provides exposure to APIs for third parties to get access to data and information, and can also contain support for direct access to applications by human users; for instance, city portal services for citizens in a smart city context, or providing necessary data visualizations to the human workforce in a particular enterprise.
 In addition to the functional layers, three functional groups cross the different layers, namely Management, Security, and IoT Data and Services. The former two are well known functions of a system solution,whereas the latter one is more specific to IoT.

Write short note on (i) data (ii) information (iii) knowledge
Explain Knowledge Reference Architecture for M2M and IoT with diagram.

8 **A) Write short note on (i) data (ii) information (iii) knowledge**

Data: Data refers to “unstructured facts and figures that have the least impact on the typical manager”. data includes both useful and irrelevant or redundant facts, and in order to become meaningful, needs to be processed.
 Information: Within the context of IoT solutions, information is data that has been contextualized, categorized, calculated, and condensed. This is where data has been carefully curated to provide relevance and purpose for the decision- makers in question. The majority of ICT solutions can be viewed as either storing information or processing data to become information.
 Knowledge: Knowledge, meanwhile, relates to the ability to understand the information presented, and using existing experience, the application of it within a certain decision making context.

B) Explain Knowledge Reference Architecture for M2M and IoT with diagram.

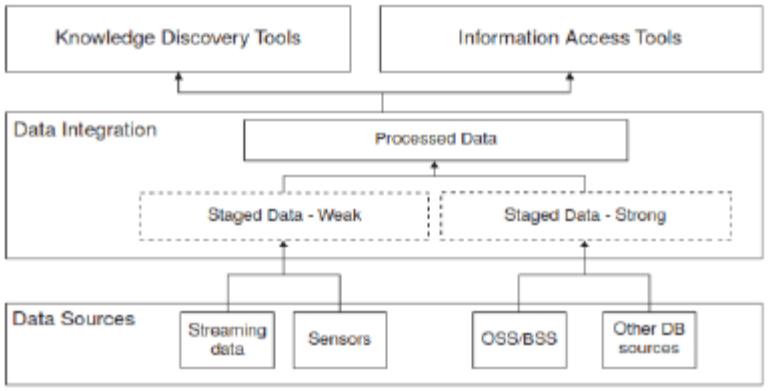


Figure No 5.17 Knowledge Reference Architecture for M2M and IoT.

Figure 5.17 outlines a high-level knowledge management reference architecture that illustrates how data sources from M2M and IoT may be combined with other types of data, for example, from databases or even OSS/ BSS data from MNOs. There are three levels to the diagram: (1) data sources, (2) data integration, and (3) knowledge discovery and information access.

 Data sources

Data sources refer to the broad variety of sources that may now be available to build enterprise solutions.

 Data integration

The data integration layer allows data from different formats to be put together in a manner that can be used by the information access and knowledge discovery tools.

 Staged Data: Staged data is data that has been abstracted to manage the rate at which it is received by the analysis platform. Essentially,"staged data" allows the correct flow of data to reach information access and knowledge discovery tools to be retrieved at the correct time. Big data and M2M analytics were discussed in detail in here we focus on the data types required for staging the data appropriately for knowledge frameworks. There are two main types of data: weak data and strong data. This definition is in order to differentiate between the manner in which data is encoded and its contents _ for example, the difference between XML and free text.

 Strong Type Data: Strong type data refers to data that is stored in traditional database formats, i.e. it can be extracted into tabular format and can be subjected to traditional database analysis techniques. Strong data types often have the analysis defined beforehand, e.g. by SQL queries written by developers towards a database.

 Weak Type Data: Weak type data is data that is not well structured according to traditional database techniques. Examples are streaming data or data from sensors. Often, this sort of data has a different analysis technique compared to strong type data. In this case, it may be that the data itself defines the nature of the query, rather than being defined by developers and created in advance. This may allow insights to be identified earlier than in strong type data.

