Chapter 6

Protocol Standardization for IoT

6.1 Web of Things versus Internet of Things

Vinton Cerf is one of the inventors of TCP/IP (transmission control protocol/Internet protocol) around 1978, which was based on his Ph.D. advisor Leonard Kleinrock's packet-switching theory published in 1961. TCP/IP became the required protocol of ARPANET (Advanced Research Projects Agency Network) in 1983. It also allowed ARPANET to expand into the Internet, facilitating features like remote login via Telnet, and later, the World Wide Web. During his tenure from 1976 to 1982 as project manager and principle scientist at DARPA (Defense Advanced Research Projects Agency), Cerf was at the center of the global network's transformation and played a key role in leading the development of the TCP/IP protocols and the Internet. Cerf is credited as father of the Internet.

Tim Berners-Lee was the man leading the development of the World Wide Web, the defining of HTML (hypertext

markup language), HTTP (hypertext transfer protocol), and URL (universal resource locator), used to create web pages. All of those developments took place between 1989 and 1991. For many people who are not tech savvy, the Internet and Web are one and the same. Many people believe Tim Berners-Lee is the father of the Internet due to the success of the World Wide Web. As the Internet existed long before the World Wide Web, Tim Berners-Lee is only "old enough" to be the father of the Web.

We need to distinguish the difference between the Internet and the World Wide Web here. The Internet is the term used to identify the massive interconnection of computer networks around the world. It refers to the physical connection of the paths between two or more computers. The World Wide Web is the general name for accessing the Internet via HTTP, thus www.anything.something. It is just one of the connection protocols that is available in the Internet, and not the only one. The Internet is the large container, and the web is a part within the container. It is common in daily conversation to discuss them as the Internet and the web, and it is a very common mistake for most people to treat the Internet and web as if they were interchangeable, although it can be argued that the World Wide Web is the most popular method of using the Internet. To be technically precise, if the Internet is the restaurant, the web is the most popular dish on the menu. However, it's *the dishes* (in Figure 6.1) that make the Internet popular, useful to everyone, and powerful.

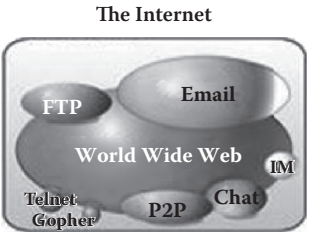


Figure 6.1 Major Internet applications.

By the same token, the key to make the Internet of Things (IoT) take off is the Web of Things (WoT)—the killer applications' platform or base of the IoT. The Web of Things is the next logical step in this IoT evolution toward global networks of sensors and actuators, enabling new applications and providing new opportunities. The Web of Things explores the layer on top of connectivity with things and addresses issues such as fast prototyping, data integration, and interaction with objects. Because the web is omnipresent and flexible enough, it has become an excellent protocol for interacting with embedded devices, and the Web of Things is a vision where things become seamlessly integrated into the web—not just through web-based user interfaces of custom applications, but by reusing the architectural principles of the web for interacting with the quickly expanding ecosystem of devices or embedded devices that are built into everyday smart objects. Well-accepted and well-understood standards and blueprints (such as uniform resource identifier [URI], HTTP, RESTful API, Atom Syndication Format) are used to access the functionality of the smart objects.

The IoT is by definition global and should be considered global in the context even that legislative and regulatory inquiries must be considered locally, regionally, nationally, and internationally. As a matter of fact, lots of IoT work has inevitably been in the WoT arena; however, it's still important to make the disctinction between IoT and WoT. One of the early prototypes mentioning the WoT concept is the Energy Visible project at ETH Zurich [101] in which sensors capable of monitoring and controlling the energy consumption of household appliances offer a RESTful API to their functionality. This API is then used to create a physical mashup. Nimbits (http://www.nimbits.com) is an open-source data historian server built on cloud computing architecture that provides connectivity between devices using data points.

There are also many other WoT applications around the world. WoT portals also started to appear just like the Internet

portals (public websites) such as Yahoo, Sina, and so forth in the early days of the Internet revolution. Some of the WoT applications are listed here. More will be discussed in the next chapter.

- Arduino (http://arduino.cc/en/): Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.
- Japan Geiger Map (http://japan.failedrobot.com/): this map visualizes crowd-sourced radiation Geiger counter readings from across Japan.
- Nanode (http://nanode.eu/): Nanode is an open-source Arduino-like board that has built-in web connectivity. It is a low-cost platform for creative development of web-connected ideas.
- The National Weather Study Project (http://nwsp.ntu.edu.sg/sensormap/): NWSP is a large-scale environmental study project deploying hundreds of mini weather stations in schools throughout Singapore.
- AgSphere: TelemetryWeb.com is launching AgSphere, a new platform that takes the complexity and pain out of connecting agricultural technology products to the web quickly and at low cost. Manufacturers of agricultural equipment can build web-connected solutions that increase margins, reduce risk, and improve efficiencies for farmers by harvesting information from the farm.

6.1.1 Two Pillars of the Web

The invention of HTML/HTTP/URL on top of TCP/IP-based Internet started the Internet revolution; however, it was not until the killer application—Netscape web browser surfaces—that the Internet revolution, symbolized by the World Wide Web, really took off. The Netscape web browser evolved from the earlier Mosaic web browser. It was co-authored by Marc Andreessen at the National Center for Supercomputing

Applications of the University of Illinois Urbana–Champaign beginning in late 1992 and released in 1993. Mosaic was also a client for earlier protocols such as file transfer protocol (FTP), network news transfer protocol (NNTP), and gopher, but HTTP with HTML/URL ruled at the end.

On the other front, the application server became the foundation that helped build widely spreading web-based applications. An application server is a software framework or middleware that provides an environment in which applications can run, no matter what the applications are or what they do. An application server acts as a set of components accessible to the software developer through an API defined by the middleware itself. For web applications, these components are usually performed in the same machine where the web server is running, and their main job is to support the construction of dynamic web pages. However, present-day application servers target much more than just web page generation: they implement services like clustering, fail-over, and load balancing, so developers can focus on implementing the business logic.

The application server is based on the three-tiered (Figure 6.2) or multitiered software architecture. The multitier architecture is a client–server architecture in which the presentation, the

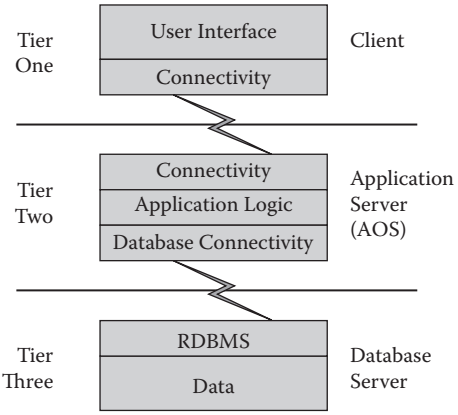


Figure 6.2 Three-tiered architecture.

application processing, and the data management are logically separate processes, which is important for distributed web applications. For example, a web application that uses middleware to service data requests between a user and a database employs multitier architecture. The most widespread use of multitier architecture is the three-tier architecture, which was first used by John Donovan for open-standards Distributed Computing Environment–based applications in Open Environment Corporation, a tools company he founded in the early 1990s.

The Java technologies developed rapidly in parallel with the web in each and every aspect. The Java EE standard-based application server architecture is shown Figure 6.3, which dominates the overall application server market as shown in the Gartner Quadrant [232].

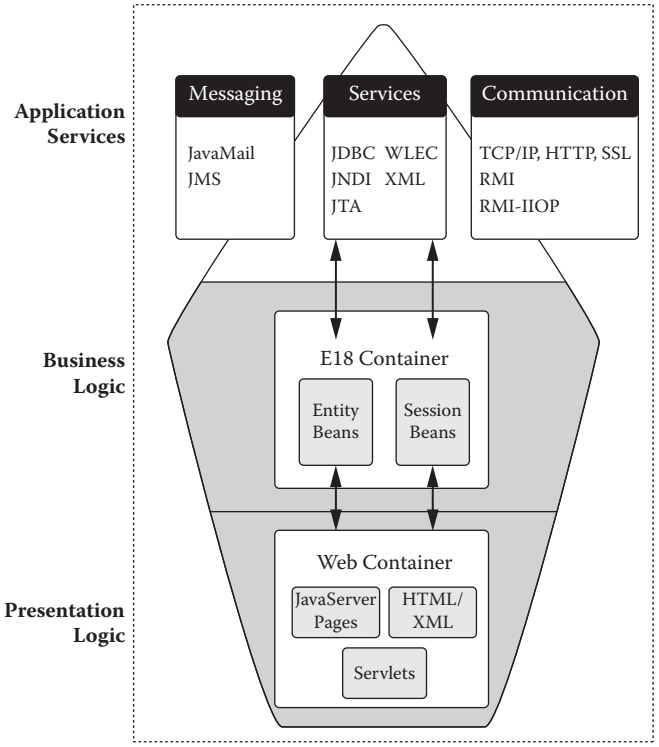


Figure 6.3 Java-based application servers.

