Module-3 M2M and IoT Technology Fundamentals

Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management.

3.1 Devices and gateways

3.1.1 Introduction

- Embedded processing is evolving, not only towards higher capabilities and processing speeds, but also towards allowing the smallest of applications to run on them.
- There is a growing market for small-scale embedded processing such as 8-, 16-, and 32-bit microcontrollers with on-chip RAM and flash memory, I/O capabilities, and networking interfaces such as IEEE 802.15.4 that are integrated on tiny System-on-a-Chip (SoC) solutions
- Such devices enable very con strained devices with a small footprint of a few mm and with a very low power consumption in the milli- to micro-Watt range, but which are capable of hosting an entire Transmission Control Protocol/Internet Protocol (TCP/IP) stack, including a small web server.
- **Device**: A device is a hardware unit that can sense aspects of it's 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).
- For several reasons, one or more of these functions are often hosted on gateway instead.
- This can be to save battery power, for example, by letting the gateway handle heavy functions such as WAN connectivity and application logic that requires a powerful processor. This also leads to reduced costs because these are expensive components.
- Another reason is to reduce complexity by letting a central node (the gateway) handle functionality such as device management and advanced applications, while letting the devices focus on sensing and actuating.

3.1.1.1 Device Types

- There are no clear criteria today for categorizing devices, but instead there is more of a sliding scale. Devices are grouped into two categories (Table 3.1):

Table 3.1 Example Characteristics of the Device Types

	CPU	Memory	Power	Comm	OS, EE
Basic	8-bit PIC, 8-bit 8051, 32-bit Cortex-M	Kilobytes	Battery	802.15.4, 802.11, Z-Wave	Main-loop, Contiki, RTOS[a]
Advanced	32-bit ARM9, Intel Atom	Megabytes	Fixed	802.11, LTE, 3G, GPRS	Linux, Java, Python

[a]*Real-time operating system.*

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.

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.

3.1.1.2 Deployment scenarios for devices

Deployment can differ for basic and advanced deployment scenarios. Example deployment scenarios for basic devices include:

- **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.

The devices and gateways of today often use legacy technologies such as KNX, Z-Wave, and ZigBee, but the vision for the future is that every device can have an IP address and be (in)directly connected to the Internet.

3.1.2 Basic Devices

- These devices are often intended for a single purpose, such as measuring air pressure or closing a valve. In some cases several functions are deployed on the same device, such as 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(Figure 3.1)

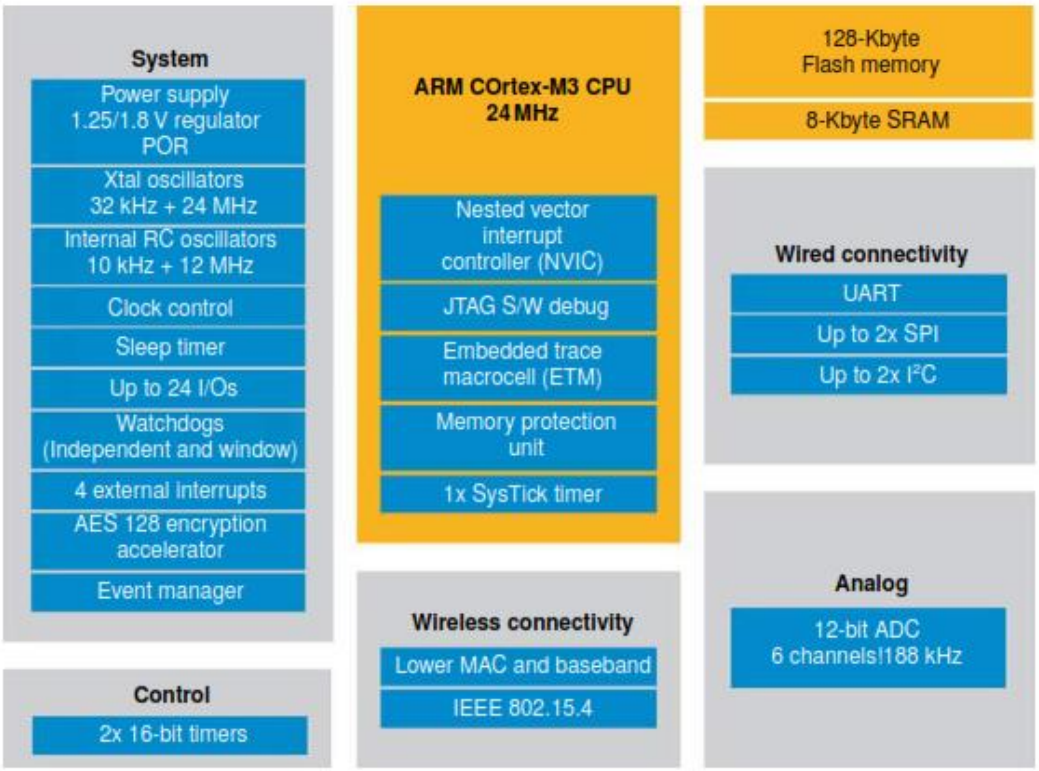

Figure 3.1 Example of a microcontroller with integrated STM32W-RFCKIT.

- The microcontroller typically 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.
- As low-power operation is paramount to battery-powered devices, the microcontroller hosts functions that facilitate sleeping, such as interrupts that can wake up the device on external and internal events, e.g. when there is activity on a GPIO port or the radio, as well as timer-based wake ups. Some devices even go as far as harvesting energy from their environment, e.g. in the form of solar, thermal, and physical energy.
- To interact with peripherals such as storage or display, it's common to use a serial interface such as SPI, IC, or UART. These interfaces can also be used to communicate with another microcontroller on the device. This is common when the there is a need

for offloading certain tasks, or when in some cases the entire application logic is put on a separate host processor.

- It's not unusual for the microcontroller to also contain a security processor, e.g. to accelerate Advanced Encryption Standard (AES). This is necessary to allow encrypted communication over the radio link without the need for a host processor.
- Because a basic device lacks a WAN interface according to our definition, a gateway of some form is necessary. The gateway together with the connected devices form a capillary network.
- The actual application logic is located on top of the OS or in the main loop. A typical task for the application logic is to read values from the sensors and to provide these over the LAN interface in a semantically correct manner with the correct units.
- For this class of devices, the constrained hardware and non-standard software limit third-party development and make development quite cost-intensive.

3.1.3 Gateways

- 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:
    - **Provisioning**: Initialization (or activation) of devices in regards to configuration and features to be enabled.
    - **Device Configuration**: Management of device settings and parameters.
    - **Software Upgrades**: Installation of firmware, system software, and applications on the device.
    - **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:
    - 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.
    - 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.
    - 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

3.1.4 Advanced devices

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.

It’s not unusual for the advanced device to also function as a gateway for local devices on the same LAN.

3.2 Local and wide area networking

3.2.1 The need for networking

- A network is created when two or more computing devices exchange data or information. The ability to exchange pieces of information using telecommunications technologies has changed the world, and will continue to do so for the foreseeable future, with applications emerging in nearly all contexts of contemporary and future living.
- Typically, devices are known as "nodes" of the network, and they communicate over "links."
- In modern computing, nodes range from personal computers, servers, and dedicated packet switching hardware, to smartphones, games consoles, television sets and, increasingly, heterogeneous devices that are generally characterized by limited resources and functionalities.
- Limitations typically include computation, energy, memory, communication (range, bandwidth, reliability, etc.) and application specificity (e.g. specific sensors, actuators, tasks), etc. Such devices are typically dedicated to specific tasks, such as sensing, monitoring, and control
- Network links rely upon a physical medium, such as electrical wires, air, and optical fibers, over which data can be sent from one network node to the next.
- It is not uncommon for these media to be grouped either as wired or wireless.
- A selected physical medium determines a number of technical and economic considerations.
- Technically, the medium selected, or more accurately, the technological solution designed and implemented to communicate over that medium, is the primary enabler of bandwidth without which, certain applications are infeasible.
- For example, consider the cost of embedding wires across a metropolitan, or larger, geographic region (e.g. electricity and legacy telephone networks).
- When direct communication between two nodes over a physical medium is not possible, networking can allow for these devices to communicate over a number of hops.
- In order to achieve this, nodes of the network must have an awareness of all nodes in the network with which they can indirectly communicate.
- This can be a direct connection over one link (edge, the transition or communication between two nodes over a link), or knowledge of a route to the desired (destination) node by communicating through cooperating nodes, over multiple edges.
- Consider Below Figure 3.2, This is the simplest form of network that requires knowledge of a route to communicate between nodes that do not have direct physical links. Therefore, if node A wishes to transfer data to node C, it must do so through node B. Thus, node B must be capable of the following: communicating with both node A and node C, and advertising to node A and node C that it can act as an intermediary.

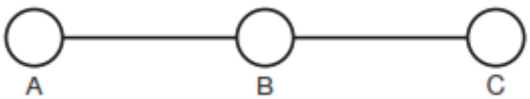

Figure 3.2 A Network

- Basic networking requirements have become explicit. It is essential to uniquely identify each node in the network, and it is necessary to have cooperating nodes capable of

linking nodes between which physical links do not exist. In modern computing, this equates to IP addresses and routing tables.

