The GS1 EPCglobal Architecture Framework

GS1 Version 1.6 dated 14 April 2014

Authors:

Ken Traub (Ken Traub Consulting LLC) kt@kentraub.com, Editor

Felice Armenio (Johnson & Johnson) FArmeni@NCSUS.JNJ.com

Henri Barthel (GS1) henri.barthel@gs1.org

Paul Dietrich (Impinj) paul.dietrich@impinj.com

John Duker (Procter & Gamble) duker.jp@pg.com

Christian Floerkemeier (MIT) floerkem@MIT.EDU

John Garrett (TESCO) john.c.garrett@uk.tesco.com

Mark Harrison (University of Cambridge) mark.harrison@cantab.net

Bernie Hogan (GS1 US) bhogan@gs1us.org

Jin Mitsugi (Keio University) mitsugi@sfc.wide.ad.jp

Josef Preishuber-Pfluegl (CISC Semiconductor) j.preishuber-pfluegl@cisc.at

Oleg Ryaboy (CVS) ORyaboy@cvs.com

Sanjay Sarma (MIT) sesarma@mit.edu

KK Suen (GS1 Hong Kong) kksuen@gs1hk.org

John Williams (MIT) jrw@mit.edu

Abstract

This document defines and describes the GS1 EPCglobal Architecture Framework. EPCglobal is an activity of the global not-for-profit standards organization GS1, and supports the global adoption of the Electronic Product Code (EPC) and related industry-driven standards to enable accurate, immediate and cost-effective visibility of information throughout the supply chain   The GS1 EPCglobal Architecture Framework is a collection of hardware, software, and data standards, together with shared network services that can be operated by GS1, its delegates or third party providers in the marketplace, all in service of this common goal. This document has several aims:

- To enumerate, at a high level, each of the hardware, software, and data standards that are part of the GS1 EPCglobal Architecture Framework and show how they are related.
- To define the top level architecture of shared network services that are operated by GS1, its delegates, and others.
- To explain the underlying principles that have guided the design of individual standards and service components within the GS1 EPCglobal Architecture Framework.
- To provide architectural guidance to end users and technology vendors seeking to implement GS1 EPCglobal standards and to use EPC Network Services.

This document exists only to describe the overall architecture, showing how the different components fit together to form a cohesive whole. It is the responsibility of other documents to provide the technical detail required to implement any part of the EPCglobal Architecture Framework.

Audience for this document

The audience for this document includes:

- Hardware developers working in the areas of developing EPC tags and EPC-enabled systems and appliances, including devices to read and write tag data.
- Software developers working in the areas of developing EPC middleware and business applications that use, create, store and/or share EPC-related information.
- Enterprise architects and systems integrators that integrate EPC-related processes and applications into enterprise architectures.
- Participants of GSMP Working Groups working on defining requirements and developing EPCglobal standards.
- Industry groups, governing organizations, and companies that are developing or overseeing business processes that rely on EPC technology.

- Members of the general public who are interested in understanding the principles and terminology of the EPCglobal Architecture Framework

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained by GS1. See `www.gs1.org` for more information.

This document is a GS1 approved document and is available to the general public.

Comments on this document should be sent to the GS1 Architecture Group mailing list `gs1ag@community.gs1.org`.

Table of Contents

1 Introduction

This document defines and describes the GS1 EPCglobal Architecture Framework (hereafter simply the ñEPCglobal Architecture Frameworkò). EPCglobal is an activity of the global not-for-profit standards organization GS1, and supports the global adoption of the Electronic Product Code (EPC) and related industry-driven standards to enable accurate, immediate and cost-effective visibility of information throughout the supply chain The EPCglobal Architecture Framework is a collection of interrelated hardware, software, and data standards (ñEPCglobal Standardsò), together with shared network services that are operated by GS1, its delegates, and others (ñEPC Network Servicesò), all in service of this common goal.

The primary beneficiaries of the EPCglobal Architecture Framework are End Users and Solution Providers. An End User is any organization that employs EPCglobal Standards and EPC Network Services as a part of its business operations. A Solution Provider is an organization that implements for End Users systems that use EPCglobal Standards and EPC Network Services. EPCglobal standards are available for use to any party, regardless of whether that party is a member of GS1. Informally, the synergistic effect of End Users and Solution Providers interacting with each other using elements of the EPCglobal Architecture Framework is sometimes called the ñEPCglobal Network,ò but this is more of an informal marketing term rather than the name of an actual network or system.

The EPCglobal Architecture Framework is the product of the GS1 Community, which not only includes GS1 members, but also includes the Auto-ID Labs, the GS1 Global Office, the GS1 Member Organizations, and government agencies and non-governmental organizations (NGOs), along with invited experts.

This document has several aims:

- To enumerate, at a high level, each of the hardware, software, and data standards that are part of the EPCglobal Architecture Framework and show how they are related. These standards are implemented by hardware and software systems, including components deployed by individual End Users as well as EPC Network Services deployed by EPCglobal, its delegates, and others.
- To define the top level architecture of EPC Network Services, which provide common services to all End Users, through interfaces defined as part of the EPCglobal Architecture Framework.
- To explain the underlying principles that have guided the design of individual standards and service components within the EPCglobal Architecture Framework. These underlying principles provide unity across all elements of the EPCglobal Architecture Framework, and provide guidance for the development of future standards and new services.
- To provide architectural guidance to end users and solution providers seeking to implement EPCglobal Standards and to use EPC Network Services, and to set expectations as to how these elements will function.

This document exists only to describe the overall architecture, showing how the different components fit together to form a cohesive whole. It is the responsibility of other documents to provide the technical detail required to implement any part of the EPCglobal Architecture Framework. Specifically:

- Individual hardware, software, and data interfaces are defined normatively by EPCglobal standards, or by standards produced by other standards bodies. EPCglobal standards are normative, and implementations are subject to conformance and certification requirements.

  An example of an interface is the radio-frequency communications protocol by which a Radio Frequency Identification (RFID) tag and an RFID reader device may interact. This interface is defined normatively by the UHF Class 1 Gen 2 Tag Air Interface Standard.

- The design of hardware and software components that implement EPCglobal standards are proprietary to the solution providers and end users that create such components. While EPCglobal standards provide normative guidance as to the behavior of interfaces between components, implementers are free to innovate in the design of components so long as they correctly implement the interface standards.

  An example of a component is an RFID tag that is the product of a specific tag manufacturer. This tag may comply with the UHF Class 1 Gen 2 Tag Air Interface Standard.

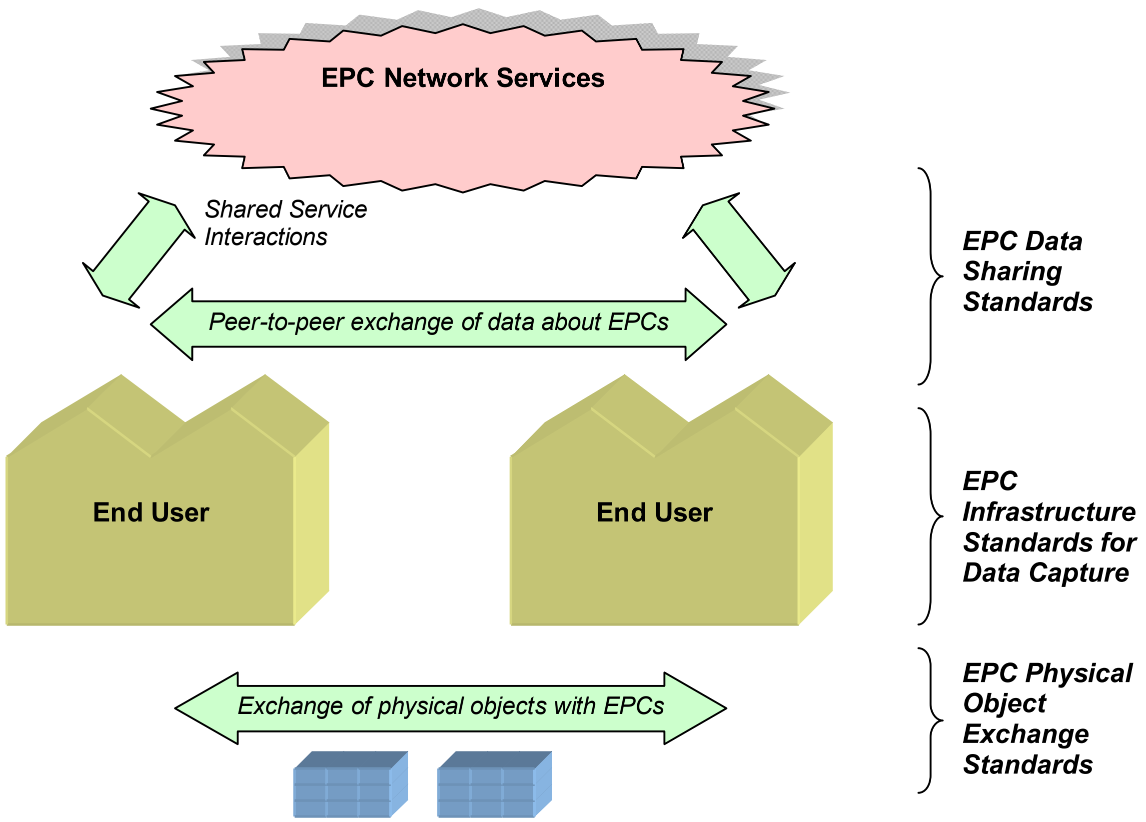
- A special case of components that implement EPCglobal standards are shared network services that are operated and deployed by EPCglobal itself (or by other organizations to which EPCglobal delegates responsibility), or by other third parties. These components are referred to as EPC Network Services, and provide services to all End Users.

  An example of an EPC Network Service is the Object Name Service (ONS), which provides a logically centralized registry through which an EPC may be associated with information services. The ONS is logically operated by GS1; from a deployment perspective this responsibility is delegated to a contractor of GS1 that operates the ONS ñrootò service, which in turn delegates responsibility for certain lookup operations to services operated by other organizations.

EPCglobal standards are a subset of the GS1 System, which includes all standards created by the GS1 Community through the GS1 Global Standards Management Process (GSMP). This document focuses on the relationships between EPCglobal standards. For an understanding of how EPCglobal standards fit into the larger universe of the GS1 System, please see the GS1 System Architecture [GS1SA] and GS1 System Landscape [GS1SL].

2 Architecture Framework Overview

The diagram below illustrates the activities carried out by End Users and the role that components of EPCglobal Architecture Framework play in facilitating those activities.



2.1 Architecture Framework Activities

In the diagram above, there are three broad activities illustrated, each supported by a group of standards within the EPCglobal Architecture Framework:

- *EPC Physical Object Exchange* End Users exchange physical objects that are identified with Electronic Product Codes (EPCs). For many End users, the physical objects are trade goods, the end users are parties in a supply chain for those goods, and physical object exchange consists of such operations as shipping, receiving, and so on. There are many other uses, like library or asset management applications that differ from this trade goods model, but still involve the unique identification and tagging of objects. The EPCglobal Architecture Framework defines EPC physical object exchange standards, designed to ensure that when one end user delivers a physical object to another end user, the latter will be able to determine the EPC of the physical object and interpret it properly.
- *EPC Data Sharing* End Users benefit from the EPCglobal Architecture Framework by sharing data with each other, increasing the visibility they have with respect to the movement of physical objects outside their four walls. The EPCglobal Architecture Framework defines EPC data sharing standards, which provide a means for end users to share data about EPCs within defined user groups or with the general public, and which also provide access to EPC Network Services and other shared services that facilitate this sharing.

- *EPC Infrastructure for Data Capture* In order to have EPC data to share, each end user carries out operations within its four walls that create EPCs for new objects, follow the movements of objects by sensing their EPCs, and gather that information into systems of record within the organization. The EPCglobal Architecture Framework defines interface standards for the major infrastructure components required to gather and record EPC data, thus allowing end users to build their internal systems using interoperable components.

This division of activities is helpful in understanding the overall organization and scope of the EPCglobal Architecture Framework, but should not be considered as extremely rigid. While in many cases, the first two categories refer to cross-enterprise interactions while the third category describes intra-enterprise operations, this is not always true. For example, an organization may use EPCs to track the movement of purely internal assets, in which case it will apply the physical object exchange standards in a situation where there is no actual cross-enterprise exchange. Conversely, an enterprise may outsource some of its internal operations so that the infrastructure standards end up being applied across company boundaries. The EPCglobal Architecture Framework has been designed to give End Users a wide range of options in applying the standards to suit the needs of their particular business operations.

2.2 Architecture Framework Standards

The following table summarizes all standards within the EPCglobal Architecture Framework in terms of the three activities described in the preceding section. A fuller description of each standard is given in Section 9. This table is intended mainly as an index of all current components of the EPCglobal Architecture Framework, not a roadmap for future work.

Activity	Standard	Status	Reference
Object Exchange	UHF Class 1 Gen 2 Tag Air Interface v1.1.0	Ratified	[UHFC1G21.1.0]
	UHF Class 1 Gen 2 Tag Air Interface v1.2.0	Ratified	[UHFC1G21.2.0]
	UHF Gen 2 Tag Air Interface v2.0.0	Ratified	[UHFC1V2]
	HF Class 1 Tag Air Interface	Ratified	[HFC1]
Object Exchange / Data Capture Infrastructure	EPC Tag Data Standard	Ratified	[TDS1.8]
Data Capture Infrastructure	Low Level Reader Protocol	Ratified	[LLRP1.1]
	Reader Management	Ratified	[RM1.0.1]
	Discovery, Configuration, and Initialization (DCI) for Reader Operations	Ratified	[DCI]

	Tag Data Translation	Ratified	[TDT1.6]
	Application Level Events (ALE)	Ratified	[ALE1.1.1]
	EPCIS Capture Interface	Ratified	[EPCIS1.0.1]
	EPCIS Data Standard	Ratified	[EPCIS1.0.1]
Data Sharing	Core Business Vocabulary	Ratified	[CBV1.0]
	EPCIS Query Interface	Ratified	[EPCIS1.0.1]
	Pedigree Standard	Ratified	[Pedigree1.0]
	EPCglobal Certificate Profile	Ratified	[Cert2.0]
	ONS	Ratified	[ONS2.0.1]
	Discovery Services	In Development	(none)

Notes for the ñStatusò column of the table above:

1. ñRatifiedò indicates a ratified EPCglobal standard.
2. ñIn developmentò indicates a standard whose development has been chartered and is underway within the GS1 standards development process

In the table above, the EPCIS Data Standard is shown as spanning the categories of infrastructure standard and data sharing standard. Likewise, the EPC Tag Data Standard is shown spanning the categories of object exchange standard and infrastructure standard, though in fact it also spans the data sharing category.

3 Goals for the EPCglobal Architecture Framework

This section outlines high-level goals for the EPCglobal Architecture Framework in terms of the benefits provided to End Users.

3.1 The Role of Standards

EPCglobal standards are created to further the following objectives:

- *To facilitate the sharing of information and physical objects between trading partners.*

  For trading partners to share information, they must have prior agreement as to the structure and meaning of data to be shared, and the mechanisms by which exchange will be carried out. EPCglobal standards include data standards and information sharing standards that form the basis of cross-enterprise sharing. Likewise, for trading partners to exchange physical objects, they must have prior agreement as to how physical objects will carry Electronic Product Codes in a mutually understandable way. EPCglobal standards include standards for RFID devices and data standards governing the encoding of EPCs on those devices.

- *To foster the existence of a competitive marketplace for system components.*

  EPCglobal standards define interfaces between system components that facilitate interoperability from components produced by different vendors (or in house). This in turn provides choice to end users, both in implementing systems that will share information between trading partners, and systems that are used entirely within four walls.

- *To encourage innovation*

  EPCglobal standards define *interfaces*, not *implementations*. Implementers are encouraged to innovate in the products and systems they create, while interface standards ensure interoperability between competing systems.

3.2 Global Standards

GS1 is committed to the creation and use of end user driven, royalty-free, global standards. This approach ensures that the EPCglobal Architecture Framework will work anywhere in the world and provides incentives for Solution Providers to support the framework. EPCglobal standards are developed for global use. GS1 is committed to making use of existing global standards when appropriate, and GS1 works with recognized global standards organizations to incorporate standards created within GS1.

3.3 Open System

The EPCglobal Architecture Framework is described in an open and vendor neutral manner. All interfaces between architectural components are specified in open standards, developed by the GS1 Community through the GS1 Global Standards Management Process or an equivalent process within another standards organization. The Intellectual Property policy of GS1 is designed to secure free and open rights to implement GS1/EPCglobal Standards in the context of conforming systems, to the extent possible.

3.4 Platform Independence

The EPCglobal Architecture Framework can be implemented on heterogeneous software and hardware platforms. The standards are platform independent meaning that the structure and semantics of data in an abstract sense is specified separately from the concrete details of data access services and bindings to particular interface protocols. Where possible, interfaces are specified using platform and programming language neutral technology (e.g., XML, SOAP messaging [SOAP1.2], and so forth).

3.5 Scalability and Extensibility

The EPCglobal Architecture Framework is designed to scale to meet the needs of each End User, from a minimal pilot implementation conducted entirely within an end-userǿs four walls, to a global implementation across many companies and many continents. The standards provide a core set of data types and operations, but also provide several means whereby the core set may be extended for purposes specific to a given industry or application area. Extensions not only provide for proprietary requirements to be addressed in a way that leverages as much of the standard framework as possible, but also provides a natural path for the standards to evolve and grow over time.

3.6 Data Ownership

The EPCglobal Architecture Framework is concerned with collecting information from a single company or across multiple companies, and making it available to those parties that have an interest in the data and are authorized to receive it. A fundamental principle is that each End User that captures data owns that data, and has full control over what other parties have access to that data.

In particular, the EPCglobal Architecture Framework does *not* presuppose that End Users will deliver their data to some shared database operated by a single third party. Instead, each End User that generates data may keep their data and only share them with whom they choose. An End User may choose to deliver the data to a shared third party database if that is the most effective way to achieve that End User's business goals, but an End User may choose instead to retain its data and share them with other parties on a point-to-point basis. ONS and Discovery Services (Section 7) are designed to help End Users find the data they need wherever it exists.

3.7 Security

For operations inside and outside a company's four walls, the EPCglobal Architecture Framework promotes environments with security precautions that appropriately address risks and protect valuable assets and information. Security features are either built into the standards, or use of an industry best security practice that is in accordance with this framework is recommended.

See Section 10 for an overview of data protection methods of current and evolving standards within the architecture framework.

3.8 Privacy

The EPCglobal Architecture Framework is designed to accommodate the needs of both individuals and corporations to protect confidential and private information. While many parties may ultimately be willing to give up some privacy in return for getting information or other benefits, all of them demand the right to control that decision. The EPCglobal Public Policy Steering Committee (PPSC) is responsible for creating and maintaining the EPCglobal Privacy Policy; readers should refer to PPSC documents for more information.

3.9 Open, Community Process

The GS1 Global Standards Management Process is designed to yield standards that are relevant and beneficial to end users. Important aspects of the process include:

- End user involvement in developing requirements through the Industry User Groups and Requirements Development Groups.

- Open process in which all GS1 Community members having relevant expertise are encouraged to join Standards Development Groups that create new standards.
- Several review milestones in which new standards are vetted by a wide community before final adoption.

