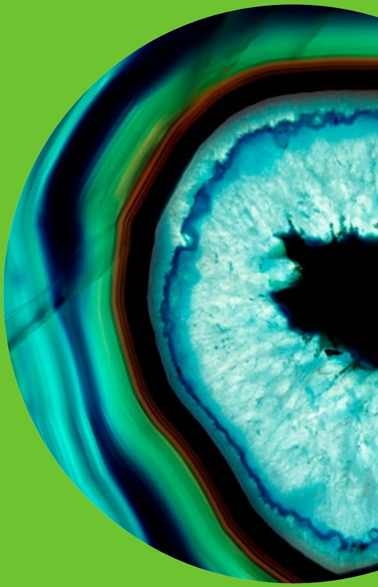
CHARTING THE FUTURE OF INNOVATION | #09·2016

ERICSSON TECHNOLOGY Review



CROSS-DOMAIN IDENTITY OF THINGS

CROSS-DOMAIN Identity OF THINGS

The rapid expansion of the Internet of Things (IoT) calls for a clearer common understanding of how identities function in the digital world. The numerous domains that make up the IoT result in single entities having multiple overlapping identities. In order to operate successfully in this environment, a company must be able to manage related identities across domains in an efficient manner. To do so, it needs to determine which cross-domain identity management solution best meets its own specific requirements.

THOMAS WEIDENFELLER, CLAUDIA BAUSCH

Identity is a concept used in fields ranging from philosophy to mathematics, with a variety of definitions. Even within the fields of ICT and IoT, the interpretation of the terms identity and identity management can vary widely, depending on the specific application and particular school of thought.

For many, identity management involves nothing more than giving a thing a traceable name or number, and perhaps adding a password or a public key certificate. For others, it means applying a consistent naming scheme or using a particular protocol to provide a computer with a host name, or a system user with a convenient sign-in experience.

According to ISO/IEC 24760-1:2011, an identity is "a set of attributes related to an entity", and an entity is defined as "an item ... that has a recognizably distinct existence" [1]. These definitions are very broad, and clearly cover more than just devices and people. For example, not only is an IoT device an entity according to this definition; all of its physical and virtual components are also entities, as are all of the actors that interact with them. The definition also covers parts and groups of such items, as long as they have a recognizably distinct existence.

From this perspective, even a small IoT device consists of many entities. Not every entity needs to have one or more identities, however; nor do all established identities need to be managed throughout the complete lifetime of the device. Defining the set of identities that need to be established, and working out how to manage them, are the result of decisions made on multiple levels at different times. For example, some identities are the result of design decisions about communication technologies or hardware component selection for a particular device.

It is also important to note that an identity does not necessarily have to be unique. For example, an identity can refer to a group of devices, such as in multicasting. An entity can also have – and typically has – more than one identity.

Entities, identities and domains

The application and validity of an identity tend to be finite, and are often dictated by technical limitations. For example, a private IP address has no global meaning; it only has meaning in a private network, and cannot be used on the internet. It is also possible to limit the applicability of an identity even further by design. The resulting domain of applicability describes where an identity may be used.

A car is an example of an entity that has multiple identities that are valid in different, partly overlapping, domains. A car receives its vehicle identification number (VIN) during the manufacturing process. The VIN is used by government agencies to track the car throughout its lifetime. The VIN's domain of applicability is typically limited to administrative purposes. However, at some point the vehicle will also receive a license plate number, which is used to identify it in public. Its domain of applicability is the public realm. Both the VIN and the license plate number identify the same entity: a particular vehicle. Both should be registered to the same owner. Depending on the type of operation to be performed, a particular one of the two identities or identifiers will be used. In some cases, both might be required. However, rarely can one identity be provided in place of the other.

> “THE APPLICATION AND VALIDITY OF AN IDENTITY TEND TO BE FINITE, AND ARE OFTEN DICTATED BY TECHNICAL LIMITATIONS”

Although they are separate, identities in different domains are related. In the car example, the relevant identity management systems (IDMSs) are designed to make it possible for government authorities to find out the license plate number from the VIN, and the VIN from the license plate number. When a license plate number is issued, identity management activities affect both domains to ensure traceability.

