


CMR INSTITUTE OF TECHNOLOGY		USN <input type="text"/>							
Internal Assessment Test – II, October 2019									
Sub:	INTERNET OF THINGS						Code:	17MCA552	
Date:	15-10-2019	Duration:	90 mins	Max Marks:	50	Sem:	V	Branch:	MCA
Answer ONE FULL QUESTION from each part								Marks	OBE
									CO RBT

Part – I

1a) Define Device and its properties	5	CO3	L2
b) Mention the differences between LAN and WAN. (OR)	5	CO3	L2
2 Explain Gateway in detail.	10	CO3	L1

Part – II

3a) List and explain the different types of devices.	5	CO3	L2
b) Discuss the deployment scenarios for devices. (OR)	5	CO3	L1
4 Explain about the different communication technologies for M2M and IOT	10	CO3	12

Part – III

5 Describe key characteristics and management of M2M data. (OR)	10	CO4	L3
--	----	-----	----

6. Discuss elaborately about the term - XaaS	10	CO3	L1
--	----	-----	----

Part – IV

7. With diagram explain CRISP-DM process in detail. (OR)	10	CO4	L1
---	----	-----	----

8a) How business processes has been distributed in IOT	10	CO4	L3
--	----	-----	----

Part – V

9 With a neat diagram, explain the overview of analytics architecture. (OR)	10	CO4	L1
--	----	-----	----

10a) Write short note on (i) data (ii) information (iii) knowledge	5	CO3	L2
b) Explain Knowledge Reference Architecture for M2M and IoT with diagram.	5	CO3	L1

Internal Assessment Test – II, October 2019 Answer Key

Sub :	INTERNET OF THINGS	Code:	17MCA552
----------	--------------------	-------	----------

Part – I

1a) Define Device and its properties

5

Device: A device is a hardware unit that can sense aspects of its environment and/or actuate, i.e. perform tasks in its environment

A device can be characterized as having several properties. including:

- **Microcontroller:** 8-, 16-, or 32-bit working memory and storage.
- **Power Source:** Fixed, battery, energy harvesting, or hybrid.
- **Sensors and Actuators:** Onboard sensors and actuators, or circuitry that allows them to be connected, sampled, conditioned, and controlled.
- **Communication:** Cellular, wireless, or wired for LAN and WAN communication.
- **Operating System (OS):** Main-loop, event-based, real-time, or full featured OS.
- **Applications:** Simple sensor sampling or more advanced applications.
- **User Interface:** Display, buttons, or other functions for user interaction.
- **Device Management (DM):** Provisioning, firmware, bootstrapping, and monitoring.
- **Execution Environment (EE):** Application lifecycle management and Application Programming Interface (API).

b) Mention the differences between LAN and WAN.

	LAN	WAN
Stands For	Local Area Network	Wide Area Network
Covers	Local areas only (e.g., homes, offices, schools)	Large geographic areas (e.g., cities, states, nations)
Definition	LAN (Local Area Network) is a computer network covering a small geographic area, like a home, office, school, or group of buildings.	WAN (Wide Area Network) is a computer network that covers a broad area (e.g., any network whose communications links cross metropolitan, regional, or national boundaries over a long distance).
Speed	High speed (1000 mbps)	Less speed (150 mbps)
Data transfer rates	LANs have a high data transfer rate.	WANs have a lower data transfer rate compared to LANs.
Example	The network in an office building can be a LAN	<u>The Internet</u> is a good example of a WAN

	LAN	WAN
Technology	Tend to use certain connectivity technologies, primarily <u>Ethernet</u> and Token Ring	WANs tend to use technologies like MPLS, ATM, Frame Relay and X.25 for connectivity over longer distances
Connection	One LAN can be connected to other LANs over any distance via telephone lines and radio waves.	Computers connected to a wide-area network are often connected through public networks, such as the telephone system. They can also be connected through leased lines or satellites.
Components	Layer 2 devices like <u>switches</u> and bridges. Layer 1 devices like hubs and repeaters.	Layers 3 devices Routers, Multi-layer Switches and Technology specific devices like ATM or Frame-relay Switches etc.
Fault Tolerance	LANs tend to have fewer problems associated with them, as there are smaller number of systems to deal with.	WANs tend to be less fault tolerant as they consist of large number of systems.
Ownership	Typically owned, controlled, and managed by a single person or organization.	WANs (like the Internet) are not owned by any one organization but rather exist under collective or distributed ownership and management over long distances.
Set-up costs	If there is a need to set-up a couple of extra devices on the network, it is not very expensive to do that.	For WANs since networks in remote areas have to be connected the set-up costs are higher. However WANs using public networks can be setup very cheaply using just software (VPN etc).
Geographical Spread	Have a small geographical range and do not need any leased telecommunication lines	Have a large geographical range generally spreading across boundaries and need leased telecommunication lines
Maintenance costs	Because it covers a relatively small geographical area, LAN is easier to maintain at relatively low costs.	Maintaining WAN is difficult because of its wider geographical coverage and higher maintenance costs.
Bandwidth	High bandwidth is available for transmission.	Low bandwidth is available for transmission.