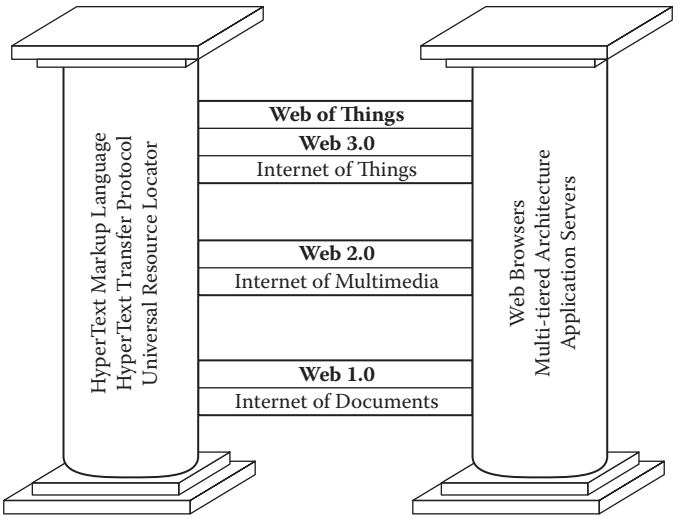


Figure 6.4 Two pillars of the web.

As the two pillars for web applications and the Internet revolution, the protocols (i.e., HTML now in its fifth version, HTML5)/HTTP/URL and the software (i.e., the web browsers and the standardized three-tiered application servers) will continue to be the two pillars of and play an important role in building WoT applications as depicted in Figure 6.4.

However, just as the web applications get more and more sophisticated, the HTML standard evolves, and a large number of standards and substandards and APIs (application programming interfaces) have been created, for example, for JavaEE and JavaME platforms. (These platforms are very relevant to IoT or machine-to-machine [M2M] applications, http://www.m2marchitect.com/what-is-m2m--2.html. There are Java Virtual Machines for all kinds of devices: JVM, CVM, KVM, CardVM, etc., as shown in Figure 6.5.) There is a need to update or augment those standards to fit the specific requirements of WoT/IoT applications, just like the wireless community has done for machine-type communication (MTC).

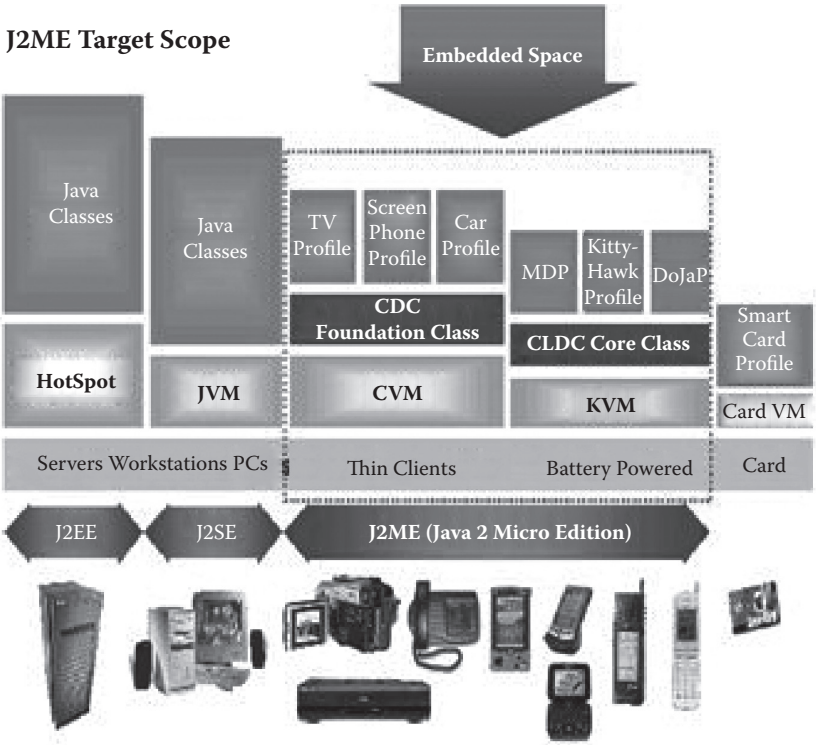


Figure 6.5 Java variants for devices.

A communications protocol is a language of digital message formats and rules for exchanging those messages in or between computing systems and/or in telecommunications. Protocols may include signaling, authentication, and error detection and correction capabilities. A protocol definition defines the syntax, semantics, and synchronization of communication. The specified behavior is typically independent of how it is to be implemented. A protocol can therefore be implemented as hardware or software or both.

Example protocols include data formats like HTML, ebXML (electronic business or e-business extensible markup language), and communication rules (or loosely called protocols) such as SOAP (simple object access protocol) and REST (representational state transfer). We will talk about horizontal

and data format (like HTML) and protocol standard efforts for WoT/IoT applications in greater detail (and propose a unified data representation approach) in this chapter, and WoT/IoT specific middleware and multitiered architecture standard efforts in the next chapter.

6.2 IoT Protocol Standardization Efforts

We have touched on the issues of IoT standardization sporadically in the previous chapters of the book. Now we are going to give a summarized description of the four pillars as well as the generic IoT standardization efforts focusing on data representations and APIs (i.e., protocols). The standards on platform architecture and middleware framework will be discussed in the next chapter. However, because in most cases, the data representation and APIs are intertwined with architecture and framework, it is hard to separate; so there may be some overlaps.

Some of the IoT projects such as the Internet of Things Strategic Research Roadmap by CERP-IoT [8] are still at the grand concept level with limited materialized results. The IoT-A (Internet of Things architecture [113]) is one of the few efforts targeting a holistic architecture for all IoT sectors. This consortium consists of 17 European organizations from nine countries. They summarized the current status of IoT standardization as follows:

- Fragmented architectures, no coherent unifying concepts, solutions exist only for application silos.
- No holistic approach to implement the IoT has yet been proposed.
- Many island solutions do exist (RFID, sensor nets, etc.).
- Little cross-sector reuse of technology and exchange of knowledge.

The author had the same observation (also one of the first who introduced the Intranet/Extranet of Things concept

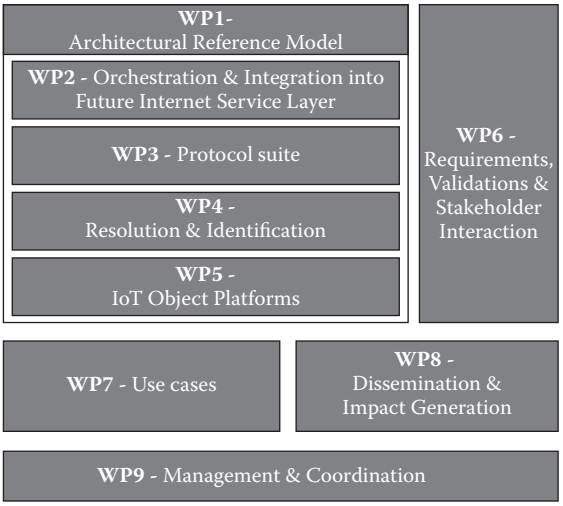
independently [74]) before 2010 based on the four-pillar classification of IoT. Even though the IoT-A consortium doesn't categorize the IoT as four pillars, they do believe solutions for radio-frequency identification (RFID), sensor nets, and so forth are island solutions. In fact, IoT-A doesn't have a systematic, clean-cut, and comprehensive classification of IoT sectors as the foundation. Their "holistic" view of IoT is based on the following scenarios, which is actually not complete and holistic currently.

The key objectives of the IoT-A consortium [103] are as follows:

- Create the architectural foundations of an interoperable Internet of Things as a key dimension of the larger future Internet
- Architectural reference model together with an initial set of key building blocks:
  - Not reinventing the wheel but federating already existing technologies
  - Demonstrating the applicability in a set of use cases
  - Removing the barriers of deployment and wide-scale acceptance of the IoT by establishing a strongly involved stakeholder group
- Federating heterogeneous IoT technologies into an interoperable IoT fabric

A WP (work package) framework of ongoing works has been proposed [103]. Also, the ITU-T has a few study groups (SGs 2, 3, 5, 9, 11, 12, 13, 15, 16, and 17, http://www.itu.int/en/ITU-T/techwatch/Pages/internetofthings.aspx) doing IoT-related works (Figure 6.6).

IPSO (Internet Protocol for Smart Objects, http://www.ipso-alliance.org/) Alliance, formed in 2008, is another effort aiming to form an open group of companies to market and educate about how to use IP for IoT smart objects based on an all-IP holistic approach [81] (Figure 6.7).



	ITU-T Study Group	Study Group Name	Activities Related to IoT
Current Standards Activities	SG 2	Operational aspects of service provision and telecommunication management	Numbering, naming and addressing
	SG 3	Tariff and accounting principles including related telecommunication economic and policy issues	
	SG 5	Environment and climate change	
	SG 9	Television and sound transmission and integrated broadband cable networks	
	SG 11	Signalling requirements, protocols and test specifications	Testing architecture for tag-based identification systems and functions
	SG 12	Performance,QoS and QoE	
	SG 13	Future networks including mobile and NGN	NGN requirements and architecture for applications and services using tag-based ID
	SG 15	Optical transport networks and access network infrastructures	
	SG 16	Multimedia coding, systems and applications	Requirements and architecture for multimedia information access triggered by tag-based ID
	SG 17	Security	Security and privacy of tag-based applications
Pre-standards	Focus Groups	Smart Grid	Smart metering, M2M
		Cloud Computing	Cloud network requirements, e.g., for IoT
		Future Networks	Describe future networks underlying the IoT
		Car Communication	

Figure 6.6 Working groups of IoT standards.