Understanding identity management

The term identity management is defined in ISO/IEC 24760–1:2011 as “the processes and policies involved in managing the lifecycle and values, type and optional metadata of attributes in identities known in a particular domain” [1].

The car example clearly illustrates that identity

Abbreviations

eUICC — embedded Universal Integrated Circuit Card | **GBA** — Generic Bootstrapping Architecture | **IDMS** — identity management system | **IIP** — identity information provider | **IoT** — Internet of Things | **LWM2M** — Lightweight M2M | **M2M** — machine-to-machine | **OAuth** — open standard for authorization | **OpenID** — open standard and decentralized authentication protocol | **SAML** — Security Assertion Markup Language | **SSO** — single sign-on | **UICC** — Universal Integrated Circuit Card | **VIN** — vehicle identification number management is not about managing the entity itself (that is, performing operations on the entity). Rather, it is about managing "a set of attributes related to an entity" – data that describes or identifies the entity. Identity management is fundamentally a security technique – not an entity management one. As such, identity management supports the identity-based decisions [1] that must be made to ensure security.

Typical identity-based decisions that are related to security include device authentication, controlling authorizations (typical authentication, authorization and accounting functions) and the categorization of data. For example, identity-based decisions can be used to ensure that the data returned by an IoT sensor (such as a temperature measurement) is associated with the correct entity (the machine from which the temperature was taken). In general, the routing of input and output data to and from an IoT device is based on identities.

The distinction between managing an entity and managing an entity's identity is important. Managing identities can have side effects that impact the entity, but won't necessarily. For instance, an attempt to manage an entity via identity management will at best be indirect, and at worst a complete failure.

For example, in geolocation applications, an entity's location might be one of its identities. The entity might even be addressed (identified) by its location. Performing a particular identity management activity could affect the location data attribute in the identity register. But this change would have no effect on the entity's actual position. In the best-case scenario, there would be additional mechanisms in place to take the identity management data and translate it into action that would in turn affect the entity itself, such as commanding it to move to the new location. This could work if the entity was a mobile machine, but would obviously fail if it were a factory building (the worst-case scenario).

The limitations of identity management are particularly significant for IoT devices. Identity management is no substitute for proper device management; rather, the two need to work in parallel. Device lifecycle changes must be supported by identity management activities.

The identity management lifecycle

Figure 1 provides an example of the lifecycle of an identity in terms of states and state transitions. This example is a modified version of the reference lifecycle model in ISO/IEC 24760–1:2011 [1]. Other lifecycle models may also be used, depending on the specific purpose of the particular identity. An IDMS supports the creation, provisioning, maintenance and decommissioning of identities throughout the lifecycle of a particular type of identity [1] following its lifecycle model. The lifecycle example in Figure 1 manages an identity within a specific domain. However, we know an entity can have more than one related identity within the same domain, or multiple identities spread over several domains. As a result, the requirements for a real-world IDMS extend beyond merely transitioning through the states for an identity.

The scenario involving multiple identities that is easiest to manage is when the related identities are within the same domain and under the control of the same authority. At the other end of the spectrum are scenarios in which the identities are in different domains, and are controlled by different authorities, and the relevant IDMSs are not able to communicate with each other.

Figure 2 illustrates the relationship between IDMS coupling and domains, and its impact on the relative difficulty of managing identity data. In cases where there is no communication between IDMSs, manual intervention and handling are mandatory. Such cases are therefore best avoided.

Cross-domain management architectures

Two common cross-domain identity management architectures are particularly relevant to IoT identity management. The first, shown in *Figure 3*, uses one IDMS for coordination, giving it special authority among its peers.

