The EPCglobal Architecture Framework

EPCglobal Final Version 1.2 Approved 10 September 2007

Authors:

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

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

Leo Burstein burstein@compuserve.com

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

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

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

Bernie Hogan (GS1 US) bhogan@gs1us.org

Oleg Ryaboy (CVS) ORyaboy@cvs.com

Sanjay Sarma (MIT) sesarma@mit.edu

Johannes Schmidt (Kraft) johannes.schmidt@kraft.com

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

Ken Traub (Ken Traub Consulting LLC) kt@alum.mit.edu, Editor

John Williams (MIT) jrw@mit.edu

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Abstract

This document defines and describes the EPCglobal Architecture Framework. The EPCglobal Architecture Framework is a collection of hardware, software, and data standards, together with core services that can be operated by EPCglobal, its delegates or third party providers in the marketplace, all in service of a common goal of enhancing business flows and computer applications through the use of Electronic Product Codes (EPCs). 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.
- To define the top level architecture of core services that are operated by EPCglobal and its delegates.
- To explain the underlying principles that have guided the design of individual standards and core service components within the EPCglobal Architecture Framework.
- To provide architectural guidance to end users and technology vendors seeking to implement EPCglobal standards and to use EPCglobal core 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 exchange EPC-related information.
- Enterprise architects and systems integrators that integrate EPC-related processes and applications into enterprise architectures.
- Participants of EPCglobal Working Groups (including Software Action Group, Hardware Action Group and all Business Action 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 at EPCglobal. See `www.epcglobalinc.org` for more information.

This document is an EPCglobal approved document and is available to the general public.

Comments on this document should be sent to the EPCglobal Architecture Review Committee mailing list `arc@lists.epcglobalinc.org`.

Table of Contents

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

1 Introduction

This document defines and describes the EPCglobal Architecture Framework. The EPCglobal Architecture Framework is a collection of interrelated hardware, software, and data standards ("EPCglobal Standards"), together with core services that are operated by EPCglobal and its delegates ("EPCglobal Core Services"), all in service of a common goal of enhancing business flows and computer applications through the use of Electronic Product Codes (EPCs).

The primary beneficiaries of the EPCglobal Architecture Framework are EPCglobal Subscribers and other Solution Providers. An EPCglobal Subscriber is any organization that uses EPCglobal Core Services, or participates in the EPCglobal Standards Development Process to develop EPCglobal Standards. EPCglobal Subscribers may be further classified as End-users or Solution Providers (or both). An End-user is an EPCglobal Subscriber that employs EPCglobal Standards and EPCglobal Core 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 EPCglobal Core Services. (A Solution Provider may or may not itself be an EPCglobal Subscriber.) Informally, the synergistic effect of EPCglobal Subscribers interacting with EPCglobal and with each other using elements of the EPCglobal Architecture Framework is called the "EPCglobal Network."

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 in the EPCglobal Network, including components deployed by individual EPCglobal subscribers as well as EPCglobal Core Services deployed by EPCglobal and its delegates.
- To define the top level architecture of EPCglobal Core Services, which provide common services to all subscribers of the EPCglobal Network, through interfaces defined as part of the EPCglobal Architecture Framework.
- To explain the underlying principles that have guided the design of individual standards and Core Service components within the EPCglobal Network. These underlying principles provide unity across all elements of the EPCglobal Architecture Framework, and provide guidance for the development of future standards and new Core Services.
- To provide architectural guidance to end users and technology vendors seeking to implement EPCglobal Standards and to use EPCglobal Core 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 specifications, or by standards produced by other standards bodies. EPCglobal specifications are developed by EPCglobal membership through the EPCglobal Standards Development Process (SDP) [SDP1.3]. EPCglobal specifications are normative, and implementations are subject to conformance and certification requirements.

  An example of an interface is the UHF Class 1 Gen 2 Tag Air Interface, that specifies a 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 Specification.

- The design of hardware and software components that implement EPCglobal specifications are proprietary to the vendors and end users that create such components. While EPCglobal specifications 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 specifications.

  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 Specification.

- A special case of components that implement EPCglobal specifications are components that are operated and deployed by EPCglobal itself (or by other organizations to which EPCglobal delegates responsibility). These components are referred to as EPCglobal Core Services, and provide services to all EPCglobal subscribers. The design of these components is the responsibility of EPCglobal or its delegates, and design details may be made public at EPCglobal's discretion.