4 Underlying Technical Principles

This section explains the design principles that underlie all parts of the EPCglobal Architecture Framework. Working Groups should take these principles into account as they develop new standards.

4.1 Unique Identity

A fundamental principle of the EPCglobal Architecture Framework is the assignment of a unique identity to physical objects, loads, locations, assets, and other entities whose use is to be tracked.[1] By ñunique identityò is simply meant a name, such that the name assigned to one entity is different than the name assigned to another entity. In the EPCglobal Architecture Framework, the unique identity is the Electronic Product Code, defined by the EPCglobal Tag Data Standard [TDS1.8].

Unique identity within the EPCglobal Architecture Framework, as embodied in the Electronic Product Code, has these characteristics:

- *Uniqueness/Serialization* The EPC assigned to one entity is different than the EPC assigned to another (but see below for exceptions). This implies that all EPC-identified entities are *serialized*; that is, they carry a unique serial number as part of the EPC.
- *Universality* EPCs comprise a single space of identifiers that can be used to identify any entity, regardless of what kind of entity it is. An EPC for an entity is globally unique across all types of entities.
- *Compatibility* EPC identifiers are designed to be compatible with existing naming systems. In particular, for every GS1 key that names a unique entity instance (as opposed to a class of entities), there is a corresponding EPC. This provides compatibility and interoperability with systems based on GS1 keys.
- *Federation* The EPC is not a single naming structure, but a federation of several naming structures. This allows existing naming structures to be incorporated into the EPC system, so that the property of universality (above) is achieved, while maintaining compatibility with existing naming structures. This attribute is extremely important to ensure wide adoption of the EPC, which would be significantly more difficult if adoption required adoption of a single naming structure.

[1] Some GS1 keys that have corresponding EPCs, particularly the GDTI and GSRN, may be used both for physical objects and for non-physical entities. The applicability of EPC standards to non-physical entities is not yet fully addressed in the EPCglobal architecture framework.

For example, both GS1 SSCC keys and GS1 GIAI keys also correspond to valid EPCs. The various concrete representations of the EPC use a system of headers (textual or binary according to the representation) to distinguish one identity scheme from another; when one EPC is compared to another, the header is always included so that EPCs drawn from different schemes will always be considered distinct. The header is always considered to be a part of the EPC, not something separate.

While the EPC is designed to federate multiple naming structures, there may be performance tradeoffs, especially with respect to RFID tag performance, when multiple naming structures are used in the same business context. For this reason, there is motivation to minimize the number of distinct naming structures used within any given industry.

- *Extensibility* The mechanisms for federating naming structures within the EPC are extensible, so that additional naming structures may be incorporated into the EPC system without invalidating existing EPCs or the GS1 system.
- *Representation independence* EPCs are defined in terms of abstract structure, which has several concrete realizations. Especially important are the binary realization that is used on RFID tags and the Universal Resource Identifier (URI) realization that is used for data sharing. Formal conversion rules exist [TDS1.8], and the Tag Data Translation Standard [TDT1.6] provides a machine-readable form of these rules.
- *Decentralized assignment* EPCs are designed so that independent organizations can assign new EPCs without the possibility of collision. This is done through a hierarchical scheme, not unlike the Internet Domain Name System though somewhat more structured. GS1 acts as the Registration Authority for the overall EPC namespace. Each naming structure that is federated within the EPC namespace has a space of codes managed by an Issuing Agency. For the EPC naming structures based on the GS1 family of keys (SGTIN, SSCC, etc, are examples of such EPC naming structures), GS1 is the Issuing Agency. An Issuing Agency allocates a portion of the EPC space to another organization, who then becomes the Issuing Organization for that block of EPCs. For GS1 keys, for example, this is done by assigning a GS1 Company Prefix to another organization, often an end user but sometimes another organization such as a GS1 Member Organization. The Issuing Organization is then free to assign EPCs within its allocated portion without any further coordination with any outside agency. (Since there are several EPC naming structures based on GS1 keys, assigning a single Company Prefix has the effect of allocating several blocks of EPCs to an Issuing Organization, one block within each GS1 coding scheme.)
- *Structure* EPCs are not purely random strings, but rather have a certain amount of internal structure in the form of designated fields. This plays a role in decentralization, as described above. More significantly, the EPCôs internal structure is essential to the scalability of lookup services such as the Object Name Service which exploit the structure of EPCs to distribute lookup processing across a scalable network of services.
- *Light Weight* EPCs have just enough structure and information to accomplish the goals above, and no more. Other information associated with EPC-bearing entities is

not encoded into the EPC itself, but rather associated with the EPC through other means.

While EPCs are intended to be globally unique in most situations, there are some varieties of EPCs that are not. In particular, a portion of EPC space may be derived from an existing coding scheme for which global uniqueness is not guaranteed. In that situation, the EPCs from that space have uniqueness guarantees which are no stronger than the original scheme. For example, GS1 SSCC keys are not unique over all time and space, but due to the limited size of the SSCC namespace they are recycled periodically. Good practice dictates that SSCCs be recycled no more frequently than the lifetime of loads within the supply chain to which the SSCCs are affixed (plus a reasonable data retention period). This eliminates the possibility that two identical SSCCs would be present on two different loads at the same time, but it might still be possible to find identical SSCCs for different loads in a long-term historical database. Applications that rely on uniqueness properties of EPCs must understand the properties of the various EPC namespaces that they might encounter, and act accordingly.

In other instances, what appears to be a single physical entity may have more than one identity, and therefore more than one EPC. A typical example is a palletized load that sits on a reusable pallet skid. In this example, there might be one EPC denoting the load, and another EPC denoting the reusable skid. (In the GS1 system, the load including the pallet skid might be given an SSCC, while the skid by itself might be given a GRAI.) During the lifetime of the palletized load these two EPCs appear to be associated with the same physical entity, but when the load is broken down the load EPC is decommissioned, while the pallet skid EPC continues to live as long as the pallet is reused. In this example, what appears to be one physical entity really consists of two separate entities from a business perspective (the pallet and the load), and so what appears to be multiple EPCs assigned to the same object is really a separate EPC for each entity.

4.1.1 Uniqueness Considerations for “Closed” Systems

It is sometimes believed that global uniqueness is not required or is prohibitively expensive when EPC technology is used for ñclosedò systems, such as proprietary use within a single company. Closer analysis suggests that this is not so, as explained below.

At the level of information systems (e.g., at the level of EPCIS), the cost of achieving global uniqueness for identifiers is extremely low, and so it is recommended even for closed systems. EPC standards use Internet Uniform Resource Identifiers (URIs) as the standard syntax for unique identifiers, and the EPC Tag Data Standard provides a URI form for Electronic Product Codes in accordance with this principle. URIs are a widely adopted mechanism for construction of globally unique identifiers, and may be used even in applications that do not use EPCs.

When RFID tags are used in a ñclosedò system, the motivation for using globally unique identifiers such as EPCs is even more significant. RFID tags communicate without line of sight from relatively long distances. It is projected that RFID/EPC technology will have substantial consumer use, proliferating the numbers of RFID tags ñin the wild.ò For these reasons, a truly ñclosedò system is in most cases not realistically achievable when RFID tags are used. If non-unique identifiers are used in RFID applications, those applications may fail to operate properly, and they may cause other applications to fail. RFID tags containing globally unique EPCs from standards-based open system will enter into closed systems, causing conflicts if those closed systems inappropriately occupy identifier space defined by standards. RFID tags containing identifiers from closed systems will enter into standards-based open systems, causing conflicts in the same way. RFID tags from one closed system will enter into other closed systems, causing conflicts if those systems happen to have chosen identical or overlapping ranges of supposed ñprivate useò identifiers.

This last example of RFID tags crossing from one closed system to another is the largest cause of concern. For example, an IT asset-tagging system with a proprietary identifier format operates properly until a second proprietary system for document tracking from another vendor, which happens to use the same ñprivate useò identifiers, is installed. Since there is no coordination between the two systems, the two systems could fail to operate in overt or subtle ways. Such issues are difficult to resolve as there is no common format among the proprietary systems or vendors to troubleshoot and coordinate the changes necessary to ensure uniqueness.

In short, there is no such thing as a ñclosedò system involving RFID tags; any RFID application must consider the possibility that tags from ñoutsideò the system may enter.

The hierarchical encoding structure within the EPC Tag Data Standard provides a globally unique identifier space for both open and closed RFID systems. The most practical method available today to assure proper operation of any system, open or ñclosed,ò is to obtain a block of EPC capacity (e.g., by obtaining a GS1 Company Prefix) and use one of the formats defined in the EPC Tag Data Standard.

4.1.2 Use of the Electronic Product Code

The Electronic Product Code is designed to facilitate business processes and applications that need to manipulate visibility data ï data about observations of physical objects. The EPC is a universal identifier that provides a unique identity for any physical object. The EPC is designed to be unique across all physical objects in the world, over all time, and across all categories of physical objects. (Though see Section 4.1, above, for situations in which an EPC may not be unique over all time.) It is expressly intended for use by business applications that need to track all categories of physical objects, whatever they may be.

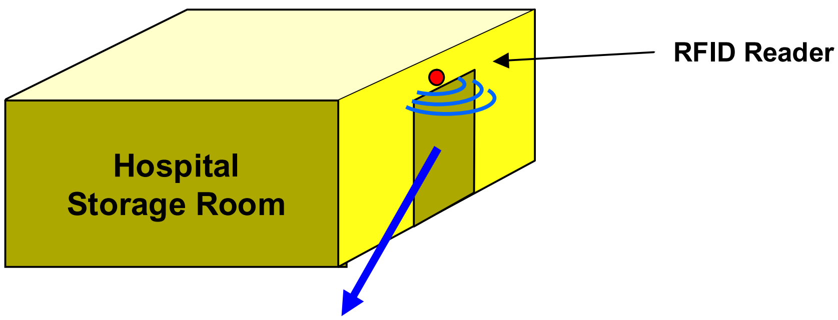
By contrast, some GS1 identification keys defined in the GS1 General Specifications [GS1GS] can identify categories of objects (GTIN), unique objects (SSCC, GLN, GIAI, GSRN), or a hybrid (GRAI, GTDI) that may identify either categories or unique objects depending on the absence or presence of a serial number. The GTIN, as the only category identification key, requires a separate serial number to uniquely identify an object but that serial number is not considered part of the identification key.

There is a well-defined correspondence between EPCs and GS1 keys. This allows any physical object that is already identified by a GS1 key to be used in an EPC context where any category of physical object may be observed. Likewise, it allows EPC data captured in a broad visibility context to be correlated with other business data that is specific to the category of object involved and which uses GS1 keys.

The remainder of this section elaborates on these points.

4.1.3 The Need for a Universal Identifier: an Example

The following example illustrates how visibility data arises, and the role the EPC plays as a unique identifier for any physical object. In this example, there is a storage room in a hospital that holds radioactive samples, among other things. The hospital safety officer needs to track what things have been in the storage room and for how long, in order to ensure that exposure is kept within acceptable limits. Each physical object that might enter the storage room is given a unique Electronic Product Code, which is encoded onto an RFID Tag affixed to the object. An RFID reader positioned at the storage room door generates visibility data as objects enter and exit the room, as illustrated below.



Visibility Data Stream at Storage Room Entrance			
Time	**In / Out**	**EPC**	**Comment**
8:23am	In	`urn:epc:id:sgtin:0614141.012345.62852`	10cc Syringe #62852 (trade item)
8:52am	In	`urn:epc:id:grai:0614141.54321.2528`	Pharma Tote #2528 (reusable transport)
8:59am	In	`urn:epc:id:sgtin:0614141.012345.1542`	10cc Syringe #1542 (trade item)
9:02am	Out	`urn:epc:id:giai:0614141.17320508`	Infusion Pump #52 (fixed asset)
9:32am	In	`urn:epc:id:gsrn:0614141.0000010253`	Nurse Jones (service relation)
9:42am	Out	`urn:epc:id:gsrn:0614141.0000010253`	Nurse Jones (service relation)
9:52am	In	`urn:epc:id:gdti:0614141.00001.1618034`	Patient Smith's chart (document)

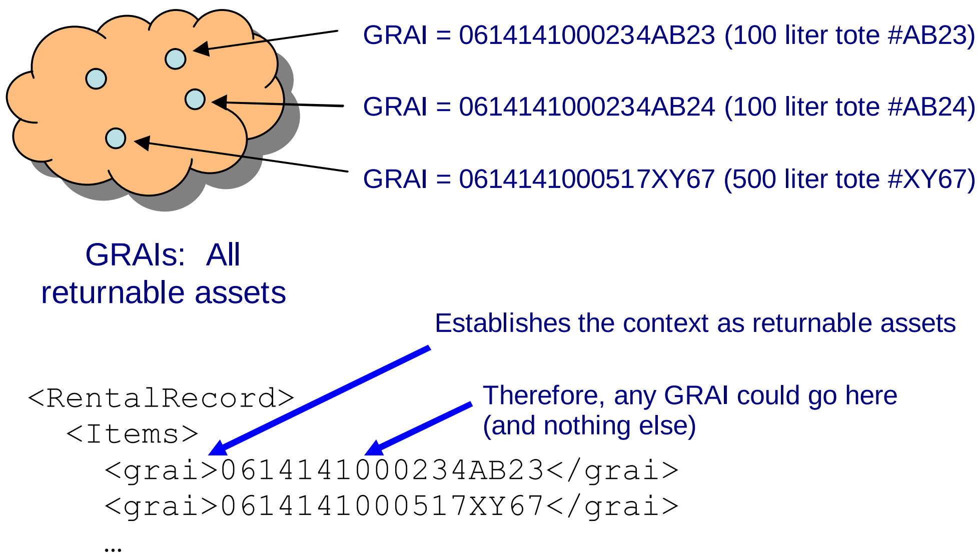
As the illustration shows, the data stream of interest to the safety officer is a series of events, each identifying a specific physical object and when it entered or exited the room. The unique EPC for each object is an identifier that may be used to drive the business process. In this example, the EPC (in Pure Identity EPC URI form) would be a primary key of a database that tracks the accumulated exposure for each physical object; each entry/exit event pair for a given object would be used to update the accumulated exposure database.

This example illustrates how the EPC is a single, *universal* identifier for any physical object. The items being tracked here include all kinds of things: trade items, reusable transports, fixed assets, service relations, documents, among others that might occur. By using the EPC, the application can use a single identifier to refer to any physical object, and it is not necessary to make a special case for each category of thing.

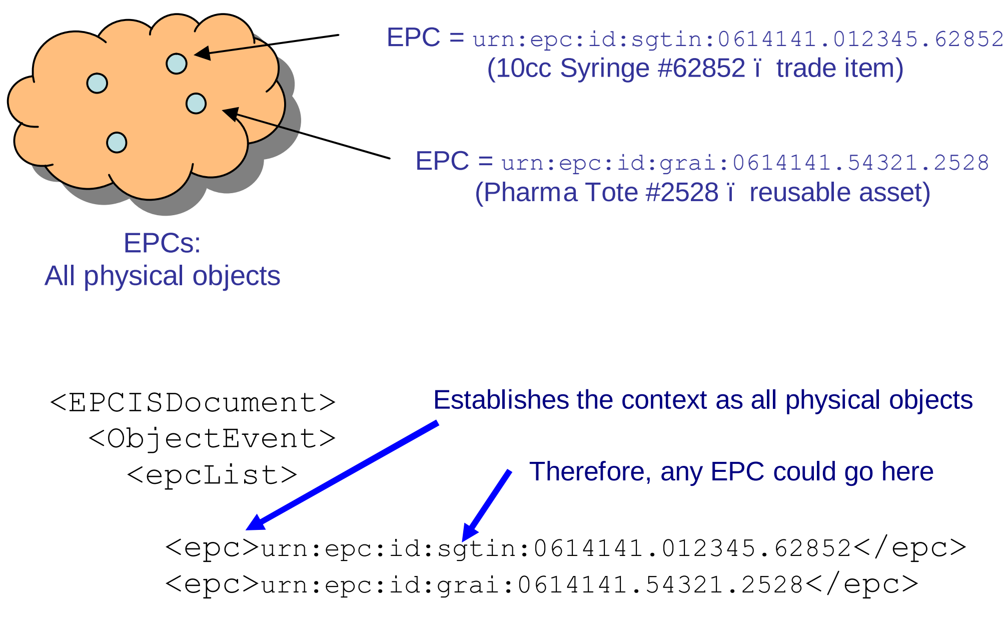
4.1.4 Use of Identifiers in a Business Data Context

Generally speaking, an identifier is a member of set (or ñnamespaceò) of strings (names), such that each identifier is associated with a specific thing or concept in the real world. Identifiers are used within information systems to refer to the real world thing or concept in question. An identifier may occur in an electronic record or file, in a database, in an electronic message, or any other data context. In any given context, the producer and consumer must agree on which namespace of identifiers is to be used; within that context, any identifier belonging to that namespace may be used.

The keys defined in the GS1 General Specifications [GS1GS] are each a namespace of identifiers for a particular category of real-world entity. For example, the Global Returnable Asset Identifier (GRAI) is a key that is used to identify returnable assets, such as plastic totes and pallet skids. The set of GRAIs can be thought of as identifiers for the members of the set ñall returnable assets.ò A GRAI may be used in a context where only returnable assets are expected; e.g., in a rental agreement from a moving services company that rents returnable plastic totes to customers to pack during a move. This is illustrated below.



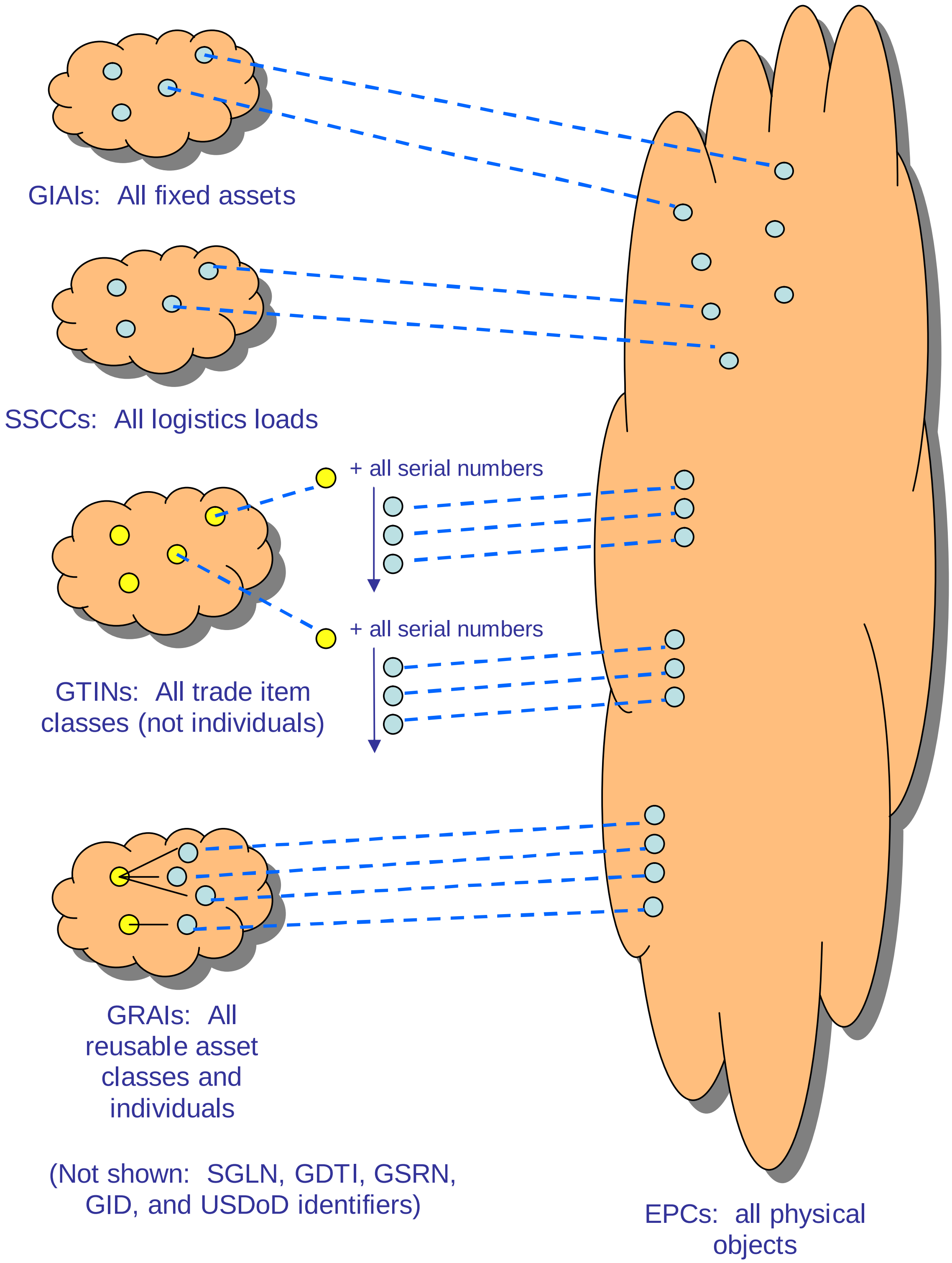
The upper part of the figure illustrates the GRAI identifier namespace. The lower part of the figure shows how a GRAI might be used in the context of a rental agreement, where only a GRAI is expected.