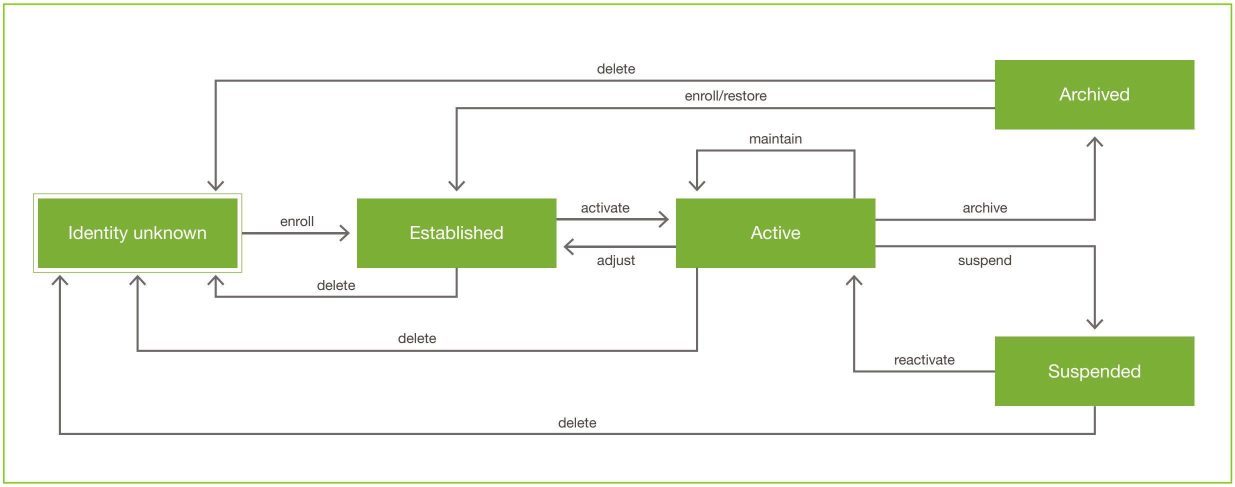


Figure 1 Example of an identity lifecycle

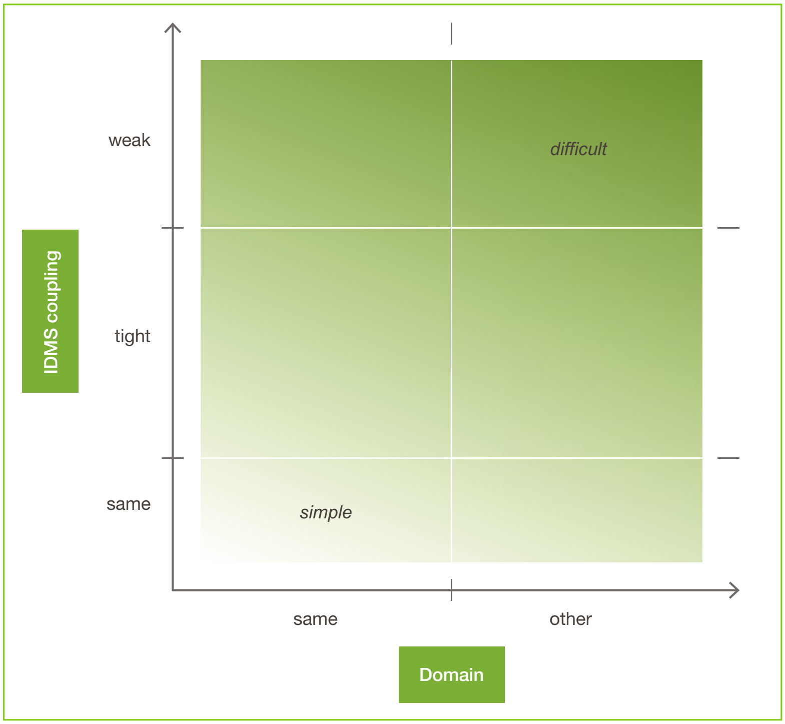


Figure 2 Relative difficulty of managing related identity data

The architecture shown in Figure 3 is similar to an architecture used in network management, in which individual element managers are each responsible for managing a particular network element, and a network management system coordinates network-wide issues above the element management layer. The architecture can be enhanced by adding hierarchy levels with intermediate coordinating IDMSs.

Figure 4 shows the second common architecture, in which the various IDMSs coordinate with other IDMSs on a peer-to-peer basis. Note that not every IDMS coordinates with every other IDMS; this depends on whether there is any need for them to coordinate, as well as technical or administrative limitations.

There are no hard and fast rules dictating which architecture is preferable. Other architectures also exist, including hybrid versions of the architectures presented in Figures 3 and 4. Practitioners need to consider their existing systems and any administrative barriers they may have, and make compromises, adapting their integrations to suit their particular circumstances. Ideally, they should establish one of the architecture options as the primary one and add diverging IDMS and management subsystems as satellite systems in isolated areas.