The emerging application space for smart objects requires scalable and interoperable communication mechanisms that support future innovation as the application space grows. IP has proven itself a long-lived, stable, and highly scalable communication technology that supports a wide range of

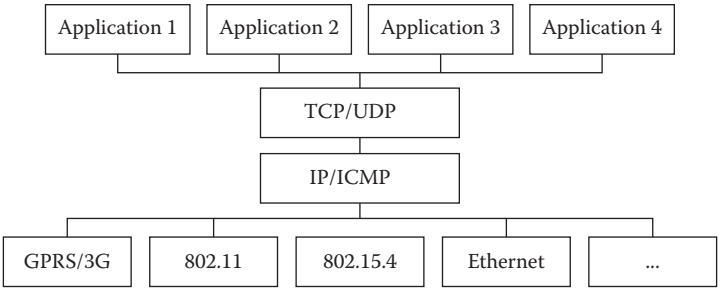


Figure 6.7 All-IP networks.

applications, devices, and underlying communication technologies. The IP stack is open, lightweight, versatile, ubiquitous, scalable, manageable, stable, and end-to-end. It can run on tiny, battery-operated embedded devices. IP therefore has all the qualities to make the Internet of Things a reality, connecting billions of communicating devices. A smart object is defined by IPSO as

- An intelligent (RFID) tag
- A sensor: device that measures a physical quantity and converts it to an analog or digital signal, such as power consumption and quality, vibration of an engine, pollution, motion detection, temperature
- An actuator: device that controls a set of equipment, such as controls and/or modulates the flow of a gas or liquid, controls electricity distribution, performs a mechanical operation
- An embedded device: a purpose-built connected device that performs a specific function, such as a factory robotic arm, vending machine, smart grid analyzer
- Any combination of the above features to form a more complex entity

The IPSO Alliance works closely with Internet Engineering Task Force (IETF), the Institute of Electrical and Electronics

Engineers (IEEE), the European Telecommunication Standard Institute (ETSI), the International Society of Automation (ISA), and others, and relies on the standards developed by them. IPv4, IPv6, and 6LoWPAN were all developed by engineers within IETF, and the role of the alliance is to ensure how they are used, deployed and provided to all potential users.

The Mobile IP protocol is a related IETF-proposed standard that provides a network layer solution to node mobility across IPv4 (Mobile IPv4) and IPv6 (Mobile IPv6) networks. Mobile IP allows a node to change its point of attachment to the Internet without having to change its IP address.

Another solution to the problem is network mobility (NEMO). NEMO is an extension of Mobile IP that enables an entire network to change its attachment point to the Internet. NEMO works by moving the mobility functionality from Mobile IP mobile nodes to a moving network's router. The router is able to change its attachment point to the Internet in a manner that is transparent to attached nodes.

SHIM6 [114], a serverless Mobile IPv6 protocol, allows two communicating nodes to overcome connection loss problems that may arise if one node changes its IP address (locator) during an established communication.

Sensinode [115], as an example, provides embedded networking software and hardware products based on IP-based 6LoWPAN technology for demanding enterprise applications. NanoStack™ 2.0 is an advanced 6LoWPAN protocol stack software product for 2.4 GHz radios. The NanoRouter™ 2.0 platform includes software and hardware solutions for 6LoWPAN-Internet routing infrastructure.

Also, since its creation in 2003, ETSI TISPAN (Telecommunications and Internet converged Services and Protocols for Advanced Networking) has been the key standardization body in creating the next-generation networks (NGN) specifications, which is a synonym of IoT.

6.2.1 M2M and WSN Protocols

Most M2M applications are developed today in a highly customized fashion, and vertical-specific industry bodies are busy crafting standards for markets ranging from the auto industry to the smart grid. A broad horizontal standard is a key requirement for the M2M industry to move from its current state of applications existing in isolated silos based on vertical market or underlying technology to a truly interconnected Internet of Things. Such a horizontal standard is expected to be the major impetus to growth in the future.

Efforts to develop broad, horizontal standards for the M2M market are gaining momentum [49,105]. The most important activity is occurring within the context of the International Telecommunication Union's (ITU) and ETSI's (M2M Technical Committee) Global Standards Collaboration (GSC), which has established the M2M Standardization Task Force (MSTF, created during the GSC-15 meeting in Beijing, China, in September 2010) to coordinate the efforts of individual standards development organizations (SDOs), including China Communications Standards Association, Telecommunications Industry Association TR-50 Smart Device, etc.

The end result of these efforts is to define a conceptual framework for M2M applications that is vertical industry and communication technology agnostic, and to specify a service layer that will enable application developers to create applications that operate transparently across different vertical domains and communication technologies without the developers having to write their own complex custom service layer [105]. The high-level M2M architecture from MSTF does include fixed and other noncellular wireless networks, which means it's a generic, holistic IoT architecture even though it is called M2M architecture (M2M and IoT sometimes are used interchangeably in the United States and in the telco-related sectors). Despite all of the positives, it seems the voices from the SCADA (supervisory control and data acquisition) and RFID

communities are relatively weak; efforts to incorporate existing SCADA standards such as OPC, ISA-95, and RFID EPCIS, ONS, and others are not seen yet. It remains to be seen whether all of the stakeholders from the four pillars of IoT will be equally included in the loop.

This is a more comprehensive approach than the 3GPP's MTC effort described in the previous chapter. Considering 3GPP is only one of the SDOs in the MSTF, this makes sense and good results are much anticipated from MSTF. Some vertical applications on top of the unified horizontal M2M architecture are already under way [105]. Companies such as Telenor Objects, Numerex, and others are building MSTF standards compliant products [104] already.

Other M2M standards activities include the following:

- Data transport protocol standards: M2MXML, JavaScript Object Notation (JSON) (originally not for IoT applications, used by the Mango open source M2M project), BiTXML [117], WMMP (shown in Figure 6.8), MDMP, open Building Information Exchange (oBIX), EEML, open M2M Information exchange (oMIX)
- Extend OMA DM to support M2M devices protocol management objects
- M2M device management, standardize M2M gateway
- M2M security and fraud detection
- Network API's M2M service capabilities
- Charging standards
- MULTI IMSI, M2M sevices that do not have MSISDN
- IP addressing issues for devices IPV6
- Remote diagnostics and monitoring, remote provisioning and discovery
- Remote management of devices behind a gateway or firewall
- Open REST-based API for M2M applications

One of the benefits of using sensor data is that the data typically can be repurposed many times, thereby reducing cost

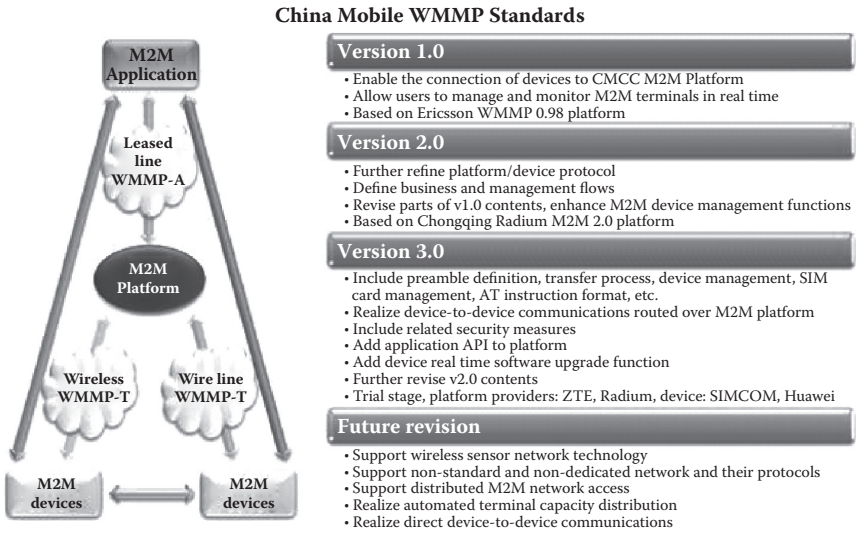


Figure 6.8 China Mobile's WMMP standard.

and maximizing benefit. For example, weather observations (temperature, wind speed and direction, humidity, and so on) can be used in climate modeling, weather forecasting, plume modeling, insurance risk analysis, ski area location decisions, and dozens of other applications. However, the ability to access and use the same sensors in multiple application domains, to share sensor data, and to maximize the full value of sensor networks and data is severely hindered by a lack of interoperability. Hundreds of sensor manufacturers build sensors for specific purposes, often using their own "language" or encodings, different metadata, and so forth. Standard data representation (together with WSN middleware) is the key to materialize data integration and increase interoperability.

There are a number of standardization bodies in the field of WSNs. The IEEE focuses on the physical and MAC layers; the IETF works on layers 3 and above. IEEE 1451 is a set of smart transducer interface standards developed by the IEEE Instrumentation and Measurement Society's Sensor Technology Technical Committee that describe a set of open, common, network-independent communication interfaces for connecting

transducers (sensors or actuators) to microprocessors, instrumentation systems, and control/field networks. One of the key elements of these standards is the definition of transducer electronic data sheets (TEDS) for each transducer. The TEDS is a memory device attached to the transducer, which stores transducer identification, calibration, correction data, and manufacturer-related information.

The IEEE 1451 family of standards includes the following:

- 1451.0-2007 Common Functions, Communication Protocols, and TEDS Formats
- 1451.1-1999 Network Capable Application Processor Information Model
- 1451.2-1997 Transducer to Microprocessor Communication Protocols & TEDS Formats
- 1451.3-2003 Digital Communication & TEDS Formats for Distributed Multi-drop Systems
- 1451.4-2004 Mixed-mode Communication Protocols & TEDS Formats
- 1451.5-2007 Wireless Communication Protocols & TEDS Formats
- 1451.7-2010 Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and TEDS Formats

The goal of the IEEE 1451 family of standards is to allow the access of transducer data through a common set of interfaces whether the transducers are connected to systems or networks via a wired or wireless means. IEEE p1451.3 is XML based and allows the manufacturer to change the contents.