In contrast, the EPC namespace is a space of identifiers for *any* physical object. The set of EPCs can be thought of as identifiers for the members of the set ñall physical objects.ò EPCs are used in contexts where any type of physical object may appear, such as in the set of observations arising in the hospital storage room example above.

4.1.5 Relationship Between GS1 Keys and EPCs

There is a well-defined relationship between GS1 keys and EPCs. For each GS1 key that denotes an individual physical object (as opposed to a class), there is a corresponding EPC. This correspondence is formally defined by conversion rules specified in the EPC Tag Data Standard [TDS1.8], which define how to map a GS1 key to the corresponding EPC value and vice versa. The well-defined correspondence between GS1 keys and EPCs allows for seamless migration of data between GS1 key and EPC contexts as necessary.



Not every GS1 key corresponds to an EPC, nor vice versa. Specifically:

- A Global Trade Identification Number (GTIN) by itself does not correspond to an EPC, because a GTIN identifies a *class* of trade items, not an individual trade item. The combination of a GTIN and a unique serial number, however, *does* correspond to an EPC. This combination is called a Serialized Global Trade Identification Number, or SGTIN. The GS1 General Specifications do not define the SGTIN as a GS1 key (though this point is under discussion and may change in a future version of the GS1 General Specifications).

- In the GS1 General Specifications, the Global Returnable Asset Identifier (GRAI) can be used to identify either a *class* of returnable assets, or an individual returnable asset, depending on whether the optional serial number is included. Only the form that includes a serial number, and thus identifies an individual, has a corresponding EPC. The same is true for the Global Document Type Identifier (GDTI).
- There is an EPC corresponding to each Global Location Number (GLN), and there is also an EPC corresponding to each combination of a GLN with an extension component. Collectively, these EPCs are referred to as SGLNs.[2]
- EPCs include identifiers for which there is no corresponding GS1 key at all. These include the General Identifier and the US Department of Defense identifier .

The following table summarizes the EPC schemes defined in the EPC Tag Data Standard and their correspondence to GS1 Keys.

EPC Scheme	Tag Encodings	Corresponding GS1 Key	Typical Use
`sgtin`	`sgtin-96` `sgtin-198`	GTIN (with added serial number)	Trade item
`sscc`	`sscc-96`	SSCC	Pallet load or other logistics unit load
`sgln`	`sgln-96` `sgln-195`	GLN (with or without additional extension)	Location
`grai`	`grai-96` `grai-170`	GRAI (serial number mandatory)	Returnable/reusable asset
`giai`	`giai-96` `giai-202`	GIAI	Fixed asset
`gdti`	`gdti-96` `gdti-113`	GDTI (serial number mandatory)	Document
`gsrn`	`gsrn-96`	GSRN	Service relation (e.g., loyalty card)
`cpid`	`cpid-96` `cpid-var`	CPID (serial number mandatory)	Component / part

[2] Both GLN without an extension and GLN with an extension identify a unique location, as opposed to a class of locations. The GLN with an extension is typically used to identify a finer-grain location, such as a particular room within a building, whereas a GLN without extension is typically used to identify a coarse-grain location, such as an entire site. The ñSò in SGLN does not stand for ñserializedò, but merely indicates that the SGLN may correspond to either a GLN without extension or a GLN with an extension.

EPC Scheme	Tag Encodings	Corresponding GS1 Key	Typical Use
`gid`	`gid-96`	[none]	Unspecified
`usdod`	`usdod-96`	[none]	US Dept of Defense supply chain
`adi`	`adi-var`	[none]	Aerospace and defense ï aircraft and other parts and items

4.1.6 Use of the EPC in EPCglobal Architecture Framework

The EPCglobal Architecture Framework includes software standards at various levels of abstraction, from low-level interfaces to RFID reader devices all the way up to the business application level.

The different forms of the EPC specified in the EPC Tag Data Standard are intended for use at different levels within the EPCglobal architecture framework. Specifically:

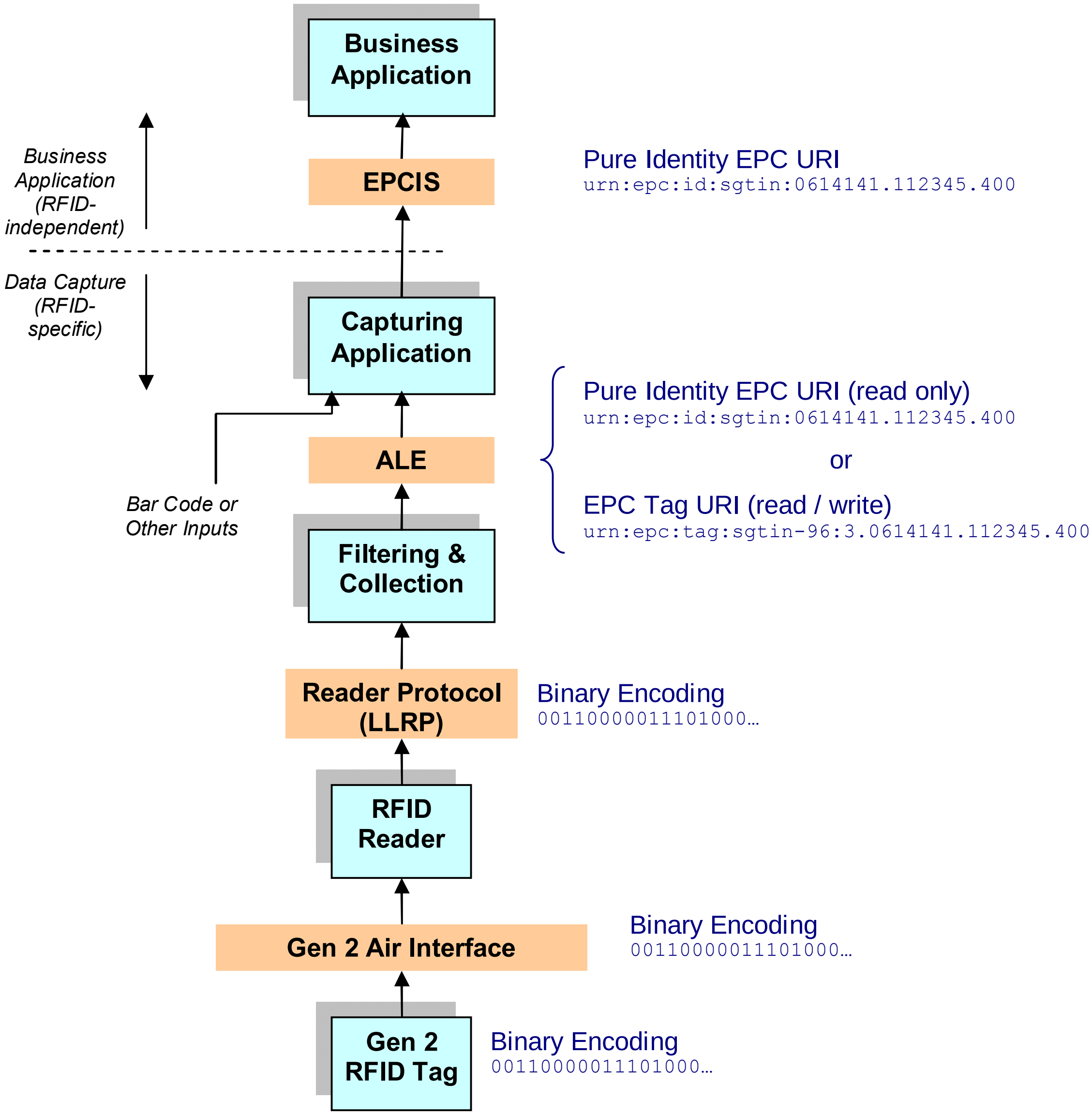
- *Pure Identity EPC URI* The primary representation of an Electronic Product Code is as an Internet Uniform Resource Identifier (URI) called the Pure Identity EPC URI. The Pure Identity EPC URI is the preferred way to denote a specific physical object within business applications. The pure identity URI may also be used at the data capture level when the EPC is to be read from an RFID tag or other data carrier, in a situation where the additional ñcontrolò information present on an RFID tag is not needed.

- *EPC Tag URI* The EPC memory bank of a Gen 2 RFID Tag contains the EPC plus additional ñcontrol informationò that is used to guide the process of data capture from RFID tags. The EPC Tag URI is a URI string that denotes a specific EPC together with specific settings for the control information found in the EPC memory bank. In other words, the EPC Tag URI is a text equivalent of the entire EPC memory bank contents. The EPC Tag URI is typically used at the data capture level when reading from an RFID tag in a situation where the control information is of interest to the capturing application. It is also used when writing the EPC memory bank of an RFID tag, in order to fully specify the contents to be written.

- *Binary Encoding* The EPC memory bank of a Gen 2 RFID Tag actually contains a compressed encoding of the EPC and additional ñcontrol informationò in a compact binary form. There is a 1-to-1 translation between EPC Tag URIs and the binary contents of a Gen 2 RFID Tag. Normally, the binary encoding is only encountered at a very low level of software or hardware, and is translated to the EPC Tag URI or Pure Identity EPC URI form before being presented to application logic.

Note that the Pure Identity EPC URI form is independent of RFID, while the EPC Tag URI and the Binary Encoding are specific to Gen 2 RFID Tags because they include RFID-specific ñcontrol informationò in addition to the unique EPC identifier.

The figure below illustrates where these forms normally occur in relation to the layers of the EPCglobal Architecture Framework. This figure is based on the architecture diagrams in Sections 6, 7, 8, and 9.



4.2 Decentralized Implementation

The EPCglobal Architecture Framework seeks to link all enterprises that have a mutual interest in sharing visibility data. Logically, the EPC Network Services that support this linkage are a common resource shared by all End Users. For many reasons it is not feasible or even advisable to literally implement this common resource as a single physical instance of a computer system operated by a central authority. The EPCglobal Architecture Framework is therefore decentralized, meaning that logically centralized functions are distributed among multiple facilities, each serving an individual End User or group of End Users. In some cases, certain of these facilities are operated by End Users themselves.

Key elements of decentralization in the EPCglobal Architecture Framework are the assignment of EPCs, and the ONS lookup service. These elements of decentralization are discussed in more detail in Sections 5.2, 7.1, and 7.3. Other elements of decentralization arise from each End User deploying its own systems that implement EPCglobal Standards. For example, the EPCglobal Architecture Framework does not include a global, centralized repository for visibility information. Instead, global visibility is achieved by each End User deploying his own systems to capture and store visibility data, and sharing that data with other End Users using the EPCIS standard.

4.3 Layering of Data Standards – Verticalization

The EPCglobal Architecture Framework includes standards for data sharing that are intended to serve the needs of many different industries. Yet, each industry has specific requirements around what data needs to be shared and what it means.

Consequently, EPCglobal standards that govern data are designed in a layered fashion. Within each data standard, there is a framework layer that applies equally to all industries that use the EPCglobal Architecture Framework. Layered on top of this are several vertical data standards that populate the general framework, each serving the needs of particular industry groups. Vertical data standards may be broad or narrow in their applicability: in many cases a vertical standard will serve several industries that share common business processes, while in other cases a vertical standard will be particular to one industry. It is even possible for a private group of trading partners to develop their own specifications atop the framework similar to a vertical standard.

The two important data standards are the EPC Tag Data Standard, and the EPCIS Data Standard. Within the EPC Tag Data Standard, the framework elements include the structure of the ñheader bitsò in the binary EPC representations and the general URI structure of the text-based EPC representations. Both of these features serve to distinguish one coding scheme from another. The vertical layer of the EPC Tag Data Standard are the specific coding schemes defined for particular industry groups.

Within the EPCIS Data Standard, the framework elements include the abstract data model that lays out a general organization for master data and visibility event data. The vertical layers of the EPCIS Data Standard define specific event types, master data vocabularies, and master data attributes used within a particular industry.

4.4 Layering of Software Standards—Implementation Technology Neutral

The EPCglobal Architecture Framework is primarily concerned with the exploitation of new data derived from the use of Electronic Product Codes and RFID technology within business processes. To foster the broadest possible applicability for EPCglobal standards, EPCglobal software standards are, whenever possible, defined using a layered approach. In this approach, the abstract content of data and/or services is defined using a technology-neutral description language such as UML. Separately, the abstract specifications are given one or more bindings to specific implementation technology such as XML, web services, and so forth. As most of the technical substance of EPCglobal standards exists in the abstract content, this approach helps ensure that even when different implementation technologies are used in different deployments there is a strong commonality in what the systems do.

4.5 Extensibility

The EPCglobal Architecture Framework explicitly recognizes the fact that change is inevitable. A general design principle for all EPCglobal Standards is openness to extension. Extensions include both enhancements to the standards themselves, through the introduction of new versions of a standard, and extensions made by a particular enterprise, group of cooperating enterprises, or industry vertical, to address specific needs that are not appropriate to address in an EPCglobal standard.

All EPCglobal Standards have identified points where extensions may be made, and provide explicit mechanisms for doing so. As far as is practical, the extension mechanisms are designed to promote both backward compatibility (a newer or extended implementation should continue to interoperate with an older implementation) and forward compatibility (an older implementation should continue to interoperate with a newer or extended implementation, though it may not be able to exploit the new features). The extension mechanisms are also designed so that non-standard extensions may be made independently by multiple groups, without the possibility of conflict or collision.

Non-standard extensions are accommodated not only because they are necessary to meet specific requirements that individual enterprises, groups, or industry verticals may have, but also because it is an excellent way to experiment with new innovations that will ultimately become standardized through newer versions of EPCglobal Standards. The extension mechanisms are designed to provide a smooth path for this migration.

5 Architectural Foundations

This section describes the key design elements at the foundations of the EPCglobal Architecture Framework. This sets the stage for the detailed description of the framework given in Sections 6, 7, and 8.

5.1 Electronic Product Code

As previously described in Section 4.1, the Electronic Product Code (EPC) is the embodiment of the underlying principle of unique identity. EPCs are assigned to physical objects, loads, locations, assets, and other entities which are to be tracked using components of the EPCglobal Architecture Framework in service of a given industry's business goals. The EPC is the thread that ties together all data that flows between End Users, and plays a central part in every role and interface within the EPCglobal Architecture Framework.

5.2 EPC Issuing Organization

As noted in Section 4.1, a key characteristic of identity as used in the EPCglobal Architecture Framework is decentralization. Decentralization is achieved through the notion of an Issuing Organization. Within this document, the term ñIssuing Organizationò refers to an organization who has been granted rights by an Issuing Agency to use a portion of the EPC namespace. That is, the Issuing Agency has effectively issued the Issuing Organization one or more blocks of Electronic Product Codes within designated coding schemes that the Issuing Organization can independently assign to physical objects and other entities without further involvement of the Issuing Agency. In many cases, the Issuing Organization is the manufacturer of a product, but this is not always the case as discussed below.

The Issuing Organization has one special responsibility within the EPCglobal Architecture Framework that distinguish it from all other End Users, with respect to the EPCs it manages:

- The Issuing Organization is responsible for ensuring that the appropriate uniqueness properties are maintained (see Section 4.1) as EPCs are allocated from the Issuing Organizationôs assigned block(s). In many cases, the Issuing Organization is also the organization that actually allocates a specific EPC and associates it with a physical object or other entity (an act called ñcommissioningò). In other cases, the Issuing Organization delegates responsibility for commissioning individual EPCs to another organization, in which case it must do so in a manner that ensures uniqueness.

Other than this responsibility, the Issuing Organization has no special responsibilities with respect to the EPCs it manages compared to any other End User. In particular, both the Issuing Organization and other end users may participate equally in the generation and sharing of EPC-related data.

5.3 EPC Hierarchical Structure

An Issuing Agency grants a block of EPCs to an Issuing Organization. An End User or other organization may be in control of multiple blocks of EPCs. The structure of all coding schemes within the Electronic Product Code definition is such that the block of EPCs is apparent by considering the first field within any given representation. The Issuing Organization for that block should not be assumed to be the product manufacturer when derived from GS1 keys (see Section 5.4.1).

Having the block of EPCs apparent in the first field within any given representation allows any system to instantly identify the Issuing Organization associated with a given EPC. This property is very important to insure the scalability of the overall system, as it allows services that would otherwise be centralized to be delegated to each Issuing Organization as appropriate.

The allocation of a block of EPCs to an Issuing Organization is actually implicit in the act of assigning the first field of the EPC, such as a GS1 Company Prefix in the case of EPCs based on GS1 keys or the CAGE/DoDAAC code in the case of USDoD and ADI EPCs. The Issuing Organization is free to commission any EPC so long as the first field within the EPC contains the assigned block number, following the EPC Tag Data Standard. The ñblockò of EPCs, therefore, simply consists of all EPCs that contain the assigned block in the first EPC field. (This is a slight simplification; see Section 5.4 for more information.)

5.4 Correspondence to Existing Codes

Most coding schemes currently defined with the EPC Tag Data Standard have a direct correspondence to existing industry coding schemes. For example, there are seven types of EPCs based on GS1 keys [GS1GS]: SGTIN, SSCC, SGLN, GRAI, GIAI, GSRN, and GDTI. In the case of these EPCs, the first field of the EPC is the GS1 Company Prefix that forms the basis of the corresponding GS1 key. The other fields of GS1-based EPCs are also derived from existing fields of the GS1 keys.