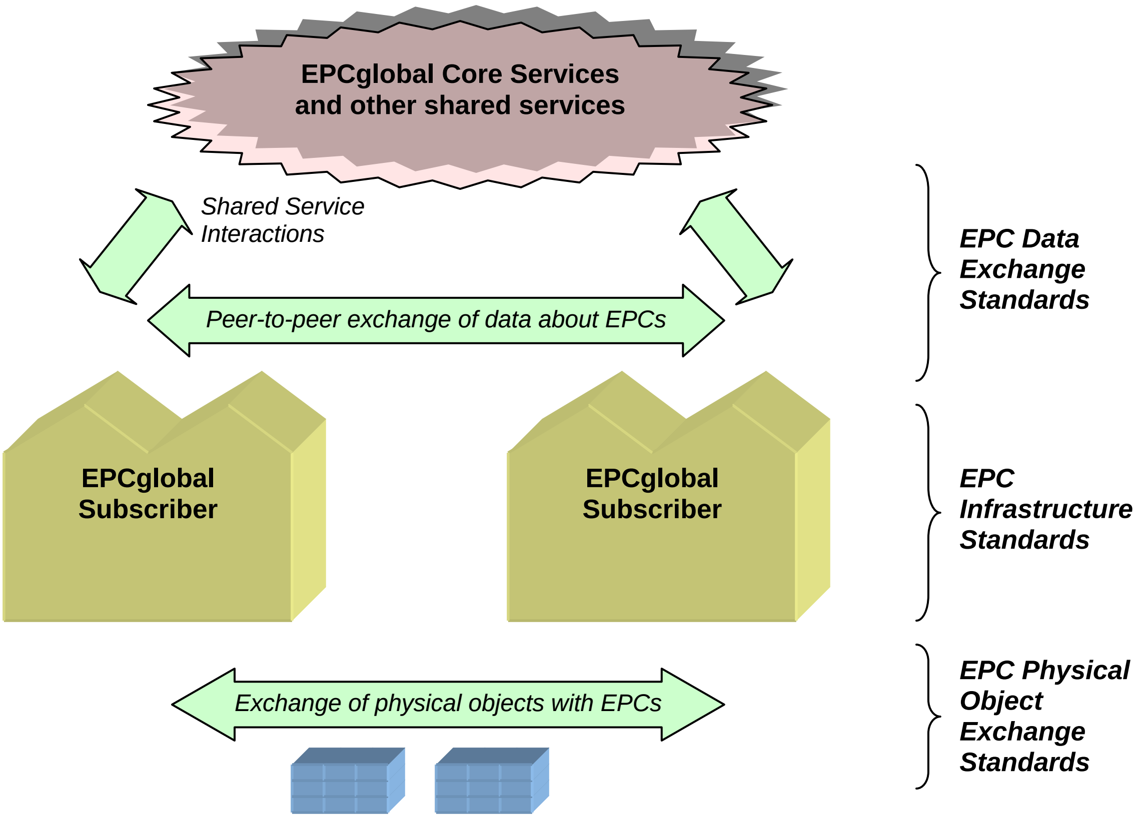
  An example of an EPCglobal Core Service is the Object Name Service (ONS), which provides a logically centralized registry through which an EPC may be associated with information services. The ONS is logically operated by EPCglobal; from a deployment perspective this responsibility is delegated to a contractor of ONS that operates the ONS "root" service, which in turn can delegate responsibility to other services operated by other organizations.

At the time of this writing, there are many parts of the EPCglobal Architecture Framework that are well understood, and for which EPCglobal standards already exist or are currently in development. There are other parts of the EPCglobal Architecture Framework that are less well understood, but where a need is believed to exist based on the analysis of known use cases. In these cases, the architectural approach has not yet been finalized, though architectural analysis is underway within the Architecture Review Committee. Developing standards or designing additional Core Services depends on the definition of a broader collection of use cases and their abstraction into general requirements. This document clearly identifies which parts of the EPCglobal Architecture Framework are understood architecturally and which parts need further work. This document will be the basis for working through and ultimately documenting the architectural decisions around the latter parts as work continues.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

2 Architecture Framework Overview

The diagram below illustrates the activities carried out by EPCglobal Subscribers 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* Subscribers exchange physical objects that are identified with Electronic Product Codes (EPCs). For many end users of the EPCglobal Network, the physical objects are trade goods, the subscribers 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 tagging of objects. The EPCglobal Architecture Framework defines EPC physical object exchange standards, designed to ensure that when one subscriber delivers a physical object to another subscriber, the latter will be able to determine the EPC of the physical object and interpret it properly.