Cross-network (e.g., between Bluetooth and ZigBee) standards are not as proliferate in the WSN community compared to other computing systems, which make most WSN systems incapable of direct communication with each other. The contents on WSN described in the previous chapters are more devices or network focused. OGC (Open Geospatial Consortium) and W3C has been doing research and standardization work following a data-focused approach [233].

The Semantic Sensor Web (SSW) [105] is an approach to annotating sensor data with spatial, temporal, and thematic semantic metadata based on OGC SWE (Sensor Web Enablement). The following data-encoding specifications have been produced by OGC SWE Working Group (in addition to the web service specifications that will be described in Chapter 7):

- SWE Common—common data models and schema
- SensorML—models and schema for sensor systems and processes surrounding measurements
- Observations & Measurements (O&M)—models and schema for packaging observation values
- Transducer Markup Language (TML)—models and schema for multiplexed data from sensor systems

The European Union SENSEI [109] project creates an open, business driven architecture that fundamentally addresses the scalability problems for a large number of globally distributed wireless sensor and actuator networks (WSAN) devices. It provides necessary network and information management services to enable reliable and accurate context information retrieval and interaction with the physical environment. By adding mechanisms for accounting, security, privacy, and trust, it enables an open and secure market space for context awareness and real-world interaction. An ambient ERP system supported the SENSEI.

Tangible results of the SENSEI project are as follows:

- A highly scalable architectural framework with corresponding protocol solutions that enable easy plug-and-play integration of a large number of globally distributed WSAN into a global system, providing support for network and information management, security, privacy and trust, and accounting

- An open service interface and corresponding semantic specification to unify the access to context information and actuation services offered by the system for services and applications
- Efficient WSAN island solutions consisting of a set of cross-optimized and energy-aware protocol stacks including an ultra-low-power multi-mode transceiver
- Pan European test platform, enabling large-scale experimental evaluation of the SENSEI results and execution of field trials, providing a tool for long-term evaluation of WSAN integration into the NGN

ISO/IEC JTC1 WG7 (Working Group on Sensor Networks), established in 2009, preceded by JTC 1 SGSN SC6, created the ISO/IEC 29182 Reference Architecture for sensor networks application and services focusing on telecommunication and information exchange between systems. The architecture is defined through the following set of documents:

- ISO/IEC 29182 Part 1: General overview and requirements
- ISO/IEC 29182 Part 2: Vocabulary/terminology
- ISO/IEC 29182 Part 3: Reference architecture views
- ISO/IEC 29182 Part 4: Entity models
- ISO/IEC 29182 Part 5: Interface definitions
- ISO/IEC 29182 Part 6: Application profiles
- ISO/IEC 29182 Part 7: Interoperability guidelines

6.2.2 SCADA and RFID Protocols

As described before, we use the SCADA term as one of the IoT pillars to represent the whole industrial automation arena in this book. Industrial automation has a variety of vertical markets and there are also many types of SCADAs.

IEEE created a standard specification, called Std C37.1™, for SCADA and automation systems [116] in 2007, targeting mostly

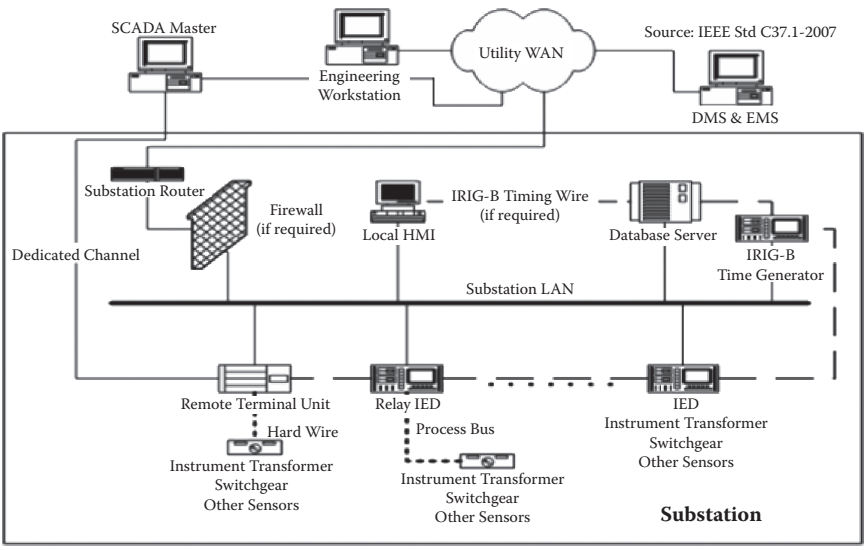


Figure 6.9 IEEE Std. C37.1 SCADA architecture.

power SCADA applications (Figure 6.9). It's recognized in the specification that in recent years, network-based industrial automation has greatly evolved with the use of intelligent electronic devices (IEDs), or IoT devices in our terms, in substations and power stations. The processing is now distributed, and functions that used to be done at the control center can now be done by the IED, that is, M2M between devices. Despite the fact that many functions can be moved to the IED, utilities still need a master station, the IoT platform, for the operation of the power system. Due to the restructuring of the electric industry, traditional vertically integrated electric utilities are replaced by many entities such as GENCO (Generation Company), TRANSCO (Transmission Company), DISCO (Distribution Company), ISO (independent system operator), RTO (regional transmission organization), and so forth. To fulfill their role, each of these entities needs a control center, that is, a substation, to receive and process data and take appropriate control actions.

This specification addressed all levels of SCADA systems and covered the technologies used and, most importantly,

the architecture of how those technologies interact and work together. However, no XML data formats and componentized architecture details are specified, which is perhaps why SCADA has long been regarded as a traditional control system market. People working in that area are often not aware of Internet-based IT innovations and cannot relate their work to a new concept such as IoT.

Wireless sensor systems have the potential to help industry use energy and materials more efficiently, lower production costs, and increase productivity. Although wireless technology has taken a major leap forward with the boom in wireless personal communications, applications for industrial field device systems must meet distinctly different challenges. That's where the ISA100, Wireless Systems for Industrial Automation, comes in. The ISA100 was developed by the standards committee of the Industrial Society for Automation, which was formed in 2005 to establish standards and related information that will define procedures for implementing wireless systems in the automation and control environment with a focus on the field level. The committee is made up of more than 400 automation professionals from nearly 250 companies around the world, lending their expertise from a variety of industrial backgrounds.

The ISA100 family of standards is designed with coexistence in mind, bringing peace of mind for the end user. We know that customers have other wireless solutions installed today and have the need for any future system to coexist with these installed systems. Therefore, the standards will feature technology to ensure the best performance possible in the presence of other wireless networks. For example, the ISA100 has created a new subcommittee to address options for convergence of the ISA100.11a and WirelessHART standards. This initiative is a key step in the mission of the ISA100 committee to develop a family of universal industrial wireless standards designed to satisfy the needs of end users across a variety of applications.

OPC, which stands for Object Linking and Embedding (OLE) for Process Control, is the original name for a standard

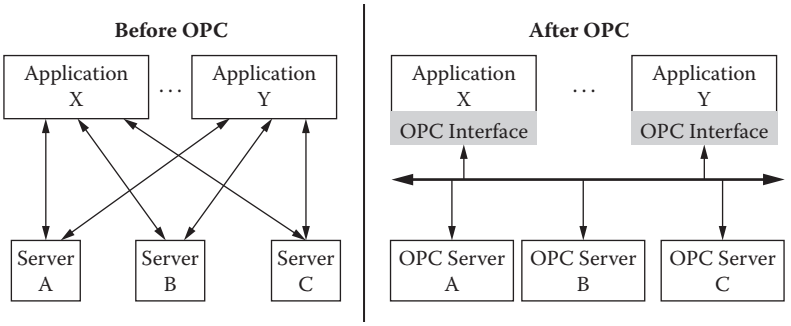


Figure 6.10 OPC standard for device connection.

specification developed in 1996 by an industrial automation industry task force. The standard specifies the communication of real-time plant data between control devices from different manufacturers (Figure 6.10). OPC is managed by the OPC Foundation [120] with more than 220 members worldwide including major firms in industrial automation, instruments manufacturers, building automation, and others.

OPC originated from the DDE (dynamic data exchange) technologies based on DOS for PCs. The introduction of Windows 3.0 in 1990 made Windows an inexpensive, mainstream computing platform, providing the ability for a PC to run multiple applications simultaneously and a standard mechanism for those applications to exchange data at runtime. Wonderware's InTouch™ SCADA software had the greatest impact for the transition from DDE to OPC. It introduced a means of networking DDE traffic (NetDDE™, which was later taken up by Microsoft) and also greatly increased the effective bandwidth of DDE by packing multiple data items into each packet or message. OLE (based on COM, common object model) and OCX (now ActiveX based on .NET) were launched in 1992. A number of SCADA vendors saw the chance to standardize the interface between the SCADA core and the device drivers that were actually responsible for acquiring the data, and the first-draft version of the OPC specification was

released in December 1995 by the OPC Foundation sponsored by Microsoft.