In general, this kind of correspondence is possible for any existing coding scheme that is based on delegating assignment through the central allocation of a unique prefix or field. The US Department of Defense, for example, has defined an EPC coding scheme based on its own CAGE and DoDAAC codes, which are issued uniquely to DoD suppliers and thus serve as the first EPC field when used to construct EPCs using the ñDoD constructò coding scheme.

In the last section, it was noted that assigning GS1 Company Prefix or a CAGE/DoDAAC code to an Issuing Organization effectively allocates a block of EPCs to the Issuing Organization. Because the Electronic Product Code federates several coding schemes, the ñblockò of EPCs implied by such assignment is not necessarily a single contiguous block of numbers, but rather a contiguous block within each EPC identity type to which the block number pertains. For example, when a GS1 Company Prefix is licensed to an Issuing Organization, the Issuing Organization is effectively granted a block of EPCs within each of the seven GS1-related EPC types (SGTIN, SSCC, SGLN, GRAI, GIAI, GSRN, and GDTI). When a US Department of Defense CAGE/DoDAAC code is assigned to an Issuing Organization, the Issuing Organization is effectively granted two blocks of EPCs, within the USDoD and ADI coding schemes.

5.4.1 A GS1 Company Prefix Does Not Uniquely Identify a Manufacturer

In the early days of the UPC, Company Prefixes were in one-to-one correspondence with trade item manufacturers. As the GS1 System has evolved, this is no longer true, for many reasons:

- Some manufacturers require more than one GS1 Company Prefix because of the number of GTINs they need to allocate. With a 7-digit Company Prefix, for example, only 100,000 distinct GTINs can be allocated.
- When one company acquires another company, the acquiring company typically ends up with both GS1 Company Prefixes. There is typically no motivation to reassign GTINs to the acquired product lines merely to reduce the number of GS1 Company Prefixes in use.
- When Company A acquires a product line from Company B (as opposed to the whole company), it may acquire specific GTINs that use the same Company Prefix as the

Company B continues to use for other products. GTIN assignment rules require Company A eventually to assign new GTINs to the acquired products, but at least for a time Company A and Company B each have products sharing the same Company Prefix. (Of course, during this time Company A is not entitled to allocate *new* GTINs using Company Bôs prefix.)

- An organization possessing a GS1 Company Prefix may subcontract the manufacture of trade items to contract manufacturers. The GTINs for these products may contain the Company Prefix of the contracting organization, not the manufacturers. This is especially typical when a retailer contracts for the manufacturer of private-label merchandise. One retailerôs Company Prefix may be used for products contracted to many different contract manufacturers, and conversely any given contract manufacturer may be manufacturing goods with many different Company Prefixes belonging to different brand owners.
- In some instances, a GS1 Company Prefix is assigned to a GS1 Member Organization (MO), which allocates individual GTINs or blocks of GTINs to end user organizations one at a time. This is especially true for MOs in smaller countries, and by all MOs when assigning GTINs suitable for use in the EAN-8 bar code symbology.

For all these reasons, the GS1 General Specifications [GS1GS] repeatedly caution against assuming that GS1 Company Prefix is usable as a unique identifier of a specific end user company (despite what the historic phrase ñcompany prefixò appears to imply). The GS1 Company Prefix should not be assumed to be the brand owner. In some situations, the GS1 Company Prefix may usefully be used as an *approximate* way to select EPCs that are related by virtue of having been assigned by the same company. For example, when searching for all EPC data pertaining to a given company, it may be a useful optimization to look for all EPC data bearing that companyôs prefix, then taking exceptions for those GTINs that do not belong to that company because they have been sold to other companies.

5.5 Class Level Data versus Instance Level Data

EPCs are assigned uniquely to physical objects and other entities, allowing data to be associated with individual objects. For example, one can associate data with a specific 24-count case of Cherry Hydro Soda by referring to its unique EPC.

In some cases, it is necessary to associate data with a class of object rather than a specific object itself. In the case of consumer goods, an object class refers to all instances of a specific product (Stock Keeping Unit, or SKU); for example, the class representing all 24-count cases of Cherry Hydro Soda. For Electronic Product Codes having a three-part structure of GS1 Company Prefix (or other block number), Object Class ID, and Serial Number, a product class is uniquely identified by the first two numbers, disregarding the Serial Number. The Serialized Global Trade Item Number (SGTIN) coding scheme is an example of an EPC having this structure. In this particular example, the GS1 Company Prefix and Object Class ID taken together are in fact in one-to-one correspondence with the GTIN that is used outside of the EPC arena to represent product classes. This is another example of how existing codes relate to the Electronic Product Code framework.

Some kinds of Electronic Product Codes are used to identify things that do not have any meaningful grouping into object classes. For example, the Serialized Shipping Container Code is a type of EPC used to identify shipping loads, where each load may contain a unique assortment of products. Codes of this kind often have a two-part structure, as the SSCC does, consisting only of an GS1 Company Prefix and a Serial Number.

5.6 EPC Information Services (EPCIS)

The primary vehicle for data sharing between End Users in the EPCglobal Architecture Framework is EPC Information Services (EPCIS). As explained below, EPCIS encompasses both interfaces for data sharing and specifications of the data itself.

EPCIS data is information that trading partners share to gain more insight into what is happening to physical objects in locations outside their own four walls. (EPCIS data may, of course, also be used within a companyôs four walls.) For most industries using the EPCglobal Architecture Framework, EPCIS data can be divided into five categories, as follows:

- *Static Data*, which does not change over the life of a physical object. This includes:
    - *Class-level Static Data*; that is, data which is the same for all objects of a given object class (see Section 5.5). For consumer products, for example, the ñclassò is the product, or SKU, as opposed to distinct instances of a given product. In many industries, class-level static data may be the subject of existing data synchronization mechanisms such as the Global Data Synchronization Network (GDSN); in such instances, EPCIS may not be the primary means of data sharing.
    - *Instance-level Static Data*, which may differ from one instance to the next within a given object class. Examples of instance-level static data include such things as date of manufacture, lot number, expiration date, and so forth. Instance-level static data generally takes the form of attributes associated with specific EPCs.
- *Transactional Data*, which does grow and change over the life of a physical object. This includes:
    - *Instance Observations*, which record events that occur in the life of one or more specific EPCs. Examples of instance observations include ñEPC X was shipped at 12:03pm 15 March 2004 from Acme Distribution Center #2,ò and ñAt 3:45pm 22 Jan 2005 the case EPCs (list here) were aggregated to the pallet EPC X at ABC Corpôs Boston factory.ò Most instance observations have four dimensions: time, location, one or more EPCs, and business process step.
    - *Quantity Observations*, which record events concerned with measuring the quantity of objects within a particular object class. An example of a quantity observation is ñThere were 4,100 instances of object class C observed at 2:00am 16 Jan 2003 in RetailMart Store #23.ò Most quantity observations have five dimensions: time, location, object class, quantity, and business process step.
    - *Business Transaction Observations*, which record an association between one or more EPCs and a business transaction. An example of a business transaction

observation is ñThe pallet with EPC X was shipped in fulfillment of Acme Corp purchase order #23 at 2:20pm.ò Most business transaction observations have four dimensions: time, one or more EPCs, a business process step, and a business transaction identifier.

The EPCIS Data Standards provide a precise definition of all the types of EPCIS data, as well as the meaning of ñeventò as used above.

Transactional data differs from static data not only because as it grows and changes over the life of a physical object, but also because transactional data for a given EPC is typically generated by many distinct end users within a supply chain. For example, consider an object that is manufactured by A, who employs transportation company B to ship to distributor C, who delivers the object by way of 3rd party logistics provider D to retailer E. By the time the object reaches E, all five companies will have gathered transactional data about the EPC. The static data, in contrast, often comes exclusively from the manufacturer A.

A key challenge faced by the EPCglobal Architecture Framework is to allow any End User to discover all transactional data to which it is authorized, from any other End User. Section 7.1 discusses how the EPCglobal Architecture Framework addresses this challenge.

6 Roles and Interfaces – General Considerations

This section and the three sections that follow define the EPCglobal Architecture Framework, describing at a high level all of the EPCglobal Standards and EPC Network Services that comprise it. The normative description of each of these is found elsewhere. In the case of an EPCglobal Standard, the normative description is or will be an EPCglobal standard document. In the case of an EPC Network Service, normative descriptions are either provided as EPCglobal Standards (for interface aspects of EPC Network Services) or in other EPCglobal documentation (for implementation aspects).

6.1 Architecture Framework vs. System Architecture

The EPCglobal Architecture Framework is a collection of interrelated standards for hardware, software, and data interfaces (EPCglobal Standards), together with shared network services that are operated by GS1, its delegates, and others (EPC Network Services). End users deploy systems that make use of these elements of the EPCglobal Architecture Framework. In particular, each end user will have a system architecture for their deployment that includes various hardware and software components, and these components may use EPCglobal Standards to communicate with each other and with external systems, and also make use of the EPC Network Services to carry out certain tasks. A given end userôs system architecture may also use alternative or additional standards, including data carriers and software interfaces beyond those governed by EPCglobal standards.

The EPCglobal Architecture Framework does not define a system architecture that end users must implement, nor does it dictate particular hardware or software components an end user must deploy. The hardware and software components within any end userôs system architecture may be created by the end user or obtained by the end user from solution providers, but in any case the definition of these components is outside the scope of the EPCglobal Architecture Framework. The EPCglobal Architecture Framework only defines interfaces that the end user's components may implement. The EPCglobal Architecture Framework explicitly avoids specification of components in order to give end users maximal freedom in designing system architectures according to their own preferences and goals, while defining interface standards to ensure that systems deployed by different end users can interoperate and that end users have a wide marketplace of components available from solution providers.

Because the EPCglobal Architecture Framework does not define a system architecture *per se*, this document does not normatively specify a particular arrangement of system components and their interconnection. However, in order to understand the interrelationship of EPCglobal Standards and EPC Network Services, it is helpful to discuss how they are used in a typical system architecture. The following sections of this document, therefore, describe a hypothetical system architecture to illustrate how the components of the EPCglobal Architecture Framework fit together. It is important to bear in mind, however, that the following description differs from a true system architecture in the following ways:

- An end user system architecture may only need to employ a subset of the EPCglobal Standards and EPC Network Services depicted here. For example, an RFID application using EPC tags that exists entirely within the four walls of a single enterprise may use the UHF Class 1 Gen 2 Tag Air Interface and the EPC Tag Data Standard, but have no need for the Object Name Service.
- The mapping between hardware and software roles depicted here and actual hardware or software components deployed by an end user may not necessarily be one-to-one. For example, to carry out a business process of shipment verification using EPC-encoded RFID tags, one end user may deploy a system in which there is a separate RFID Reader (a hardware device), Filtering & Collection middleware (software deployed on a server), and EPCIS Capturing Application (software deployed on a different server). Another end user may deploy an integrated verification portal device that combines into a single package all three of these roles, exposing only the EPCIS Capture Interface. For this reason, this document is careful to refer to *roles* rather than *components* when talking about system elements that make use of standard interfaces.
- In the same vein, roles depicted here may be carried out by an end user's legacy system components that may have additional responsibilities outside the scope of the EPCglobal Architecture Framework. For example, it is common to have enterprise applications such as Warehouse Management Systems that simultaneously play the role of EPCIS Capturing Application (e.g., receiving EPC observations during the loading of a truck), an EPCIS-enabled Repository (e.g., recording case-to-pallet associations), and an EPCIS Accessing Application (e.g., carrying out business decisions based on EPCIS-level data).

The overall intent of the EPCglobal Architecture Framework is to provide end users with great flexibility in creating system architectures that meet their needs.

6.2 Cross-Enterprise versus Intra-Enterprise

As discussed in Section 2, elements of the EPCglobal Architecture Framework can be categorized as pertaining to EPC Data Sharing between enterprises, EPC Object Exchange between enterprises, or EPC Infrastructure deployed within a single enterprise. Clearly, all End Users will find relevance in the first two categories, as use of these standards is necessary to interact with other end users. An end user has much more latitude, however, in its decisions surrounding adoption of the EPC Infrastructure standards, as those standards do not affect parties outside the end userôs own four walls.

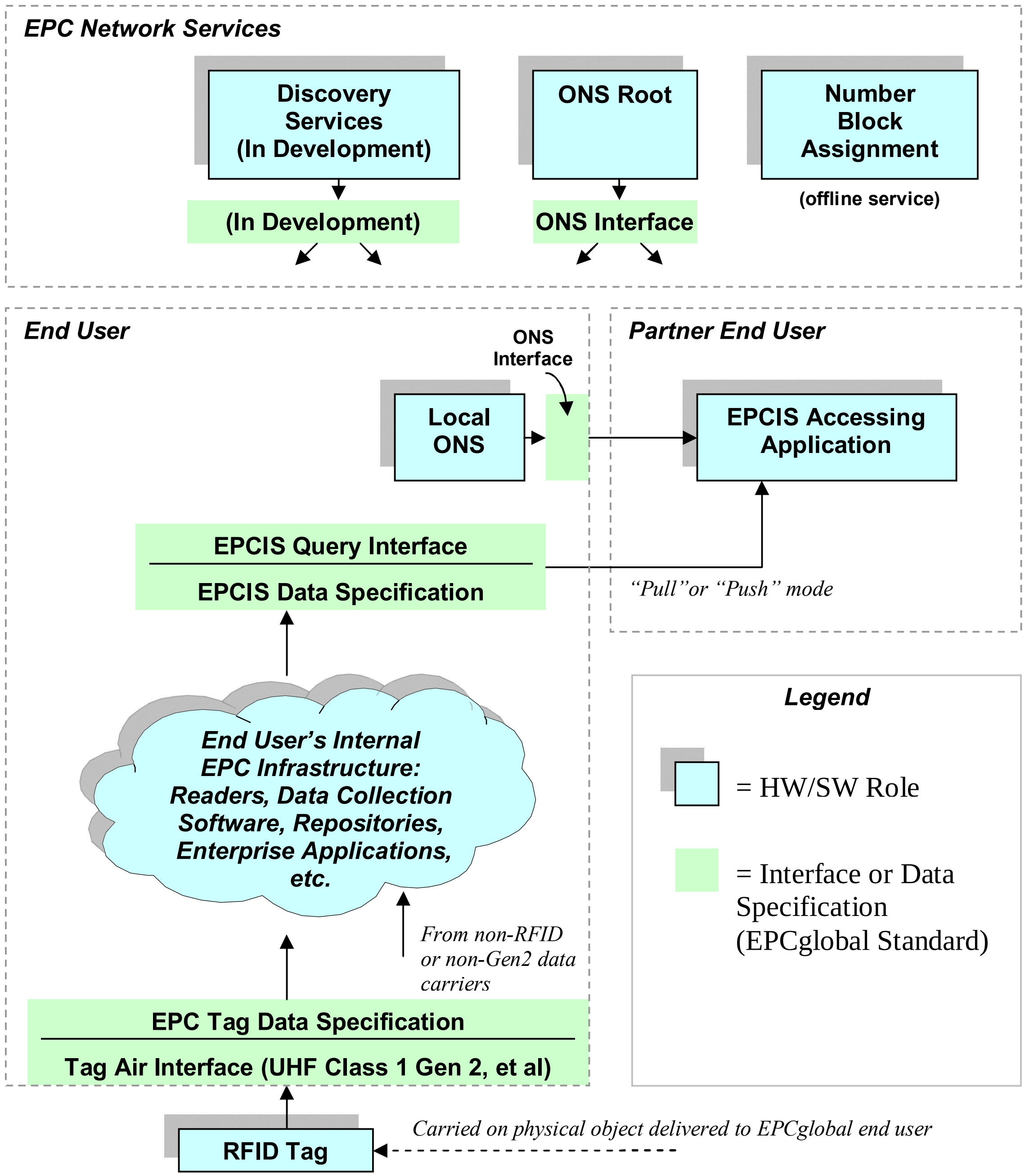
For this reason, the following discussion of roles and interfaces within the EPCglobal Architecture Framework is divided into two sections, the first dealing with cross-enterprise elements (EPC Data Sharing and EPC Object Exchange), and the second dealing with intra-enterprise elements (EPC Infrastructure). As explained in Section 2, however, it should be borne in mind that the division between cross-enterprise and intra-enterprise standards is not absolute, and a given enterprise may employ cross-enterprise standards entirely within its four walls or conversely use intra-enterprise standards in collaboration with outside parties.

7 Data Flow Relationships – Cross-Enterprise

This section provides a diagram showing the relationships between EPCglobal Standards, from a data flow perspective. This section shows only the EPCglobal Standards that are typically used between end users, namely those categorized as ñEPC Object Exchange Standardsò or ñEPC Data Sharing Standardsò in Section 2. EPCglobal Standards that are primarily used within the four walls of a single end user (ñEPC Infrastructure Standardsò from Section 2) are described in Section 8. Most End Users will implement the architecture given in this section.

In the following diagram, the plain green bars denote interfaces governed by EPCglobal standards, while the blue ñshadowedò boxes denote roles played by hardware and software components of a typical system architecture. As emphasized in Section 6.1, in any given end userôs deployment the mapping of roles in this diagram to actual hardware and software components may not be one-to-one, nor will every end userôs deployment contain every role shown here.

To emphasize how EPCglobal Standards are employed to share data between partners, this diagram shows one end user (labeled ñEnd Userò in the diagram) who observes a physical object having an EPC on an RFID tag, and shares data about that observation with a second end user (labeled ñPartner End Userò). This interaction is shown as one way, for clarity. In many situations, the Partner End User may also be observing physical objects and sharing that data with the first End User. If that is the case, then the full picture would show a mirror-image set of roles, interfaces, and interactions.



1038

1039 A formal definition of each of the roles and interfaces in this diagram may be found in
1040 Section 9. The remainder of this section provides a more informal illustration of how the
1041 roles and interfaces interact in typical scenarios of using the EPCglobal Architecture
1042 Framework.

7.1 Data Sharing Interactions

The top part of the diagram shows the roles and interfaces involved in data sharing. The Partner End User has an ñEPCIS Accessing Applicationò (role), which is some application specific to the Partner End User that is interested in information about a particular EPC.

The first thing the EPCIS Accessing Application needs to do is to determine where it can go to obtain data of interest. This is generally not a trivial task, because the source of information may vary from EPC to EPC, and the network address where information is available cannot be derived from the EPC itself. In general, there are several ways an EPCIS Accessing Application may locate the data of interest:

- The EPCIS Accessing Application may know in advance exactly where to find the information. This often arises in simple two-party supply chain scenarios, where one party is given the network address of the other partyôs EPCIS service as part of a business agreement.
- The EPCIS Accessing Application may know where to find the information it seeks based on information obtained previously. For example, in a three-party supply chain consisting of parties A, B, and C, party C may know how to reach Bôs service as part of a business agreement, and in obtaining information from B it learns how to reach Aôs service (which B knows as part of its business agreement with A). This is sometimes referred to as ñfollowing the chain.ò
- The EPCIS Accessing Application may use the Object Name Service (ONS) to locate the EPCIS service of the End User who commissioned the EPC of the object in question.
- The EPCIS Accessing Application may use Discovery Services to locate the EPCIS services of all End Users that have information about the object in question, including End Users other than the one who commissioned the EPC of the object. This method is required in the general case of multi-party supply chain, when the participants are not known to the EPCIS Accessing Application in advance and when it is not possible or practical to ñfollow the chain.ò (Discovery Services are TBD at the time of this writing, so the precise architecture of roles and interfaces involved in Discovery Services is not yet known ï the box in the diagram is just a placeholder.)

