Edge Networks & Devices for the Internet of Things

Peter T. Kirstein

Abstract: This paper considers how existing concepts underlying the development of the Internet are being strained in the emerging Internet of Things (IoT). It also explores how the well-known and tried Domain Name Service concepts, which map hierarchic names to addresses, can be extended for the IoT. The extension greatly broadens the concept of name/address mapping to digital objects with identifier/attribute database mapping for physical objects, applications, and data. Finally, this essay discusses the properties of the identifier management systems, showing how scalability, security, and flexibility can be supported in the IoT.

PETER T. KIRSTEIN, a Foreign Honorary Member of the American Academy since 2002, is Professor of Computer Systems at University College London. He is the author of *Space-Charge Flow* (1967) and has recently published articles in *Sensors*, *Information Systems*, and *International Journal of Informatics Society*.

The initial aim of the Internet was to develop a system that would allow computers to connect together, irrespective of their location or individual method of connection. That system grew to connect the several billion systems in use today. The Internet is now becoming the *Internet of Things* (IoT), embracing the hundreds of billions (or trillions) of digital devices that can sense or activate aspects of our lives. The IoT is still in its infancy: the state of networks and variety of equipment types in the IoT today is comparable to that at the onset of the Internet, from 1975 – 1980. In this essay, we explore some of the theory behind the design of the Internet, and consider the ways in which the needs of the IoT fundamentally differ. At the same time, we will examine similarities between the development and growth of the Internet and the IoT. Of course, Internet protocols (IP) have developed hugely since the Internet's youth. We will not consider the core of the Internet in this paper, but will focus on what new demands the IoT may make on that core.

Even thirty-five years ago, it was clearly important to link together the many network deployments of different architectures. But there was a question whether to choose to *adapt* between network types or wait for universal *adoption* of the same type. Adoption eventually won out, though it took fifteen years.

doi:10.1162/DAED_a_00363

The design of the early Internet assumed that any compatible networks would use the Internet protocol IPv4 at the network level. Because the ARPANET (Advanced Research Projects Agency Network) could address only 256 computers with permanent identities, computer scientists considered the increase to four billion machines, as allowed by IPv4, as more than would ever be needed. Within about fifteen years, however, it was apparent that even this would be inadequate. Various short-term measures were taken to allow the mechanisms to cope, including introducing private addresses that allowed address space to be reused, albeit with a loss of flexibility, which made it very difficult for a device to have more than one address. While these measures allowed the Internet to continue to grow without adopting a new universal network type, it had become clear that the existing structure could not cope with the huge numbers envisioned even without the advent of the IoT.

In order to plan for the inevitable address crunch, the Internet community decided to specify a new protocol: IPv6. This resolved the addressing problem and fixed a number of other shortcomings: allowing multiple addresses for a single interface, group operations, limiting the scopes of addresses, and improvements in mobility support, among other gains. Although aids have encouraged transition, moving all new customers, let alone existing customers, from IPv4 to IPv6 has proven to be a long and difficult process. Yet this is now occurring on an increasing scale, particularly for newer applications and in contexts in which customers are running out of IPv4 address space. While there are various attempts to design completely different network architectures and components, IPv6 will prevail.

To cope with the relatively large number of computers that the initial Internet intended to connect, it was necessary to define some human-friendly directory of computers; hence, the *Domain Name Service* (DNS) was defined to connect user-friendly names to Internet addresses.[1] Engineers deployed a scalable architecture, which has lasted through the introduction of IPv6. The system is hierarchic, meaning the owner of the domain has almost complete freedom at any level to allocate address ranges – indicated by a "." – to the domains below it. For kirstein.cs.ucl.ac.uk, for example, the .uk domain is allocated a large block of addresses: it has jurisdiction over a number of domains including commercial (.com), nonprofit (.org), government (.gov), and academic (.ac). The registered owner of .ac allocates from his range of addresses a set to each university (such as University College London: .ucl), which in turns allocates a range of its addresses to each department (such as computer science: .cs). The Internet assumes that all end points obey the Internet protocols (IPv4 or IPv6), and so the only value that has to be returned from a query to the DNS is the IP address. There is very limited security: the owner of a domain like cs.ucl.ac.uk enters certain security features to ensure that only authorized entities may insert name/address pairs. The implementation of the DNS has evolved over the last thirty-five years; it is highly distributed and many parts are replicated for resilience. The lowest levels are often near the end systems, and the number of entries on a particular platform is kept reasonably low to maintain performance. The system has continued to perform with the few billion names it now contains. Individual user processes often cache the information of often-used end devices to minimize further access delays.