 Processed data

Processed data is combined data from both strong and weak typed data that has been combined within an IoT context to create maximum value for the enterprise in question. There are various means by which to do this processing _ from stripping data separately and creating relational tables from it or pooling relevant data together in one combined database for structured queries. Examples could include combining the data from people as they move around the city from an operator's business support system with sensor data from various buildings in the city. A health service could then be created analyzing the end-users routes through a city and their overall health _ such a system may be used to more deeply assess the role that air pollution may play in health factors of the overall population.

Illustrate Internet Engineering Task Force architecture fragments in detail

1) Illustrate Internet Engineering Task Force architecture fragments in detail (10)

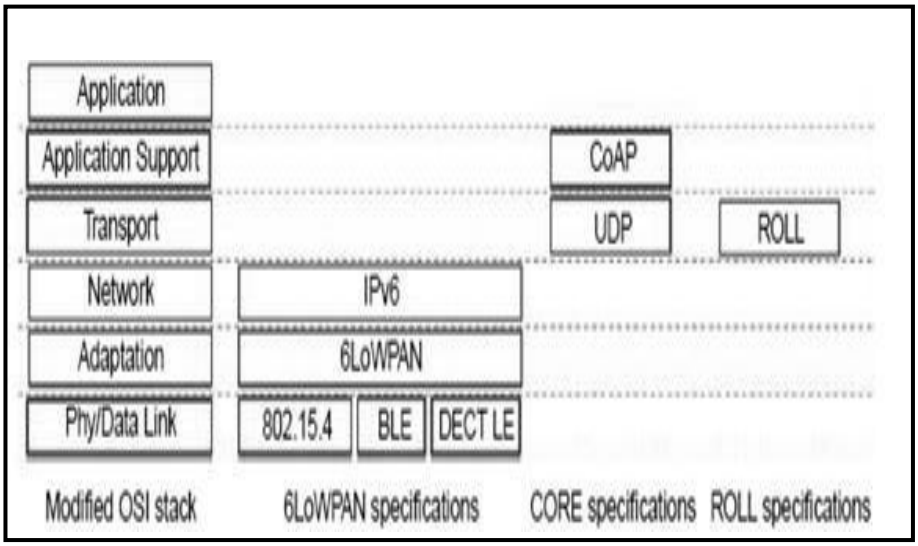


Figure 4.7: IETF Working Groups and Specification Scope

The modified Open Systems Interconnection (OSI) model in which several layers are merged because of the implementation on constrained devices.This is illustrated in Figure 4.7 as one layer called Application Support which includes the Presentation and Session Layers combined.

Moreover, one intermediate layer is introduced: the Adaptation Layer positioned between the Physical/Data Link and the Network Layer and whose main function is to adapt the Network Layer packets to Phy/Link layer packets among others.

An example of an adaptation layer is the 6LoWPAN layer designed to adapt IPv6 packets to IEEE 8021.5.4/Bluetooth Low Energy (BLE)/DECT Low Energy packets. An example of an Application Support Layer is IETF Constrained Application Protocol (CoAP), which provides reliability and RESTful operation support to applications.

Apart from the core of the specifications, the IETF CoRE workgroup includes several other draft specifications that sketch parts of an architecture for IoT. The CoRE Link Format specification describes a discovery method for the CoAP resources of a CoAP server.

The IETF CoRE working group has also produced a draft specification for a Resource Directory. A Resource Directory is a CoAP server resource (/rd) that maintains a list of resources, their corresponding server contact information (e.g. IP addresses or fully qualified domain name, or FQDN), their type, interface, and other information similar to the information

that the CoRE Link Format document specifies (Figure 4.8a).

An RD plays the role of a rendezvous mechanism for CoAP Server resource descriptions, in other words, for devices to publish the descriptions of the available resources and for CoAP clients to locate resources that satisfy certain criteria such as specific resource types.

A Mirror Server (Vial 2012) is a rendezvous mechanism for CoAP Server resource presentations. A Mirror Server is a CoAP Server resource (/ms) that maintains a list of resources and their cached representations (Figure 4.8b). A CoAP Server registers its resources to the Mirror Server, and upon registration a new mirror server resource is created on the Mirror Server with a container (mirror representation) for the original server representation. The original CoAP Server updates the mirror representation either periodically or when the representation changes. A CoAP Client that retrieves the mirror representation receives the latest updated representation from the original CoAP Server.