Techniques to build a coordinating system

There are technical and administrative issues to overcome when building a coordinating system. The technical issues begin with the communication layer. The individual IDMSs that should take part in cross-domain identity management as shown in Figures 3 and 4 need to communicate in some way – typically via the TCP/IP suite. When faced with legacy protocols on the network layer [2], an adaptation to IP should be considered. Which protocols to use on layers above the transport layer (particularly the application layer) is both a technical and an administrative decision.

Administration of cross-domain identity management includes the creation of an identity federation: "[an] agreement between two or more domains specifying how identity information will be exchanged and managed for cross-domain identification purposes." [1] The system that is subsequently built according to this agreement is typically also known as an identity federation.

Single sign-on identity federation

One highly sought-after feature when building identity federations – especially when humans are involved – is single sign-on (SSO). With SSO, the identity of an entity in one domain can be used for authentication of the same entity in another domain. The purpose of SSO is to avoid having to perform identity management in two or more domains in parallel. This is achieved by having fully automated protocols and processes in the identity federation agreement for handling the data processing and exchange between the domains.

Enterprise and cloud system architectures are good examples of how cryptography-based identity federations can be used to provide SSO services. SAML, OpenID and OAuth 2.0 (with or without additional application programming interfaces like OpenID Connect) are typical protocols used to build SSO identity federations for authentication or authorization purposes in this context. Essentially, these protocols are used to exchange trust in an identity – and by association, an entity or groups of entities – between domains.

For humans, SSO is a highly valued convenience feature that removes tasks like remembering user login credentials. But for non-human IoT entities, which connect to a rather limited number of services, the use of identities and identity-based decisions in IoT device communication does not necessarily require SSO.

A typical IoT device might, for example, make use of the following services:

- a network service that provides basic communication
- a device management service provided via the Lightweight M2M (LWM2M) management protocol [3]
- a service management service provided via LWM2M, either separate from or in cooperation with the device management service
- a payload or application service to which the IoT device delivers data and from which it receives application information.