- Beyond the basic ability to transfer data, the speed and accuracy with which data can be transferred is of critical importance to the application. Irrespective of the ability to link devices, without the necessary bandwidth, some applications are rendered impossible.
- Consider the differences between streaming video from a surveillance camera, for example, and an intrusion-detection system based on a passive sensor.
- A Local Area Network (LAN) was traditionally distinguishable from a Wide Area Network (WAN) based on the geographic coverage requirements of the network, and the need for third party, or leased, communication infrastructure. In the case of the LAN, a smaller geographic region is covered, such as a commercial building, an office block, or a home, and does not require any leased communications infrastructure.
- WANs provide communication links that cover longer distances, such as across metropolitan, regional, or by textbook definition, global geographic areas. In practice, WANs are often used to link LANs and Metropolitan Area Networks (MAN) _ where LAN technologies cannot provide the communications ranges to otherwise interconnect _ and commonly to link LANs and devices (including smart phones, Wi-Fi routers that support LANs, tablets, and M2M devices) to the Internet. Quantitatively, LANs tended to cover distances of tens to hundreds of meters, whereas WAN links spanned tens to hundreds of kilometers.
- There are differences between the technologies that enable LANs and WANs. In the simplest case for each, these can be grouped as wired or wireless. The most popular wired LAN technology is Ethernet. Wi-Fi is the most prevalent wireless LAN (WLAN) technology.
- Wireless WAN (WWAN), as a descriptor, covers cellular mobile telecommunication networks, a significant departure from WLAN in terms of technology, coverage, network infrastructure, and architecture. The current generation of WWAN technology includes LTE (or 4G) and WiMAX.
- Considering M2M and IoT applications, there are likely to exist a combination of traditional networking approaches. The need exists to interconnect devices (generally integrated microsystems) with central data processing and decision support systems, in addition to one another. The business logic and requirements for each embodiment will differ on a case-by-case basis.
- The “Internet of Things,” as a term, originated from Radio Frequency Identification (RFID) research, wherein the original IoT concept was that any RFID-tagged “thing” could have a virtual presence on the “Internet.” In reality, there is little conceptual dissimilarity between RFID and bar codes, or more recently, QR codes _ they simply use different technological means to achieve the same result (i.e. an “object” has an online presence).

3.2.2 Wide area networking

- WANs are typically required to bridge the M2M Device Domain to the backhaul network, thus providing a proxy that allows information (data, commands, etc.) to traverse heterogeneous networks. This is seen as a core requirement to provide communications services between the M2M service enablement and the physical deployments of devices in the field. Thus, the WAN is capable of providing the bi-

directional communications links between services and devices. This, however, must be achieved by means of physical and logical proxy.

- The proxy is achieved using an M2M Gateway Device. Depending on the situation, there are, in general, a number of candidate technologies to select from. As before, the M2M Gateway Device is typically an integrated microsystem with multiple communications interfaces and computational capabilities. It is a critical component in the functional architecture, as it must be capable of handling all of the necessary interfacing to the M2M Service Capabilities and Management Functions.
- This device is now capable of acting as a physical proxy between the LR-WPAN, or M2M Device Domain, and the M2M Network Domain. The latest ETSI M2M Functional Architecture is illustrated in Figure 5.3.

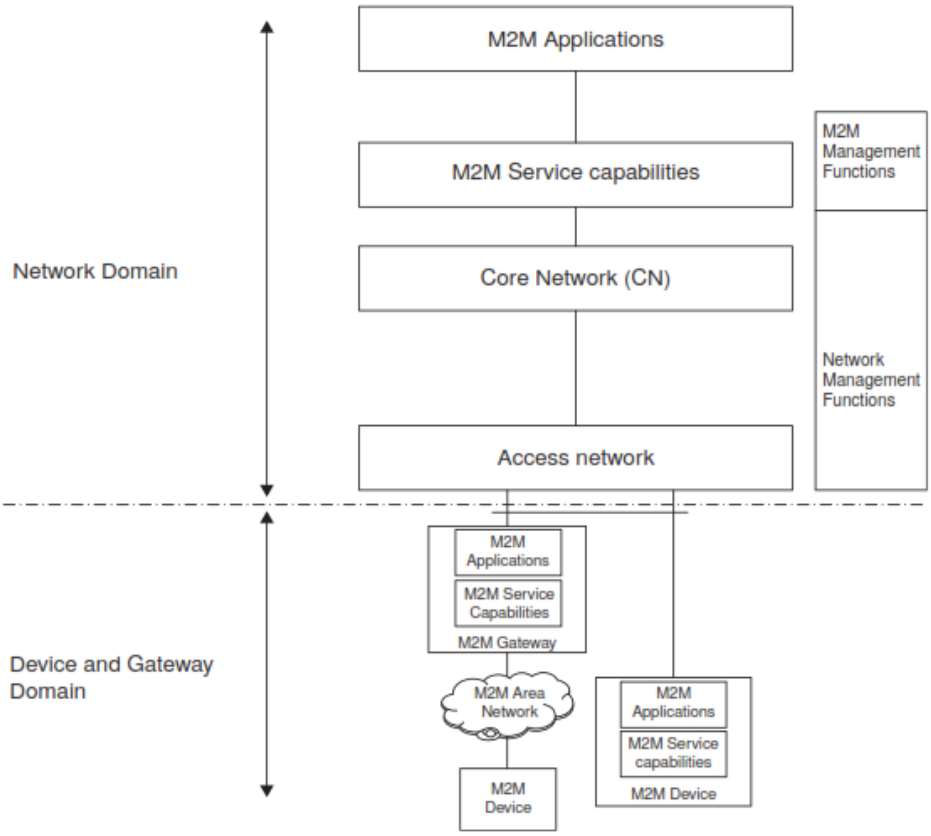

Figure 5.1 ETSI M2M Functional Architecture

- The Access and Core Network in the ETSI M2M Functional Architecture are foreseen to be operated by a Mobile Network Operator (MNO), and can be thought of simply as the "WAN" for the purposes of interconnecting devices and backhaul networks (Internet), thus, M2M Applications, Service Capabilities, Management Functions, and Network Management Functions.
- The WAN covers larger geographic regions using wireless (licensed and un-licensed spectra) as well as wire-based access. WAN technologies include cellular networks (using several generations of technologies), DSL, WiMAX, Wi-Fi, Ethernet, Satellite, and so forth.
- The WAN delivers a packet-based service using IP as default. However, circuit-based services can also be used in certain situations.

In the M2M context, important functions of the WAN include:

- The main function of the WAN is to establish connectivity between capillary networks, hosting sensors, and actuators, and the M2M service enablement. The default connectivity mode is packet-based using the IP family of technologies.
  Many different types of messages can be sent and received. These include messages originating as, for example, a message sent from a sensor in an M2M Area Network and resulting in an SMS received from the M2M Gateway or Application (e.g. by a relevant stakeholder with SMS notifications configure ed for when sensor readings breach particular sensing thresholds.).
- Use of identity management techniques (primarily of M2M devices) in cellular and non-cellular domains to grant right-of-use of the WAN resource.
- The following techniques are used for these purposes:
  - MCIM (Machine Communications Identity Module) for remote provisioning of SIM targeting M2M devices.
  - xSIM (x-Subscription Identity Module), like SIM, USIM, ISIM.
  - Interface identifiers, an example of which is the MAC address of the device, typically stored in hardware.
  - Authentication/registration type of functions (device focused).
  - Authentication, Authorization, and Accounting (AAA), such as RADIUS services.
  - Dynamic Host Configuration Protocol (DHCP), e.g. employing deployment-specific configuration parameters specified by device, user, or application-specific parameters residing in a directory.
  - Subscription services (device-focused).
  - Directory services, e.g. containing user profiles and various device (s) parameter(s), setting(s), and combinations thereof. M2M-specific considerations include, in particular:
  - MCIM (cf. 3GPP SA3 work).
  - User Data Management (e.g. subscription management).
  - Network optimizations (cf. 3GPP SA2 work).

- There may be many suppliers of WAN functionality in a complete M2M solution. It follows that an important function in the M2M Service Enablement domain will be to manage westbound business-to-business (B2B) relations between a number of WAN service providers.

3.2.2.1 3rd generation partnership project technologies and machine type communications

Machine Type Communications (MTC) is heavily referred to in the ETSI documentation. MTC, however, lacks a firm definition, and is explained using a series of use cases. Generally speaking, MTC refers to small amounts of data that are communicated between machines (devices to back-end services and vice versa) without the need for any human intervention. In the 3rd Generation Partnership Project (3GPP), MTC is used to refer to all M2M communication (Jain et al. 2012). Thus, they are interchangeable terms.

3.2.3 Local area networking

- Capillary networks are typically autonomous, self-contained systems of M2M devices that may be connected to the cloud via an appropriate Gateway.