- *EPC Data Exchange* Subscribers benefit from the EPCglobal Network by exchanging data with each other, increasing the visibility they have with respect to the movement of physical objects outside their four walls. The EPCglobal

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Architecture Framework defines EPC data exchange standards, which provide a means for subscribers to share data about EPCs within defined user groups or with the general public, and which also provide access to EPCglobal Core Services and other shared services that facilitate these exchanges.

- *EPC Infrastructure* In order to have EPC data to share, each subscriber carries out operations within its four walls that create EPCs for new objects, follow the movements of objects by sensing their EPC codes, 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 subscribers 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 EPCglobal Subscribers 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 0 Gen 1 Tag Air Interface	(Note 4, below)	[UHFC0]
	UHF Class 1 Gen 1 Tag Air Interface	(Note 4, below)	[UHFC1G1]
	HF Class 1 Gen 1 Tag Air Interface	(Note 5, below)	[HFC1]
	UHF Class 1 Gen 2 Tag Air Interface v1.0.9	Ratified	[UHFC1G21.0.9]
	UHF Class 1 Gen 2 Tag Air Interface v1.1.0	In Development	[UHFC1G21.1.0]

	UHF Class 1 Gen 2 Tag Air Interface v1.2.0	In Development	[UHFC1G21.2.0]
	HF Class 1 Version 2 Tag Air Interface	In Development	[HFC1V2]
Infrastructure	EPC Tag Data Specification	Ratified	[TDS1.3]
	Low Level Reader Protocol	Ratified	[LLRP 1.0]
	Reader Protocol	Ratified	[RP1.1]
	Reader Management	Ratified	[RM1.0]
	Discovery, Configuration, and Initialization (DCI) for Reader Operations	In Development	[DCI]
	Tag Data Translation	Ratified	[TDT1.0]
	Application Level Events (ALE)	Ratified	[ALE1.0]
	Application Level Events (ALE)	In development	[ALE1.1]
	EPCIS Capture Interface	Ratified	[EPCIS1.0]
Data Exchange	EPCIS Data Specification	Ratified	[EPCIS1.0]
	EPCIS Query Interface	Ratified	[EPCIS1.0]
	Pedigree Specification	Ratified	[Pedigree1.0]
	EPCglobal Certificate Profile	Ratified	[Cert1.0]
	ONS	Ratified	[ONS1.0]
	EPCIS Discovery	TBD (Note 3)	(none)
	Subscriber Authentication	TBD (Note 3)	(none)

Notes for the "Status" column of the table above:

1. "Ratified" indicates a ratified EPCglobal specification.
2. "In development" indicates a specification whose development has been chartered and is underway within the EPCglobal standards development process
3. "TBD" indicates a technical area that is expected to be addressed within the EPCglobal Architecture Framework but where requirements are still under study within the Business Action Groups or the Architecture Review Committee.

4. Prior to the launch of EPCglobal in November 2003, the former Auto-ID Center published two UHF Tag Air Interface specifications, referred to herein as UHF Class 0 Gen 1 and UHF Class 1 Gen 1. These specifications, which are not EPCglobal standards, are superseded by the UHF Class 1 Gen 2 Tag Air Interface which was ratified by EPCglobal in December 2004.

5. Prior to the launch of EPCglobal in November 2003, the former Auto-ID Center also published an HF Tag Air Interface specification referred to herein as HF Class 1. This specification, which is not an EPCglobal standard, will be superseded by the HF Class 1 Version 2 Tag Air Interface.

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

3.1 The Role of Standards

EPCglobal standards are created to further the following objectives:

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

  For trading partners to exchange information, they must have prior agreement as to the structure and meaning of data to be exchanged, and the mechanisms by which exchange will be carried out. EPCglobal standards include data standards and information exchange standards that form the basis of cross-enterprise exchange. 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 specifications 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 exchange 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

EPCglobal 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. EPCglobal is committed to making use of existing global standards when appropriate, and EPCglobal works with recognized global standards organizations to ratify standards created within EPCglobal.