In the original Internet, it was generally assumed that each interface to a computer was attached to a unique network and had a unique name. Thus, the name/address could be unique. More recently, with the advent of both wireless networks and IPv6, this uniqueness has been put into doubt.

The same wireless interface can be seen by a number of different overlapping networks, and the same interface can have a number of different names (as seen by different application operators).

Over the years, the DNS was modified in three important respects: the capability for adding descriptions of services (DNS-RR),[2] the ability to search for services (DNS-SD),[3] and the capability to authenticate DNS entries (DNSSEC).[4] While parts of the RR (Resource Records) can be encrypted, access to all attributes of the entries has remained open. All entries to the DNS are assumed to obey the IP suite, though the RR gives some information on how services can be accessed.

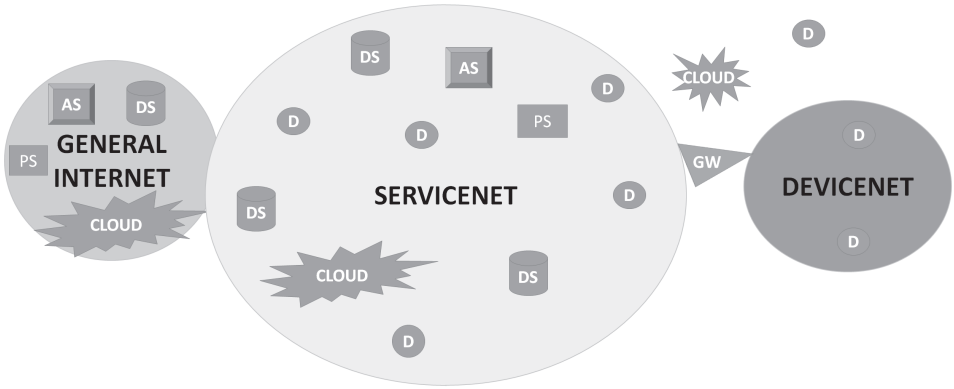
With the move toward the IoT, many of the fundamental assumptions of the Internet are overturned. It was initially assumed that we would only communicate with computers at the edge of the networks that make up the Internet. More recently, the scope of edge devices has broadened, including personal computers, telephones, printers, and SatNavs (satellite navigation). However, the digital controllers in these devices all obey the Internet protocols. If there are changes in these protocols, we can assume that most of the devices will evolve so that their successors can incorporate the changes. With the IoT, this may be the case, but some areas, such as building automation, bridges, and ships, may have much slower rates of change. There are already many standards for automation systems, which often now use Internet interfaces, though these usually just allow the same procedures to be carried out remotely as were previously performed locally. In these systems, the whole concept of network, edge device, and network technology is much broader.

Figure 1 illustrates a particular application (using IP technology) running over a specific network we call the *ServiceNet*. The application will be connected to the Internet, but also to many different devices (D). Some devices may be IP-enabled, some not. The latter are shown connected through a gateway (GW) to a network called the *DeviceNet* (T). The ServiceNet is related to a physical deployment of devices, gateways, data storage elements (DS), process servers (PS), and application servers (AS). Some of these may be free-standing; others may be on processors in a computing cloud (CLOUD). The various processing elements may be directly related to the deployment; then they are shown as being located on the ServiceNet. They may be much more remote entities, shown in Figure 1 as on the general Internet. It is important to understand that while there may be a large number of real objects deployed in the IoT, the ServiceNet is a virtual network of the subset used in a particular application.