The Mirror Server is useful when the CoAP Server is not always available for direct access.

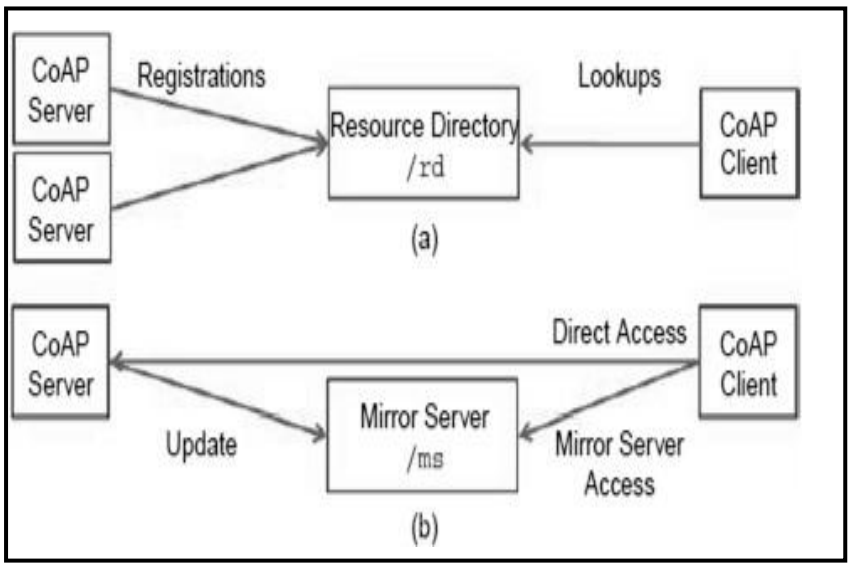


Figure 4.8: IETF CoRE Functional Components a) Resource Directory b) Mirror Server

The figure 4.9 The interworking issues appear when an HTTP Client accesses a CoAP Server through an HTTP-CoAP proxy or when a CoAP Client accesses an HTTP Server through a CoAP-HTTP proxy (Figure 4.9a).

The mapping process is not straightforward for a number of reasons. The main is the different transport protocols used by the HTTP and CoAP: HTTP uses TCP while CoAP uses UDP. The guidelines focus more on the HTTP-to-CoAP proxy and recommend addressing schemes (e.g.

how to map a CoAP resource address to an HTTP address),mapping between HTTP and CoAP response codes, mapping between different media types carried in the HTTP/CoAP payloads, etc.

a) Possible configurations b) example layer interaction upon a request from HTTP client to a CoAP server via a HTTP Proxy.

- As an example, consider the case that an HTTP Client sends an HTTP request to a CoAP server (Figure 4.9a) through a Gateway Device hosting an HTTP-CoAP Cross Proxy.
- The Gateway Device connects to the Internet via an Ethernet cable using a LAN, and on the CoAP side the CoAP server resides on a Sensor/Actuator (SAN) based on the IEEE 802.15.4 PHY/MAC.
- The HTTP request needs to include two addresses, one for reaching the Cross Proxy and one for reaching the specific CoAP Server in the SAN. The default recommended address mapping is to append the CoAP resource address (e.g. coap://s.coap.example.com/foo)to the Cross proxy address (e.g. http://p.example.com/.well-known/core/), resulting in http://p.example.com/.well-known/core/coap://s.coap.example.com/foo.
- The request is in plain text format and contains the method (GET). It traverses the IPv4 stack of the client, reaches the gateway, traverses the IPv4 stack of the gateway and reaches the Cross proxy.
- The request is translated to a CoAP request (binary format) with a destination CoAP resource coap://s.coap.example.com/foo, and it is dispatched in the CoAP stack of the gateway, which sends it over the SAN to the end device.A response is sent from the end device and follows the reverse path in the SAN order to reach the gateway.