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 community through the EPCglobal Standards Development Process or an equivalent process within another standards organization. The Intellectual Property policy of EPCglobal is designed to secure free and open rights to implement 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 specifications 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., SOAP messaging [SOAP1.2]).

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-user's four walls, to a global implementation across entire supply chains. The specifications 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 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 specifications, or best security practice is recommended.

See Section 11 for an overview of data protection methods of current and evolving standards within the architecture framework.

3.7 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.

3.8 Industry Architectures and Standards

The EPCglobal Architecture Framework is designed to work with and complement existing industry-wide architectures and standards. For example, if the automotive or healthcare industry has registries, data exchanges, or data pools, it should be able to utilize and leverage the EPCglobal Network. The same holds true for Fast Moving Consumer Goods (FMCG) industries.

A specific example is the significant investment that FMCG companies have made in data synchronization, which will continue for the foreseeable future. Depending on the industry, participation in these and other enablers of e-commerce may be viewed as a prerequisite for or as complementary to use of the EPCglobal Network in its goals for improved supply chain operations.

3.9 Open, Community Process

The EPCglobal standards development 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 Business Action Groups.
- Open process in which all EPCglobal subscribers having relevant expertise are encouraged to join working 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 specifications.

4.1 Unique Identity

A fundamental principle of the EPCglobal Network Architecture is the assignment of a unique identity to physical objects, loads, locations, assets, and other entities whose use is to be tracked. By “unique identity” 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

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Network Architecture, the unique identity is the Electronic Product Code, defined by the EPCglobal Tag Data Specification [TDS 1.3].

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

- *Uniqueness* The EPC assigned to one entity is different than the EPC assigned to another (but see below for exceptions).
- *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 EPC is a universal identifier. 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.

  For example, both GS1 SSCC codes and GS1 GIAI codes are also 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.
- *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 exchange. Formal conversion rules exist [TDS1.3], and the Tag Data Translation Standard [TDT1.0] 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. EPCglobal 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 codes (SGTIN, SSCC, etc), for example, GS1 is the Issuing Agency. An Issuing Agency allocates a portion of the EPC space to another organization, who then becomes the "EPC Manager" for that block of codes. For GS1 codes, for example, this is done by assigning a Company Prefix to another organization, often a subscriber but sometimes another organization such as a GS1 Member Organization. The EPC Manager 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 codes, assigning a single Company Prefix has the effect of allocating several blocks of codes to an EPC Manager, 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 EPC's 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 codes 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 might be given an SSCC EPC, while the skid might be given a GRAI EPC.) 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. One example of multiple EPCs assigned to the same object would occur when the ownership or the chain of custody for the underlying physical item changes. This would constitute another exception to the principle of EPC uniqueness.

Another instance where global uniqueness may not be required is when EPC technology is used for closed systems, such as proprietary use within a single company. In most cases, the cost of achieving global uniqueness (that is, in obtaining an EPCglobal Manager Number) is so low that doing so is recommended even for a closed system. Nevertheless, the Tag Data Standards Working Group is currently considering whether any special provision for closed systems should be made in a future version of the EPC Tag Data Standard.

4.2 Decentralized Implementation

The EPCglobal Network seeks to link all enterprises together in a single global network. Logically, the EPCglobal Network is a common resource shared by all EPCglobal Subscribers. 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 EPCglobal Subscriber or group of Subscribers. In some cases, certain of these facilities are operated by EPCglobal Subscribers themselves.

The key elements of decentralization in the EPCglobal Architecture Framework are the assignment of EPC codes, and the ONS lookup service. These elements of decentralization are discussed in more detail in Sections 5.2, 7.1, and 7.3 .

4.3 Layering of Data Standards – Verticalization

The EPCglobal Architecture Framework includes standards for data exchange that are intended to serve the needs of many different industries. Yet, each industry has specific requirements around what data needs to be exchanged 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 Network. 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 framework layers tend to be developed by EPCglobal technical action groups, while the requirements for vertical standards tend to be developed by appropriate industry groups.

The two important data standards are the EPC Tag Data Specification, and the EPCIS Data Specification. Within the EPC Tag Data Specification, the framework elements include the structure of the “header bits” 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 Specification are the specific coding schemes defined for particular industry groups.