Figure 2 highlights the distinction between a *DeploymentNet* (DNET) and a ServiceNet. There may be a number of different deployments, each characterized by a single owner and database. An example might be individual smart buildings, in which the DNET refers to the entities that are a subset of those within that building. An application might refer just to one such deployment: for instance, all the temperature gauges or lights in that building. It might, however, refer to entities in several buildings: such as the set of fire alarms or electricity meters on the whole street. This is indicated in Figure 2 by calling the ServiceNet an *Application-ServiceNet* (APP-SERVICENET). The network connecting all the devices in specific applications is thus a *virtual network*. Different applications may be concerned with different subsets of devices in the different deployments.

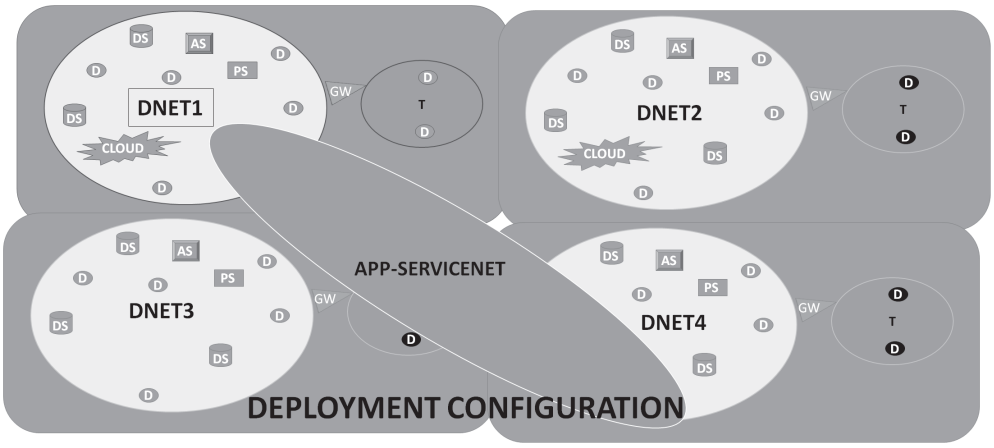
This is indicated in Figure 3 by the different APP-SERVICENETs shown. In light of this, one view of the deployments is the *deployment configuration*: the collection of all the physical devices deployed. Normally, there would not be a single database or description illustrating this collection; rather,

Figure 1
Basic Network Diagram



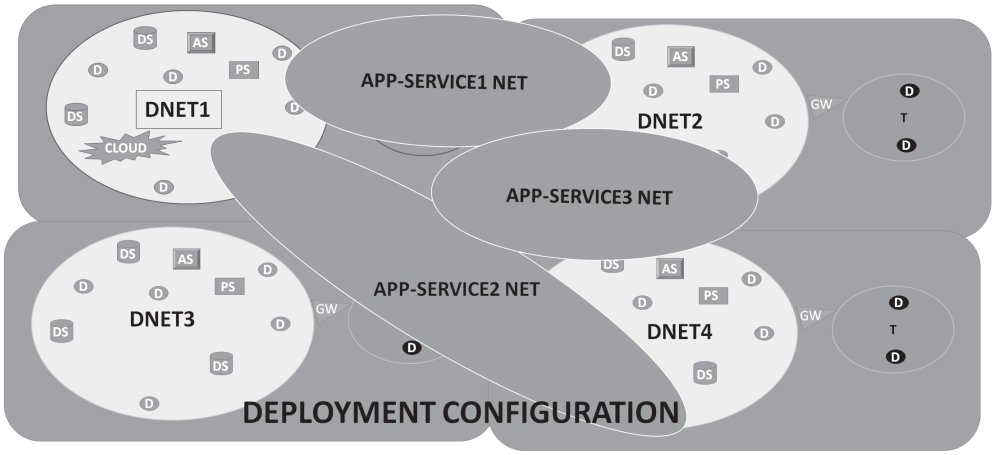
Relevant IP-enabled devices are located on the ServiceNet; those that are not so enabled are on the DeviceNet via a gateway. The servers are on the ServiceNet or general Internet. Key: devices (D); gateway (GW); data storage elements (DS); process servers (PS); application servers (AS); computing cloud (CLOUD). Source: Figure prepared by the author.