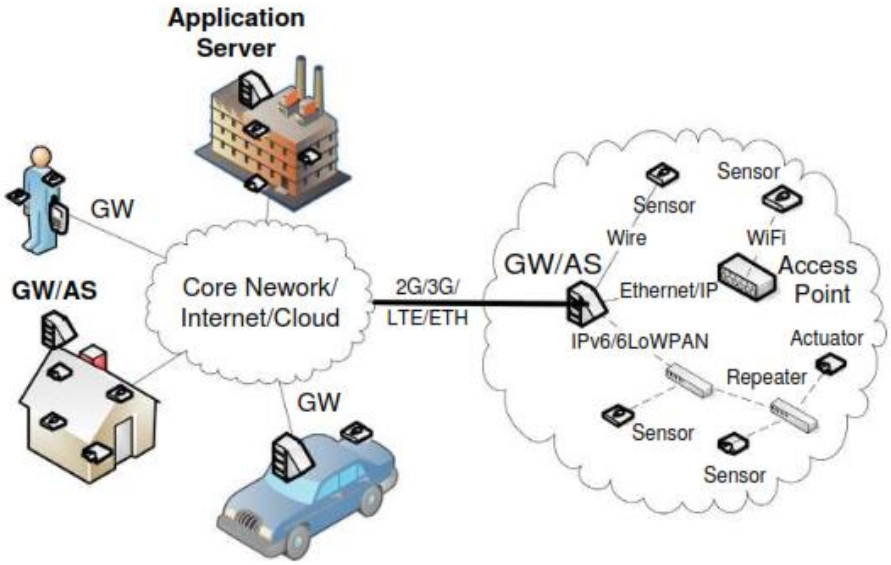

Figure 5.2 Capillary networks and their inside view

- They are often deployed in controlled environments such as vehicles, buildings, apartments, factories, bodies, etc. (Figure 5.4) in order to collect sensor measurements, generate events should sensing thresholds be breached, and sometimes control specific features of interest (e.g. heart rate of a patient, environmental data on a factory floor, car speed, air conditioning appliances, etc.). There will exist numerous capillary networks that will employ short-range wired and wireless communication and networking technologies.
- For certain application areas, there is a need for autonomous local operation of the capillary network. That is, not everything needs to be sent to, or potentially be controlled via, the cloud.
- In the event that application-level logic is enforceable via the cloud, some will still need to be managed locally. The complexity of the local application logic varies by application. For example, a building automation network may need local control loop functionality for autonomous operation, but can rely on external communication for configuration of control schemas and parameters.
- The M2M devices in a capillary network are typically thought to be low-capability nodes (e.g. battery operated, with limited security capabilities) for cost reasons, and should operate autonomously. For this reason, a GW/application server will naturally also be part of the architected solution for capillary networks.
- More and more (currently closed) capillary networks will open up for integration with the enterprise back end systems. For capillary networks that expose devices to the cloud/Internet, IP is envisioned to be the common waist. IPv6 will be the protocol of choice for M2M devices that operate a 6LoWPAN-based stack. IPv4 will still be used for capillary networks operating in non-6LoWPAN IP stacks (e.g. Wi-Fi capillary networks).

3.2.3.1 Deployment considerations

- The nature of the intended application plays a significant role in determining the appropriate technological solution. Typically, these are defined by the business logics that motivate initial deployment. There are increasing numbers of innovative IoT applications (hardware and software) marketed as consumer products. These range from intelligent thermostats for effectively managing comfort and energy use in the home, to precision gardening tools (sampling weather conditions, soil moisture, etc.). At scale, similar solutions are, and will continue to be, applied in and across industry.
- Scaling up for industrial applications and moving from laboratories into the real world creates significant challenges that are not yet fully under- stood. Low-rate, low-power communications technologies are known to be "lossy." The reasons for this are numerous. They can relate to environmental factors, which impact upon radio performance (such as time varying stochastic wireless propagation characteristics), technical factors such as performance trade-offs based on the characteristics of medium access control and routing protocols, and physical limitations of devices and practical factors such as maintenance opportunities (scheduled, remote, accessibility, etc.)

3.2.3.2 Key technologies

- This section details a number of the standards and technologies currently in use and under development that enable ad hoc connectivity between the devices that will form the basis of the IoT.
- These are the communications technologies that are considered to be critical to the realization of massively distributed M2M applications and the IoT at large.
    - *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.
    - *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.
    - *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.

- *Low-Rate, Low-Power Networks*
  - are another key technology that form the basis of the IoT.

- *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.
- *6LoWPAN*
  - (IPv6 Over Low Power Wireless Personal Area Networks) was developed initially by the 6LoWPAN 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

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

3.3 Data Management

3.3.1 Introduction

- Modern enterprises need to be agile and dynamically support multiple decision-making processes taken at several levels. In order to achieve this, critical information needs to be available at the right point in a timely manner, and in the right form.
- Some of the key characteristics of M2M data include.
  - **Big Data:** Huge amounts of data are generated, capturing detailed aspects of the processes where devices are involved.
  - **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.
  - **Real-World Data:** The overwhelming majority of the M2M data relates to real-world processes and is dependent on the environment they interact with.
  - **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.
- **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.
- **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.
- **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.
- **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.

3.3.2 Managing M2M data

- 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.

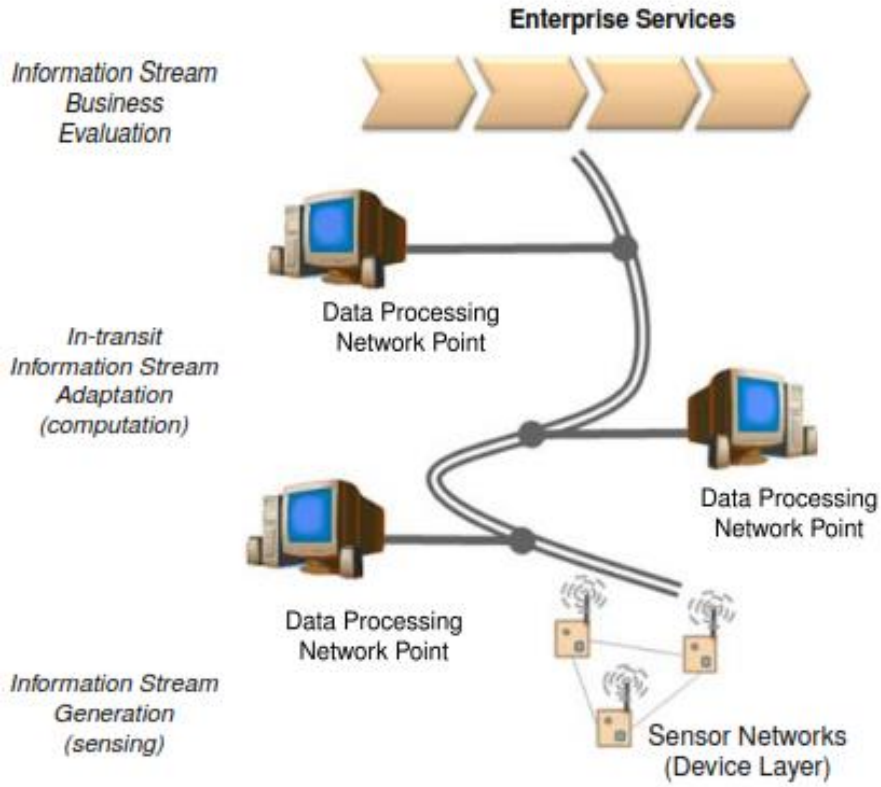

Figure 3.5 M2M data from point of generation to business assessment.

3.2.3.1 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.

3.2.3.2 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 over- whelming 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.

3.2.3.2 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.

3.2.3.5 Data processing

Data processing enables working with the data that is either at rest (already stored) or is 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, re-order 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.

3.2.3.6 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.

3.2.3.7 Data Analysis

Data available in the repositories can be subjected to analysis with the aim to obtain the information they encapsulate and use it for supporting decision-making processes. The analysis of data at this stage heavily depends on the domain and the context of the data. For instance, business intelligence tools process the data with a focus on the aggregation and key performance indicator assessment. Data mining focuses on discovering knowledge, usually in conjunction with predictive goals. Statistics can also be used on the data to assess them quantitatively (descriptive statistics), find their main characteristics (exploratory data analysis), confirm a specific hypothesis (confirmatory data analysis), discover knowledge (data mining), and for machine learning, etc. This stage is the basis for any sophisticated applications that take advantage of the information hidden directly or indirectly on the data, and can be used, for example, for business insights, etc. M2M has the potential to revolutionize modern businesses.

3.3.3 Considerations for M2M data

However, the real paradigm change the M2M data will enforce depends on a single aspect: data sharing. Although there are benefits acquired by processing the M2M data at local loops, their real benefit is brought into the foreground when these are shared at large scale. The latter can act as an enabler to better understand complex systems of systems, and better manage them. The Cooperating Objects vision, which assumes cooperation among devices and systems as the key driving force for interaction, sheds some light on the benefits and challenges that will emerge in all layers of such an M2M infrastructure. As an indicative example, the smart city can be used where huge data from its infrastructure, citizens, businesses, and individual assets need to be considered, analyzed, and after decisions are taken, enforced.