OPC was designed to provide a common bridge for Windows-based software applications and process control hardware. Standards define consistent methods of accessing field data from plant floor devices. This method remains the same regardless of the type and source of data. An OPC server for one hardware device provides the same methods for an OPC client to access its data as each and every other OPC server for that same or another hardware device. The aim was to reduce the amount of duplicated effort required from hardware manufacturers and their software partners, and from the SCADA and other HMI producers, in order to interface the two. When a hardware manufacturer had developed their OPC server for the new hardware device, their work was done to allow anyone to access their device; and when the SCADA producer had developed their OPC client, their work was done to allow access to any hardware, existing or yet to be created, with an OPC-compliant server.

OPC has achieved great success in many application areas, most of them closely related to or part of IoT applications. However, OPC's success story is accompanied by some caveats. For example, standard OPC DA (data access) is based on Microsoft's COM and DCOM technology and is consequently restricted to the Windows operating system. In addition, DCOM communication is easily blocked by firewalls that prevent OPC clients from accessing data over a wide-area network and the World Wide Web. New approaches, such as XML-DA and United Architecture (UA) [234], have been developed to make OPC technology available on other platforms or accessible by other systems.

The RFID protocols and data formats are relatively well defined, mostly by EPCglobal, and unified compared with protocols and formats of the other three pillars of IoT. The RFID protocols (such as PML, Object Naming Service [ONS],

Edgeware, EPC Information Service [EPCIS], Application Level Event [ALE], etc.) have been described in the previous chapters, so we will talk only about protocols for the related contactless smart cards here.

The smart cards with contactless interfaces (RFID is a subset) are becoming increasingly popular for payment and ticketing applications such as mass transit and stadiums. Visa and MasterCard have agreed to an easy-to-implement version deployed in the United States. Smart cards are also being introduced in personal identification and entitlement schemes at regional, national, and international levels. Citizen cards, drivers' licenses, and patient card schemes are becoming more prevalent. Some examples of widely used contactless smart cards are Taiwan's EasyCard, Hong Kong's Octopus card, Shanghai's Public Transportation Card, and Beijing's Municipal Administration and Communications Card.

The standard for contactless smart card communications is ISO/IEC 14443. It defines two types of contactless cards (A and B) and allows for communications at distances up to 10 cm. An alternative standard for contactless smart cards is ISO/IEC 15693, which allows communications at distances up to 50 cm (Figure 6.11).

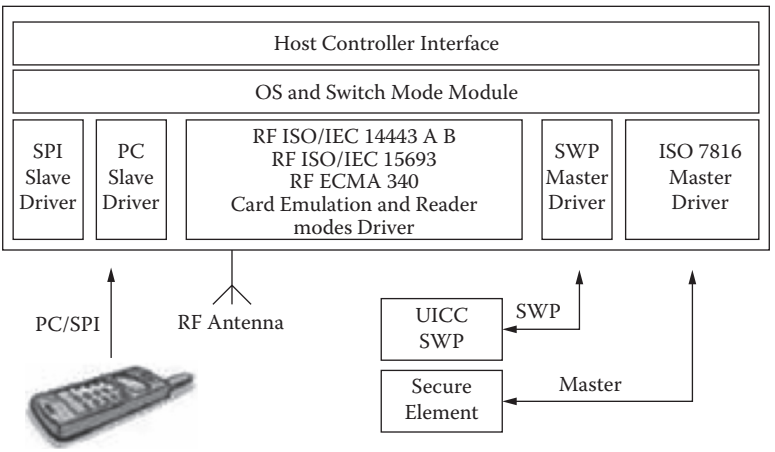


Figure 6.11 ISO/IEC 14443/15693 smart card standards.

6.2.3 Issues with IoT Standardization

Apart from the standardization efforts that can be categorized as one of four pillars, there are also standardization efforts from major vertical IoT applications such as smart grid and telematics. For example, we have described the NGTP (Next Generation Telematics Protocol) standard for telematics in Chapter 2. The participating SDOs of GridWise (Smart Grid) include almost all of the SDOs in the information and communications technology (ICT) industry.

It should be noted that not everything about standardization is positive. Standardization is like a double-edged sword: critical to market development, but it may threaten innovation and inhibit change when standards are accepted by the market. Standardization and innovation are like yin and yang, and they could be contradictory to each other in some cases, even though this observation is debatable.

We have also noted in the previous chapter that among the four pillar segments of IoT, for example, in ETSI/3GPP's M2M/MTC and EPCglobal's RFID standardization efforts, different consortia, forums, and alliances have been doing standardization in their own limited scope covering mostly the area they are familiar with. For example, 3GPP covers only cellular wireless networks and EPCglobal's middleware covers only RFID events. Even within the same segment, there are more than one consortium or forum doing standardization without enough communication with each other, and some are even competing with each other.

Some people believe that the IoT concept is well established; however, some gray zones remain in the definition, especially on which technologies should be included, such as the four pillars described in this book, in order not to pose a limit too strict to the system.

Even though some of the IoT standard organizations have cooperation and interaction, as shown in Figure 5 of Jacobs et al. [102], it is limited and not open enough. The following

two issues for the IoT standardization in particular and the ICT standardization in general may never have answers:

- ICT standardization is a highly decentralized activity. How can the individual activities of the network of extremely heterogeneous standards-setting bodies be coordinated?
- It will become essential to allow all interested stakeholders to participate in the standardization process toward the IoT and to voice their respective requirements and concerns. How can this be achieved?

The only, or at least better, possible solution to address these chaotic situations is to try to standardize the omnipresent middleware and the XML-based data representation from across-industry organizations such as World Wide Web Consortium (W3C), Organization for the Advancement of Structured Information Standards (OASIS), and others.

OASIS and W3C are web-oriented standard organizations. Their expertise makes them capable of doing high-level, segment-independent WoT standardization. They are now actually participants of ETSI/3GPP and other efforts, but they are currently more like observers instead of active participants. Most other IoT SDOs are more qualified to do IoT (communication layers) standardization instead of WoT standardization since they often lack a high-level view and experiences of the system across the globe and across industries.

6.3 Unified Data Standards: A Challenging Task

We have talked about the two pillars of the Internet in this and previous chapters and pointed out that the HTML/HTTP combination of data format and exchange protocol is the foundation pillar of the World Wide Web as depicted in Figure 6.12 [74].

We have also listed and described a great number of data standards and protocols proposed for the four pillar domains

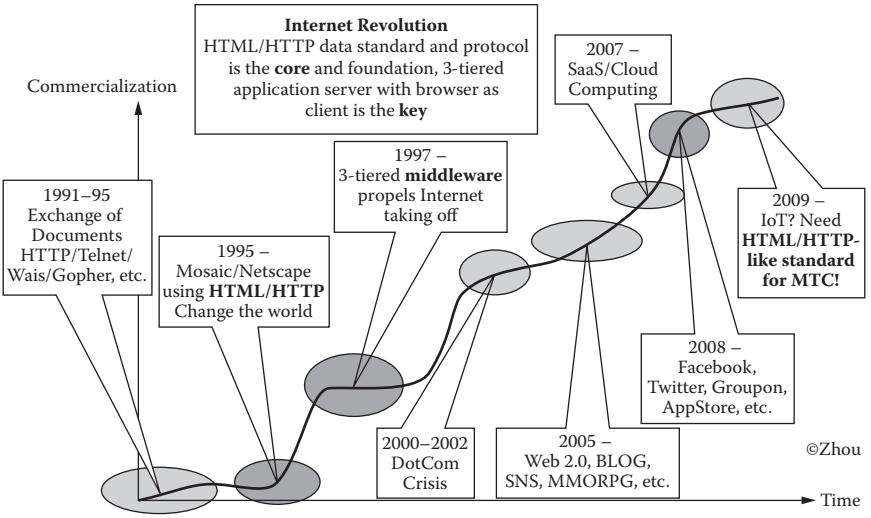


Figure 6.12 Evolution of the web.

of the IoT in the previous sections of this chapter and previous chapters. Many issues still impede the development of IoT and especially the WoT vision.

Many standardization efforts have been trying to define a unified data representation and protocol for IoT. This is the right direction even though some of the approaches are limited by their scope of application domains and technologies used as described and discussed before.

Before the Internet of Things, the Internet was actually an Internet of documents or of multimedia documents. The two pillars of the Internet including HTML/HTTP turned the Internet into the World Wide Web. By the same token, we need to turn the Internet of Things into the Web of Things to make sense of everything. What will it take to make this to happen?

- Do we need a new HTML/HTTP-like standard for MTC and WoT? If there is no need to reinvent the wheel, what extensions do we need to build on top of HTML/HTTP or HTML5?

- The browser is intended for humans, so do we need a new browser for machines to make sense of the ocean of machine-generated data? If not, what extensions do we need to make to the existing browsers?
- Today, most new protocols are built on top of XML. For OS there must be XML-based data format standards or a metadata standard to represent the machine-generated data (MGD). Is it possible to define such a metadata standard that covers everything?

There are many different levels of protocols, but the ones that most directly relate to business and social issues are the ones closest to the top, the so-called application protocols such as HTML/HTTP for the web. The web has always been a visual medium, but a restricted one at best. Until recently, HTML developers were limited to CSS and JavaScript in order to produce animations or visual effects for their websites, or they would have to rely on a plug-in like Flash. With the addition of technologies like the canvas element, Web GL, and scalable vector graphics (SVG) images in HTML5, this is no longer the case. In fact, many new features deal with graphics on the web with HTML5: 2-D Canvas, WebGL, SVG, 3-D CSS transforms, and Synchronized Multimedia Integration Language (SMIL).