Figure 2
Single Applications in Complex Configuration



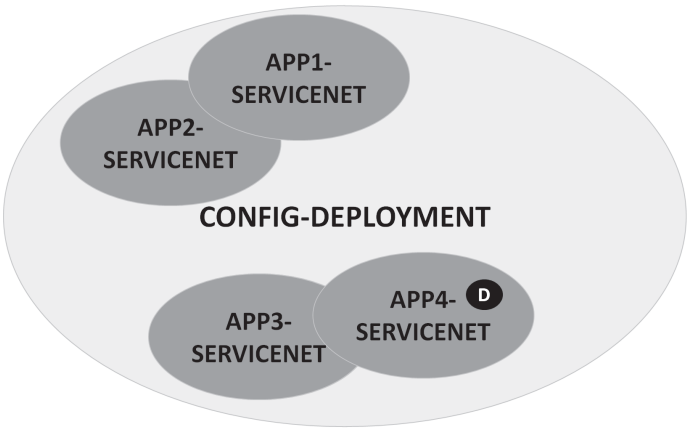
Deployments belonging to different entities are shown on different DeploymentNets. An Application-ServiceNet will use a subset of devices that may be on several DeploymentNets. Key: devices (D); gateway (GW); data storage elements (DS); process servers (PS); application servers (AS); computing cloud (CLOUD); DeploymentNet (DNET); DeviceNet (T); ServiceNet/Application-ServiceNet (APP-SERVICENET). Source: Figure prepared by the author.

Figure 3
Multiple Applications in a Complex Configuration



Several different applications can use the same DeploymentNet configuration. Sometimes several applications can use the same devices in different ways. Key: devices (D); gateway (GW); data storage elements (DS); process servers (PS); application servers (AS); computing cloud (CLOUD); DeploymentNet (DNET); DeviceNet (T); ServiceNet/Application-ServiceNet (APP-SERVICENET). Source: Figure prepared by the author.

Figure 4
Multiple Applications on Given Deployment



An alternate representation of Figure 3, showing a general deployment and several applications using the same or different devices that use subsets of that deployment. Key: devices (D); ServiceNet/Application-ServiceNet (APP-SERVICENET); configuration deployment (CONFIG-DEPLOYMENT). Source: Figure prepared by the author.

there will be separate deployments and databases referring to different management units, as is shown clearly in Figure 2.

If we now consider how applications may interact with physical deployments, the picture becomes even more confusing. In fact, there is a third type of entity in the IoT: namely, *data*. In some forms of deployment, sensors produce data as a result of interactions from an application. In others, the deployment is such that the data are produced spontaneously and continually. Each block of data may then contain certain *metadata* about the data, which may contain attributes that indicate the source of the data, the time it was produced, the location of the producer, both authentication information to ensure it was produced as stated and security information to indicate how it may be used, and even ownership information. The task of dealing with these superpositions of deployments and applications is too complex in real space, particularly if we try to depict the use of the same device or data by different deployments or applications. It is much more effective to work in cyberspace, provided that we can show clearly how to relate the cyberspace to the physical space that we think we better understand (see Figure 3).

The situation illustrated in Figure 3 may be generalized, as shown in Figure 4. Here the set of all devices and subsystems in a particular environment is called a configuration deployment (CONFIG-DEPLOYMENT). The different applications form specific ServiceNets. Sometimes the applications are quite distinct and use different devices; sometimes, however, they overlap by using some devices in several applications. For example, the collection of air conditioners in a building might be managed in one application. The set of all air conditioners of a particular type in a city may be managed for servicing purposes by another entity. Provided the details of the deployments are known, it is possible to plan whether a particular application is feasible and useful. Moreover, by slightly extending the deployment, whole new applications might be achievable at marginal cost.