2 Explain Gateway in detail.

10

A gateway serves as a translator between different protocols, e.g. between IEEE 802.15.4 or IEEE 802.11, to Ethernet or cellular.

□ There are many different types of gateways, which can work on different levels in the protocol layers. Most often a gateway refers to a device that performs translation of the physical and link layer, but

application layer gateways (ALGs) are also common. The latter is preferably avoided because it adds complexity and is a common source of error in deployments.

- Some examples of ALGs include the ZigBee Gateway Device (ZigBee Alliance 2011), which translates from ZigBee to SOAP and IP, or gateways that translate from Constrained Application Protocol (CoAP) to Hyper Text Transfer Protocol/Representational State Transfer (HTTP/REST).
- For some LAN technologies, such as 802.11 and Z-Wave, the gateway is used for inclusion and exclusion of devices.
- This typically works by activating the gateway into inclusion or exclusion mode and by pressing a button on the device to be added or removed from the network.
- For very basic gateways, the hardware is typically focused on simplicity and low cost, but frequently the gateway device is also used for many other tasks, such as data management, device management, and local applications. In these cases, more powerful hardware with GNU/Linux is commonly used.
- The following sections describe these additional tasks in more detail.

3.1.3.1 Data Management

- Typical functions for data management include performing sensor readings and caching this data, as well as filtering, concentrating, and aggregating the data before transmitting it to back-end servers.

3.1.3.2 Local applications

- Examples of local applications that can be hosted on a gateway include closed loops, home alarm logic, and ventilation control, or the data management.
- The benefit of hosting this logic on the gateway instead of in the network is to avoid downtime in case of WAN connection failure, minimize usage of costly cellular data, and reduce latency. The execution environment is responsible for the lifecycle management of the applications, including installation, pausing, stopping, configuration, and uninstallation of the applications.
- A common example of an execution environment for embedded environments is OSGi, which is based on java applications are built as one or more Bundles, which are packaged as Java JAR files and installed using a so-called Management Agent. The Management Agent can be controlled from, for example, a terminal shell or via a protocol such as CPE WAN Management Protocol (CWMP).
- Bundle packages can be retrieved from the local file system or over HTTP.
- The benefit of versioning and the lifecycle management functions is that the OSGi environment never needs to be shut down when upgrading, thus avoiding downtime in the system.

3.1.3.3 Device Management

- Device management (DM) is an essential part of the IoT and provides efficient means to perform many of the management tasks for devices:
 - o **Provisioning:** Initialization (or activation) of devices in regards to configuration and features to be enabled.
 - o **Device Configuration:** Management of device settings and parameters.
 - o **Software Upgrades:** Installation of firmware, system software, and applications on the device.
 - o **Fault Management:** Enables error reporting and access to device status
- In the simplest deployment, the devices communicate directly with the DM server. This is, however, not always optimal or even possible due to network or protocol constraints, e.g. due to a firewall or mismatching protocols.
- In these cases, the gateway functions as mediator between the server and the devices, and can operate in three different ways:
 - o If the devices are visible to the DM server, the gateway can simply forward the messages between the device and the server and is not a visible participant in the session.
 - o In case the devices are not visible but understand the DM protocol in use, the gateway can act as a proxy, essentially acting as a DM server towards the device and a DM client towards the server.
 - o For deployments where the devices use a different DM protocol from the server, the gateway can represent the devices and translate between the different protocols (e.g. TR-069, OMA-DM, or CoAP). The devices can be represented either as virtual devices or as part of the gateway

3a) List and explain the different types of devices.

Basic Devices

Devices that only provide the basic services of sensor readings and/or actuation tasks, and in some cases limited support for user interaction. LAN communication is supported via wired or wireless technology, thus a gateway is needed to provide the WAN connection.

- Intended for a single purpose - measuring air pressure or closing a valve. Some cases several functions are deployed on the same device- monitoring humidity, temperature, and light level.
- The requirements on hardware are low, both in terms of processing power and memory.
- The main focus is on keeping the bill of materials (BOM) as low as possible by using inexpensive microcontrollers with built-in memory and storage, often on an SoC integrated circuit with all main components on one single chip.
- The microcontroller hosts a number of ports that allow integration with sensors and actuators, such as General Purpose I/O (GPIO) and an analog-to-digital converter (ADC) for supporting analog input.
- For certain actuators, such as motors, pulse-width modulation (PWM) can be used.

Advanced devices

In this case the devices also host the application logic and a WAN connection. They may also feature device management and an execution environment for hosting multiple applications. Gateway devices are most likely to fall into this category.

- The distinction between basic devices, gateways, and advanced devices is not cut in stone, but some features that can characterize an advanced device are the following:
 - A powerful CPU or microcontroller with enough memory and storage to host advanced applications, such as a printer offering functions for copying, faxing, printing, and remote management.
 - A more advanced user interface with, for example, display and advanced user input in the form of a keypad or touch screen.
 - Video or other high bandwidth functions.

b) Discuss the deployment scenarios for devices.