Developers are taking advantage of these features: a flood of HTML graphics demos have been showing up on the web, ranging from implementations of old two-dimensional graphics algorithms, to brand-new techniques created specifically for the modern web. Using graphics to display real-world behavior of things is a very important feature of IoT systems; for example, sophisticated graphic display of device behavior and process is a must in most SCADA-based industrial automation systems. The use of SVG technologies can reduce the footprint of a graphic by up to 90 percent. On the left of Figure 6.13 is an oil and gas industrial automation application using SVG built by the author's team in 2008 before HTML5 was announced. A tool or IDE (integrated development

Figure 6.13 SVG graphics of ezM2M.

environment) is normally required as a companion product for SCADA to create and configure graphics in large volume efficiently and productively based on a large graphic (parts) library.

Mango is an open-source software system for M2M applications. It enables users to access and control electronic sensors, devices, and machines over multiple protocols simultaneously. It relies heavily on JavaScript to render its graphical pages. While rendering, massive amounts of data are being transferred between the Mango server and the browser. Furthermore, because of the continuous polling for new data, it can easily hog the central processing unit of the computer displaying said data. SVG, supported by HTML5 as well as major browsers such as Internet Explorer 9, Safari (Apple doesn't support Adobe Flash), and others, with embedded scripting capabilities can be a very useful technology; however, enhancement to HTML/HTTP and the browsers is still required for MTC support. Human-oriented browsers may also have to be enhanced for processing massive MGDs similar to the mobile browser on audio devices. Content management is a big market sector

of the Internet and web; future IoT contents may also require similar technologies for sensor content management.

The Resource Description Framework (RDF) is a family of W3C specifications originally designed as a metadata model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats. It could be investigated and used as a metadata model for WoT applications.

An RDF browser is a piece of technology that enables you to browse RDF data sources by way of data link traversal. The key difference between this approach and traditional browsing is that data links are typed (they possess inherent meaning and context just like IoT data), whereas traditional links are not typed. There are a number of RDF browsers including Tabulator, DISCO (Hyperdata Browser), and OpenLink RDF Browser.

SOAP and RESTful protocol frameworks are extensions on top of HTTP for web services. They are more than protocols or data formats but rather the so-called protocol frameworks. SOAP and REST frameworks can be used to provide data exchange protocols for IoT applications, which will be discussed in the next chapter.

At the back-end server side or deep down in the ETSI/3GPP-defined M2M/IoT protocol stack, a unified IoT data format and protocol can borrow and leverage the standards proposed for e-commerce or e-business, especially B2B (business-to-business) standards. To clarify, EAI (enterprise application integration) is the integration of legacy software systems within an organization to allow the systems to have a more complete and consistent worldview. This is essentially an internal matter. B2B is about cross-organization integration, the creation of public interfaces to allow partners and customers to interact with internal systems in a programmatic fashion.

E-commerce comprises the B2B, business-to-consumer (B2C), and consumer-to-consumer (C2C) business models, which describe who the target buyer market the target seller market are. B2B application integration bridged the gap

between legacy IT infrastructures and emerging B2B collaboration frameworks and allows the IT infrastructure to provide greater adaptability to the business of the enterprise and easier management of constantly evolving business processes. The same principle and technologies apply to legacy IoT infrastructure and emerging Internet- and web-based IoT collaboration frameworks also.

There are two important enabling technologies: electronic data interchange (EDI) and XML. EDI describes the rigorously standardized format of electronic documents. The EDI standards were designed to be independent of communication and software technologies. EDI can be transmitted using any methodology agreed to by the sender and recipient. This includes a variety of technologies, including modem (asynchronous and synchronous), FTP, e-mail, HTTP, AS1, AS2, and so forth. XML is a more recent invention for exchanging information between computer systems. XML is a markup language used to create smart data and documents for applications.

A newer standard such as ebXML incorporates as part of its design solution some borrowed ideas from both EDI and XML. It offers businesses the opportunity to build an interoperable e-commerce infrastructure. In a computer system, ebXML specifies the business rules for how two different systems talk to each other. Those systems need to be written using any application programming language (such as XML, Java, C, C++, or Visual Basic), executed in a specific middleware (like JavaEE or COM+; the author has worked in the BEA Weblogic Integration team developing Java software frameworks based on ebXML and RosettaNet protocols for e-commerce applications), and designed using a specific modeling language (UML). To model B2B business processes, an abstract computer-modeling language such as UML or the XML language–specific business process modeling language (BPML) is used. BPML is an XML-based meta-language for modeling, deploying, and managing business processes such as order management, customer care, demand planning, product development, and strategic outsourcing.

XML or ebXML coexists with the popular web formatting language HTML. HTML tells us how the data should look, but XML tells us what it means. XML enables complex linking (using XPointer and XLink) and allows users to define their own elements (using a document type definition [DTD] or schema). It also provides a style sheet for formatting documents (using XSL). The key issue of IoT applications is also about integration and interoperability, so the HTML/ebXML approaches still apply and new HTML-based, ebXML-like standards should be the solution for the Internet of Missed Things and the focus of IoT data representation standards for WoT applications.

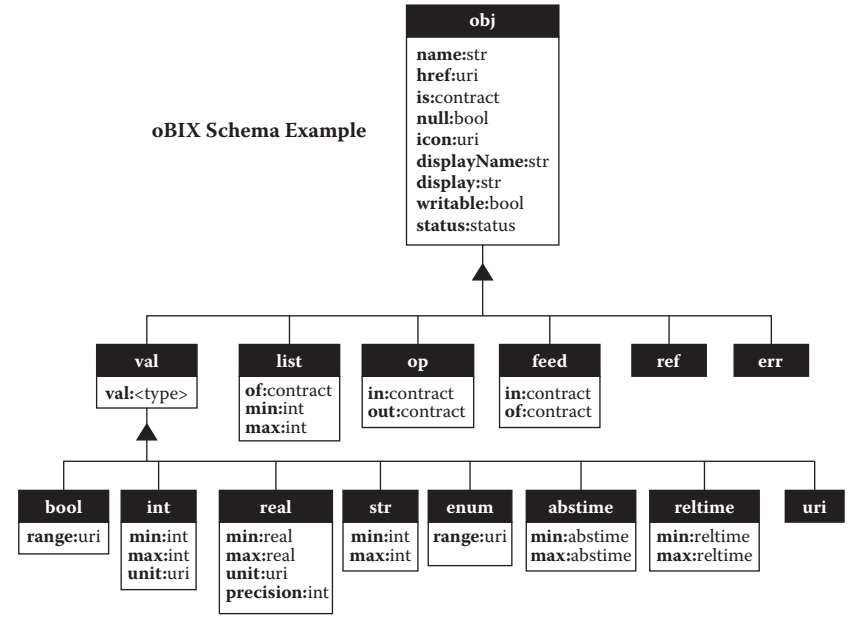
There are a few specifications for the WoT data format mentioned before. The following is a longer list, which is summarized in *Smarter Earth* [74]:

- BITXML, data format defined by BITX Inc.
- CBRN, format for Chemical, Biological, Radiological and Nuclear data
- CAP, Common Alerting Protocol, of EXDL
- EDDL, Electronic Device Description Language
- EEML, Extended Environments Markup Language from Pachube
- EXDL, Emergency Data Exchange Language of OASIS
- FDT, Field Device Tool
- IRIG, Inter-Range Instrumentation Group
- MDMP, M2M protocol of China Telecom
- M2MXML, Machine-to-Machine XML
- NGTP, Next-Generation Telematics Protocol
- oBIX, open Building Information eXchange
- OMA SyncML, Open Mobile Alliance Synchronization Markup Language
- oMIX, open Machine Information eXchange, proposed by the author's team
- OPC, OLE for Process Control
- PML, Physical Markup Language of EPCglobal
- SensorML, Sensor XML of OGC

- TEDS/IEEE 1451, transducer electronic data sheets of IEEE
- TransducerML, Transducer Markup Language of OGC
- WMMP, Wireless Machine Management Protocol of China Mobile

Figure 6.14 shows the example schema of oBIX (open Building Information eXchange).

There are many ongoing and in some cases overlapping efforts to develop the CBRN standards within industrial, federal,




```
<obj name="IN7" is="obix:BoolPoint obix:Point" display="Group Number 7"
    displayName="Working Coffee Machine Sensor" writable="true">
    <bool name="value" val="true" display="Heating the coffee"
    displayName="I/O GROUP" />
    <obj name="status" is="obix:Status" display="group status">
            <bool name="overridden" val="false" />
            <bool name="disabled" val="false" />
            <bool name="fault" val="false" />
            <bool name="down" val="false" />
            <bool name="inAlarm" val="false" />
            <bool name="unackedAlarm" val="false" />
            <bool name="historyStart" val="false" />
            <bool name="historyEnd" val="false" />
    </obj>
    <abstime name="timestamp" val="1970-01-01T02:00:00.000+02:00" />
 </obj>
```

Figure 6.14 oBIX schema.

and international standards organizations. Oak Ridge National Laboratory, Tennessee (ORNL), where the author has worked, has invested a significant amount of research and development into implementing, testing, deconflicting, and harmonizing these efforts to establish an overarching set of working interoperability standards to connect CBRN sensors, detectors, and data to emergency response, homeland security, and defense applications (Table 6.1).

EDI for e-commerce is like OPC for WoT/SCADA—it's a legacy standard. A new, unified, open, cross-pillar, and usable standard like ebXML for WoT is needed, or efforts must be made to harmonize [123] the existing standards and to make them interoperable such as those in Figure 6.15 before a general sensor information model or a metadata XML schema can be established eventually.