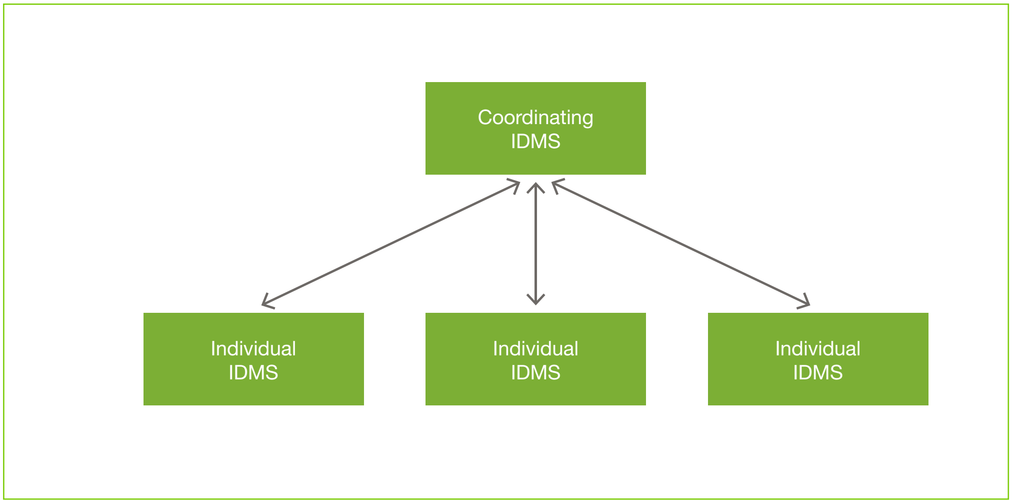


Figure 3 Centrally coordinated IDMSs

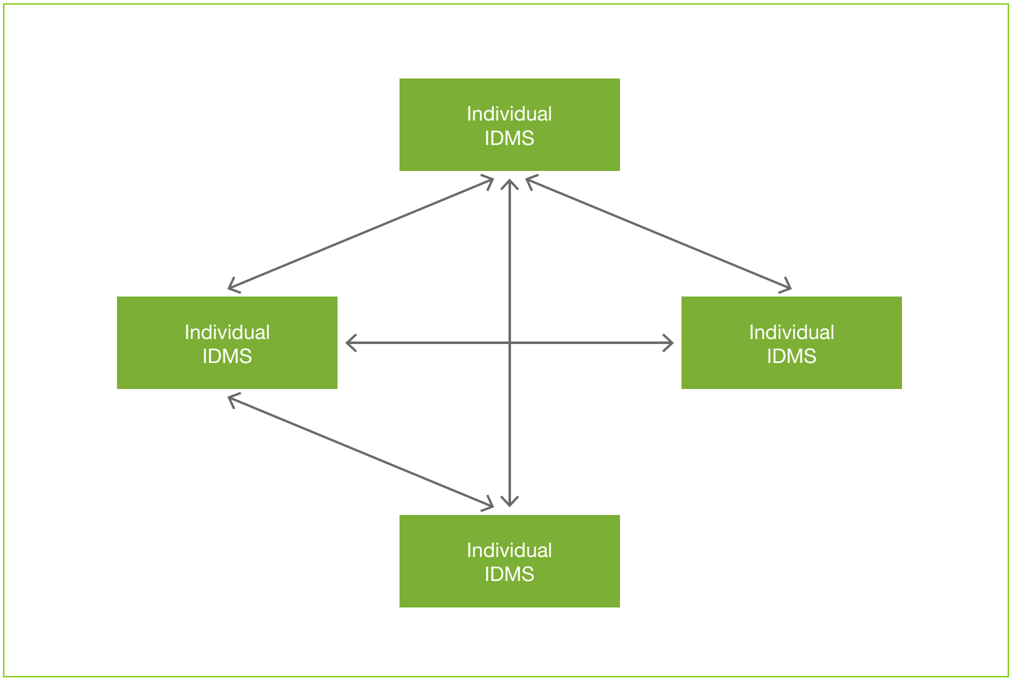


Figure 4 Peer-to-peer coordination

> THE 3GPP IDENTITY AND GBA ARE CURRENTLY ASSOCIATED WITH CELLULAR NETWORKS, [BUT] THIS TECHNOLOGY CAN ALSO BE USED FOR DEVICES CONNECTED TO A NETWORK USING OTHER, NON-3GPP TECHNOLOGIES

Since the number of services used is relatively static over the lifetime of the IoT device, and there is no human convenience advantage, an SSO-capable identity federation is not absolutely necessary in this type of case. In fact, for small IoT devices, the use of enterprise SSO protocols adds considerable overhead to the device firmware. When SSO is needed on an IoT device, lightweight SSO protocols should be considered instead.

The Generic Bootstrapping Architecture (GBA) [4] is a mobile network technology that makes it possible to reuse an identity from within the mobile network domain in other domains. Solutions based on the GBA architecture make use of mobile network subscribers' identities, associated cryptographic key material and cryptographic algorithms to establish a temporary, cryptographically-secured security association between an IoT device and a service in the application layer, for example. The security association can then be used for tasks such as authenticating the IoT device before granting access to the service. One promising realization of an identity management solution using GBA as a federation technique is a trial project for agricultural applications known as the Connected Vineyards project [5].

GBA uses well-known mobile network identity information providers (IIPS). A UICC/eUICCwith a SIM application suitable for GBA is used in the IoT device, while the corresponding identity information on the mobile network side is provided by the Home Location Register/Home Subscriber Server.

Note that, although the 3GPP identity and GBA are currently associated with cellular networks, this technology can also be used for devices connected to a network using other, non-3GPP technologies. The identity credential (shared secret) and associated software may in this case be protected by hardware-specific isolation and protection mechanisms to avoid the extra cost of (e)UICC in IoT devices. GBA can also be used to extend the federation beyond SSO – for example, to provide cryptographically derived, temporary pre-shared keys to secure communication.

It is important to recognize that setting up an identity federation, for SSO purposes or otherwise, requires effort. The need to manage identities in multiple domains is replaced with the need to manage the federation. More importantly, an identity federation requires trust. An enrollment in one domain affects all federated domains, which means that improper identity proofing in one domain creates a potential security risk in all federated domains. However, in some cases – such as GBA – a mobile network operator with an established track record of managing signup and access to network services is in a good position to provide the necessary trust.