Whatever method is used, the net result is that the EPCIS Accessing Application has located the EPCIS service of the End User from whom it will obtain data to which the EPCIS Accessing Application is authorized. The EPCIS Accessing Application then requests information directly from the EPCIS service of the other end user. Two EPCglobal Standards govern this interaction. The EPCIS Query Interface defines how data is requested and delivered from an EPCIS service. The EPCIS Data Standard defines the format and meaning of this data. The EPCIS Query Interface is designed to support both on-demand or ñpullò modes of data transfer, as well as asynchronous or ñpushò modes. Several transport bindings are provided, including on-line transport as well as disconnected (store and forward) transport.

When an EPCIS Accessing Application of the Partner End User accesses the EPCIS service of the first End User, the first End User will usually want to authenticate the identity of the Partner End User in order to determine what data the latter is authorized to receive. The EPCglobal Architecture Framework allows the use of a variety of authentication technologies across its defined interfaces. It is expected, however, that the X.509 authentication framework will be widely employed by End Users. If X.509 certificates are used, they should comply with the standards defined in the EPCglobal X.509 Certificate Profile [Cert2.0], which provides a minimum level of cryptographic security and defines and standardizes identification parameters for users, services/servers and devices. In some situations, an End User may grant EPCIS access to another party whose identity is not authenticated or authenticated by means other than those facilitated by EPCglobal. This is a policy decision that is up to each End User to make.

7.2 Object Exchange Interactions

The lower part of the diagram illustrates how the first End User interacts with physical objects it receives from other end users. A physical object is received by the End User, bearing an RFID tag that contains an EPC. The End User reads the tag using RFID Readers deployed as part of its internal EPC infrastructure. Two EPCglobal Standards govern this interaction. A Tag Air Interface defines how data is communicated via radio signals between RFID Tags and RFID Readers. The EPC Tag Data Standard defines the format and meaning of this data, including the EPC and other data on the Tag.

Within the End User's internal EPC infrastructure, there may be many hardware and software components involved in obtaining and processing the tag read, integrating the tag read into an ongoing business process, and ultimately using the tag read to help in creating an EPCIS event that can be made available to a Partner End User via EPCIS as previously described. A single tag read could in theory result in a new EPCIS event by itself; far more commonly, each EPCIS event results from many tag reads together with other information derived from the business context in which the tag (or tags) were read. Some scenarios of how this takes place are illustrated in Section 8.

7.3 ONS Interactions

In Section 7.1, it was mentioned that one End User may locate the EPCIS service of the organization that commissioned a given EPC by using the Object Name Service, or ONS. This section describes in somewhat more detail how this takes place as a collaboration between an EPC Network Service and a service provided by an individual end user.

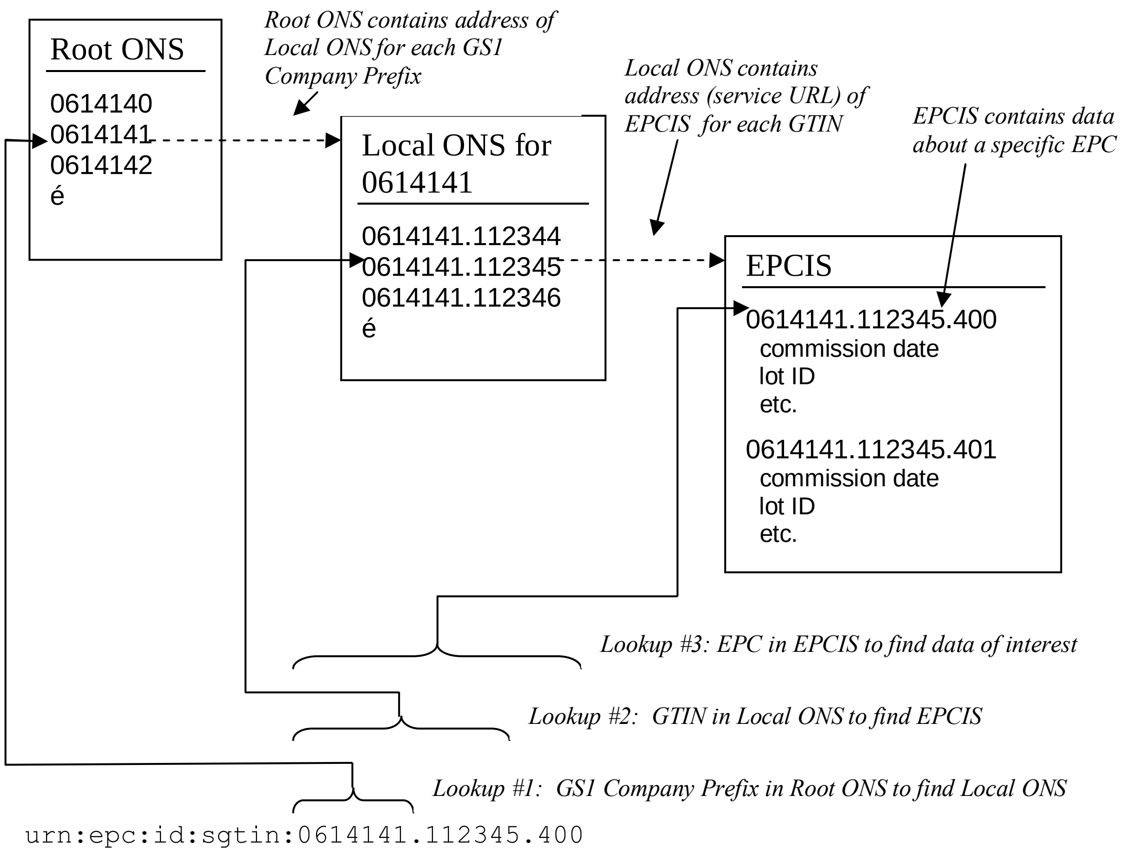
The Object Name Service can be thought of as a simple lookup service that takes an EPC as input, and produces as output the address (in the form of a Uniform Resource Locator, or URL) of an EPCIS service designated by the Issuing Organization of the EPC in question. (An Issuing Organization may actually use ONS to associate several different services, not just an EPCIS service, with an EPC. All of the following discussion applies equally regardless of which type of service is looked up.) In general, there may be many different object classes that fall under the authority of a single Issuing Organization, and it may not be the case that all object classes of a given Issuing Organization will have information provided by the same EPCIS service. This is especially true when the Issuing

Organization delegates the commissioning of EPCs to other organizations; for example, a retailer who contracts with different manufacturing partners for different private-label product lines. Therefore, ONS requires a separate entry for each object class. (The current design of ONS does not, however, permit different entries for different serial numbers of the *same* object class. For coding schemes which do not have a field corresponding to object class, such as the SSCC, GIAI, and GSRN keys, the ONS entry is at the Issuing Organization level.)

Conceptually, this is a single global lookup service. It would not be practical, however, to implement ONS as one gigantic directory, both for reasons of scalability and in consideration of the difficulty of each Issuing Organization having to maintain records for its object classes in a shared database. Instead, ONS is architected as an application of the Internet Domain Name System (DNS), which is also a single global lookup service conceptually but is implemented as a hierarchy of lookup services.

ONS works as follows. When an End User application wishes to locate an EPCIS service, it presents a query to its local DNS resolver (typically provided as part of the computer's operating system). The DNS resolver is responsible for carrying out the query procedure, and returning the result to the requesting application. From the application's point of view, the lookup appears to be a single operation.

Inside the resolver, however, a multi-step lookup is performed as follows. First, it consults a Root ONS service operated by a party authorized by GS1 to provide an ONS Root service (typically a GS1 Member Organization). The Root ONS service identifies the Local ONS service of the Issuing Organization organization for that EPC, possibly delegating to a different Root ONS service if the first root tried is not able to resolve this particular Issuing Organization. The End User then completes the lookup by consulting the Local ONS service, which provides the pointer to the EPCIS service in question. This multi-step lookup procedure is illustrated below.



Note that the Local ONS might return a pointer to an EPCIS service operated by a *different* organization. For example, in a contract manufacturing scenario Company A is the Issuing Organization for the block of EPCs and operates the local ONS, but the commissioning of individual tags is done by Company B, the contract manufacturer to which Company A has delegated the work of commissioning EPCs. In that example, Company A operates the Local ONS for Company Aôs GS1 Company Prefix, but for contract-manufactured products it returns pointers to Company Bôs EPCIS service. The table below illustrates the relationships between the lookup stages, the underlying services, and the data involved.

Lookup Step	**Lookup Service Employed**	**Who Maintains the Service**	**What Data is Retrieved**
1	Root ONS	GS1 Member Organization or other authorized Root ONS service provider	Address of Local ONS for given GS1 Company Prefix or CAGE/DoDAAC

Lookup Step	Lookup Service Employed	Who Maintains the Service	What Data is Retrieved
2	Local ONS for given GS1 Company Prefix or CAGE/DoDAAC	Holder of GS1 Company Prefix or CAGE/DoDAAC	Address of EPCIS Service for given EPC Class (e.g., GTIN)
3	EPCIS	End user responsible for commissioning EPC	Commissioning data about the EPC

ONS is implemented as an application of the Internet Domain Name System (DNS), simply by specifying a convention whereby an EPC is converted to an Internet Domain Name in a domain specified by an ONS Root service. Any such root domain may be used. For example, given an EPC:

`urn:epc:id:sgtin:0614141.112345.400`

and a choice of initial root ONS domain, `onsepc.com`, an ONS lookup is performed by transforming the EPC into the following Internet Domain Name (essentially, by converting to a GS1 key, dropping the serial number, dropping the check digit and indicator digit, reversing what remains and inserting dots, and adding the root domain `onsepc.com`):

`5.4.3.2.1.1.4.1.4.1.6.0.sgtin.id.onsepc.com`

This domain name is then looked up in the Internet DNS following ordinary DNS rules, using a type of lookup designed to retrieve service records (so-called ñNAPTRò records). An ñONS service,ò therefore is nothing more than an ordinary DNS nameserver that happens to be part of the domain name tree rooted at one of several possible ONS root domains. This has several implications:

- The "Root ONS service" and "Local ONS service" as used above may each be implemented by multiple redundant servers, as DNS allows more than one server to be listed as the provider of DNS service for any particular domain name. This increases the scalability and reliability of the overall system.
- Each Root ONS service is actually itself several levels down in a hierarchy of lookups, which has its true root in the worldwide DNS root.
- ONS benefits from the DNS caching mechanism, which means that in practice a given ONS lookup does not actually need to consult each of the services in the hierarchy, as in most cases the higher-level entries are cached locally.

More information may be found in the DNS specifications [RFC1034, RFC1035], and in the ONS Standard [ONS2.0.1].

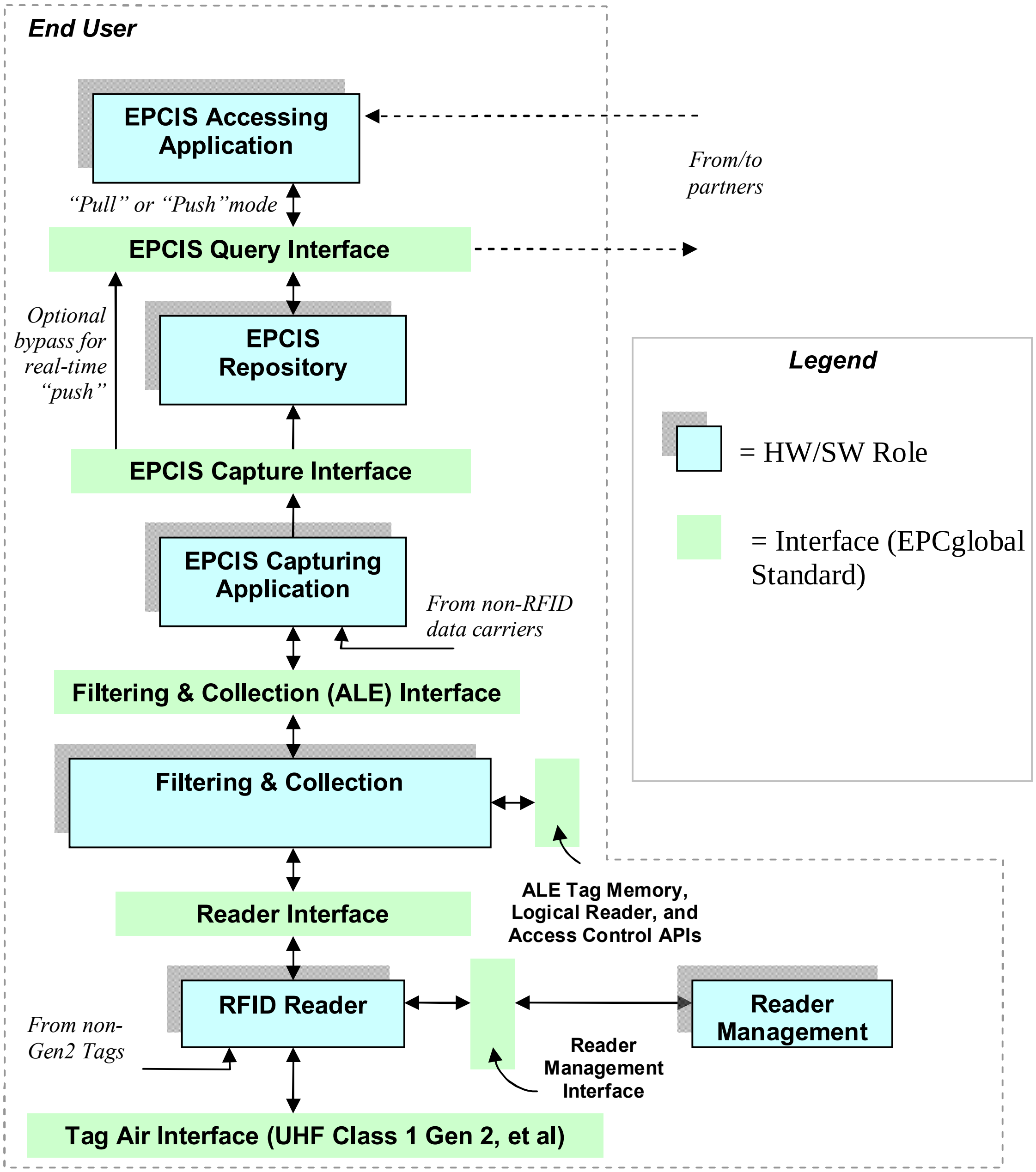
7.4 Number Assignment

The foregoing text has described every role and interface in the diagram at the beginning of this Section 7, except for Number Block Assignment. This role simply refers to GS1ôs service of issuing unique GS1 Company Prefixes to each Issuing Organization that requests one, in its capacity as the Issuing Agency for GS1 keys (see Section 4.1). By insuring that every GS1 Company Prefixes that is issued is unique, the uniqueness of EPCs assigned by individual End Users is ensured. (Number assignment for coding schemes other than GS1 keys is carried out by Issuing Agencies other than EPCglobal, and so GS1ôs Number Block Assignment Service does not apply in those cases.)

8 Data Flow Relationships – Intra-Enterprise

This section provides a diagram showing the relationships between EPCglobal Standards, from a data flow perspective. In contrast to Section 7, this section shows only the EPCglobal Standards that are typically used within the four walls of a single end user, namely those categorized as ñEPC Infrastructure Standardsò in Section 2. This section expands the ñcloudò in the diagram from Section 7. Because this cloud is completely internal to a given enterprise, an end user has much more latitude to deviate from this picture when appropriate to that end userôs unique business conditions. EPCglobal sets standards in this area, however, to encourage solution providers to create interoperable system components from which end users may choose.

As in Section 7, the plain green bars in the diagram below denote interfaces governed by EPCglobal standards, while the blue ñshadowedò boxes denote roles played by hardware and software components of a typical system architecture. As emphasized in Section 6.1, in any given end userôs deployment the mapping of roles in this diagram to actual hardware and software components may not be one-to-one, nor will every end userôs deployment contain every role shown here.



Between the EPC Object Exchange interfaces and the EPC Data Sharing interfaces in the figure from Section 7 is a ñcloudò of internal infrastructure whose purpose is to create EPCIS-level data from RFID observations of EPCs and other data sources. The figure above shows a typical approach to architecting this infrastructure, showing the role that EPCglobal standards play.

Several steps are shown in the figure, each mediated by an EPCglobal standard interface. At each step progressing from raw tag reads at the bottom to EPCIS data at the top, the semantic content of the data is enriched. Following the data flow from the bottom of the figure to the top:

- *Readers* Make multiple observations of RFID tags while they are in the read zone.
- *Reader Interface* Defines the control and delivery of raw tag reads from Readers to the Filtering & Collection role. Events at this interface say ñReader A saw EPC X at time T.ò
- *Filtering & Collection* This role filters and collects raw tag reads, over time intervals delimited by events defined by the EPCIS Capturing Application (e.g. tripping a motion detector).
- *Filtering & Collection (ALE) Interface* Defines the control and delivery of filtered and collected tag read data from Filtering & Collection role to the EPCIS Capturing Application role. Events at this interface say ñAt Location L, between time T1 and T2, the following EPCs were observed,ò where the list of EPCs has no duplicates and has been filtered by criteria defined by the EPCIS Capturing Application.
- *EPCIS Capturing Application* Supervises the operation of the lower EPC elements, and provides business context by coordinating with other sources of information involved in executing a particular step of a business process. The EPCIS Capturing Application may, for example, coordinate a conveyor system with Filtering & Collection events, may check for exceptional conditions and take corrective action (e.g., diverting a bad case into a rework area), may present information to a human operator, and so on. The EPCIS Capturing Application understands the business process step or steps during which EPCIS data capture takes place. This role may be complex, involving the association of multiple Filtering & Collection events with one or more business events, as in the loading of a shipment. Or it may be straightforward, as in an inventory business process where there may be ñsmart shelvesò deployed that generate periodic observations about objects that enter or leave the shelf. In the latter case, the Filtering & Collection-level event and the EPCIS-level event may be so similar that no actual processing at the EPCIS Capturing Application level is necessary, and the EPCIS Capturing Application merely configures and routes events from the Filtering & Collection interface directly to an EPCIS-enabled Repository.
- *EPCIS Capture Interface* The interface through which EPCIS data is delivered to enterprise-level roles, including EPCIS Repositories, EPCIS Accessing Applications, and data sharing with partners. Events at this interface say, for example, ñAt location X, at time T, the following contained objects (cases) were verified as being aggregated to the following containing object (pallet).ò
- *EPCIS Accessing Application* Responsible for carrying out overall enterprise business processes, such as warehouse management, shipping and receiving, historical throughput analysis, and so forth, aided by EPC-related data.
- *EPCIS Repository* Software that records EPCIS-level events generated by one or more EPCIS Capturing Applications, and makes them available for later query by EPCIS Accessing Applications.