5

- **Home Alarms:** Such devices typically include motion detectors, magnetic sensors, and smoke detectors.
- **Smart Meters:** The meters are installed in the households and measure consumption of, for example, electricity and gas. A concentrator gateway collects data from the meters, performs aggregation, and periodically transmits the aggregated data to an application server over a cellular connection.
- **Building Automation System(BASs):** Such devices include thermostats, fans, motion detectors, and boilers, which are controlled by local facilities, but can also be remotely operated.
- **Standalone Smart Thermostats:** These use Wi-Fi to communicate with web services. Examples for advanced devices, meanwhile, include: Onboard units in cars
- **Onboard units in cars** that perform remote monitoring and configuration over a cellular connection.
- **Robots and autonomous vehicles** such as unmanned aerial vehicles that can work both autonomously or by remote control using a cellular connection
- **Video cameras** for remote monitoring over 3G and LTE.
- **Oil well monitoring** and collection of data points from remote devices.
- **Connected printers** that can be upgraded and serviced remotely.

4 Explain about the different communication technologies for M2M and IOT 10

These are the communications technologies that are considered to be critical to the realization of massively distributed M2M applications and the IoT at large.

o *Power Line Communication*

□ (PLC) refers to communicating over power (or phone, coax, etc.) lines. This amounts to pulsing, with various degrees of power and frequency, the electrical lines used for power distribution. PLC comes in numerous flavors. At low frequencies (tens to hundreds of Hertz) it is possible to communicate over kilometers with low bit rates (hundreds of bits per second).

□ Typically, this type of communication was used for remote metering, and was seen as potentially useful for the smart grid. Enhancements to allow higher bit rates have led to the possibility of delivering broadband connectivity over power lines.

o LAN (and WLAN)

□ Continues to be important technology for M2M and IoT applications.

□ This is due to the high bandwidth, reliability, and legacy of the technologies. Where power is not a limiting factor, and high bandwidth is required, devices may connect seamlessly to the Internet via Ethernet (IEEE 802.3) or Wi-Fi (IEEE 802.11). The utility of existing (W) LAN infrastructure is evident in a number of early IoT applications targeted at the consumer market, particularly where integration and control with smartphones is required.

o Bluetooth Low Energy

□ (BLE; “Bluetooth Smart”) is a recent integration of Nokia’s Wibree standard with the main Bluetooth standard. It is designed for short-range (50 m) applications in healthcare, fitness, security, etc., where high data rates (millions of bits per second) are required to enable application functionality. It is deliberately low cost and energy efficient by design, and has been integrated into the majority of recent smartphones.

o Low-Rate, Low-Power Networks

□ are another key technology that form the basis of the IoT.

o IPv6 Networking

□ making the fact that devices are networked, with or without wires, with various capabilities in terms of range and bandwidth, essentially seamless.

□ It is foreseeable that the only hard requirement for an embedded device will be that it can somehow connect with a compatible gateway device.

o 6LoWPAN

□ (IPv6 Over Low Power Wireless Personal Area Networks) was developed initially by the LoWPAN Working Group (WG) of the IETF

□ The 6LoWPAN concept originated from the idea that “the Internet Protocol could and should be applied even to the smallest devices”, and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things

o RPL

□ IPv6 **Routing Protocol** for Low-Power and Lossy Networks. Abstract Low-Power and Lossy Networks (LLNs) are a class of network in which both the routers and their interconnect are constrained. LLN routers typically operate with constraints on processing power, memory, and energy (battery power).

o CoAP

□ Constrained Application Protocol (CoAP) is a protocol that specifies how low-power compute-constrained devices can operate in the internet of things (IoT).

5 Describe key characteristics and management of M2M data.

10

Big Data: Huge amounts of data are generated, capturing detailed aspects of the processes where devices are involved.

O Heterogeneous Data: The data is produced by a huge variety of devices and is itself highly heterogeneous, differing on sampling rate, quality of captured values, etc.

O Real-World Data: The overwhelming majority of the M2M data relates to realworld processes and is dependent on the environment they interact with.

O Real-Time Data: M2M data is generated in real-time and overwhelmingly can be communicated also in a very timely manner. The latter is of pivotal importance since many times their business value depends on the real-time processing of the info they convey.

Temporal Data: The overwhelming majority of M2M data is of temporal nature, measuring the environment over time.

O Spatial Data: Increasingly, the data generated by M2M interactions are not only captured by mobile devices, but also coupled to interactions in specific locations, and their assessment may dynamically vary depending on the location.

O Polymorphic Data: The data acquired and used by M2M processes may be complex and involve various data, which can also obtain different meanings depending on the semantics applied and the process they participate in.