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

4.4 Layering of Software Specifications—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. Most of the content of EPCglobal standards does not rely on specific implementation technology such as web services, XML, AS2, EDI, and so on. Each enterprise has its own requirements and preferences for underlying technologies such as these, and these inevitably change over time.

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 specifications 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 specification.

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 is the embodiment of the underlying principle of unique identity. Electronic Product Codes are assigned to physical objects, loads, locations, assets, and other entities which are to be tracked through the EPCglobal Network in service of a given industry's business goals. The Electronic Product Code is the thread that ties together all data that flows within the EPCglobal Network, and plays a central part in every role and interface within the EPCglobal Architecture Framework.

5.2 EPC Manager

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 EPC Manager. Within this document, the term "EPC Manager" 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 EPC Manager one or more blocks of Electronic Product Codes within designated coding schemes that the EPC Manager can independently assign to physical objects and other entities without further involvement of the Issuing Agency. The EPC Manager is said to be the "managing authority" for the EPCs in this block.

The EPC Manager has two special responsibilities within the EPCglobal Architecture Framework that distinguish it from all other EPCglobal subscribers, with respect to the EPC codes it manages:

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

- The EPC Manager is responsible for maintaining the Object Name Service (ONS) records associated with blocks of EPCs it manages. ONS records are the point of entry for certain types of global lookup operations as described in later sections. (This responsibility is limited to those blocks of EPCs that are allocated by the EPC Manager for objects that are exchanged with other Subscribers; any EPC blocks reserved for internal use by the EPC Manager need not be reflected in ONS. Also, the EPC Manager may elect not to provide ONS lookup for any or all of its EPCs, in which case there is obviously no requirement to maintain ONS records for those EPCs.)

Other than these two responsibilities, the EPC Manager has no special responsibilities with respect to the EPCs it manages compared to any other EPCglobal Subscriber. In particular, both the EPC Manager and other subscribers may participate equally in the generation and exchange of EPC-related data.

5.3 EPC Manager Number

The way that an Issuing Agency grants a block of EPCs to an EPC Manager is by issuing the EPC Manager a single number, called the EPC Manager Number. An End-User Subscriber or other organization may hold multiple Manager Numbers, and therefore be in control of multiple blocks of EPCs. The structure of all coding schemes within the Electronic Product Code definition is such that the EPC Manager Number appears as a distinct field within any given representation. The EPC Manager Number should not be assumed to be the owner when derived from GS1 codes.

Having the EPC Manager Number as a distinct field within any given representation, allows any system to instantly identify the EPC Manager 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 EPC Manager as appropriate. For example, an ONS lookup is conceptually a lookup in a single large table that maps any EPC to the location of an EPCIS service, but having the EPC Manager Number as a distinct field allows ONS to be implemented as a collection of tables, each maintained by the EPC Manager for a given block of EPCs (see Section 7.3 for more information on ONS specifically).

The allocation of a block of EPC codes to an EPC Manager is actually implicit in the act of assigning an EPC Manager Number. The EPC Manager is simply free to commission any EPC code so long as the EPC Manager Number field within the code contains the assigned EPC Manager Number, following the EPC Tag Data Specification. The “block” of codes, therefore, simply consists of all EPCs that contain the assigned EPC Manager Number in the EPC Manager Number field. (This is a slight simplification; see Section 5.4 for more information.)

5.4 Embedding of Existing Codes

Most coding schemes currently defined with the EPC Tag Data Specification are based on existing industry coding schemes. For example, there are five types of EPCs based on the GS1 family of codes [GS1GS]: SGTIN, SSCC, SGLN, GRAI, and GIAI. In the case of these GS1 codes, the EPC Manager Number is one and the same as the GS1 Company Prefix that forms the basis of the GS1 codes. The other fields of GS1-based EPCs are also derived from existing fields of the GS1 codes.

In general, this kind of embedding is possible for any existing coding scheme that has a manager-like structure; that is, when the existing coding scheme 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 EPC

Manager Numbers when used to construct EPCs using the "DoD construct" coding scheme.

In the last section, it was noted that assigning an EPC Manager Number to an EPC Manager effectively allocates a block of codes to the EPC Manager. Because the Electronic Product Code federates several coding schemes, the "block" of EPC codes implied by the assignment of an EPC Manager Number is not necessarily a single contiguous block of numbers, but rather a contiguous block within each EPC identity type to which the EPC Manager Number pertains. For example, when an EPC Manager Number is a GS1 Company Prefix, the EPC Manager is effectively granted a block of EPCs within each of the five GS1-related EPC types (SGTIN, SSCC, SGLN, GRAI, and GIAI). But when an EPC Manager Number is a US Department of Defense CAGE/DoDAAC code, the EPC Manager is effectively granted a single block of EPCs, within the "DoD Construct" coding scheme.