The interfaces within this stack are designed to insulate the higher levels of the stack from unnecessary details of how the lower levels are implemented. One way to understand this is to consider what happens if certain changes are made:

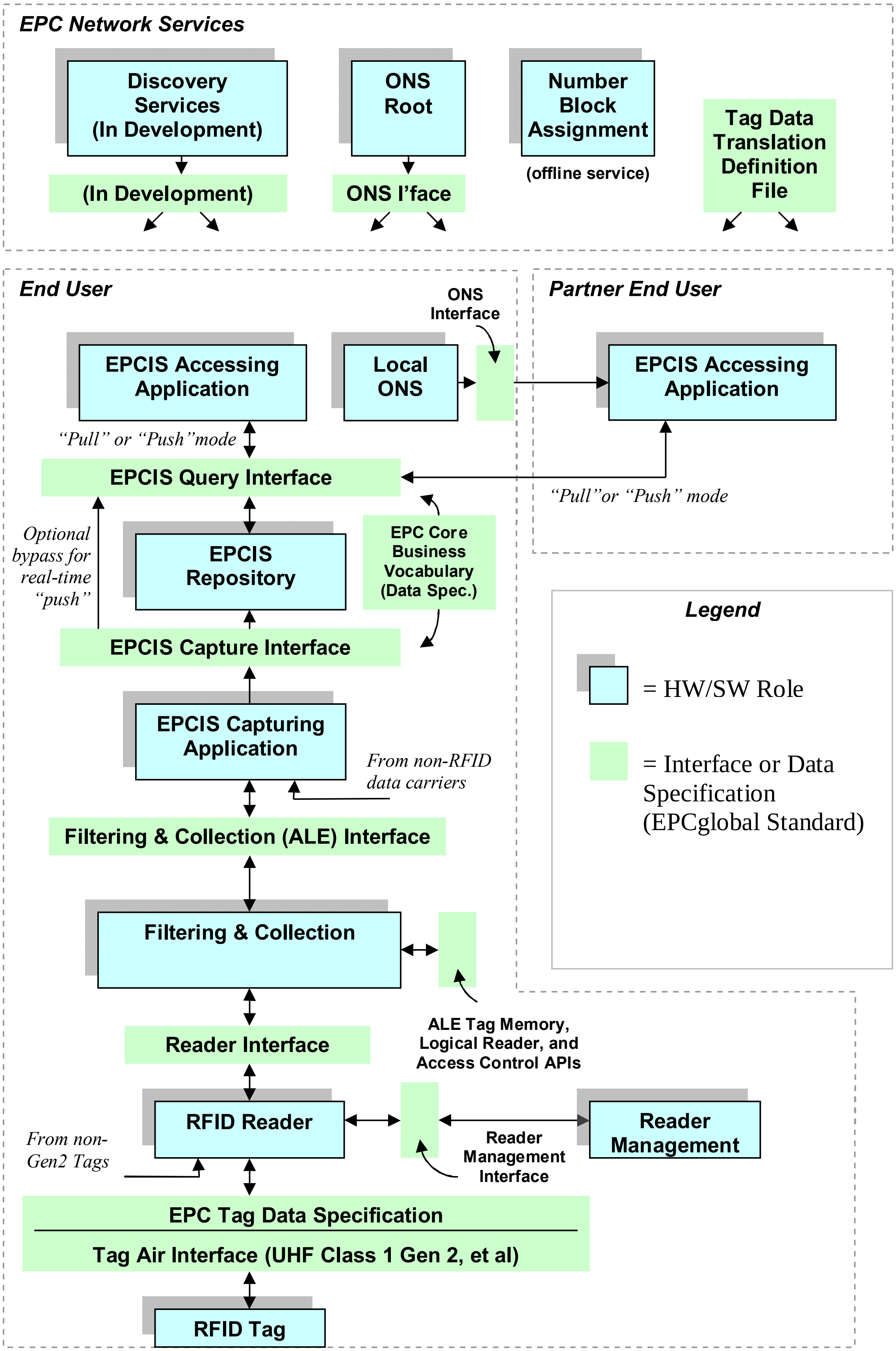
- The Reader Interface insulates the higher layers from knowing what reader makes/models have been chosen. If a different reader is substituted, the information at the Reader Interface remains the same. The Reader Interface may, to some extent, also provide insulation from knowing what Tag Air Interfaces are in use, though obviously not when one tag type or Tag Air Interface provides fundamentally different functionality from another.
- The Filtering & Collection Interface insulates the higher layers from the physical design choices made regarding how tags are sensed and accumulated, and how the time boundaries of events are triggered. If a single four-antenna reader is replaced by a constellation of five single-antenna ñsmart antennaò readers, the events at the Filtering & Collection level remain the same. Likewise, if a different triggering mechanism is used to mark the start and end of the time interval over which reads are accumulated, the Filtering & Collection event remains the same.
- The EPCIS interfaces insulate enterprise applications from understanding the details of how individual steps in a business process are carried out at a detailed level. For example, a typical EPCIS event is ñAt location X, at time T, the following cases were verified as being on the following pallet.ò In a conveyor-based business implementation, this likely corresponds to a single Filtering & Collection event, in which reads are accumulated during a time interval whose start and end is triggered by the case crossing electric eyes surrounding a reader mounted on the conveyor. But another implementation could involve three strong people who move around the cases and use hand-held readers to read the EPCs. At the Filtering & Collection level, this looks very different (each triggering of the hand-held reader is likely a distinct Filtering & Collection event), and the processing done by the EPCIS Capturing Application is quite different (perhaps involving an interactive console that the people use to verify their work). But the EPCIS event is still the same.

In summary, the different steps in the data path correspond to different semantic levels, and serve to insulate different concerns from one another as data moves up from raw tag reads towards EPCIS.

Besides the data path described above, there is also a control path responsible for managing and monitoring of the infrastructure. This includes the Reader Management standard, the Discovery, Configuration, and Initialization (DCI) standard, and the control interfaces in the Application Level Events (ALE) standard.

9 Roles and Interfaces – Reference

This section provides a complete reference to all roles and interfaces described in Sections 7 and 8, describing each in more formal terms. For convenience, the following diagram combines the figures from the two previous sections into a single figure. As in Sections 7 and 8, the plain green bars in the diagram below denote interfaces governed by EPCglobal standards, while the blue ñshadowedò boxes denote roles played by hardware and software components of a typical system architecture. As emphasized in Section 6.1, in any given end user's deployment the mapping of roles in this diagram to actual hardware and software components may not be one-to-one, nor will every end user's deployment contain every role shown here.


EPC Network Services
Discovery Services (In Development)
(In Development)
ONS Root
ONS I'face
Number Block Assignment
(offline service)
Tag Data Translation Definition File
End User
ONS Interface
EPCIS Accessing Application
Local ONS
Partner End User
EPCIS Accessing Application
"Pull" or "Push"mode
EPCIS Query Interface
"Pull"or "Push" mode
Optional bypass for real-time "push"
EPCIS Repository
EPC Core Business Vocabulary (Data Spec.)
EPCIS Capture Interface
Legend
= HW/SW Role
= Interface or Data Specification (EPCglobal Standard)
EPCIS Capturing Application
From non-RFID data carriers
Filtering & Collection (ALE) Interface
Filtering & Collection
ALE Tag Memory, Logical Reader, and Access Control APIs
Reader Interface
RFID Reader
Reader Management
Reader Management Interface
From non-Gen2 Tags
EPC Tag Data Specification
Tag Air Interface (UHF Class 1 Gen 2, et al)
RFID Tag

The next section explains the roles and interfaces in this diagram in more detail.

9.1 Roles and Interfaces – Responsibilities and Collaborations

This section defines each of the roles and interfaces shown in the diagram above.

9.1.1 RFID Tag (Role)

RFID tags compliant with GS1 EPCglobal Air Interface standards include the following minimum features:

- An EPC identifier, optionally writeable.
- A Tag Identifier (TID) that indicates the tagôs manufacturer identity and mask ID.
- A ñkillò function that permanently disables the Tag This feature may involve additional data stored on the tag such as a kill password.

In addition, tags may include the following optional features:

- Extended TID that may include a unique serial number and information describing the capabilities of the tag.
- Recommissioning of the Tag
- Password-protected access control.
- User memory (for application data apart from the EPC).
- Authenticated access control
- Read-range reduction and/or hiding portions of tag memory
- Sensors, with or without sensor data logging
- A power source that may supply power to the Tag or to its sensors

9.1.2 EPC Tag Data Standard (Data Specification)

Normative references:

- Ratified EPCglobal Standard: [TDS1.8]
- Standard in development: [TDS1.9]

Responsibilities:

- Defines the overall structure of the Electronic Product Code, including the mechanism for federating different coding schemes.
- Defines specific EPCglobal coding schemes.
- For each EPCglobal coding scheme, defines binary representations for use on RFID tags, text representations for use within information systems (in particular, at the ALE level and higher in the EPCglobal Architecture Framework, including EPCIS and

Discovery Services), and rules for converting between one representation and another.

- For EPCs that are in correspondence with GS1 keys, defines rules for traversing this correspondence in both directions.
- Defines the encoding of TID memory for Gen2 Tags, which encodes information about the Tag itself as opposed to the object to which the Tag is affixed. This information may include the capabilities of the Tag (such as how much memory it contains, whether it implements optional features, etc). It also may include a globally unique serial number assigned at Tag manufacture time.
- Defines the encoding of User Memory for Gen2 Tags, which may be used to store additional data elements beyond the EPC.

9.1.3 Tag Air Interface (Interface)

There are two EPCglobal Tag Air Interfaces, which differ primarily in the frequency band of operation. .

Normative references:

- Ratified EPCglobal Standard: [UHFC1G21.1.0], [UHFC1G21.2.0], [UHFG2V2], [HFC1]

Responsibilities:

- Communicates a command to a tag from an RFID Reader.
- Communicates a response from a tag to the RFID Reader that issued the command.
- Provides means for a reader to singulate individual tags when more than one is within range of the RFID Reader.
- Provides means for readers and tags to minimize interference with each other.

9.1.4 RFID Reader (Role)

Responsibilities:

- Reads the EPCs of RFID Tags within range of one or more antennas (via a Tag Air Interface) and reports the EPCs to a host application (via the Reader Interface).
- When an RFID Tag allows the EPC to be written post-manufacture, writes the EPC to a tag (via a Tag Air Interface) as commanded by a host application (via the Reader Interface).
- When an RFID Tag provides additional user data apart from the EPC, reads and writes user data (via a Tag Air Interface) as directed by a host application (via the Reader Interface).
- When an RFID Tag provides additional features such as kill, lock, etc, operates those features (via a Tag Air Interface) as directed by a host application (via the Reader Interface).

- May provide additional processing such as filtering of EPCs, aggregation of reads, and so forth. See also the Filtering & Collection Role, Section 9.1.8.

9.1.5 Reader Interface (Interface)

A Reader Interface provides the means for software to control aspects of RFID Reader operation, including the capabilities implied by features of the Tag Air Interfaces. The EPCglobal Low Level Reader Protocol (LLRP) standard is designed to provide complete access to all capabilities of the UHF Class 1 Gen 2 Tag Air Interface, including reading, writing, locking, and killing tags, as well as providing control to clients over the use of the RF channel and protocol-specific tag features such as Gen2 inventory sessions

Normative references:

- Ratified EPCglobal Standard: [LLRP1.1]

Responsibilities[3]*:*

- Provides means to command an RFID Reader to inventory tags (that is, to read the EPCs carried on tags), read tags (that is, to read other data on the tags apart from the EPC), write tags, manipulate tag user and tag identification data, and access other features such as kill, lock, etc.
- Provides means to access RFID Reader management functions including capability discovery, firmware/software configuration and updates, health monitoring, connectivity monitoring, statistics gathering, antenna connectivity, transmit power level, and managing reader power consumption.
- Provides means to control RF aspects of RFID Reader operation including control of RF spectrum utilization, interference detection and measurement, modulation format, data rates, etc.
- Provides means to control aspects of Tag Air Interface operation, including protocol parameters and singulation parameters.
- Provides access to processing features such as filtering of EPCs, aggregation of reads, and so forth. For features that require converting between different representations of EPCs, may use the Tag Data Translation Interface (Section 9.1.21) to obtain machine-readable rules for doing so.

9.1.6 Reader Management Interface (Interface)

Normative references:

- Ratified EPCglobal Standards: [RM1.0.1]] [DCI]

Responsibilities:

[3] Several of these responsibilities are described using text adapted from [SLRRP], which the authors gratefully acknowledge.

- Provides means to query the configuration of an RFID Reader, such as its identity, number of antennas, and so forth.
- Provides means to monitor the operational status of an RFID Reader, such as the number of tags read, status of communication channels, health monitoring, antenna connectivity, transmit power levels, and so forth.
- Provides means for an RFID Reader to notify management stations of potential operational problems.
- Provides means to control configuration of an RFID Reader, such as enabling/disabling specific antennas or features, and so forth.
- May provide means to access RFID Reader management functions including device discovery, identification and authentication, network connectivity management, firmware/software initialization, configuration and updates, and managing reader power consumption.

Note: While we consider certain reader configuration functions (as outlined below) to be part of the reader management protocol, the current version of the Reader Management standard [RM 1.0.1] addresses only reader monitoring functions.

The Reader Management standard [RM 1.0.1] focuses on monitoring reader's operational status and on notifying management stations of potential operational problems. The Discovery, Configuration, and Initialization (DCI) for Reader Operations standard focuses on reader discovery identification, configuration and network connectivity management. These two standards fulfill different and complementary responsibilities of the reader management interface.

Management of roles above the RFID Reader role is not currently addressed by EPCglobal standards, but may be considered in the future as warranted.

9.1.7 Reader Management (Role)

Responsibilities:

- Monitors the operational status of one or more RFID Readers within a deployed infrastructure.
- Provides mechanisms for RFID Readers to alert management stations of potential issues
- Manages the configuration of one or more RFID Readers.
- Carries out other RFID Reader management functions including device discovery, authentication, firmware/software configuration and updates, and managing reader power consumption.

9.1.8 Filtering & Collection (Role)

The Filtering & Collection role coordinates the activities of one or more RFID Readers that occupy the same physical space and which therefore have the possibility of radio-frequency interference. It also raises the level of abstraction to one suitable for application business logic.

Responsibilities:

- Receives raw tag reads from one or more RFID Readers.
- Carries out processing to reduce the volume of EPC data, transforming raw tag reads into streams of events more suitable for application logic than raw tag reads. Examples of such processing include filtering (eliminating some EPCs according to their identities, such as eliminating all but EPCs for a specific object class), aggregating over time intervals (eliminating duplicate reads within that interval), grouping (e.g., summarizing EPCs within a specific object class), counting (reporting the number of EPCs rather than the EPC values themselves), and differential analysis (reporting which EPCs have been added or removed rather than all EPCs read).
- Carries out an applicationôs requirements for writing, locking, killing, or otherwise operating upon tags by performing writes or other operations on one or more RFID Readers.
- Determines which processing operations as described above may be delegated to the RFID Reader, and which must be performed by the Filtering & Collection role itself. Implicit in this responsibility is that the Filtering & Collection role knows the capabilities of associated RFID Readers.
- Decodes raw tag values read from tags into URI representations defined by the Tag Data Standard, and conversely encodes URI representations into raw tag values for writing. May use the Tag Data Translation Interface (Section 9.1.21) to obtain machine-readable rules for doing so.
- Maps between ñlogical reader namesò and physical resources such as reader devices and/or specific antennas.
- May provide decoding and encoding of non-EPC tag data in Tag user memory or other memory banks.
- When the Filtering & Collection role is accessed by more than one client application, mediates between multiple client application requests for data when those requests involve the same set or overlapping subsets of RFID Readers.
- May set and control the strategy for finding tags employed by RFID Readers.
- May coordinate the operation of many readers and antennas within a local region in which RFID Readers may affect each other's operation; e.g., to minimize interference. For example, this role may control when specific readers are activated so that physically adjacent readers are not activated simultaneously. In another example, this role may make use of reader- or Tag Air Interface-specific features, such as the ñsessionsò feature of the UHF Class 1 Gen 2 Tag Air Interface, to minimize interference.

The Filtering & Collection role has many responsibilities. The EPCglobal Architecture Framework currently provides standard interfaces to access some, but not all, of these responsibilities. Specifically:

- The Filtering & Collection (ALE) 1.1 Interface (Section 9.1.9), provides standard interfaces that support use cases in which tags are inventoried, read, written or killed, in which the kill or lock passwords are maintained, and in which ñuser dataò or TID memory on the tags is read or written. It also provides management interfaces for maintaining mappings between logical reader names and physical resources, for defining symbolic names for tag data fields, and for securing the use of the ALE interface by clients.
- Other aspects of managing the Filtering & Collection role are not addressed by any EPCglobal standard. This includes controlling aspects of coordinating the activities of multiple readers to minimize interference, setting parameters that govern inventorying strategies, control over Tag Air Interface-specific features, and so on. Products of Solution Providers that implement the ALE 1.1 Interface may provide these features through vendor extensions to the ALE 1.1 Interface or through proprietary interfaces.

9.1.9 Filtering & Collection (ALE) Interface (Interface)

The Filtering & Collection (ALE) 1.1 Interface provides standard interfaces to the Filtering & Collection role.

Normative references:

- Ratified EPCglobal Standard: [ALE1.1.1]

Responsibilities ("data plane"):

- Provides means for one or more client applications to request EPC data from one or more Tag sources.
- Provides means for one or more client applications to request that a set of operations be carried out on Tags accessible to one or more Tag sources. Such operations including writing, locking, and killing.
- Insulates client applications from knowing how many readers/antennas, and what makes and models of readers are deployed to constitute a single, logical Tag source.
- Provides declarative means for client applications to specify what processing to perform on EPC data, including filtering, aggregation, grouping, counting, and differential analysis, as described in Section 9.1.8.
- Provides a means for client applications to request data or operations on demand (synchronous response) or as a standing request (asynchronous response).
- Provides means for multiple client applications to share data from the same reader or readers, or to share readersô access to Tags for carrying out other operations, without prior coordination between the applications.

- Provides a standardized representation for client requests for EPC data and operations, and a standardized representation for reporting filtered, collected EPC data and the results of completed operations.

Responsibilities ("control plane"):

- Provides a means for client applications to query and configure the mapping between logical reader names as used in read/write requests and underlying physical resources such as RFID Readers.
- Provides a means for client applications to configure symbolic names for Tag data fields.
- Provides a means for management applications to secure client access to the ALE interface.

9.1.10 EPCIS Capturing Application (Role)

Responsibilities:

- Recognizes the occurrence of EPC-related business events, and delivers these as EPCIS data.
- May coordinate multiple sources of data in the course of recognizing an individual EPCIS event. Sources of data may include filtered, collected EPC data obtained through the Filtering & Collection Interface, other device-generated data such as bar code data, human input, and data gathered from other software systems.
- May control the carrying out of actions in the physical environment, including writing RFID tags and controlling other devices. The EPCIS Capturing Application may use the Filtering & Collection Interface to carry out some of these responsibilities.

9.1.11 EPCIS Capture Interface (Interface)

Normative references:

- Ratified EPCglobal standard: [EPCIS1.0.1]
- Standard in development: [EPCIS1.1]

Responsibilities:

- Provides a path for communicating EPCIS events generated by EPCIS Capturing Applications to other roles that require them, including EPCIS Repositories, internal EPCIS Accessing Applications, and Partner EPCIS Accessing Applications.