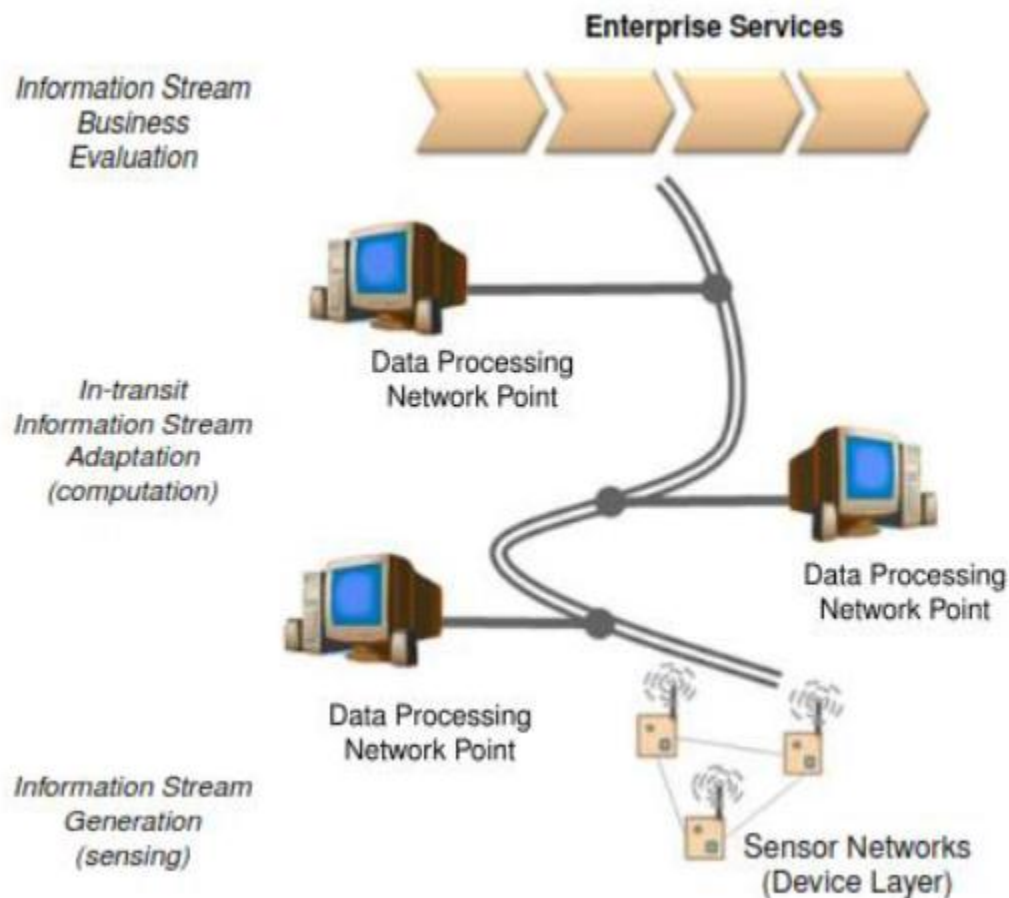
O Proprietary Data: Up to now, due to monolithic application development, a significant amount of M2M data is stored and captured in proprietary formats. However, increasingly due to the interactions with heterogeneous devices and stakeholders, open approaches for data storage and exchange are used.

O Security and Privacy Data Aspects: Due to the detailed capturing of interactions by M2M, analysis of the obtained data has a high risk of leaking private information and usage patterns, as well as compromising security.

Data Management

The data flow from the moment it is sensed (e.g. by a wireless sensor node) up to the moment it reaches the backend system has been processed manifold, either to adjust its representation in order to be easily integrated by the diverse applications, or to compute on it in order to extract and associate it with respective business intelligence (e.g. business process affected, etc.).

- As in figure 3.5 we see a number of data processing network points between the machine and the enterprise that act on the data stream (or simply forwarding it) based on their end-application needs and existing context.



Data generation

- Data generation is the first stage within which data is generated actively or passively from the device, system, or as a result of its interactions.
- The sampling of data generation depends on the device and its capabilities as well as potentially the application needs.
- Usually default behaviors for data generation exist, which are usually further configurable to strike a good benefit between involved costs, e.g. frequency of data collection vs. energy used in the case of WSNs, etc. Not all data acquired may actually be communicated as some of them may be assessed locally and subsequently disregarded, while only the result of the assessment may be communicated.

Data acquisition

- Data acquisition deals with the collection of data (actively or passively) from the device, system, or as a result of its interactions. The data acquisition systems usually communicate with distributed devices over wired or wireless links to acquire the needed data, and need to respect security, protocol, and application requirements. The nature of acquisition varies, e.g. it could be continuous monitoring, interval-poll, event-based, etc. The frequency of data acquisition overwhelming depends on, or is customized by, the application requirements.
- The data acquired at this stage (for non-closed local control loops) may also differ from the data actually generated. In simple scenarios, due to customized filters deployed at the device, a fraction of the generated data (e.g. adhering to the time of interest or over a threshold) may be communicated.
- Additionally, in more sophisticated scenarios, data aggregation and even on-device computation of the data may result in communication of key performance indicators of interest to the application, which are calculated based on a device's own intelligence and capabilities.

Data validation

- Data acquired must be checked for correctness and meaningfulness within the specific operating context. The latter is usually done based on rules, semantic annotations, or other logic.
- Data validation in the era of M2M, where the acquired data may not conform to expectations, is a must as data may be intentionally or unintentionally corrupted during transmission, altered, or not make sense in the business context. As real-world processes depend on valid data to draw business-relevant decisions, this is a key stage, which sometimes does not receive as much attention as it should.
- Several known methods are deployed for consistency and data type checking; for example, imposed range limits on the values acquired, logic checks, uniqueness, correct time-stamping, etc. In addition, semantics may play an increasing role here, as the same data may have different meanings in various operating contexts, and via semantics one can benefit while attempting to validate them. Another part of the validation may deal with fallback actions such as requesting the data again if checks fail, or attempts to "repair" partially failed data.
- Failure to validate may result in security breaches. Tampered-with data fed to an application is a well-known security risk as its effects may lead to attacks on other services, privilege escalation, denial of service, database corruption, etc., as we have witnessed on the Internet over the last decades. As full utilization of this step may require significant computational resources, it may be adequately tackled at the network level (e.g. in the cloud), but may be challenging in direct M2M interactions, e.g. between two resource constrained machines communicating directly with each other.