Table 6.2 from *Smarter Earth* [74] summarizes the IoT data and protocol standardization efforts with the author's analysis, views, and suggestions about future developments.

As described before, there are many efforts to create a unified, cross-segment, overarching data representation standard for WoT. Due to domain knowledge differences, this is a great technological challenge or even mission impossible. Looking at the issues from a different angle, it is probably more realistic to create interoperability standards to integrate WoT systems between the four-pillar IoT systems. Even within a pillar segment, it's not an easy task to create a unified data standard. However, it's worth a try, especially at the early development stage of WoT before the IoT "information islands" are formed, as is the situation in many existing IT systems.

6.3.1 Unified Identification of Objects

One of the key issues of unified data format for IoT is the unique identification of objects. When the IoT application is within the intranet or extranet of an organization, which is the case most often currently, the identification is not an issue.

Table 6.1 CBRN Data Standard Efforts: Standards Activities for CBRN Sensors

Department of Defense	*DHS*	*Institute of Electrical and Electronics Engineers (IEEE)*	*OASIS*	*Open Geospatial Consortium (OGC)*
POC Activity				
JPEO-CBD	Standards Portfolio S&T Directorate	Sensor Interface Standards	Emergency Interoperability Consortium	Sensor Web Enablement
Prof. Tom Johnson, NPS	Dr. Bert Coursey, DHS S&T	Mr. Kang Lee, NIST	Ms. Elysa Jones, OASIS	Mr. Sam Bacharach, OGC
Standards				
JPM-IS Data CBRN	ANSI N42.32 ANSI N42.33	IEEE 1451.0 IEEE 1451.1	Common Alerting Protocol	Sensor Observation Services
Common Data Model	ANSI N42.34 ANSI N42.35 ANSI N42.38	IEEE 1451.2 IEEE 1451.3 IEEE 1451.4	Emergency Data	Sensor Planning Service Sensor Alerting Service
NATO NBC Standards (Allied Tactical Publication 45B)	ANSI N42.42 ASTM E54	IEEE 1451.5 IEEE 1451.6	Exchange Language	Geospatial Markup Language
STANAG 5523	AOAC International			Web Feature Services

Source: Courtesy of ORNL.

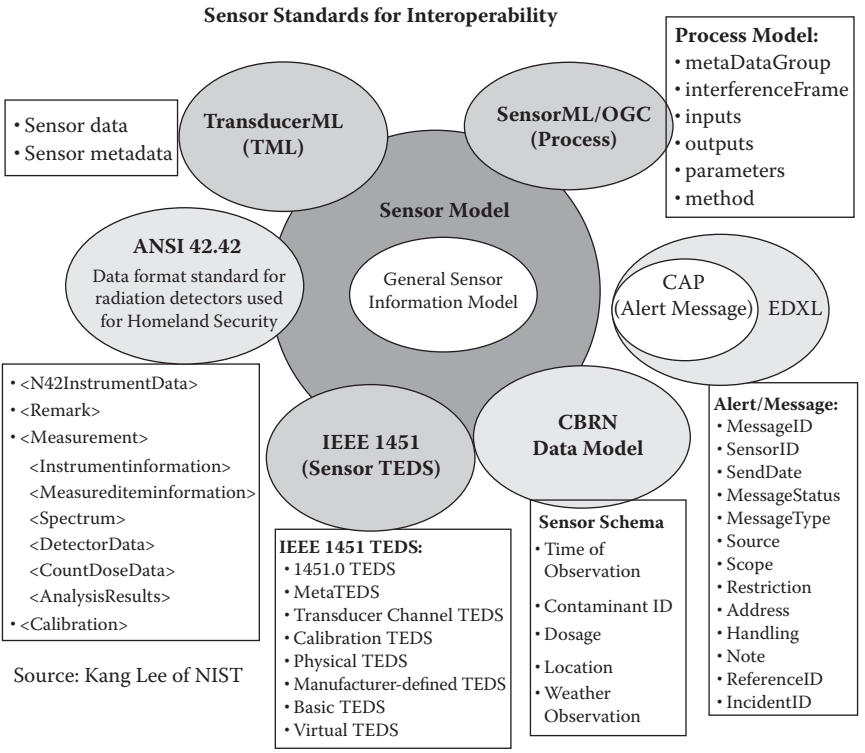


Figure 6.15 Unified Data Standard approaches.

However, when the WoT applications prevail in the future, globally unique identification of objects becomes a serious issue. Object identification can essentially encompass the naming, addressing, or both of an asset or device. In the web, the identification of a resource that represents some form of information has been achieved by the development of the universal resource identifier (URI), which is a global agreement on the identification of a particular resource based on specified schemes. In IoT, similar to the Internet and the web, objects need to have common naming and addressing schemes and also discovery services to enable global reference and access to them. In this section, we review common identification, naming, and addressing schemes and frameworks that can contribute to designing a naming and addressing scheme for IoT/WoT.

Table 6.2 Unified IoT Data Standard Based on Existing Data Formats and Protocols

IoT Standards Matrix ©Zhou	*Existing Data formats and Protocols*		*Unified New IoT Standards*		
			Goals	*Necessity*	*Feasibility*
Application Layer (M)	Data Formats	BITXML, EXDL, MDMP, M2MXML, NGTP, oBIX, oMIX, ONS/PML, OPC, SyncML, WMMP, etc.	Create new, unified, open, cross-sector, usable data standards including formats, exchange protocols, processing and modeling frameworks.	**High** Enable easier integration and interoperability.	**Medium** Enhanced HTML and ebXML-like standards, hard to create a unified data format due to domain differences.
	Software Framework	ArchestrA, CoAP, DRM, ECF, ezM2M, HYDRA, IDM, MDM, OSGi, PaaS, RESTful, SaaS, Sedona, SOA, SODA, SOAP, etc.	Data standards compliant SaaS/ PaaS 3-tiered platform middleware, support new paradigms such as DRM.	**High** Enable easier integration, new MAI paradigm, etc.	**High** Enhanced 3-tiered application servers, OSGi middleware for server-side.

continued

Table 6.2 (continued) Unified IoT Data Standard Based on Existing Data Formats and Protocols

IoT Standards Matrix ©*Zhou*	*Existing Data formats and Protocols*		*Unified New IoT Standards*		
			Goals	*Necessity*	*Feasibility*
Transmission Layer Protocols (C)	Wired Long Distance	IP(TCP/UDP/HIP), IP over Everything/ Everything over IP, Ethernet, IPv6, ATM, Frame Relay, SDH, FDDI, Fiber Channel, ISDN, SS7, PSTN, VPN, VoIP, Cable/xDSL, etc.	"3-network" convergence, all-IP networks, IPv6 should be leveraged for IoT applications, existing networks OK for most IoT applications.	**Medium** MTC support enhancements and optimizations.	**Medium** It takes time for all-IP, IPv6 to prevail.

	Wired Short Range	ANSI C12.18, AS-i, BACNet, CanBus, CC-Link, ControlNet, Dali, DeviceNet, DF-1, DLMS/IEC 62056, Dupline, FF, FlexRay, HART, HomePNA, IEC 61107, InterBus, LIN, LonWorks, KNX, ModBus, MOST, MTConnect, P-Net, ProfiBus, SwiftNet, Vnet/IP, WorldFIP, CC-Link, PLC, Industrial Ethernet, RS232, RS485, VAN, etc.	Ruggedness enhancements, few new protocols are required, no need to reinvent the wheel.	**Low** Few or no new protocols required.	**Low** Few or no new protocols required.

continued

Table 6.2 (continued) Unified IoT Data Standard Based on Existing Data Formats and Protocols

IoT Standards Matrix ©Zhou	*Existing Data formats and Protocols*		*Unified New IoT Standards*		
			Goals	*Necessity*	*Feasibility*
	Wireless Long Distance	2G: GSM, CDMA, etc.; 3G:WCDMA,EV-DO,HSUPA, EV-DOrA, UMTS, etc.; 2.5G: GPRS, EDGE,HSCSD, etc.; 4G:EV-DOrB, LTE, WiMAX, UMB/UWB, TD-SCDMA, etc. Satellite M2M, GPS, etc.	All-IP, Mobile IP, etc., helpful but not required, MTC enhancements for low bandwidth, low latency IoT applications, backend BOSS system enhancements.	**Medium** Dedicated packet switch MTC network helpful but not required.	**Medium** Few or no new protocols required, optimization focus.

	Wireless Short Range	Bluetooth, BSN, DECT, DSAH 7, EDACS, EnOcean, HyperLan, HyperMAN, 6LoWPAN, HomeRF, HomeIR, InfiNET, Insteon, IrDA, IRIG, ISA 100.11a, LMDS, NFC, OpenSky, OSIAN, RFID, TETRA, TransferJet, WAVE, Wavenis, WiFi/WAPI, WirelessHART, Zigbee, Z-Wave, etc.	Few new protocols required, focus should be on embedded OS or middleware, TinyOS, MagnetOS, Contiki, Mantis, SINA, SensorWare, etc.	**Medium** Enhancements on embedded OS and middleware.	**Low** Few or no new protocols needed, leverage existing protocols.
	Sensor Layer (D)	TEDS/IEEE 1451, CBRN, TransducerML, SensorML, IRIG, EXDL/CAP, AutomationML, OpenPLC XML, EDDL, FDT, CANOpen, etc.	Optimized and minimized version of application layer XML data standards, supported by embedded OS and middleware. Universal OSGi middleware for device-side hardware.	**High** Enable easier integration and interoperability.	**Medium** Minimized ebXML-like standards, it's hard to create a unified standard due to small footprint.