It is clear we need a more holistic description of devices, deployments, applications, and data than has been attempted so far. The key to this expansive description is to consider not specific devices or networks, but just *digital objects* (DOs). We may work with DOs, their identifiers, and their attributes and denote this as working in cyberspace. Digital objects are a much broader concept than physical objects. The different perspectives of the same deployment described above become simpler here: it is now represented by different DOs. A DO is defined by its identifier; this is a string of bits that represent the name of the object, which usually has some hierarchic structure. The identifier is associated with a set of attributes that describe its properties. Thus, the air conditioning unit as seen by the building's operator can have an entirely different identifier from that seen by the service engineer. But DOs need not be associated only with physical objects; they can also be associated with processes and data objects. It all depends on the attributes associated with the DO. All the discussion on DOs, particularly about the properties desired, is heavily based on the work of electrical engineer Bob Kahn and the Corporation for National Research Initiatives (CNRI), who have implemented the Handle System.

With every DO, one must associate an identifier: a handle by which we can refer to, and work with, the DO. The most general way to describe the DO's properties is through a set of attributes, each described by a *type/value pair*. The first describes the nature of the attribute, the latter describes some value. Of course, there must be a description elsewhere of what is meant by that *type* and how it is represented by the *value*. The identifier may be associated also with

metadata that describe how the attributes may be accessed – and possibly with its owner. To describe a large number of DOs, there should be some form of registry of types, which may even reveal the way the values of DOs will be described. The attributes should be stored in a searchable database, allowing applications to ascertain which DOs they might wish to use. Thus, we can make DOs useful by associating them with an identifier management system (IMS) comprising three parts: an identifier resolution system (IRS), an identifier attribute store (IAS), and an identifier type registry (ITR). For each physical object deployed, a DO is defined. Of course, different stakeholders may have different views of the physical object – as with the air conditioner – thus, it may be associated with several identifiers.

It is possible, and useful, to define applications and data elements as DOs. This allows the metadata either to explain how the DO can be accessed or to reveal attributes of the DO itself. Thus, applications may be stored generically, including a template. A specific application can thus be defined in terms of inserting the details of a particular deployment into the template of the generic application. Similarly, data may be stored as a DO with the appropriate attributes; applications may even simply reprocess the data for a new purpose, providing, of course, that it is authorized to do so. Just as an application might use a subset of the physical deployment, it may use a subset of the relevant stored data for its purposes.

Much of the activity involving DOs can be processed via normal computer clouds; sometimes, however, fast processing is required. In that case, one requires substantially more local processing, o r computer clouds with definable quality of service (QoS) standards of performance.

The IoT has Internet in its makeup, so our analysis should use as many of the tried properties of the Internet as possible. The DNS employs a hierarchical structure, and its implementation architecture has shown that it can be distributed at will; the properties of control of the IoT DO identifiers fulfill exactly the same need as does the DNS. While the DNS was deliberately open for all entities to access, this is not necessarily desirable in the identifier attribute store. While universal accessibility is appealing, the deployment owner may require that access to some of the attributes be limited. Thus, while the original Internet deliberately introduced a minimum of security in the DNS, the IoT would benefit from constraining operations on the IAS. Indeed, it would be desirable to constrain the authorization to *create*, *delete*, *modify*, or *access* both identifiers and attributes, achievable through association with appropriate metadata. We may go further still and ensure that any values of attributes transmitted from the IAS must be encrypted.

There are already many large component databases whose data could, if made available, describe directly the DOs of digital devices; for this reason, it is useful to define one attribute as being the ID in any other such database of interest. We have already given examples of how some DOs may themselves represent complex systems; thus, having another *attribute type* allows the system to be recursive. To tie the cyberspace representation to the physical world, we must have another type of IP address (or name). Provided IPv6 is used, there is no reason why the same physical object cannot be represented by DOs with different identifiers and IP addresses.