Mapping

The SSO identity federation protocols presented above all rely on sound cryptographic principles. The original identity data, including passwords and cryptographic material, are not copied between the domains. Only the trust in some identity – an identity assertion [1] – is exchanged, enabling a federated domain to authenticate an entity, and, if desired, bootstrap its own cryptographic material. This is not the only way to build an identity federation, however.

Another common way to build an identity federation is by mapping. Identity data valid in one domain is mapped to some other identity data in another domain. The mapping can be 1:1 (the data is copied as is) or with some adaptations. For example, the mapping could include adding supplementary identity data, or adding an identifier as an attribute to one's own identity data.

One way to perform mapping is to synchronize at regular intervals. At certain points in time, the contents of two or more IIPS are compared with each other. Algorithms are then used to resolve any detected discrepancies and generate a consistent state across domains.

Tracking changes is another way to perform mapping. When this method is used, each of the state transitions shown in Figure 1 is communicated to the federated IDMSs. The IDMSs then map the received event data and add the result to their identity registers. A message bus is one possible software architecture that can be used for communication and exchange of events between the IDMSs.

Regardless of which of these two mapping methods is chosen, it is vital to address the issue of concurrent changes to the mapped data in the federated domains. This can be dealt with by considering one domain to be the master for particular identity data. That is, one domain always has precedence, or may even be the only domain in which the data is allowed to actively be changed. If this solution is not possible in a particular case, operational transformation techniques can be used to handle issues of concurrent changes, especially in the case of tracking changes. Three-way merge or differential synchronization are other techniques for resolving issues when tracking changes or synchronizing.

Identity management domains in the IoT

The selection criteria for identity domains in an IoT IDMS are largely technical, but they are also

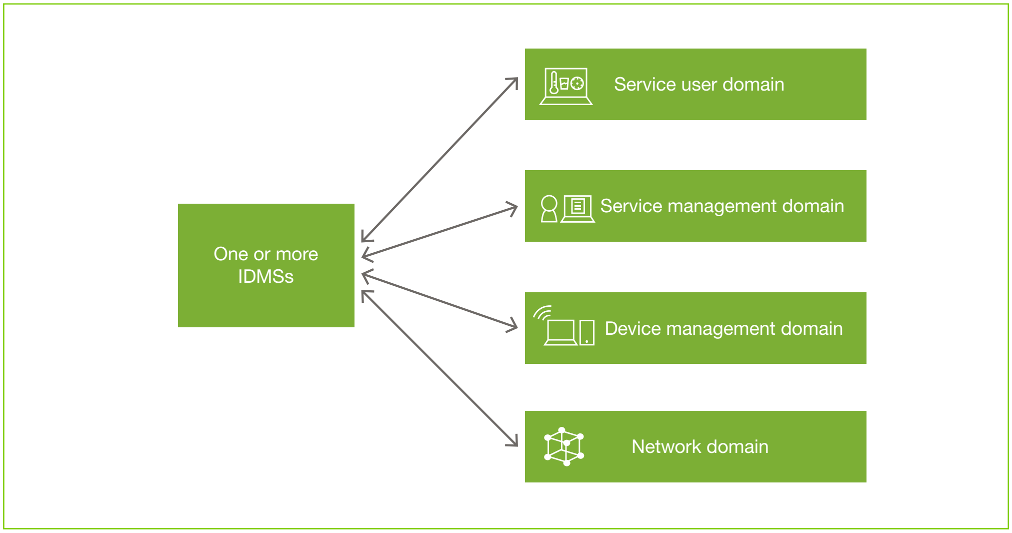


Figure 5 Four identity management domains in the IoT

influenced by organizational factors and sometimes even individual preferences. Domains can be quite small or rather broad, containing only a few or many different types of identity data.

Figure 5 illustrates four identity management domains that capture the technical and organizational properties of an IoT system at a high level:

- **service user domain** – where the IoT system is exploited for benefits. Services on top of the IoT device(s) are provided here. They supply a machine or a human with accumulated data and value-added services.
- **service management domain** – where the application(s) and/or service(s) running on the IoT device are managed, along with their association with enterprise application servers responsible for dealing with the payload data. A service delivery platform would work in this domain, for example.
- **device management domain** – where basic device functions are managed, including the device lifecycle and firmware (operating system). Services based on the LWM2M protocol would run here, for example.
- **network domain** – the "I" in IoT, where the communication happens, such as a cellular network or another type of WAN, or a LAN.