The M2M infrastructure in place heavily depends on real-world processes, implying also that a big percentage of data will be generated by machines that interact with the real-world environment, while the rest will be purely virtual data. For the first part, where machines are involved, there is a real cost for the infrastructure that has to be met. Hence, it is expected that stakeholders in the future will further diversify, and we will see the emergence of infrastructure providers who will operate and manage many of the machines generating this data, which can then be communicated to others (e.g. analytics specialists to take advantage of the insights offered)

The end-beneficiaries might acquire information, but do not necessarily need to have access or to process the data by themselves. Hence, as we see, there is a rise of specialists in the various stages of M2M data management that will cooperate with application providers, users, etc. for the common benefit. Such ecosystems are expected to be of key importance in the future IoT era. This transition is already at an early stage, and boldly contradicts the existing initial M2M efforts, where the application developer, the data collector, and the infrastructure operator roles are largely performed by the same stakeholder (or a very small number of them).

Because of expected wide sharing of data and usage in multiple applications, security and trust are of key importance. Security is mandatory for enabling confidentiality, integrity, availability, authenticity, and non- repudiation of data from the moment of generation to consumption. Due to the large-scale IoT infrastructure, heterogeneous devices, and stakeholders involved, this will be challenging. In addition, trust will be another major

3.3.4 Conclusions

Data and its management hold the key to unveiling the true power of M2M and IoT. To do so, however, we have to think and develop approaches that go beyond simple data collection, and enable the management of their whole lifecycle at very large scale, while in parallel considering the special needs and the usage requirements posed by specific domains or applications. Mastering the challenges of data management will enable data analysis to flourish, and this in turn will empower new innovative approaches to be realized for the benefit of citizens, business, and society.

3.4 Business processes in IoT

3.4.1 Introduction

- A business process refers to a series of activities, often a collection of interrelated processes in a logical sequence, within an enterprise, leading to a specific result.
- There are several types of business processes such as ***management***, ***operational***, and ***supporting***, all of which aim at achieving a specific mission objective.
- As business processes usually span several systems and may get very complex, several methods and techniques have been developed for their modeling, such as the Business Process Model and Notation (BPMN), which graphically represents business processes in a business process model.
- Several key business processes in modern enterprise systems heavily rely on interaction with real-world processes, largely for monitoring, but also for some control, in order to take business-critical decisions and optimize actions across the enterprise. The introduction of modern ICT has significantly changed the way enterprises interact with the real world.
- *In Figure 3.6*, we have witnessed a paradigm change with the dramatic reduction of the data acquisition from the real world this was attributed mostly to the automation offered by machines embedded in the real world.
- Initially all these interactions were human-based (e.g. via a keyboard) or human-assisted (e.g. via a barcode scanner); however, with the prevalence of RFID, WSNs, and advanced networked embedded devices, all information exchange between the real-world and enterprise systems can be done automatically without any human intervention and at blazing speeds.
- In the M2M era, connected devices can be clearly identified, and with the help of services, this integration leads to active participation of the devices to the business processes. This direct integration is changing the way business processes are modeled and executed today as new requirements come into play.
- Existing modeling tools are hardly designed to specify aspects of the real world in modeling environments and capture their full characteristics.

- The industrial adoption of IoT (e.g. of wireless sensor networks) is hampered by the lack of integration of WSNs with business process modeling languages and back-end systems.
- There are, however, promising approaches such as the one provided by makeSense (Tranquillini et al. 2012), which tackles this problem space with a unified programming framework and a compilation chain that, from high-level business process specifications, generates code ready for deployment on WSN nodes. A layered approach for developing, deploying, and managing WSN applications that natively interact with enterprise information systems such as a business process engine and the processes running therein is proposed and assessed.M2M and IoT empower business processes to acquire very detailed data about the operations, and be informed about the conditions in the real world in a very timely manner. Subsequently, better business intelligence (Spiess & Karnouskos 2007) and more informed decision-making can be realized. The latter enables businesses to operate more efficiently, which translates to a business competitive advantage.

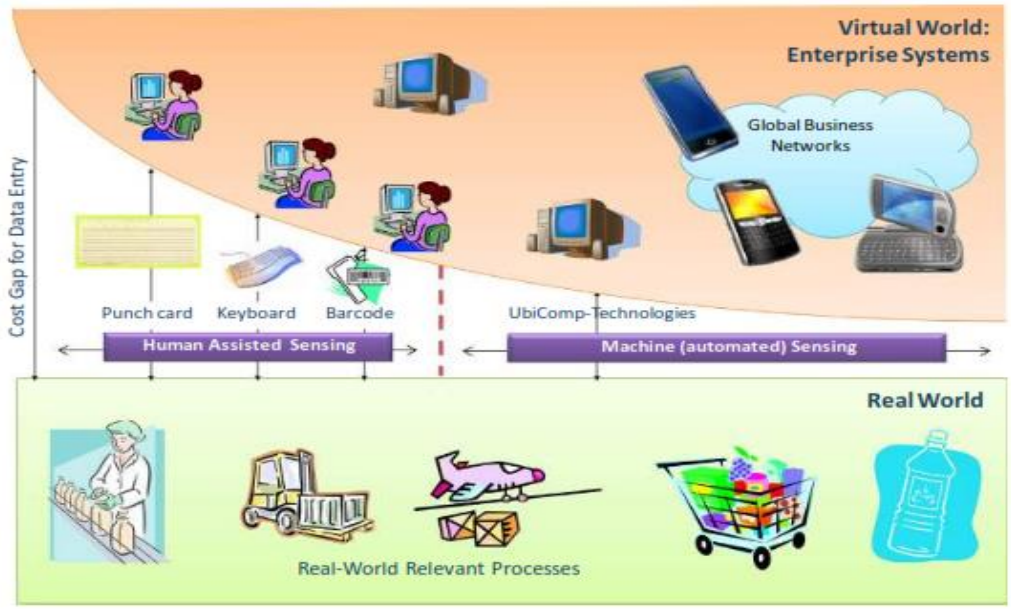

Figure No 3.6: The decreasing cost of information exchange between the real-world and enterprise systems with the advancement of M2M

3.4.2 IoT integration with enterprise systems

- M2M communication and the vision of the IoT pose a new era where billions of devices will need to interact with each other and exchange information in order to fulfill their purpose. Much of this communication is expected to happen over Internet technologies and tap into the extensive experience acquired with architectures and experiences in the Internet/Web over the last several decades. More sophisticated, though still overwhelmingly experimental, approaches go beyond simple integration and target more complex interactions where collaboration of devices and systems is taking place.
- As shown in Figure 3.7, cross-layer interaction and cooperation can be pursued:
    - at the M2M level, where the machines cooperate with each other (machine-focused interactions), as well as

- at the machine-to-business (M2B) layer, where machines cooperate also with network-based services, business systems (business service focus), and applications.