The ubiquitous ID (uID) framework [124] was developed in Japan. uID or ucode is the identification number assigned to individual objects. The ucode is a 128-bit fixed-length identifier system. Moreover, a mechanism to extend the ucode length in units of 128 bits has been prepared to meet the future demands so that codes longer than 128 bits also can be defined.

In the field of RFIDs, EPCglobal [51] has promoted the adoption and standardization of electronic product code (EPC), which has been used to uniquely identify RFID tags. It is based on the URI model. ID@URI, developed by the DIALOG research project [125], is another identification model that takes the same properties of the EPC/ONS standard but can be manifested in bar codes as well. EPC (recognized in the United States and Europe) is a competing standard of uID (used in Japan), affected by national or regional interests, so compatibility and interoperability is always an issue politically instead of technologically.

In the mobile telecoms domain, the international mobile equipment identity (IMEI) [126] provides a means for unique identification of mobile phones. IMEI is formed through a set of digits that represent the manufacturer, the unit itself, and the software installed on it. IMSI conforms to the recommendation of ITU-T E.212 stored in the SIM card and often used as a key in the home location register. For public switched telephone network (PSTN), the operator has the possibility to identify uniquely the resource with the E.163/E.164 addresses (a.k.a., telephone numbers). Besides, operators provide the mobile subscriber ISDN number (MSISDN) following the ITU-T recommendation E.164. This unique number is used for routing calls in the operator networks.

The following unique ID schemes refer to addresses and names of electronic objects at various levels of the OSI stack along with their related protocols: MAC address, IP address on the Internet, e-mail address, uniform resource name (URN), URI, URL, and others. IP address is certainly a straightforward

unique ID scheme; however, on January 19, 2010, the Numbers Resource Organization, the entity tasked with protecting the unallocated pool of remaining IPv4 Internet addresses, issued a statement indicating that less than 10 percent of IPv4 addresses remain unallocated. Obviously, if millions to hundreds of millions of new devices are going to be networked in an Internet of Things in the coming years, this shortage of IPv4 addresses poses a challenge, particularly for countries outside of North America that were allocated comparatively fewer IPv4 addresses to begin with. The long-term solution is IPv6, which enables orders of magnitude larger numbers of available IP addresses. Most mobile network operators (MNOs) are in the planning stages for this transition to IPv6 or have already made the transition. M2M-optimized mobile infrastructure can help with the transition by future-proofing applications through the use of techniques such as IPv6 tunneling over IPv4. Essentially, this capability would enable remote M2M devices to use native IPv6 addresses that are translated to IPv4.

In the software domain, the UUID (universal unique identifier, as shown in Figure 6.16) was proposed in the early 1980s. It's an identifier standard used in software construction, standardized by the Open Software Foundation (later called the Open Group) as part of the Distributed Computing Environment. In 1996, it became part of ISO/IEC 11578 documents, and more recently documented in ITU-T Rec. X.667 | ISO/IEC 9834-8:2005. The IETF has published Standards Track

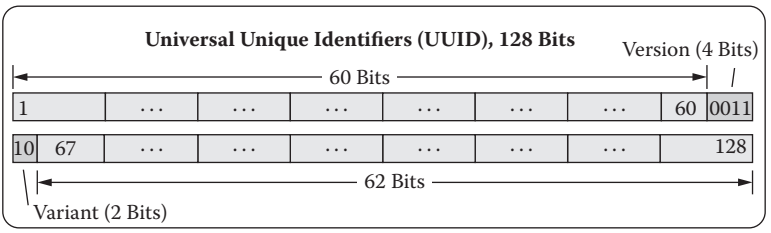


Figure 6.16 Structure of UUID.

RFC 4122, which is technically equivalent with ITU-T Rec. X.667 | ISO/IEC 9834-8. The intent of UUIDs is to enable distributed systems to uniquely identify information without significant central coordination. UUIDs are widely used in distributed middleware such as Tuxedo (the author used to work in the team), CORBA, and JavaEE. One widespread use of this standard is in Microsoft's globally unique identifiers (GUIDs, a different name for UUID). Other significant uses include Linux's ext2/ext3 file systems, GNOME, KDE, and Mac OS X, all of which use implementations derived from the UUID library found in the e2fsprogs (Ext2 file systems utilities) package. UUID was also used in the Bluetooth standard.

The ASN.1 project was established in February 2001 by ITU-T Study Group 7 to assist existing users of ASN.1 within and outside of ITU-T, and to promote the use of ASN.1 across a wide range of industries and standards bodies. Since September 2001, the responsibility for the ASN.1 project resides with Study Group 17 and the project now also encompasses object identifiers (OIDs) and registration authorities.

In an open and international world such as the one of telecommunications and information technologies, one often needs to reference an object in a unique and universal way. Many standards define certain objects for which unambiguous identification is required. This is achieved by assigning OID to an object in a way that makes the assignment available to interested parties. It is carried out by a registration authority. The naming structure of OID is a tree structure that allows the identification of objects in a local or international context, without being limited by the registration authority or by the number of objects they can register (Figure 6.17). Each new node is associated with a name and a number that will be used for data transfers. An OID is semantically an ordered list of object identifier components, for example, {joint-iso-itu-t(2) ds(5) attributeType(4)distinguishedName(49)}, or for short 2.5.4.49. OID and ASN.1 is also widely used in X.400/X.500,

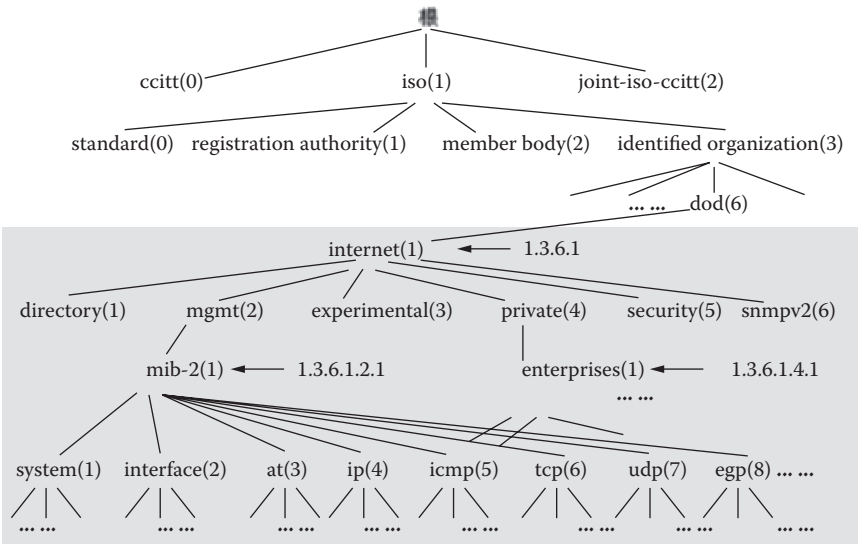


Figure 6.17 Hierarchy of OID.

H.323, SNMP, and in many wired or wireless network protocols such as UMTS, etc.

OID is a flexible, extensible framework. It can also be used together with other ID schemes such as UUID, uID, EPC, and others. For example, the member body number in China as a country is 156. An IoT object's locally (such as Beijing with a number 10) unique ID such as 66666666 can be prefixed with 1.2.156 to form a globally unique ID 1.2.156.10.66666666, just like the phone number prefixed with a country number.

OID is a good identification candidate for IoT objects considering it's a mature scheme and supported by both ISO and ITU. However, it's a bit complex to use (since it is part of ASN.1 involving registration processes etc.) compared with other schemes such as UUID, EPC, or uID. Considering EPC and uID are not compatible with each other due to the aforementioned reasons, UUID is a widely accepted scheme used in distributed environments including IoT (which is a distributed system by itself) and it already exists in many software systems, so UUID is a

better scheme for IoT (not necessary on the Internet), especially WoT applications.

Compared with creating a unified, cross-segment, overarching data standard for WoT, it is possible and a must to create a global unique identification. The hurdles are more about interest considerations of related parties rather than technological difficulty.

EPC, uID, UUID, and so forth are basically fixed-length IDs, while OID and others are variable-length IDs. OID is more flexible in intranet and extranet IoT applications. However, as described in the next chapter, the software industry has been an object-oriented world for a long time. Object-oriented programming (where *objects* are the *Things* of IoT) has a profound root in software representation and programming, so using UUID/GUID, already widely used in object-oriented systems, as the identification of Things would be a breeze to transition from object-oriented to real-world objects and the integration of IoT systems with existing IT systems.

6.4 Summary

In this chapter, we talked about the difference between WoT and IoT as well as the web and the Internet. To build WoT, the standardization of communication protocols, especially data formats, plays a crucial and important role as evidenced by the invention and dominance of the HTML/HTTP standard. One of the main value propositions or suggestions to the IoT industry in this book is to focus on protocol standardization, especially data format standardization, instead of standardization on other layers of the value chain, such as creating or modifying existing communication protocols such as Zigbee and others.

Various kinds of existing and emerging protocols in all four pillar segments are investigated and analyzed to support the proposition. However, the feasibility of creating a unified XML data format including a global identification scheme of objects

for all IoT applications that cover the four pillar segments is still under investigation by some of the IoT projects worldwide, particularly in Europe. Issues existing in current standardization efforts are also discussed.

In the next chapter, we will talk about existing IoT architectures and the unified architectural framework for IoT.