5.4.1 A GS1 Company Prefix Does Not Uniquely Identify a Company when the Manager Number is Derived from GS1 Codes

In the early days of the UPC code, 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 GTIN codes 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 B's prefix.)
- An organization possessing a GS1Company Prefix may subcontract the manufacture of trade items to contract manufacturers. The GTINs for these products 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 retailer's 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.

- 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 barcode 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 "company prefix" appears to imply). Therefore, the EPC Manager Number should not be assumed to be the owner when derived from GS1 codes

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 EPC Manager 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 used in the consumer packaged goods industry is an example of an EPC having this structure. In this particular example, the EPC Manager Number and Object Class ID are in fact convertible to the GTIN code that is used outside of the EPC arena to represent product classes. This is another example of how existing codes are embedded within 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 EPC Manager Number and a Serial Number.

5.6 EPC Information Services (EPCIS)

The primary vehicle for data exchange between EPCglobal Subscribers in the EPCglobal Architecture Framework is EPC Information Services (EPCIS). As explained below, EPCIS encompasses both interfaces for data exchange 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 not under their direct control. (EPCIS data may, of course, also be used within a company's four walls.) For most industries using the EPCglobal Network, 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 0). For consumer products, for example, the "class" 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 exchange.
- *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 "EPC X was shipped at 12:03pm 15 March 2004 from Acme Distribution Center #2," and "At 3:45pm 22 Jan 2005 the case EPCs (list here) were aggregated to the pallet EPC X at ABC Corp's Boston factory." 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 "There were 4,100 instances of object class C observed at 2:00am 16 Jan 2003 in RetailMart Store #23." 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 "The pallet with EPC X was shipped in fulfillment of Acme Corp purchase order #23 at 2:20pm." Most business transaction observations have four dimensions: time, one or more EPCs, a business process step, and a business transaction identifier.

The EPCIS Data Specifications provide a precise definition of all the types of EPCIS data, as well as the meaning of "event" 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 enterprises 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 Network is to allow any participant in the supply chain to discover all transactional data to which it is authorized, from any other participant. 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 EPCglobal Core 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 specification document. In the case of an EPCglobal Core Service, normative descriptions are either provided as EPCglobal Standards (for interface aspects of Core Services) or in other EPCglobal documentation (for implementation aspects of Core Services).

As noted in Section 2, a specific EPCglobal Standard is either ratified, in development within an EPCglobal technical Working Group, or TBD meaning that requirements are still under discussion within EPCglobal Business Action Groups or the Architecture Review Committee. Where ratified specifications exist, this document provides citations to the specification document, which provides the normative description. Otherwise, details beyond what is described herein are only available to EPCglobal Subscribers who have joined the appropriate EPCglobal Working Group or Action Group.

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 core services that are operated by EPCglobal and its delegates (EPCglobal Core 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 EPCglobal Core Services to carry out certain tasks.

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 user's system architecture may be created by the end user or obtained by the end user from vendors, 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 vendor-provided components from which to choose.

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 Core 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 Core 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 Standards, 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., detecting EPCs during product movement during truck loading), 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 Exchange between enterprises, EPC Object Exchange between enterprises, or EPC Infrastructure deployed within a single enterprise. Clearly, all EPCglobal Subscribers will find relevance in the first two categories, as use of these standards is necessary to interact with other subscribers. A subscriber has much more latitude, however, in its decisions surrounding adoption of the EPC Infrastructure standards, as those standards do not affect parties outside the subscriber's own four walls.