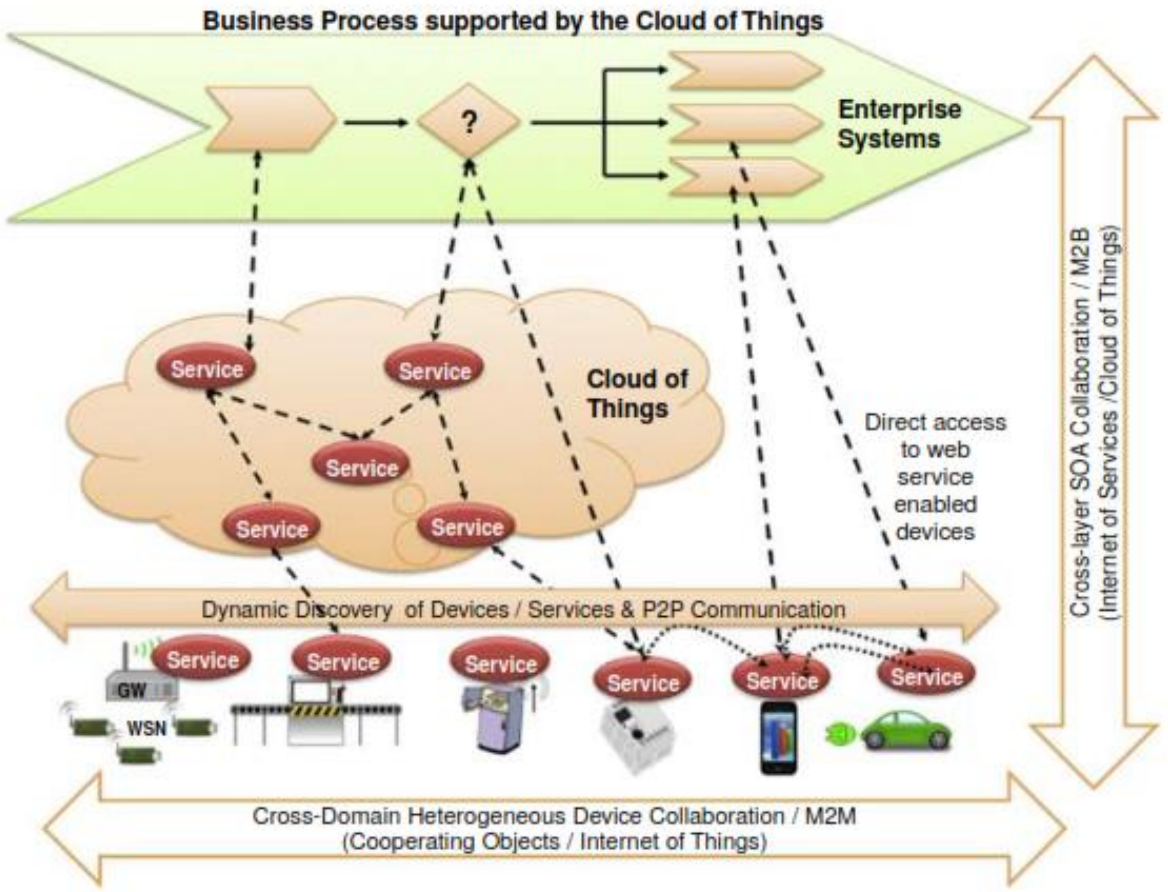

Figure No 3.7 A collaborative infrastructure driven by M2M and M2B

- As depicted in *Figure 3.7*, we can see several devices in the lowest layer. These can communicate with each other over short-range protocols (e.g. over ZigBee, Bluetooth), or even longer distances (e.g. over Wi-Fi, etc.).
- Some of them may host services (e.g. REST services), and even have dynamic discovery capabilities based on the communication protocol or other capabilities (e.g. WS-Eventing in DPWS). Some of them may be very resource constrained, which means that auxiliary gateways could provide additional support such as mediation of communication, protocol translation, etc. Independent of whether the devices are able to discover and inter- act with other devices and systems directly or via the support of the infrastructure, the M2M interactions enable them to empower several applications and interact with each other in order to fulfill their goals.
- Promising real-world integration is done using a service-oriented approach by interacting directly with the respective physical elements, for example, via web services running on devices or via more lightweight approaches such as REST. In the case of legacy systems, gateways and service mediators are in place to enable such integration challenges.
- Many of the services that will interact with the devices are expected to network services available, for example, in the cloud. The main motivation for enterprise services is to

take advantage of the cloud characteristics such as *virtualization*, *scalability*, *multi-tenancy*, *performance*, *lifecycle management*, etc.

- we are moving towards an infrastructure where the cloud and its services (as depicted in *Figure 3.8*) take a prominent position towards empowering modern enterprises and their business processes.
- A key motivator is the minimization of communication overhead with multiple endpoints by, for example, transmission of data to a single or limited number of points in the network, and letting the cloud do the load balancing and further mediation of communication.
- To this end, the data acquired by the device can be offered without overconsumption of the device's resources, while in parallel, better control and management can be applied.
- Typical examples include enabling access to the full historical data, preprocessing of information, transparently upgrading the cloud services, or even not providing access to internal systems for security reasons. This clear decoupling of "things" and the usage of their data is expected to further empower information-driven business processes and applications that can operate over federated infrastructures.

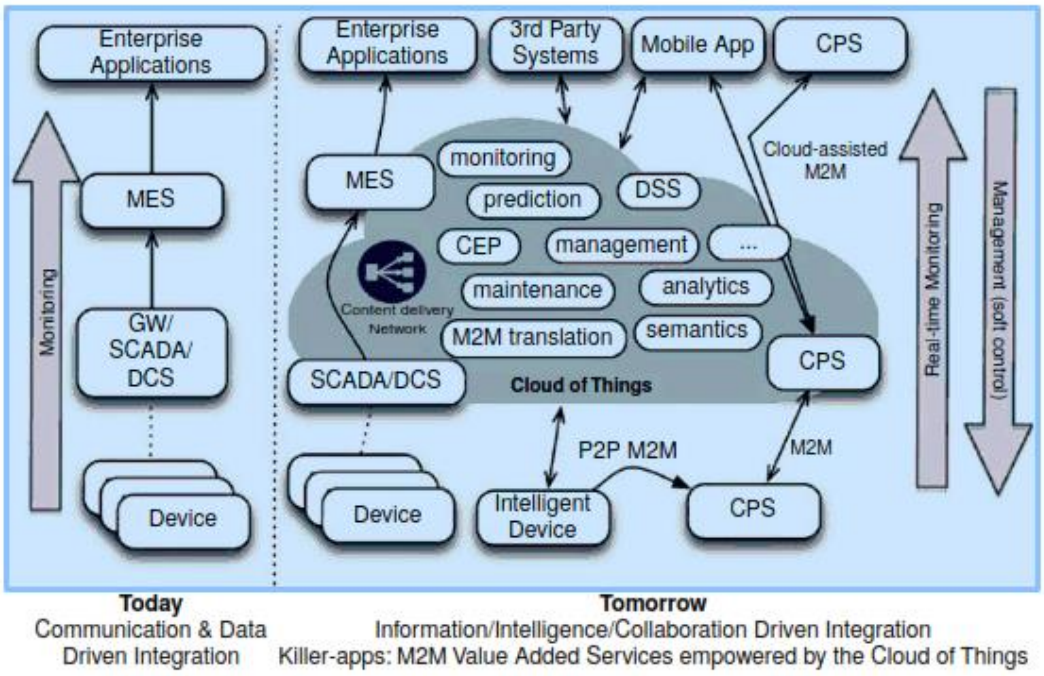

Figure No 3. 8 The Cloud of Things as an enabler for new value added services & apps.

3.4.3 Distributed business processes in IoT

- Today, as seen on the left part of *Figure 3.9*, 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.

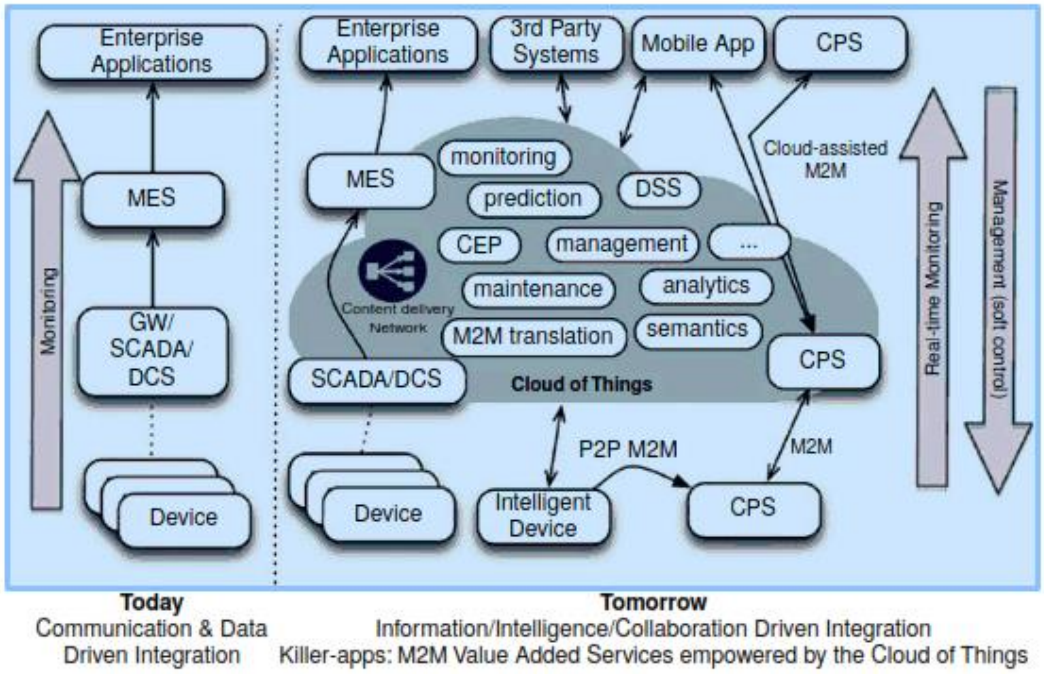

Figure No. 3.9 Distributed Business Processes in M2M era

- 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 3.9*.
- 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 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 3.10*. 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.