3.2.3.4 Data Storage

- The data generated by M2M interactions is what is commonly referred to as “Big Data.” Machines generate an incredible amount of information that is captured and needs to be stored for further processing. As this is proving challenging due to the size of information, a balance between its business usage vs. storage needs to be considered; that is, only the fraction of the data relevant to a business need may be stored for future reference.
- This means, for instance, that in a specific scenario, (usually for on-the-fly data that was used to make a decision) once this is done, the processed result can be stored but not necessarily the original data. However, one has to carefully consider what the value of such data is to business not only in current processes, but also potentially other directions that may be followed in the future by the company as different assessments of the same data may provide other, hidden competitive advantages in the future.
- Due to the massive amounts of M2M data, as well as their envisioned processing (e.g. searching), specialized technologies such as massively parallel processing DBs, distributed file systems, cloud computing platforms, etc. are needed.

Data processing

- Data processing enables working with the data that is either at rest (already stored) or in motion (e.g. stream data). The scope of this processing is to operate on the data at a low level and “enhance” them for future needs.
- Typical examples include data adjustment during which it might be necessary to normalize data, introduce an estimate for a value that is missing, reorder incoming data by adjusting timestamps, etc. Similarly, aggregation of data or general calculation functions may be operated on two or more data streams and mathematical functions applied on their composition.
- Another example is the transformation of incoming data; for example, a stream can be converted on the fly (e.g. temperature values are converted from F to C), or repackaged in another data model, etc. Missing or invalid data that is needed for the specific time-slot may be forecasted and used until, in a future interaction, the actual data comes into the system.
- This stage deals mostly with generic operations that can be applied with the aim to enhance them, and takes advantage of low-level (such as DB stored procedures) functions that can operate at massive levels with very low overhead, network traffic, and other limitations.

Data remanence

- M2M data may reveal critical business aspects, and hence their lifecycle management should include not only the acquisition and usage, but also the end-of-life of data. However, even if the data is erased or removed, residues may still remain in electronic media, and may be easily recovered by third parties _ often referred to as data remanence.
- Several techniques have been developed to deal with this, such as overwriting, degaussing, encryption, and physical destruction. For M2M, points of interest are not only the DBs where the M2M data is collected, but also the points of action, which generate the data, or the individual nodes in between, which may cache it.
- At the current technology pace, those buffers (e.g. on device) are expected to be less at risk since their limited size means that after a specific time has elapsed, new data will occupy that space; hence, the window of opportunity is rather small. In addition, for large-scale infrastructures the cost of potentially acquiring “deleted” data may be large; hence, their hubs or collection end-points, such as the DBs who have such low cost, may be more at risk.
- In light of the lack of cross industry M2M policy-driven data management, it also might be difficult to
- not only control how the M2M data is used, but also to revoke access to it and “delete” them from the Internet once shared.

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:

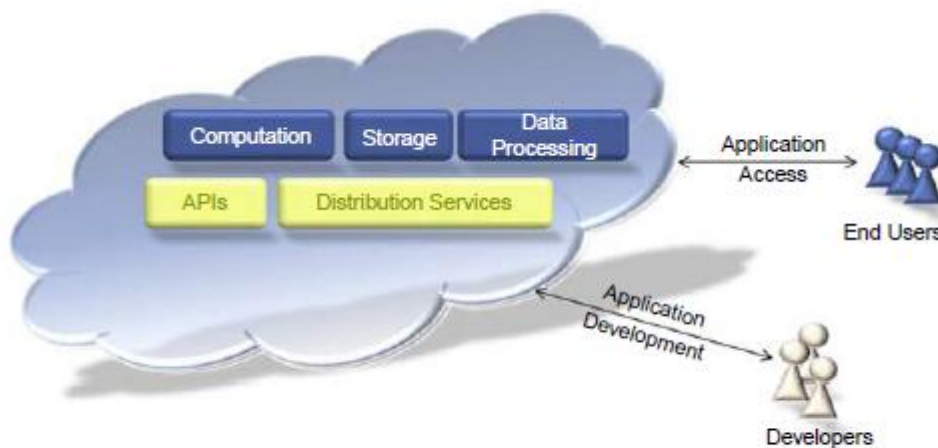


FIGURE 5.11

Conceptual Overview of Cloud Computing.

O 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.

O 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).

O 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).

7. With diagram explain CRISP-DM process in detail.

10

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.