With the Internet, the planning of deployments has not been a major part of the network engineer's work. With the IoT, the configuration process and the application design and implementation are central concerns. Further, during the physical deployment phase, the population of the IAS is of vital importance. Most physical de-

ployments follow some domain-specific procedures. For example, in the construction and functioning of a smart building, the architect, installation engineer, building supervisor, security officer, and service engineer each have unique roles. Normally, drawings and specifications are produced as part of a business process; processing algorithmically the models of the physical systems into DO form will be a major aid to implementing large-scale deployments in the future.

This will require the development of tools that can populate the IAS algorithmically based on the data already extant in different domains. This process requires the provision of security tokens; thus, every physical entity that may need to be secured on actuation will need one or more security tokens and an associated list of authorized entities. Similarly, the data of every sensor that provides information that may require authentication should be signed by the authentication token associated with the device. During the setup phases of configurations, attributes required for authentication or actuation should be put in the IAS. In some cases, such as when asymmetric encryption is used, the entry is not sensitive. In others, it is critical that it be stored only in an encrypted form. During operations, the IMS may be used to control proxy security operations for devices too constrained to do them locally. In order for these tools to work well, it is likely preferable to be able to define some *templates* on what attributes are permitted and needed by the entity using the tool. Note that any time there is a change to the configuration – for instance, if a sensor is replaced – it may be necessary to update the IMS; if only to provide a new security token.

For many situations, these physical deployments will be constant. Thus, for example, each building, traffic light system, or surveillance system connected to the IoT may have different physical models that need to be processed to populate the IMS. An application will often deal with a subset of the whole configuration. This might be termed a *virtual deployment*. The individual deployments may belong to different entities. Because there may need to be a negotiation regarding the terms by which an application can use parts of a physical deployment, one part of the metadata associated with an identifier may have to be its ownership. In a typical life cycle, an application will be designed, implemented, deployed, and put into operation. Having determined the usage rights for a deployment, the information in the IAS is used to define the application in the design and implementation phases. Some devices in the IoT may require special processes to access them; this will be specified in attributes stored in the IMS.

It is important to note that during the phases of designing, implementing, and deploying physical infrastructure, data is put into the IAS as part of the deployment process. During the design and implementation of applications, data in the IAS is used to define the virtual configuration appropriate for the application.

Some massive applications do not require access to physical deployments and their related applications. It is adequate to access only the data previously stored. Indeed, this property is at the heart of many of the current generation of large start-up enterprises like Google and Facebook. Their data are produced from a different set of applications and deployments; it is only the authorization to use and deep-mine the data that their applications require.

In the IoT, certain compound operations can be very convenient, such as *reading all sensors on a floor* or *notifying all cars in a particular location of a nearby accident*. Of course, it is possible to define such operations in an application; however, it may make both the design of the application and its opera-

tion simpler if the relevant operations can be carried out in physical space. Similarly, it would be convenient for applications to use network addresses located in the address space of the owner of the application. As explained above, we can associate physical space with cyberspace by defining one attribute of an identifier to be its IP address. If the ServiceNet shown in Figure 1 is in IPv6, then both features are supported at the network level. The group operations can be supported by *multicast*, allowing operations at the network level to be performed on a group of objects. And in this form of network, there is no problem associating different IPv6 addresses with the same object in use in different applications. Neither of these functions are essential, but they certainly ease application design and operation.

There are many extant identity management systems; it is unlikely that they will all adopt the same implementation in the near future. Electronic components in particular, but increasingly also types of subsystems – lifts, cars, automation subsystems – will be stored in an identifier database complete with all their properties. Ideally, another type of attribute is the identity of a given subsystem in a different database.