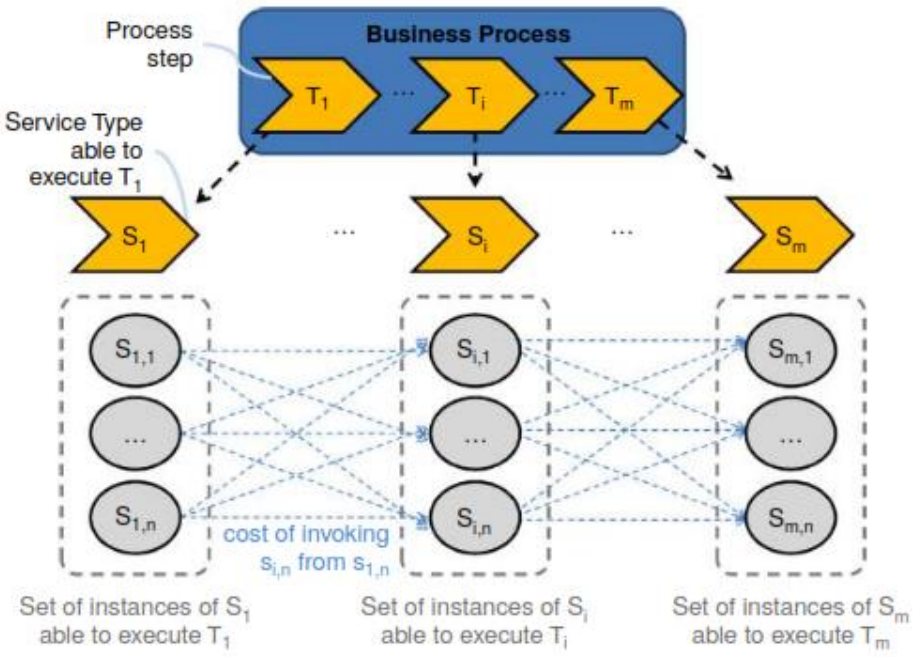

Figure No. 3.10 On-Device and in-network Business Process Composition and runtime execution.

3.4.4 Considerations

- Existing tools and approaches need to be extended to the make the business processes IoT aware. The current terminology in modeling tools is focused on the enterprise context and does not include the notation of physical entities such as devices as these are considered in IoT.
- The current terminology in modeling tools is focused on the enterprise context and does not include the notation of physical entities such as devices as these are considered in IoT.
- Although distributed execution of processes exists (e.g. in BPMN), additional work is needed to be able to select the devices in which such processes execute and consider their characteristics or dynamic resources, etc. The dynamic aspect is of key importance in the IoT, as this is mobile and availability is not guaranteed, which means that availability in modeling time does not guarantee availability at runtime and vice-versa.
- Even if the latter holds true, this might again change during the execution of a business process; hence, fault tolerance needs to be considered.
- IoT infrastructures are expected to be of large scale. Hence, scalability is an aspect that needs to be considered in the business process modeling and execution. In addition, event-based interactions among the processes play a key role in IoT, as a business process flow may be influenced by an event, or as its result, trigger a new event.

3.4.5 Conclusions

- Modern enterprises operate on a global scale and depend on complex business processes. Business continuity needs to be guaranteed, and therefore efficient

information acquisition, evaluation, and interaction with the real world are of key importance. The infrastructure envisioned is a heterogeneous one, where millions of devices are interconnected, ready to receive instructions and create event notifications, and where the most advanced ones depict self-behavior (e.g. self-management, self-healing, self-optimization, etc.) and collaborate. This can lead to a paradigm change as business logic can now be intelligently distributed to several layers such as the network, or even the device layer, creating new opportunities, but also challenges that need to be assessed.

- Future Enterprise systems will be in position to better integrate state and events of the physical world in a timely manner, and hence to lead to more diverse, highly dynamic, and efficient business applications.

3.5 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 3.11*). Several essential characteristics of cloud computing have been defined by NIST (2011) as follows:
    - **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.
    - **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.
    - **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 (NIST 2011):

- **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.
- **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 (NIST 2011).

- **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.

- **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).

3.6 M2M and IoT Analytics

3.6.1 Introduction

- Traditionally, M2M data has been sent from specific devices to specific services, which store the data of interest. This approach uses semantically well-defined data for specific purposes, and only requires storing the data that is needed for the explicit use cases, and only for as long as it's required. For the most part, the applications have been monitoring, reporting, and rule-based actions.
- To further increase the speed of M2M deployments, it's important to look at methods to extract additional value from these devices. Given the enormous amounts of data that will be generated by the IoT and the advancements within the area of Big Data, new opportunities arise from the possibility to reuse data from devices for multiple purposes, many of which will not even be imagined at the time of deployment. The opportunities of using M2M data for advanced analytics and business intelligence are very promising. By transforming raw data into actionable intelligence, it's possible to improve many areas, such as enhancement of existing products, cost-savings, service quality, as well as operational efficiency.
- By applying technologies from the Big Data domain, it is possible to store more data, such as contextual and situational information, and given a more open approach to data, such as the open-data government initiatives (e.g. Data.gov and Data.gov.uk), even more understanding can be derived, which can be used to improve everything from Demand/Response in a power grid to wastewater treatment in a city. This requires a migration from the data silos of today to an architecture where it's possible to cross-analyze data residing in many different locations, which implies that the location of the stored data will not necessarily be the same as the location where the analytics will take place. From a practical standpoint, this is a problem that needs to be handled when we are talking about extreme amounts of data.
- Descriptive statistics can take you a long way from raw data to actionable intelligence. Other opportunities are provided by data mining and machine learning, with no clear distinction between the three, although data mining can be described as the automatic or semiautomatic task of extracting previously unknown information from a large quantity of data, while machine learning is focused on finding models for specific tasks, e.g. separate spam from non-spam email.
- For M2M data, traditional data warehousing and analytics will for many cases not be up to the task. Big Data technologies such as MapReduce for massively parallel analytics, as well as analytics on online streaming data where the individual data item is not necessarily stored, will play an important role in the management and analysis of large-scale M2M data.

- To handle the analytical needs related to M2M and IoT, it's expected that in the near term, vendors of Big Data solutions will provide for the needs of in-house analytics. In the long term, new niches are likely to appear, such as cloud storage providers, data brokers, and Analytics-as-aService providers. Apart from the software and services provided for analytics, a major uptake in professional services for consultancy within M2M analytics is expected *(Figure 3.12)*
- The revenues within M2M analytics are expected to grow rapidly for most industry verticals (*Figure 3.13*).

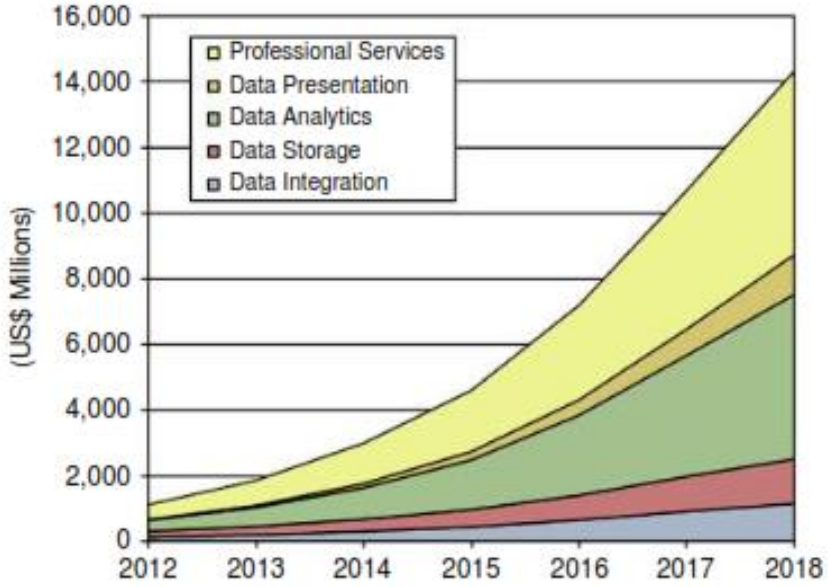

Figure No. 3.12 M2M Analytics Revenues by Segment, World Market, Forecast: 2012 to 2018

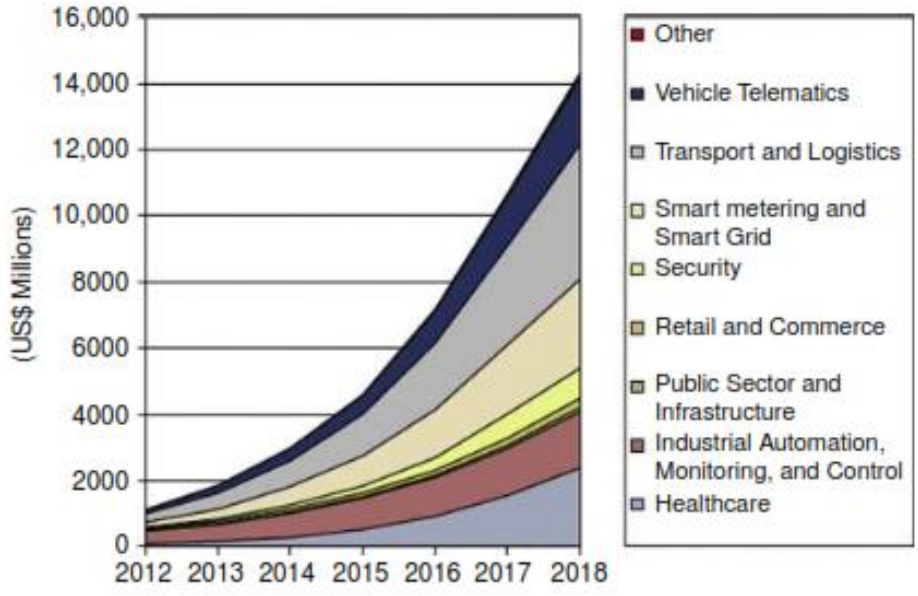

Figure No 3.13 M2M Analytics Revenues by Industry, Vertical World Market, Forecast: 2012 to 2018.