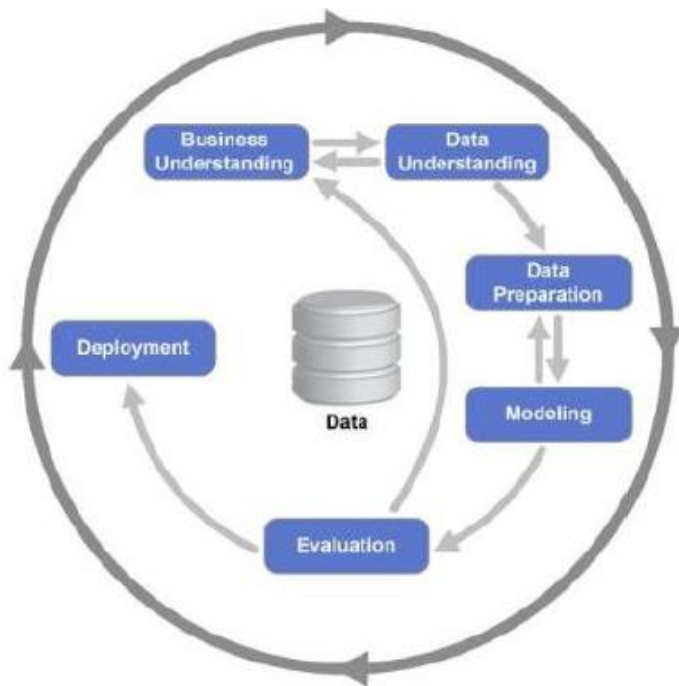


Figure 5. 15 CRISP-DM Process Diagram.

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.

8. How business processes have been distributed in IOT

10

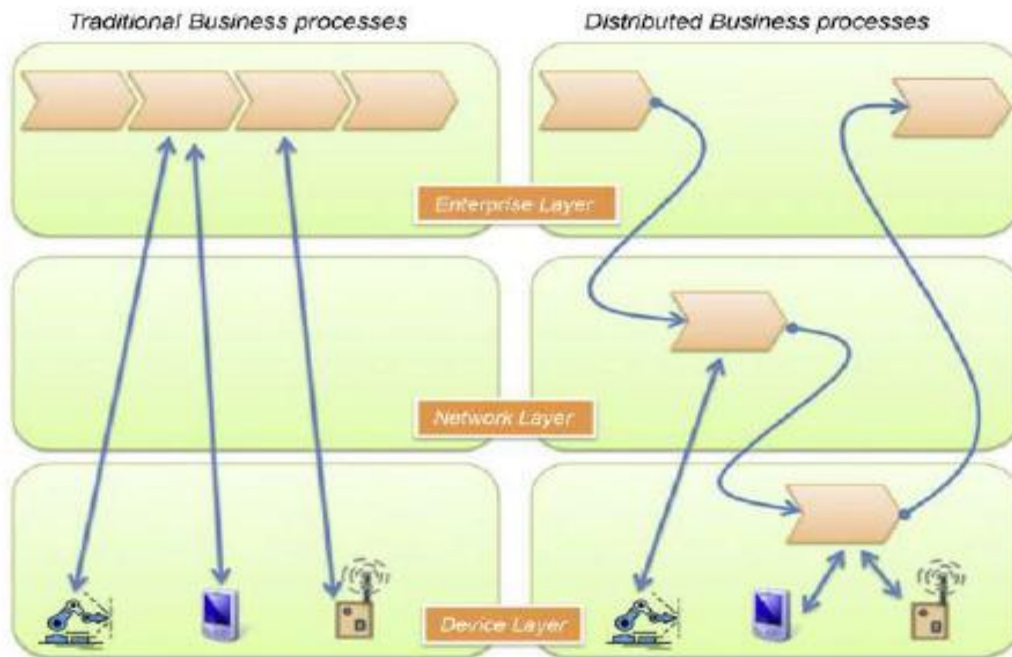


FIGURE 5.9

Distributed Business Processes in M2M era.

- As seen on the left part of [Figure 5.9](#), the integration of devices in business processes merely implies the acquisition of data from the device layer, its transportation to the backend systems, its assessment, and once a decision is made, potentially the control (management) of the device, which adjusts its behavior. However, in the future, due to the large scale of IoT, as well as the huge data that it will generate, such approaches are not viable.
- Transportation of data from the “point of action” where the device collects or generates them, all the way to the backend system to then evaluate their usefulness, will not be practical for communication reasons, as well as due to the processing load that it will incur at the enterprise side; this is something that the current systems were not designed for.
- Enterprise systems trying to process such a high rate of non- or minor-relevancy data will be overloaded. As such, the first strategic step is to minimize communication with enterprise systems to only what is relevant for business. With the increase in resources (e.g. computational capabilities) in the network, and especially on the devices themselves (more memory, multi-core CPUs, etc.), it makes sense not to host the intelligence and the computation required for it only on the enterprise side, but actually distribute it on the network, and even on the edge nodes (i.e. the devices themselves), as depicted on the right side of [Figure 5.9](#).
- Partially outsourcing functionality traditionally residing in backend systems to the network itself and the edge nodes means we can realize distributed business processes whose sub-processes may execute outside

the enterprise system. As devices are capable of computing, they can either realize the task of processing and evaluating business relevant information they generate by themselves or in clusters.