Identity management and security

There is another point that must be considered when coupling IDMSs to manage identities across domains. Identity management itself needs to be performed securely to fulfill the promise of helping to secure systems. It can only do so when identity management is performed in such a way that the managed identities are not compromised. For example, during enrollment, the right entity must be paired with the right identity. This is the most important aspect of this activity.

The basic security requirements for identity management are nearly identical to the security requirements of modern ICT systems. Both data at rest (storage) and data in motion (communication) need to be protected; and in each case, common ICT security techniques and technologies are relevant. This applies particularly to the exchange of identity information in identity federations, in those cases where identity data (such as access credentials) are simply copied or mapped from one domain to another.

There can be additional security requirements for identity management, depending on the particular domain or system, and on the system providers' level of commitment to offering a secure system. In general, the security of the management process and the security of the IDMS will have a direct impact on the trustworthiness of the managed identities.

Conclusion

With the spread of IoT systems to almost all areas of life, IoT security is set to become one of the most important technology development areas in the coming years [6]. IoT systems will need to be able to support large-scale field applications comprising a diversity of connected things. This will require massive enrollments of identities at an early stage of the device lifecycle, as well as the maintenance of those identities throughout the devices' lifetimes. The use of technologies like GBA and specific identity management systems for the IoT will substantially reduce the complexity of these activities.

It is clear that identity management systems – based on sound identity principles and intra-domain identity lifecycle models – have an important role to play in ensuring IoT security. Due to the heterogeneous setup of IoT end-to-end solutions, an IDMS that can only support one domain is not adequate for the complete identity management of IoT devices. Devices that must be identified in multiple domains need to have their identities managed across them. There are several ways to achieve this, depending on the systems and technologies available, and the relationship between the domains and the domain-specific identity data. ✱

References

1. **International Organization for Standardization, ISO/IEC 24760-1:2011, Information technology – Security techniques – A framework for identity management – Part 1: Terminology and concepts, available at:** *http://standards.iso.org/ittf/PubliclyAvailableStandards/c057914_ISO_IEC_24760-1_2011.zip*
2. **International Organization for Standardization, ISO/IEC 7498-1:1994, Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, available at:** *http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip*
3. **Open Mobile Alliance, Lightweight Machine to Machine Technical Specification. OMA-TS-LightweightM2M-V1_0-20160407-D. Draft Version 1.0. 07 April 2016, available at:** *http://member.openmobilealliance.org/ftp/Public_documents/DM/LightweightM2M/Permanent_documents/OMA-TS-LightweightM2M-V1_0-20160407-D.zip*
4. **3GPP, Generic Bootstrapping Architecture (GBA). 3GPP TS 33.220, available at:** *https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2280*
5. **Ericsson, Connected Vineyards, available at:** *http://www.ericsson.com/res/docs/2015/iot-connected-vineyards.pdf*
6. **Gartner, Gartner Identifies the Top 10 Internet of Things Technologies for 2017 and 2018, available at:** *http://www.gartner.com/newsroom/id/3221818*

THE AUTHORS

Thomas Weidenfeller

◆ is a master systems designer at Portfolio & Systems within Customer Group Industry & Society. He has more than 20 years of experience at Ericsson, starting in telecommunication management systems. Over the years, he has worked in such diverse areas as software design, systems management, mobile packet backbone design and software architectures. He is currently working on IoT security issues. He holds a degree in electrical engineering from the Cologne University of Applied Sciences (now called the Technical University of Cologne), Germany.

Claudia Bausch

◆ joined Ericsson in 1998. She holds a degree in computer science from RWTH Aachen University, Germany. Her expertise covers several areas of software design, configuration management and project management. She is currently working as senior systems designer at Portfolio & Systems on IoT studies and end-to-end solutions within the Customer Group Industry & Society.

The authors would like to acknowledge the support and inspiration they received from their colleagues Per Ståhl, Patrik Teppo, Dhruvin Patel and Gustavo Tanoni.

ISSN 0014-0171
284 23-3293 | Uen

Ericsson
SE-164 83 Stockholm, Sweden
Phone: +46 10 719 0000