Even the Internet has relied on a consistent management structure that defines protocols, allocates address space, and specifies security features. While there have been political concerns that international governance bodies such as the Internet Assigned Number Authority (IANA), the Internet Advisory Board (IAB), and the Internet Engineering Task Force (IETF) have not been appointed in the conventional manner, they have nevertheless functioned well. In the context of the IoT, even more governance is likely to be required. It is probable that when the identifier systems outlined here are globally accepted, there will be entities in each application domain that assist in the governance of, at least, the identifier space, and likely also of the attribute types that are standardized for the domain. In this realm, other bodies will be concerned with standardization across domains.

Security is the great challenge posed to those working toward stable governance, authorization, and user responsibility. On the one hand, those responsible for specific installations may need to organize their own security trust chains; on the other hand, the chains may need to be regulated by an official third party. Clearly, in terms of access to the data objects in the IAS, we are moving toward the general considerations of privacy of data, ownership of data, and permitted usage. Here we stray well beyond the past and future of the Internet. However, the provision of these broad classes of DOs inevitably leads to very difficult cases of who is entitled to what access to the IAS and under which conditions.

In retrospect, the concept of the Internet was simple compared to the Internet of Things. At the time, it seemed a daunting task to persuade less than a dozen major suppliers to change completely their protocols for connecting computers together. But it was successful because the concept was so straightforward. Of course, the Internet evolved to deal with issues of scale, heterogeneity, and performance; but the foundational concepts remained relatively stable. Three early adjuncts to the basic Internet protocols were vital: keeping heterogeneity on the edge of the network, restricting security to the edges, and setting up a highly distributed system for name/address mapping. For the IoT, many more large industrial and political players must be persuaded to adopt a common approach. Moreover, the number of edge devices in use in the IoT are many orders of magnitude greater, the governance more challenging, and trustable security more vital than with the Internet. However, the experience gained through the introduction and deployment

of the Internet gives us a much clearer indication of what is required in advancing the IoT.

The way in which the deployment of physical devices is almost orthogonal to the development and deployment of applications provides a clue as to how to proceed. The imperative of being able to scale to much larger numbers of devices, while keeping the size of individual deployment authorities and applications operators manageable, gives a second. How the complex nature of trustable authentication and authorization must be provided for edge devices that have limited computing power and memory capacity is a third. Finally, the need to be sufficiently flexible to allow different communities to adopt myriad ways of working is inevitable. The third orthogonal category of DOs fits naturally into the same basic technology; it is clearly another natural aspect of the IoT. While fundamental to the benefits, and dangers, of the IoT, it leads to whole new deployments, uses and reuses, security and privacy concerns, responsibility and liability, and domains of regulation and control.

The above considerations make it important to work conceptually (in cyberspace) as much as possible. This is particularly so in the case of physical deployments and maintenance. All maintenance of physical devices can be recorded in cyberspace, where the authentication and authorization attributes can also be maintained. This allows applications to be developed on virtual deployments, or even using existing data, which are a subset of the physical deployments and/or data derived from cyberspace databases. The scalability, with manageable subsets, can be assured by adopting the structure of the domain name service, the power of the identifier management system, and the flexibility of allowing attributes to refer to an arbitrary set of other identifier systems. Finally, the growth of cloud computing allows most of the cyberspace work to be carried out in the computer cloud, while operational concerns are carried out in local services, which probably also adopt local clouds with specifiable characteristics. Authorities have stressed the importance of deploying ServiceNets based on the newer Internet protocol IPv6 because of its larger address space capacity. We go further, having considered how use of IPv6 gives important advantages in multistakeholder use of shared interfaces and in enabling the group operations common in the IoT.

ENDNOTES

1 P. Mockapetris, "Domain Names – Implementation and Specification," *IETF RFC 1035* (November 1987), https://www.ietf.org/rfc/rfc1035.txt.

2 The TCP/IP Guide, "DNS Message Resource Record Field Formats," http://www.tcpipguide.com/free/t_DNSMessageResourceRecordFieldFormats.htm.

3 S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," *IETF RFC 6763* (February 2013), https://tools.ietf.org/html/rfc6763.

4 R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Protocol Modifications for the DNS Security Extensions," *IETF RFC 4035* (March 2005), https://tools.ietf.org/html/rfc4035.