- Distributing the computational load in the layers between enterprises and the real-world infrastructure is not the only reason; distributing business intelligence is also a significant motivation. Business processes can bind during execution of dynamic resources that they discover locally, and integrate them to better achieve their goals. Being in the world of service mash-ups, we will witness a paradigm change not only in the way individual devices, but also how clusters of them, interact with each other and with enterprise systems.
- Modeling of business processes (Spiess et al. 2009) can now be done by focusing on the functionality provided and that can be discovered dynamically during runtime, and not on the concrete implementation of it; we care about what is provided but not how, as depicted in Figure 5.10.
- As such, we can now model distributed business processes that execute on enterprise systems, in-network, and on-device. The vision (Spiess et al. 2009) is to additionally consider during runtime the requirements and costs associated with the execution in order to select the best of available instances and optimize the business process in total according to the enterprise needs, e.g. for low impact on a device's energy source, or for highspeed communication, etc.

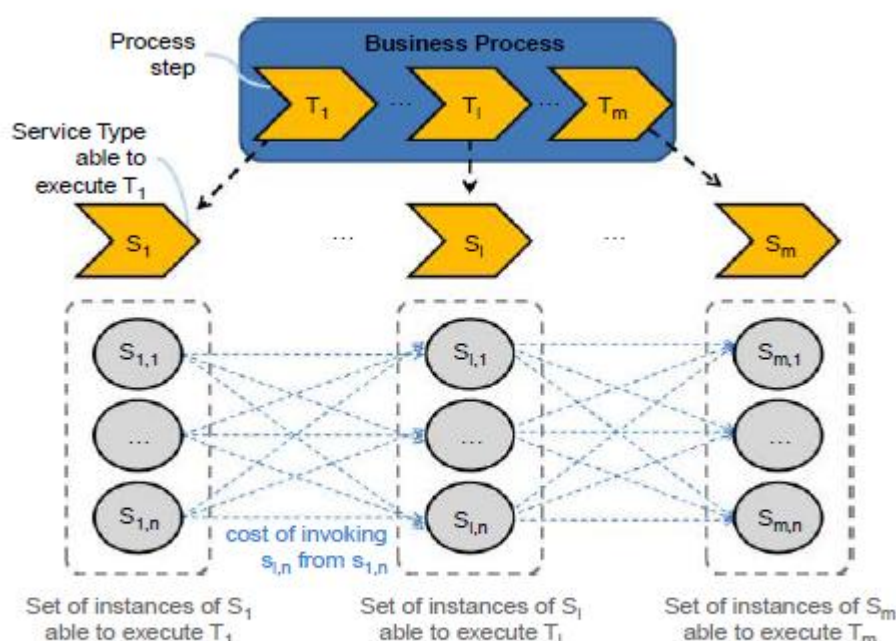


FIGURE 5.10

On-Device and in-network Business Process Composition and runtime execution.

9 With a neat diagram, explain the overview of analytics architecture.

10

An architecture for analytics needs to take a few basic requirements into account (Figure 5.14) One of these is to serve as a platform for data exploration and modeling by data scientists and other advanced information consumers performing business analytics and intelligence.

□ As much time is spent on data preparation before any analytics can take place, this is also an integral part of the architecture to facilitate. Finally, efficient means of building and viewing reports, as well as integrating with back-end systems and business processes, is of importance. These requirements concern batch analytics, but should also be considered for stream analytics.

□ Note that an analytics architecture is not intended for general-purpose data storage, although sometimes it's efficient to co-locate these two functions into one architecture. Risks of affecting production must, however, be taken into consideration if this is done instead of importing the data into an analytics sandbox where analysts can work on the data independently.

- Another benefit with an analytics sandbox is also that this environment offers a full suite of analytical tools that normally cannot be found in a traditional database. It also offers a development platform with the necessary computing resources required to perform complex analytics on very big data sets.
- A sandbox for Big Data analytics can be realized in a number of ways, of which the Hadoop ecosystem is probably the best known.
- Other alternatives include:
 - o Columnar databases such as HP Vertica, Actian ParAccel MPP, SAP Sybase IQ, and Infobright.
 - o Massively Parallel Processing (MPP) architectures such as Pivotal Greenplum and Teradata Aster.
 - o In-memory databases such as SAP Hana and QlikView.
- All of the above focus on batch-oriented analytics, where all data is available for the model generation.
- A complimentary method is to perform analytics on the live data streams (i.e. stream analytics), which means that the data does not need to be stored after it has been processed. This in turn limits the available algorithms to those that can handle incremental model building. The most common technologies in this segment are Event Stream Processing (e.g. Twitter Storm and Apache S4) and Complex Event Processing (e.g. EsperTech Esper and SAP Sybase Event Stream Processor). An analytical architecture should preferably also provide:
 - Authentication and authorization to access data.
 - Failover and redundancy features
 - Management facilities.
 - Efficient batch loading of data and support self-service.
 - Scheduling of batch jobs, such as data import and model training.
 - Connectors to import data from external sources.
 - The core of Hadoop is the MapReduce programming model, which allows processing of large data sets by deploying an algorithm, written as a program, onto a cluster of nodes.
 - A MapReduce job reads data from the Hadoop File System (HDFS), and runs on the same nodes as the deployed algorithm. This allows the Hadoop framework to utilize data locality as much as possible to avoid unnecessary transfer of data between the nodes. MapReduce is batch-oriented and intended for very large jobs that typically take an hour or more to execute. The nodes and services in a Hadoop cluster are coordinated by ZooKeeper, which serves as a central naming and configuration service.
 - Although it's not unusual for developers to use MapReduce directly, there exist a number of technologies that provide further abstraction levels, such as:
 - o **HBase**: A column-oriented data store that provides real-time read/write access to very large tables distributed over HDFS.