3.6.2 Purposes and Considerations

Regardless of whether you call it statistics, data mining, or machine learning, there exist a multitude of methods to extract different types of information from data. The information can be used in everything from static reports to interactive decision support systems, or even fully automated real-time systems.
Some examples of methods and purposes are as follows:

- **Descriptive Analytics:** Use of means, variances, maxima, minima, aggregates, and frequencies, optionally grouped by selected characteristics.
    - Create Key Performance Indicators (KPI's) that enable better understanding of the performance of complex systems such as cellular networks or oil pipelines.
- **Predictive Analytics:** Use current and historical facts to predict what will happen next.
    - Forecast demand and supply in a power grid and train a model to predict how price affects electric usage to optimize the performance and minimize peaks in the electricity consumption.
    - Predictive maintenance on electromechanical equipment in a nuclear power plant by modeling the relationship between device health characteristics measured by sensors and historic failures.
    - Understand how electricity and water consumption relates to regional demographics.
    - Model the effects of traffic lights on a city's road network based on data from cars and sensors in the city to minimize congestion.
- **Clustering:** Identification of groups with similar characteristics.
    - Perform customer segmentation or find behavioral patterns in a large set of M2M devices.
    - Mine time series data for recurring patterns that can be used in predictive analytics to detect, for example, fraud, machine failures, or traffic accidents.
- **Anomaly Detection:**
    - Detect fraud for smart meters by checking for anomalous electricity consumption compared to similar customers, or historic consumption for the subscriber.

These can be divided into two categories, enterprise specific data and public data, of which the former can be efficiently accessed using common formatting, processing, and storage. The latter is most often accessed using public APIs, and has commonalities to a much lesser degree.

M2M data fulfills all the characteristics of Big Data, which is usually described by the four "Vs":

- **Volume**: To be able to create good analytical models it's no longer enough to analyze the data once and then discard it. Creating a valid model often requires a longer period of historic data. This means that the amount of historic data for M2M devices is expected to grow rapidly.
- **Velocity:** Even though M2M devices usually report quite seldom, the sheer number of devices means that the systems will have to handle a huge number of transactions per second. Also, often the value of M2M data is strongly related to how fresh it is to be able provide the best actionable intelligence, which puts requirements on the analytical platform.
- **Variation:** Given the multitude of device types used in M2M, it's apparent that the variation will be very high. This is further complicated by the use of different data formats as well as different configurations for devices of the same type (e.g. where one device measures temperature in Celsius every minute, another device measures it in

Fahrenheit every hour). The upside is that the data is expected to be semantically well-defined, which allows for simple transformation rules.

- **Veracity:** It's imperative that we can trust the data that is analyzed. There are many pitfalls along the way, such as erroneous timestamps, non-adherence to standards, proprietary formats with missing semantics, wrongly calibrated sensors, as well as missing data. This requires rules that can handle these cases, as well as fault-tolerant algorithms that, for example, can detect outliers (anomalies).

Last but not least are the consequences of the need for user privacy. This means that data will often be anonymized both in terms of removing user identities, as well as uniqueness of user data. This limits the possibilities of cross-referencing different data sources.

3.6.3 Analytics architecture

- 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:
    - Columnar databases such as HP Vertica, Actian ParAccel MPP, SAP Sybase IQ, and Infobright.
    - Massively Parallel Processing (MPP) architectures such as Pivotal Greenplum and Teradata Aster.
    - 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).

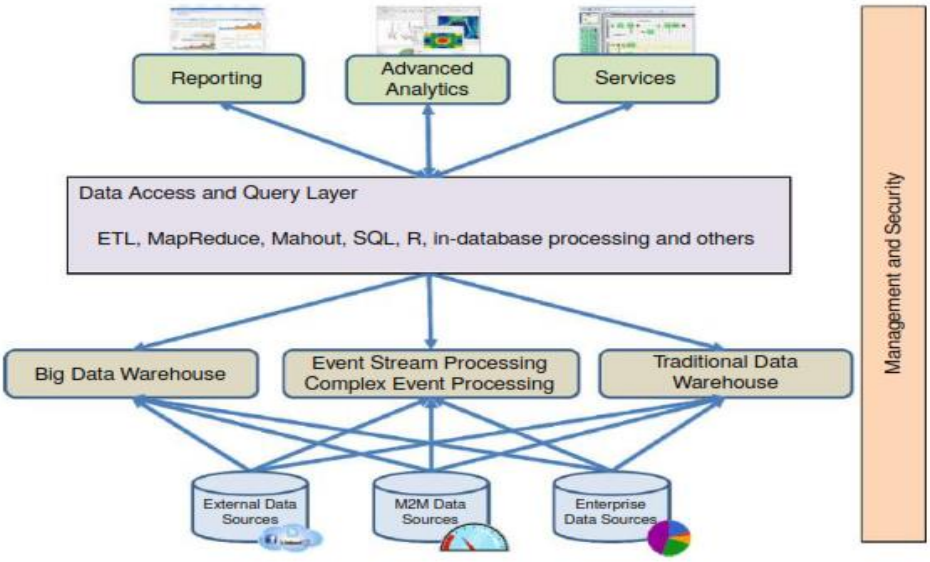

Figure 3.14 Analytics Architectural Overview.

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:
    - **HBase**: A column-oriented data store that provides real-time read/write access to very large tables distributed over HDFS.
    - Mahout: A distributed and scalable library of machine learning algorithms that can make use of MapReduce.
    - **Pig**: A tool for converting relational algebra scripts into MapReduce jobs that can read data from HDFS and HBase.
    - **Hive**: Similar to Pig, but offers an SQL-like scripting language called HiveQL instead.
    - **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.

3.6.4 Methodology

- Knowledge discovery and analytics can be described as a project methodology, following certain steps in a process model. To perform efficient analytics and find answers to important questions, it's paramount to involve the right people with the necessary business understanding at the beginning of a project.
- When the goals have been understood, the next step is to gather the necessary data and to understand it in terms of characteristics and quality. When this is done, it's possible to build the models that can answer the previously stated questions, although quite commonly the data needs to be transformed first. Before a model is deployed in the organization, it's also important to evaluate the performance of the model, i.e. How well does it fare in the real world?
- A model that has been developed can be used in a multitude of scenarios, such as an executive performing Enterprise Performance Management, information consumers using fixed reports, or an automated process deciding on individual transactions using a knowledge-driven business process.
- There exist several process models that include some or all parts of the steps mentioned above, such as the Knowledge Discovery in Databases (KDD) process, or the industrial standards Sample, Explore, Modify, Model and Assess (SEMMA), and Cross Industry Standard Process for Data Mining (CRISP-DM) (Table 5.2). The most commonly used process model of these is CRISP-DM (Kdnuggets 2007)

Table 3.2 Summary of the Correspondences between KDD, SEMMA, and CRISP-DM

KDD	SEMMA	CRISP-DM
Pre-KDD	—	Business understanding
Selection	Sample	Data understanding
Pre-processing	Explore	
Transformation	Modify	Data preparation
Data mining	Model	Modeling
Interpretation/Evaluation	Assessment	Evaluation
Post-KDD	—	Deployment

Source: Azevedo & Santos (2008).

- 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.

3.6.4.1 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.

3.6.4.1.1 Predictive Maintenance example

- It has been decided to start a project with the main objective to study ways to reduce the frequency of costly unplanned emergency repair work and downtime in pumping stations by predicting future pump failures or necessary maintenance work in good time to allow for organized maintenance to take place.
- The project will be determined a success if the study evaluation shows that:
  - Pump station downtime is reduced by 10%.
  - Maintenance costs for pump stations is reduced by 15%.
  - The project is concluded within time and kept on budget.
- These business requirements are translated into more practical data mining goals, such as evaluating two different approaches to predicting maintenance actions:
- **Action Forecasting:** Train a model that can predict needed maintenance actions based on vibrations and other pump characteristics. Apply forecasting methods on the vibration measurement sensors to predict future maintenance actions.
- **Similar Case Recommendations:** Use information about pumps, such as manufacturer, model, and age, as well as information about working conditions, such as workload, water corrosiveness, and percentage of sand and grit, to define groups of similar pumps. Use data from prior pump failures of similar pumps, as well as prior maintenance decisions, to recommend actions to take.

3.6.4.2 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.

3.6.4.2.1 Predictive Maintenance example

In this example, there are several data sources that could be of interest, including:

- Trouble Reports from the Ticket System: These will allow us to correlate prior failures with historic data from the pumps.
- Work Orders: Information from these enable us to understand what actions were taken in regards to failing/failed pumps.
- Pump Information: Descriptive data such as water source, manufacturer, model, and age of pump and it's major parts, as well as information about working conditions (e.g. water corrosiveness, percentage of sand and grit).
- Sensor Data: Historic measurements of pump revolutions per minute (RPM) and vibration measurements.