9.1.12 EPCIS Query Interface (Interface)

Normative references:

- Ratified EPCglobal standard: [EPCIS1.0.1]
- Standard in development: [EPCIS1.1]

Responsibilities:

- Provides means whereby an EPCIS Accessing Application can request EPCIS data from an EPCIS Repository or an EPCIS Capturing Application, and the means by which the result is returned.
- Provides a means for mutual authentication of the two parties.
- Reflects the result of authorization decisions taken by the providing party, which may include denying a request made by the requesting party, or limiting the scope of data that is delivered in response.

9.1.13 EPCIS Accessing Application (Role)

Responsibilities:

- Carries out overall enterprise business processes, such as warehouse management, shipping and receiving, historical throughput analysis, and so forth, aided by EPC-related data.

9.1.14 EPCIS Repository (Role)

Responsibilities:

- Records EPCIS-level events generated by one or more EPCIS Capturing Applications, and makes them available for later query by EPCIS Accessing Applications.

9.1.15 Core Business Vocabulary (Data Specification)

Normative references:

- Ratified EPCglobal Standard: [CBV1.0]
- Standard in development: [CBV1.1]

Responsibilities:

- Provides standardized identifiers for use in EPCIS data to denote business steps, dispositions, and business transaction types.
- Specifies syntax templates that end users may use to create identifiers for physical objects, locations, and business transactions, for use in EPCIS data.

9.1.16 Drug Pedigree Messaging (Interface)

In an attempt to help ensure only authentic pharmaceutical products are distributed through the supply chain, some regulatory agencies, have implemented or are considering provisions requiring a ñpedigreeò for drug products. Drug Pedigree Messaging is a data sharing interface intended to standardize the sharing of electronic pedigree documents. Although this standard is initially intended to meet regulatory requirements in certain U.S. states, this interface could be extended to meet the needs of other geographies and regulatory agencies in the future. Flexibility was built into the pedigree schema to allow for multiple interpretations of the existing and possible future, state, federal and even international laws.

A pedigree is a certified record that contains information about each distribution of a prescription drug. It records the creation of an item by a pharmaceutical manufacturer, any acquisitions and transfers by wholesalers or re-packagers, and final transfer to a pharmacy or other entity administering or dispensing the drug. The pedigree contains product information, transaction information, distributor information, recipient information, and signatures.

It is important to point out that the use of ePedigree schema does not require an EPC. The schema can be used even if products are not serialized.

It is also important to note that a complete ePedigree document will not be created by issuing a query to the product network and assembling it from various components; rather, it will travel through the supply chain together with the product and gather the required digitally signed information along the way.

Normative references:

- Ratified EPCglobal Standard: [Pedigree1.0]

Responsibilities:

- Specifies a formal collection of XML schemas and associated usage guidelines under a Drug Pedigree Standard that can be adopted by members of the pharmaceutical supply chain.

9.1.17 Object Name Service (ONS) Interface (Interface)

Normative references:

- Ratified EPCglobal Standard: [ONS2.0.1]

Responsibilities:

- Provides a means for looking up a reference to an EPCIS service or other service associated with an EPC. The list of services associated with an EPC is maintained by the Issuing Organization for that EPC, and typically includes services operated by the organization that commissioned the EPC (often, but not always, the manufacturer; see Section 5.2).

9.1.18 Local ONS (Role)

Responsibilities:

- Fulfills ONS lookup requests for EPCs within the control of the enterprise that operates the Local ONS; that is, EPCs for which the enterprise is the Issuing Organization.

See also the discussion of ONS in Section 7.3.

9.1.19 ONS Root (EPC Network Service)

Responsibilities:

- Provides the authoritative source of data for the root of the hierarchical ONS lookup.
- May provide the initial point of contact for ONS lookups, if the information is not available locally in the DNS resolver cache.
- In most cases, delegates the remainder of the data authority and lookup operation to a Local ONS operated by the Issuing Organizationfor the requested EPC.
- May completely fulfill ONS requests in cases where there is no local ONS to which to delegate a lookup operation.

See also the discussion of ONS in Section 7.3.

9.1.20 Number Block Assignment (EPC Network Service)

Responsibilities:

- Ensures global uniqueness of EPCs by associating an Issuing Agency with each EPC scheme.
- Ensures global uniqueness of EPCs by requiring each Issuing Agency to maintain uniqueness of EPC number blocks assigned to End Users
- Each Issuing Agency assigns new EPC blocks as required by End Users.

9.1.21 Tag Data Translation (Interface and Data Specification)

Normative references:

- Ratified EPCglobal Standard: [TDT1.6]

Responsibilities:

- Provides machine-readable files that define how to translate between EPC encodings defined by the EPC Tag Data Standard (Section 9.1.2). EPCglobal provides these files for use by End Users, so that components of their infrastructure may automatically become aware of new EPC formats as they are defined.

9.1.22 Discovery Services (EPC Network Service – In Development)

At the time of writing, Discovery standards are still under technical development within EPCglobal and it is expected that the standard will not be ratified until late 2011. The EPCglobal Community has completed drafting requirements for the Discovery standards and services, following the GS1 Global Standards Management Process. This has resulted in over sixty specific user requirements and fundamental principles for Discovery Services, organized in ten categories, covering Trust in the Network, Data Integrity & Confidentiality, Data Ownership & Management, Data in Discovery Services, Query Framework, Query Criteria, Identifiers and Pointers, End-to-end traceability and resilience, Scalability and Communication and Access Control.

As a placeholder in this document, ñDiscovery Servicesò is labeled an EPC Network Service, but the final set of responsibilities may be addressed by a combination of EPC Network Services and EPCglobal Standards leading to services operated by End Users and independent Solution Providers. A fundamental principle in the Data Discovery requirements is that end users should have a choice of Discovery Service providers and that there should be mechanisms to allow independent auditing of Discovery Service operators, as well as mechanisms to allow users to migrate their data and access control policies from one Discovery Service provider to another.

Discovery provides a means to locate EPCIS Services and other kinds of EPC-related information resources in the most general situations arising from multi-party supply chains or product lifecycles, in which several different organizations may have relevant data about an EPC but the identities of those organizations are not known in advance. The responsibilities of Discovery include the following.

Responsibilities:

- Facilitate visibility by providing a lookup mechanism to help find multiple sources of information related to serial-level unique identifiers (e.g., EPCs), particularly when that information is provided by multiple parties, is commercially sensitive and/or not published in the public domain.
- The results of a Discovery Service query will typically provide a set of one or more URLs, each accompanied by an indication of the type of service to which they correspond; such service types may indicate EPCIS interfaces, web pages, web services, additional Discovery Services as well as other kinds of services.
- Provides a means to allow parties to mutually identify and authenticate each other.
- Provides a means to share information necessary for authorizing access to EPCIS service listings and EPCIS data. May provide a means to securely pass authorization rules among parties.
- May provide a cache for selected EPCIS data for the purposes of resilient traceability or avoiding unnecessary cascading of queries.

As described above, the Object Name Service (ONS) (Section 9.1.16) is a lookup service useful to find the address of the EPCIS service designated by the Issuing Organization of an EPC. ONS does not address the issues of discovering the set of EPCIS data sources that may contain information about a particular EPC or set of EPCs. ONS and Discovery co-exist and serve different roles in the EPCglobal architecture.

Discovery does not address the storage, sharing, access authorization, or reporting of EPC observation data provided by EPCIS, except as noted above. However, because of the commercial sensitivity of serial-level data, particularly when it is held within a service to which multiple parties have access, a flexible and granular security framework will be developed for Discovery Services, wherever possible leveraging existing standards and state of the art technologies. The technical work group envisages a modular internal architecture for Discovery Services, providing the possibility of interfacing with external security services, where necessary.

10 Data Protection in the EPCglobal Architecture Framework

10.1 Overview

This section describes and assesses the data protection and security mechanisms within the EPCglobal architecture. It provides general information for EPCglobal members wishing to gain a basic understanding of the data protection provisions within the EPCglobal Architecture Framework.

This document does not contain a security analysis of the EPCglobal architecture or any systems based on the EPCglobal architecture. Security analysis requires not only detailed knowledge of the data communications standards, but also the relevant use cases, organizational process, and physical security mechanisms. Security analyses are left to the owners and users of the systems built using the EPCglobal Architecture Framework.

Section 10.2 introduces security concepts. Section 10.3 describes the data protection mechanisms defined within the existing EPCglobal ratified standards.

10.2 Introduction

Security is the process by which an organization or individual protects its valuable assets. In general, assets are protected to reduce the risk of an attack to acceptable levels, with the elimination of risk an often unrealizable extreme. Because the level of acceptable risk differs widely from application to application, there is no standard security solution that can apply to all systems. The EPCglobal architecture framework cannot be pronounced secure or insecure, nor can an individual standard or service.

Data security is commonly subdivided into attributes: confidentiality, integrity, availability, and accountability. Data confidentiality is a property that ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes. Data integrity is the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner during transport or storage. Data availability is a property of a system or a system resource being accessible and usable upon demand by an authorized system entity. Accountability is the property of a system (including all of its system resources) that ensures that the actions of a system entity may be traced uniquely to that entity, which can be held responsible for its actions [RFC2828].

Security techniques like encryption, authentication, digital signatures, and non-repudiation services are applied to data to provide or augment the system attributes described above.

As ñsecurityò cannot be evaluated without detailed knowledge of the entire system, we focus our efforts to describe the data protection methods within the EPCglobal Standards. That is, we describe the mechanisms that protect data when it is stored, shared and published within EPCglobal Standards and relate these mechanisms to the system attributes described above.

10.3 Existing Data Protection Mechanisms

This section summarizes the existing data protection mechanism within the standards and standards forming the EPCglobal Architecture Framework.

10.3.1 Network Interfaces

Many of the standards within the EPCglobal framework are based on network protocols that communicate EPC information over existing network technology including TCP/IP networks. This section summarizes the data protection mechanisms described within the interface standards.

Some network standards within EPCglobal rely on Transport Layer Security [RFC2246] [RFC4346] as part of their underlying data protection mechanism. TLS provides a mechanism for the client and server to select cryptographic algorithms, exchange certificates to allow authentication of identity, and share key information to allow encrypted and validated data sharing. Mutual authentication within TLS is optional. Typically, TLS clients authenticate the server, but the client remains unauthenticated or is authenticated by non-TLS means once the TLS session is established. The protection provided by TLS depends critically on the cipher suite chosen by the client and server. A Cipher suite is a combination of cryptographic algorithms that define the methods of encryption, validation, and authentication.

Some EPCglobal Standards rely on HTTPS (HTTP over TLS) for data protection. HTTPS [RFC2818] is a widely used standard for encrypting sensitive content for transfer over the World Wide Web. In common web browsers, the ñsecurity lockò shown on the task bar indicates that the transaction is secured using HTTPS. HTTPS is based on TLS (Transport Layer Security). A HTTPS client or endpoint acting as the initiator of the connection, initiates the TLS connection to the server, establishes a secure and authenticated connection and then commences the HTTP request. All HTTP data is sent as application data within the TLS connection and is protected by the encryption mechanism negotiated during the TLS handshake. The HTTPS specification defines the actions to take when the validity of the server is suspect. Using HTTPS, client and server can mutually authenticate using the mechanisms provided within TLS. However, another approach (and the one more frequently used) is for the client to authenticate the server within TLS, and then the server authenticates the client using HTTP-level password-based authentication carried out over the encrypted channel established by TLS.

All of the data protection methods below are specified as optional behaviors of devices that comply with the relevant network interface standards. An enterprise must make the specific decision on whether these data protection mechanisms are valuable within their systems.

10.3.1.1 Application Level Events 1.1 (ALE)

The ALE 1.1 standard describes the interface to the Filtering and Collection Role within the EPCglobal architecture framework. It provides an interface to obtain filtered, consolidated EPC data from variety of EPC sources. For a complete description of the ALE 1.1 standard, see [ALE1.1.1].

ALE is specified in an abstract manner with the intention of allowing it to be carried over a variety of transport methods or bindings. The ALE 1.1 standard provides a SOAP [SOAP1.2] binding of the abstract protocol compliant with the Web Services Interoperability (WS-I) Basic Profile version 1.0 [WSI]. SOAP provides a method to share structured and typed information between peers. WS-I provides interoperability guidance for web services. SOAP is typically carried over HTTP and security based on HTTPS is permitted by the WS-I Basic Profile. ALE can utilize this SOAP/HTTPS binding for the ALE messages and responses to provide authentication and transport encryption. Authentication and encryption mechanisms together provide for confidentiality and integrity of the shared data.

The ALE interface also provides a callback interface for events that are delivered asynchronously. . Several protocol bindings for callbacks are specified. The HTTPS binding of the callback interface provides for delivery of reports in XML via the HTTP protocol using POST operation secured via TLS. The HTTPS protocol provides link-level security, and optionally mutual authentication between an ALE implementation and its callback receivers.

ALE 1.1 specifies an Access Control API over which administrative clients may define the access rights of other clients to use the facilities provided by the other ALE APIs. This API provides a standardized, role-based way to associate access control permissions with ALE client identifiers. This API can be used to restrict the operations that can be performed by clients (e.g. defining an event cycle) and also can restrict the data available to a client (e.g. restrict EPC data to a subset of the available logical readers).

10.3.1.2 Reader Protocol 1.1 (RP)

The current RP 1.1 standard provides a standard communication link between device providing services of a reader, and the device proving Filtering and Collection (F & C) of RFID data. For a complete description, see [RP1.1]

The RP protocol supports the optional ability to encrypt and authenticate the communications link between these two devices when using certain types of communication links (transports). For example, HTTPS can be used as an alternative to HTTP when desiring a secure communication link between reader and host for Control Channels (initiated by a host to communicate with a reader) and/or Notification Channels (initiated by a reader to communicate with a host). This information is relevant to the authentication of the RP communications as the cipher suite provided requires only server authentication. The RP standard provides information and guidance for those desiring secure communication links when using other defined transports; see the RP standard for more details.

10.3.1.3 Low Level Reader Protocol 1.1 (LLRP)

The LLRP protocol supports the optional ability to encrypt and authenticate the communications link between these two devices using TLS. If X.509 certificates are used for authentication, LLRP requires certificates compliant with X.509 Certification Profile. Using TLS for LLRP Reader and Client communications provides the following protections:

- Readers only talk to authorized clients
- Clients only talk to authorized readers
- No other party can read the LLRP messages (privacy protection) or inject/modify messages without being detected (integrity protection).

Note that the strength of the protection depends on the negotiated cipher suites.

10.3.1.4 Reader Management 1.0.1 (RM)

The reader management standard describes wire protocol used by management software to monitor the operating status and health of EPCglobal compliant tag Readers. For a complete description, see [RM1.0.1].

RM divides its standard into three distinct layers: reader layer, messaging layer, and transport layer. The reader layer specifies the content and abstract syntax of messages exchanged between the Reader and Host. This layer is the heart of the Reader Management Protocol, defining the operations that Readers expose to monitor their health. The messaging layer specifies how messages defined in the reader layer are formatted, framed, transformed, and carried on a specific network transport. Any security services are supplied by this layer. The transport layer corresponds to the networking facilities provided by the operating system or equivalent.

The current RM standard defines two implementations of the messaging layer or message transport bindings: XML and (Simple Network Management Protocol) SNMP. The XML binding follows the same conventions as RP described in section 10.3.1.2. The RM SNMP MIB is specified using SMIv2 allowing use of SNMP v2 [RFC1905] or SNMP v3 [RFC3414]. SNMP v2c has weak authentication using community strings which are sent in plain-text within the SNMP messages. SNMP v2c contains no encryption mechanisms. SNMP v3 has strong authentication and encryption methods allowing optional authentication and optional encryption of protocol messages.

10.3.1.5 EPC Information Services 1.0.1 (EPCIS)

EPCIS provides EPC data sharing services between disparate applications both within and across enterprises. For a complete description of EPCIS, see [EPCIS1.0.1]

EPCIS contains three distinct service interfaces, the EPCIS capture interface, the EPCIS query control interface, and the EPCIS query callback interface (The latter two interfaces are referred to collectively as the EPCIS Query Interfaces). The EPCIS capture interface and the EPCIS query interfaces both support methods to mutually authenticate the partiesô identities.

Both the EPCIS capture interface and the EPCIS query interface allow implementations to authenticate the client's identity and make appropriate authorization decisions based on that identity. In particular, the query interface specifies a number of ways that authorization decisions may affect the outcome of a query. This allows companies to make very fine-grain decisions about what data they want to share with their trading partners, in accordance with their business agreements.

The EPCIS standard includes a binding for the EPCIS query interface (both the query control and query callback interfaces) using AS2 [RFC4130] for communication with external trading partners. AS2 provides for mutual authentication, data confidentiality and integrity, and non-repudiation. The EPCIS standard also includes WS-I compliant SOAP/HTTP binding for the EPCIS query control interface. This may be used with HTTPS to provide security. The EPCIS standard also includes an HTTPS binding for the EPCIS query callback interface.

10.3.2 EPC Network Services

EPCglobal and other organizations provide EPC Network Services. The following section describes the data protection methods employed by these services.

10.3.2.1 Object Name Service 2.0 (ONS)

The ONS service is based on the current internet Domain Name System (DNS). ONS provides authoritative lookup of information about an electronic identifier. See [ONS2.0.1] for a complete description.

Users query the ONS server with an EPC (represented as a URI and translated into a domain name). ONS returns the requested data record which contains address information for services that may contain information about the particular EPC value. ONS does not provide information for individual EPCs; the lowest granularity of service is based on the object class of the EPC. ONS delivers only address information. The corresponding services are responsible for access control and authorization.

The current Internet DNS standard provides a query interface. Users query the DNS server for information about a particular domain name, and the domain server returns information for the domain name in question. The system is a hierarchical set of DNS servers, culminating at the root DNS, serving addresses for the entire Internet community. As the DNS infrastructure is designed to provide address lookup service for all users of the internet, there is no encryption mechanism built into DNS/ONS. Any user wishing to gain Internet address information, can query DNS/ONS directly, hence the encryption of DNS traffic would have little or no benefit.

New records are added to ONS manually, by electronic submission via a web interface. These submissions are protected by ACL (access control list) and by shared secret (password).

For a complete security analysis of DNS, see [RFC3833].

10.3.2.2 Discovery Services

Discovery Services are currently under development, and so the security mechanisms are still to be determined. Detailed user requirements have been captured and documented by the Data Discovery JRG, regarding Data Integrity & Confidentiality, Data Ownership and Access Control. The Data Discovery JRG took particular care to consider the perspectives of both the information provider (and the sensitivity of revealing the link between a specific EPC and a specific EPCIS resource) and also the sensitivity of the client's query to a Discovery Service (which itself may indicate which EPCs a specific company is handling).

The technical work group for Discovery Services is using these requirements as the foundation for its work on the security framework for Discovery Services and, wherever possible, is leveraging established tried and tested best practices and existing open standards for security.

10.3.2.3 Number Assignment

Number assignment is provided as an EPC Network Service. These documents are provided as standard text files on a public web site operated by GS1. Currently, these files contain only a list of the assigned GS1 Company Prefixes, and do not contain any information on the assignee of each ID.

10.3.3 Tag Air Interfaces

A Tag Air Interface specifies the Radio Frequency (RF) communications link between a reader device and an RFID tag. This interface is used to write and read data to and from an RFID tag.

In general, transmitted RF energy is susceptible to eavesdropping or modification by any device within range of the intended receiver. To this end, each Tag Air Interface may have various countermeasures to protect the data transmitted across the interface specific to the application of the particular standard.