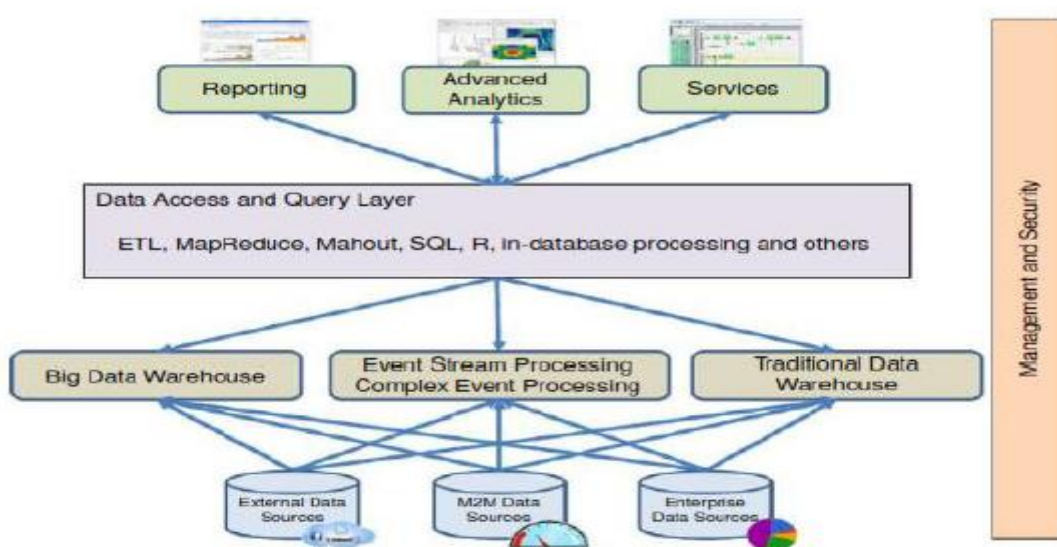


Figure 3.14 Analytics Architectural Overview.

- o Mahout: A distributed and scalable library of machine learning algorithms that can make use of MapReduce.
- o Pig: A tool for converting relational algebra scripts into MapReduce jobs that can read data from HDFS and HBase.
- o Hive: Similar to Pig, but offers an SQL-like scripting language called HiveQL instead.
- o Impala: Offers low-latency queries using HiveQL for interactive exploratory analytics, as compared to Hive, which is better suited for long running batch-oriented tasks.

10a) Write short note on (i) data (ii) information (iii) knowledge

5

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.

□ For IoT solutions to be practicable, the data management and information presentation within them needs to take into consideration real-time performance, complexity, and the human-data interface. Knowledge management in this context needs to perform a careful balancing act between the sheer speed of incoming data sets and the provision of a user-centric presentation view. Due to the nature of big data, as we discussed in previous sections, two key issues emerge:

o Managing and storing the temporal knowledge created by IoT solutions. IoT solutions data will evolve rapidly over time, the temporal nature of the “knowledge” as understood at a particular point in time will have large implications for the overall industry. For example, it could affect insurance claims if the level of knowledge provided by an IoT system could be proven to be inadequate.

o Life-cycle management of knowledge within IoT systems. Closely related to analytics, the necessity to have a lifecycle plan for the data within a system is a strong requirement.

□ Having covered the differences between data, information, and knowledge, we now move to outlining a reference architecture for knowledge management in IoT solutions. Existing knowledge management frameworks have previously focused on clearly structured data, generally found in databases that can be stored in a form that is easily analyzed via various well-established tools

b) Explain Reference Architecture for M2M and IoT with diagram

5

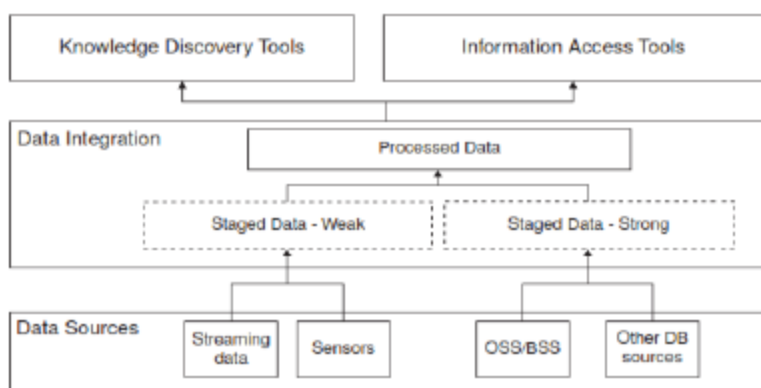


Figure No 5.17 Knowledge Reference Architecture for M2M and IoT.

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.