The different data sets are then explored and described accordingly:

- An inspection of the data shows that there are several million records of sensor data, spanning from approximately a two-year time period.
- Most of the data, apart from the sensor data, is symbolic, e.g. manufacturer and equipment identifiers. Some fields are numerical, e.g. percentage of sand, equipment age, and vibrations. There is also some unstructured data in the form of free text descriptions in the trouble reports and work orders.
- The pump information sometimes contains a pump identifier in the form of a numeric key, and sometimes using a textual name. The same problem is observed for the water source identifier. These are key fields that are used for mapping the different data sources with each other, and some work to fix them will be needed.
- In some cases there are missing data, both from sensor data (e.g. when a station has been services) and from the contextual data, e.g. missing pump information.
- It is noted that the measurements from the sensors indicates that the sensors in some cases have not been calibrated correctly.

3.6.4.3 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.

3.6.4.3.1 Predictive Maintenance example

Several operations are performed to prepare the data, and three data sets are constructed:

- **Vibration Time Series**
    - Time series data for the pump vibrations are at the core of the analytical work. Missing values are estimated and imputed to create complete time series representations. The measurement values are adjusted to account for incorrectly calibrated sensors.

- **Workload Time Series**

- Pump RPM measurements are used to construct a new time series with attributes that describe the pump workload at a given date, e.g. average daily workload, standard deviation of the workload, maximal daily workload, and workload trend.

- **Pump Records**
  - Information needed for grouping similar pumps is included and joined with the newly created workload data.
- **Action Records**
  - Trouble reports and work orders are joined to create one action record for each maintenance task. Some of the data is excluded and transformed to create new attributes that indicate what kind of action was performed, e.g. bearing replacement, oil lubricate on, or motor replacement.
  - The action records are merged with the pump records, as they were at the time the action was performed.

3.6.4.3 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.

3.6.4.4.1 Predictive Maintenance example

With the business goal of finding models that help to reduce downtime of pump stations and maintenance costs, as well as avoiding unnecessary maintenance work, we form the hypothesis that pumps with similar characteristics will follow a similar pattern in terms of maintenance needs, and that it's possible to reuse knowledge from prior cases to make decisions. To make use of this hypothesis and test it, we create three models.

- **Action Prediction Model:** The action records are used to create a classification model that can predict what actions to take given a certain set of input data, e.g. the pump is vibrating strongly, the water is corrosive, it has been 14 months since it was serviced, and given prior cases, replacing the bearing and lubricating the pump with oil are likely to be the best actions. A decision tree-based model is selected since the data is highly heterogeneous and contains many categorical values. An assessment shows that the best performing model is based on the Random Forests method.
- **Forecasting Model:** To be able to predict future failures and needed maintenance in advance, a forecasting model is applied to the historic vibration sensor measurements. Two models, one based on the ARIMA (Autoregressive Integrated Moving Average) method, and the other on the ETS (Error_Trend_Seasonal) method performs well, and after assessment, the ETS-based model is selected.

- **Similar Pump Model:** To create a model that can be used to determine similarity between pumps, there exists a number of similarity and clustering techniques, such as k-nearest-neighbor and k-means. After some reasoning, it is decided to use a k-means-based model. These models require the number of clusters to be set as a parameter, and to determine the most appropriate number of clusters a decision tree is trained to classify which cluster a pump should belong to. The benefit of this is that trained pump maintenance experts are able to inspect the decision trees to determine which number of clusters produces the most realistic decision tree.

3.6.4.4 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.

3.6.4.5.1 Predictive Maintenance example

- To evaluate how well the models perform in the real world, a set of example cases are selected and the results are studied and verified by maintenance staff with several years of experience from working in the field. Several variations of the models, with slightly different parameter settings, are evaluated and studied. Especially the action prediction model is analyzed to find which version recommends the most correct actions compared to what the experts would recommend.
- The two different approaches are evaluated:
    - **Action Forecasting:** This approach proved to provide stable results that are easily interpreted by both humans and machines. A discussion was undertaken as to whether this could be used to automatically create work orders if the forecast was within certain bounds of confidence.
    - **Similar Pump Recommendations:** This approach was much appreciated since it provided the staff with empirical data about how pumps under similar conditions have evolved, i.e. failed or been subjected to early maintenance. A decision was made to deploy both approaches and combine them in one report.

3.6.4.6 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.

3.6.4.6.1 Predictive Maintenance example

- Data from the pump stations is read automatically every day. A new batch job is deployed that is triggered when all readings have been collected. The batch job performs all the necessary data transformations and data loading needed before applying the models.
- A new routine was implemented that generates a 30-day forecast for all pumps, and then evaluates the action prediction model on each pump and its forecasted data.
- For those pumps that have actions predicted, a rule set is evaluated that checks the results against given thresholds to check for, for example, confidence of predicted actions. If these checks are positive, similar cases are retrieved and a report is generated and sent to the right people.
- The case retrieval method is implemented by first looking up the pumps that belong to the same cluster according to the similar pump model. For each similar pump, a scan is executed to find matches in time when the vibration characteristics were the same as the pump currently being evaluated. For each match, a lookup in the action records is performed to check what kind of actions and failure have occurred within 30 days after the date of the match. The matches that have logged actions are then added to the report.
- Another batch job is also implemented, with the tasks of updating the pump and action records, as well as retraining the models periodically. This job is also responsible for evaluating the efficiency and correctness of the models as the system evolves over time. If the models seem to deteriorate, an administrator is notified.
- Finally, it's decided to inspect the system with the help of experts once a year to evaluate its performance.

3.7 Knowledge Management

- Covered analytics in the context of M2M. Here, we investigate Knowledge Management Frameworks. Firstly, we must look at the concept of knowledge, which in every day usage relates to information, understanding, or skill you get from experience or education. Within the context of ICT systems, the term "knowledge" mostly arises from the application of two other concepts: data and information, illustrated in *Figure 5.16*. We discuss here the relationships between these terms, and in the next section we discuss reference architecture for knowledge management within M2M and IoT solutions.

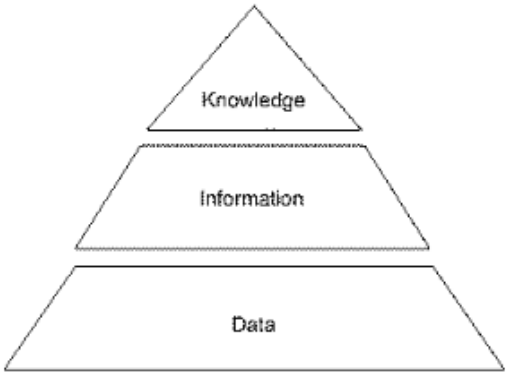

Figure No: 3.16 Data, Information, and Knowledge

3.7.1 Data, information, and knowledge

- For our purposes, we use the following definitions of data:Data: Data refers to "unstructured facts and figures that have the least impact on the typical manager". With regards to IoT solutions, however, 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:
    - 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.
    - 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

3.7.2 A Knowledge management reference architecture

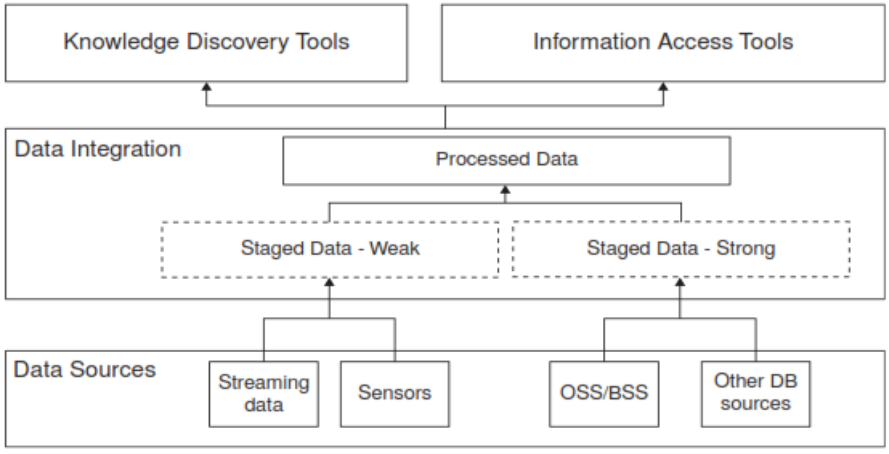

Figure No 5.17 Knowledge Reference Architecture for M2M and IoT.

- 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.

3.7.3 Retrieval layer

- Once data has been collated and processed, it is time to develop insights from the data via retrieval.
- This can be of two main forms: *Information Access and Knowledge Discovery.*

Information access tools

- Information access relates to more traditional access techniques involving the creation of standardized reports from the collation of strong and weak typed data. Information access essentially involves displaying the data in a form that is easily understandable and readable by end users. A variety of information access tools exist, from SQL visualization to more advanced visualization tools.

Knowledge discovery tools

- Knowledge Discovery, meanwhile, involves the more detailed use of ICT in order to create knowledge, rather than just information, from the data in question. Knowledge Discovery means that decisions may be able to be taken on such outputs _ for example, in the case where actuators (rather than just sensors) are involved, Knowledge Discovery Systems may be able to raise an alert that a bridge or flood control system may need to be activated.