10.3.3.1 UHF Class 1 Generation 2 (C1G2 or Gen2)

The Class 1 Generation 2 Tag Air Interface standard specifies a UHF Tag Air Interface between readers and tags. The interface provides a mechanism to write and read data to and from an RFID tag respectively. A tag complying with the Gen2 standard can have up to four memory areas which store the EPC and EPC related data: EPC memory, User memory, TID memory, and reserved memory. For a complete description of the Gen2 Tag Air Interface see [UHFC1G21.2.0].

The Gen2 Tag Air Interface, as its name professes, is the second generation of Class 1 Tag Air Interfaces considered by EPCglobal. To this end, many of the security concerns of previous generation Tag Air Interfaces were well understood during the development of Gen2.

The following describes the key data protection features of the Gen2 Tag Air Interface.

10.3.3.1.1 Pseudonyms

Class 1 Tags are passive devices that contain no power source. Tags communicate by backscattering energy sent by the interrogator or reader device. This phenomenon leads to an asymmetric link, where a very high energy signal is sent on the forward link from the interrogator to the tag. The tag responds by backscattering a very small portion of that energy on the reverse link, which can be detected by the interrogator, forming a bi-directional half-duplex link.

Depending on the regulatory region, antenna characteristics, and propagation environment, the high power forward link can be read hundreds to thousands of meters away from the interrogator source. The much lower power reverse link, often with only one millionth the power of the forward link, can typically be observed only within 10ôs of meters of the RFID tag.

To prevent the transmission of EPC information over the forward link, the Gen2 standard employs pseudonyms, or temporary identities for communication with tags. A pseudonym for a tag is used only within a single interrogator interaction. The interrogator uses this pseudonym for communication with the tag rather than the tagôs EPC or other tag data. The EPC is only presented in the interface on the backscatter link, limiting the range of eavesdropping to the range of backscatter communications. Eavesdroppers are still able to obtain EPC information during tag singulation, but cannot obtain this information from the high power forward link.

Gen2 provides a select command which allows an interrogator to identify a subset of the total tag population for inventory. Using the select command requires the interrogator to transmit the forward link the bit pattern to match within the tag memory. Forward link transmission of this bit pattern may compromise the effectiveness of the pseudonym.

10.3.3.1.2 Cover Coding

For the same reasons described above, it may be undesirable to transmit non-EPC tag data on the forward link. To this end, Gen2 includes a technique called cover coding to obscure passwords and data transmitted to the tag on the forward link. Cover coding uses one-time-pads, random data backscattered by the tag upon request from the interrogator. Before sending data over the forward link, the interrogator requests a random number from the tag, and then uses this one-time-pad to encrypt a single word of data or password sent on the forward link.

An observer of the forward communications link would not be able to decode data or passwords sent to the tag without first ñguessingò the one-time-pad. Gen2 specifies that these pads can only be used a single time.

An observer of the forward and reverse link would be able to observe the one-time-pads backscattered by the tag to the interrogator. This, in combination with the encryption method specified in Gen2 would allow this observer to decode all data and passwords sent on the forward link from the interrogator to the tag.

Gen2 specifies an optional Block Write command which does not provide cover coding of the data sent over the forward link. Block write enables faster write operations at the expense of forward link security.

10.3.3.1.3 Memory Locking

Gen2 contains provisions to temporarily or permanently lock or unlock any of its memory banks.

User, TID, and EPC memory may be write locked so that data stored in these memory banks cannot be overwritten. Reading of the TID, EPC and User memory banks are always permitted. There is no method to read-lock these memory banks. This memory can be temporarily or permanently locked or unlocked. Once permanently locked, memory cannot be written. When locked but not permanently locked, memory can be written, but only after the interrogator provides the 32-bit access password.

Reserved memory currently specifies the location of two passwords: the access password and kill password. In order to prevent unauthorized users from reading these passwords, an interrogator can individually lock their contents. Locking of a password in reserved memory renders it un-writeable and un-readable. The read locking and write locking of password memory is not independent, e.g. memory cannot be write-locked without also being read-locked. A password can be temporarily or permanently locked or unlocked. Once permanently locked, memory cannot be written or read. When locked but not permanently locked, memory can be read and written only after the interrogator furnishes the 32-bit access password.

10.3.3.1.4 Kill Command

Gen2 contains a command to ñkillò the tag. Killing a tag sets it to a state where it will never respond to the commands of an interrogator. To kill a tag, an interrogator must supply the 32-bit kill passwords. Tags with a zero-valued kill password cannot be killed. By perma-locking a zero valued kill password, tags can be rendered un-killable. By perma-unlocking the kill password, a tag can be rendered always killable.

10.3.4 Data Format

10.3.4.1 Tag Data Standard (TDS)

The Tag Data Standard, currently Version 1.6, specifies the data format of the EPC information, both in its pure identity URI format and the binary format typically stored on an RFID tag. The TDS standard provides encodings for numbering schemes within an EPC, and does not provide encodings or standard representations for other types of data. For a complete description of the TDS standard, see [TDS1.8]

RFID users are sometimes concerned with transmitting or backscattering EPC information that can directly infer the product or manufacturer of the product. Current Tag Air Interface standards do not provide mechanisms to secure the EPC data from unauthorized reading.

TDS allows for the encoding of data types that contain manufacturer or company prefix, object class, and serial number. TDS also specifies encoding of formats that contain company prefix and serial number, but do not contain object class information.

The TDS standard does not provide any encoding formats that standardize the encryption or obstruction of the manufacturer, product identification, or any other information stored on the RFID tag.

10.3.5 Security

Several EPCglobal Standards were created specifically to address security issues of shared data.

10.3.6 EPCglobal X.509 Certificate Profile

The authentication of entities (end users, services, physical devices) serves as the foundation of any security function incorporated into the EPCglobal Architecture Framework. The EPCglobal Architecture Framework allows the use of a variety of authentication technologies across its defined interfaces. It is expected, however, that the X.509 authentication framework will be widely employed. To this end, the EPCglobal Security 2 Working Group produced the EPCglobal X.509 Certificate profile. The certificate profile serves not to define new functionality, but to clarify and narrow functionality that already exists. For a complete description, see [Cert2.0]

The certificate profile provides a minimum level of cryptographic security and defines and standardizes identification parameters for users, services/server and device.

10.3.7 EPCglobal Electronic Pedigree

EPCglobal electronic pedigree provides a standard, interoperable platform for supply chain partner compliance with state, regional and national drug pedigree laws. It provides flexible interpretation of existing and future pedigree laws.

In the United States, current legislation in multiple states dictates the creation and updating of electronic pedigrees at each stop in the pharmaceutical supply chain. Each state law specifies the data content of the electronic pedigree and the digital signature standards but none of them specifies the actual format of the document. The need for a standard electronic document format that can be updated by each supply chain participant is what has driven the creation of the standard.

The Standard does not identify exactly how pedigree documents must be transferred between trading partners. Any mechanism chosen must provide document immutability, non-repudiation and must be secure and authenticated. Although the scope of the standard focuses on the pedigree and pedigree envelope interchange formats, secure transmission relies on the recommendations for securing pedigree transmissions defined by the HLS Information Work Group.

11 References

[ALE1.1.1] EPCglobal, ñThe Application Level Events (ALE) Specification, Version 1.1; Part 1: Core Specificationò EPCglobal Ratified Standard, March 2009, http://www.gs1.org/gsmp/kc/epcglobal/ale/ale_1_1_1-standard-core-20090313.pdf.

[CBV1.0] EPCglobal, ñCore Business Vocabulary Specification, Version 1.0,ò EPCglobal Ratified Standard, October 2010, http://www.gs1.org/gsmp/kc/epcglobal/cbv/cbv_1_0-standard-20101013.pdf.

[CBV1.1] EPCglobal, ñCore Business Vocabulary Specification, Version 1.1,ò EPCglobal standard in development.

[Cert2.0] EPCglobal, ñEPCglobal Certificate Profile 2.0,ò EPCglobal Ratified Standard, August 2010, http://www.gs1.org/gsmp/kc/epcglobal/cert/cert_2_0-standard-20100610.pdf.

[CLASS1] Engels, D.W. and Sarma S.E, ñStandardization Requirements within the RFID Class Structure Frameworkò, MIT Auto-ID Labs Technical Report, January 2005.

[DCI] EPCglobal, ñDiscovery, Configuration, and Initialization (DCI) for Reader Operationsò, EPCglobal Ratified Standard, June 2009, http://www.gs1.org/gsmp/kc/epcglobal/dci/dci_1_0-standard-20090610.pdf.

[EPCIS1.0.1] EPCglobal, ñEPC Information Services (EPCIS) Version 1.0.1 Specification,ò EPCglobal Ratified Standard, September 2007, http://www.gs1.org/gsmp/kc/epcglobal/epcis/epcis_1_0_1-standard-20070921.pdf.

[EPCIS1.1] EPCglobal, ñEPC Information Services (EPCIS) Version 1.1 Specification,ò EPCglobal standard in development.

[GS1GS] GS1, ñGeneral Specifications Version 13,ò January 2013.

[GS1SA] GS1, ñGS1 System Architecture,ò March 2013, http://www.gs1.org/docs/gsmp/architecture/GS1_System_Architecture.pdf

[GS1SL] GS1, ñGS1 System Landscape,ò March 2013, http://www.gs1.org/docs/gsmp/architecture/GS1_System_Landscape.pdf

[HFC1] EPCglobal, ñEPC Radio-Frequency Identity Protocols EPC Class-1 HF RFID Air Interface Protocol for Communications at 13.56MHz,Version 2.0.3,ò EPCglobal Ratified Standard, September, 2011, http://www.gs1.org/sites/default/files/docs/epcglobal/epcglobal_hf_2_0_3-standard-20110905r3.pdf

[ISO19762-3] ISO/IEC, ñInformation technology ð Automatic identification and data capture (AIDC) techniques ð Harmonized vocabulary ð Part 3: Radio frequency identification (RFID),ò ISO/IEC International Standard, March, 2005.

[LLRP1.1] EPCglobal, ñEPCglobal Low Level Reader Protocol (LLRP), Version 1.1ò, Ratified EPCglobal Standard, October 2010, http://www.gs1.org/gsmp/kc/epcglobal/llrp/llrp_1_1-standard-20101013.pdf.

[ONS2.0.1] EPCglobal, ñEPCglobal Object Naming Service (ONS), Version 2.0,ò EPCglobal Ratified Standard, December 2012, http://www.gs1.org/sites/default/files/docs/epcglobal/standards/ONS-2_0_0-Standard-i1%202012Dec20.pdf.

[Pedigree1.0] EPCglobal, ñPedigree Ratified Standard, Version 1.0,ò EPCglobal Ratified Standard, January, 2007, http://www.gs1.org/gsmp/kc/epcglobal/pedigree/pedigree_1_0-standard-20070105.pdf.

[RFC1034] P. V. Mockapetris, ñDomain names ï concepts and facilities.ò RFC1034, November 1987, http://www.ietf.org/rfc/rfc1034.

[RFC1035] P. V. Mockapetris, ñDomain names ï implementation and specification.ò RFC1035, November 1987, http://www.ietf.org/rfc/rfc1035.

[RFC1905] J. Case, K. McCloghrie, M. Rose, S. Waldbusser, ñProtocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)ò, RFC 1905, January 1996.

[RFC2246] T. Dierks, ñThe TLS Protocol Version 1.0ò, RFC 2246, January 1999, http://www.ietf.org/rfc/rfc2246.

[RFC2818] P. Rescorla, ñHTTP Over TLSò, RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.

[RFC2828] R. Shirey, ñInternet Security Glossaryò, RFC 2828, May 2000, http://www.ietf.org/rfc/rfc2828.

[RFC3414] U. Blumenthal, ñUser-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)ò, RFC 3414, December 2002 http://www.ietf.org/rfc/rfc3414.

[RFC3833] D Atkins, ñThreat Analysis of the Domain Name System (DNS)ò, RFC 3833, August 2004, http://www.ietf.org/rfc/rfc3833.

[RFC4130] D. Moberg and R. Drummond, ñMIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP, Applicability Statement 2 (AS2),ò RFC4130, July 2005, http://www.ietf.org/rfc/rfc4130.

[RFC4346] T. Dierks, ñThe Transport Layer Security (TLS) Protocol Version 1.1ò, RFC 4346, April 2006, http://www.ietf.org/rfc/rfc4346.

[RM1.0.1] ñReader Management 1.0.1,ò EPCglobal Ratified Standard, May 2007, http://www.gs1.org/gsmp/kc/epcglobal/rm/rm_1_0_1-standard-20070531.pdf.

[SLRRP] P. Krishna, D. Husak, ñSimple Lightweight RFID Reader Protocol,ò IETF Internet Draft, June 2005.

[SOAP1.2] M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. F. Nielsen, ñSOAP Version 1.2,ò W3C Recommendation, June 2003, http://www.w3.org/TR/soap12.

[TDS1.8] EPCglobal, ñEPCglobal Tag Data Standards Version 1.8,ò EPCglobal Ratified Standard. February 2014, http://www.gs1.org/sites/default/files/docs/tds/TDS_1_8_Standard_20140203.pdf

[TDS1.9] EPCglobal, ñEPCglobal Tag Data Standards Version 1.9,ò EPCglobal standard in development

[TDT1.6] EPCglobal, ñEPCglobal Tag Data Translation (TDT) 1.6,ò EPCglobal Ratified Standard, October 2011, http://www.gs1.org/gsmp/kc/epcglobal/tdt/tdt_1_6_RatifiedStd-20111012-i2.pdf.

[UHFC1G21.1.0] EPCglobal, ñEPCÊ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz ï 960 MHz Version 1.1.0,ò EPCglobal Ratified Standard, October 2007, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_1_0-standard-20071017.pdf.

[UHFC1G21.2.0] EPCglobal, ñEPCÊ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz ï 960 MHz Version 1.2.0,ò EPCglobal Ratified Standard, May 2008, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf.

[UHFC1V2] EPCglobal, ñEPCÊ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz ï 960 MHz Version 2.0.0 Ratified,ò GS1 Ratified Standard, November 2013, http://www.gs1.org/sites/default/files/docs/uhfc1g2/uhfc1g2_2_0_0_standard_20131101.pdf.

[WSI] K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, P. Yendluri, ñBasic Profile Version 1.0,ò WS-I Final Material, April 2004, http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html

12 Glossary

This section provides a summary of terms used within this document. For fuller definitions of these terms, please consult the relevant sections of the document. See also the whole of Section 9, which defines all roles and interfaces within the EPCglobal Architecture Framework.

Term	Section	Meaning
EPCglobal Architecture Framework	1	A collection of interrelated standards (ñEPCglobal Standardsò), together with services operated by GS1, its delegates, and others (ñEPC Network Servicesò), all in service of a common goal of enhancing business flows and computer applications through the use of Electronic Product Codes (EPCs).
EPCglobal Standards	1	Specifications for hardware and software interfaces through which components of the EPCglobal Architecture Framework interact. EPCglobal Standards are developed by the EPCglobal Community through the EPCglobal Standards Development Process. EPCglobal standards are implemented by systems deployed by End Users. Such systems may be developed by or deployed with the aid of Solution Providers, or they may be developed in-house by End Users themselves. EPCglobal Standards are also implemented by EPC Network Services.
EPC Network Services	1	Network-accessible services, operated by GS1, its delegates, and others, that provide common services to all end users, through interfaces defined as part of the EPCglobal Architecture Framework.

Term	Section	Meaning
EPCglobal Network	1	An informal marketing term used to refer loosely to End Users and their interaction with each other, where that interaction takes place directly through the use of EPCglobal Standards and indirectly through EPC Network Services.
End User	1	A company or other organization that employs EPCglobal Standards and EPC Network Services as a part of its business operations. An End User may or may not be a GS1 member.
Solution Provider	1	A company or other organization that develops products or services that implement EPCglobal Standards, or that implements EPCglobal Standards-compliant systems on behalf of End Users. A Solution Provider may or may not itself be an End User.
EPCglobal Community	1	Collective term for all organizations that participate in developing EPCglobal Standards through the EPCglobal Standards Development Process. The EPCglobal Community includes GS1 members, Auto-ID Labs, the GS1 Global Office, GS1 Member Organizations, and government agencies and NGOs, along with invited experts from other standards organizations and other institutions.
Electronic Product Code (EPC)	1	A unique identifier for a physical object, unit load, location, or other identifiable entity playing a role in business operations. Electronic Product Codes are assigned following rules designed to ensure uniqueness despite decentralized administration of code space, and to accommodate legacy coding schemes in common use. EPCs have multiple representations, including binary forms suitable for use on RFID tags, and text forms suitable for data sharing among enterprise information systems.
Registration Authority	4.1	The organization responsible for the overall structure and allocation of a namespace. In the case of the Electronic Product Code, the Registration Authority is EPCglobal. The Registration Authority delegates responsibility for allocating portions of the namespace to an Issuing Agency.
Issuing Agency	4.1	An organization responsible for issuing blocks of codes within a predefined portion of a namespace. For Electronic Product Codes, Issuing Agencies include GS1 (for GS1 keys such as SGTIN, SSCC, etc) and the US Department of Defense (for DoD codes). An Issuing Agency issues a block of EPCs to an Issuing Organization, who may then commission individual EPCs without further coordination.

Term	Section	Meaning
Issuing Organization	5.2	An End User that has been allocated a block of Electronic Product Codes by an Issuing Agency.
Object Class	5.5	A group of objects that differ only in being separate instances of the same kind of thing; for example, a product type or SKU.
Tag Air Interface	9.1.3	ñA conductor-free medium, usually air, between a transponder and a reader/interrogator through which data communication is achieved by means of a modulated inductive or propagated electromagnetic field.ò [ISO19762-3]

13 Acknowledgements

The following former members of the EPCglobal Architecture Review Committee contributed to earlier versions of this document:

Greg Allgair (formerly of EPCglobal), Leo Burstein (formerly of Gillette), Bryan Rodrigues (formerly of CVS), Johannes Schmidt (formerly of Kraft), Chuck Schramek (formerly of EPCglobal), Roger Stewart (formerly of Intelleflex and AWiD),

The authors would like to thank the following persons and organizations for their comments on earlier versions of this document:

John Anderla (Kimberly Clark), Chet Birger (ConnecTerra), Judy Bueg (Eastman Kodak), Curt Carrender (Alien Technogies), Chris Diorio (Impinj), Andreas Füßler (GS1 Europe), Lim Joo Ghee (Institute for Infocomm Research), Graham Gillen (VeriSign), Sue Hutchinson (EPCglobal), Osamu Inoue (EPCglobal Japan), P. Krishna (Reva Systems), Shinichi Nakahara (NTT), Mike OôShea (Kimberly Clark), Andrew Osborne (GS1 Technical Steering Team), Hidenori Ota (Fujitsu), Tom Pounds (Alien Technologies), Steve Rehling (Procter & Gamble), Steve Smith (Alien Technologies), Suzanne Stuart-Smith (GS1 UK), Hiroyasu Sugano (Fujitsu), Hiroki Tagato (NEC), Neil Tan (UPS), Joseph Tobolski (Accenture), Nicholas Tsougas (US Defense Logistics Agency), Mitsuo Tsukada (NTT), Shashi Shekhar Vempati (Infosys), Ulrich Wertz (MGI METRO Group), Gerd Wolfram (MGI METRO Group), and Ochi Wu (CODEplus).