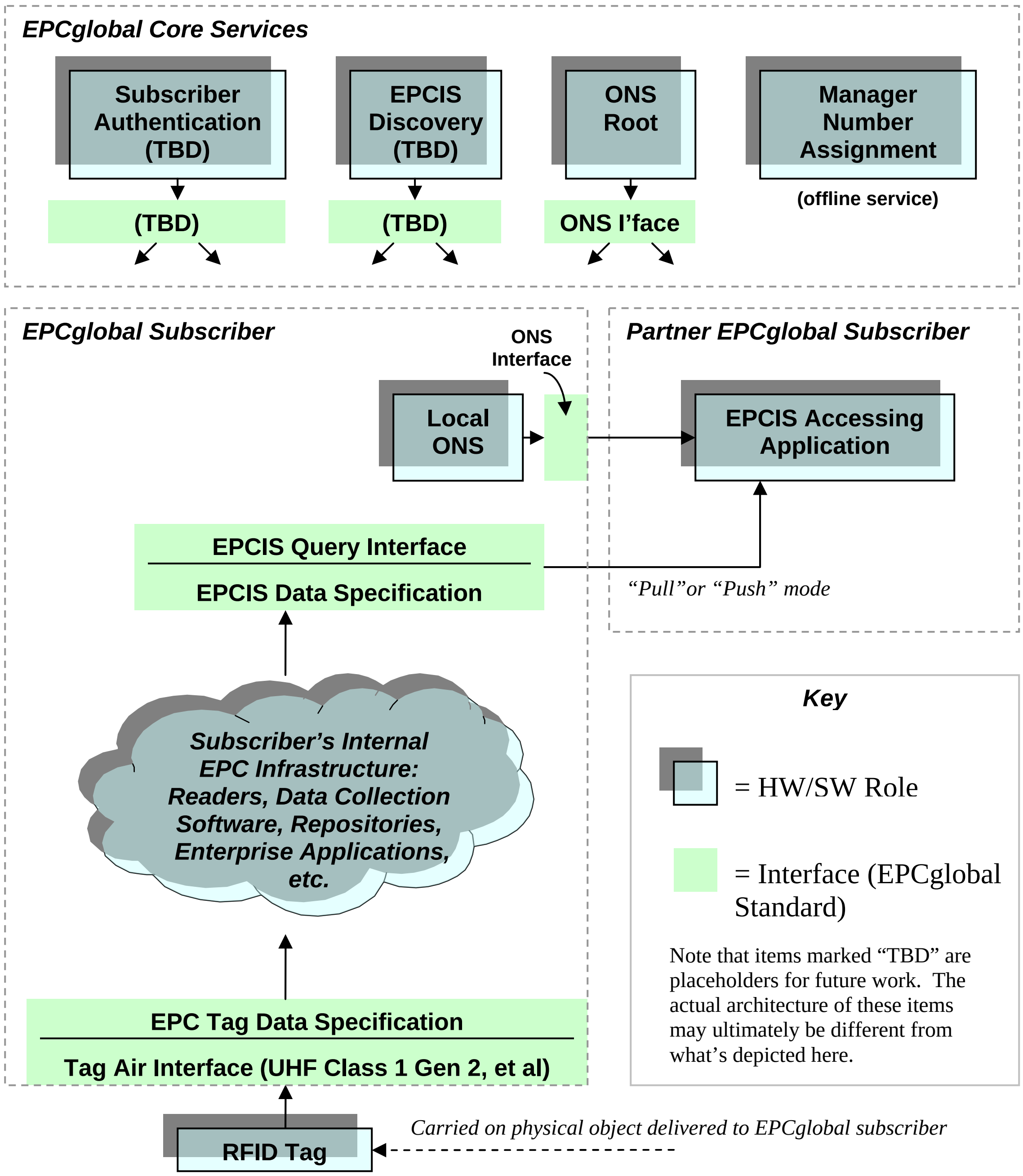
For this reason, the following discussion of roles and interfaces within the EPCglobal Architecture Framework is divided into two sections, the first dealing with cross-enterprise elements (EPC Data Exchange and EPC Object Exchange), and the second dealing with intra-enterprise elements (EPC Infrastructure). 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 subscribers, namely those categorized as "EPC Object Exchange Standards" or "EPC Data Exchange Standards" in Section 2. EPCglobal Standards that are primarily used within the four walls of a single subscriber ("EPC Infrastructure Standards" from Section 2) are described in Section 8. Most EPCglobal Subscribers will implement the architecture given in this section in order to fully participate in the EPCglobal Network.

In the following diagram, the plain green bars denote interfaces governed by EPCglobal standards, while the blue "shadowed" 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.

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



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

7.1 Data Exchange Interactions

The top part of the diagram shows the roles and interfaces involved in data exchange. The Partner EPCglobal Subscriber has an "EPCIS Accessing Application" (role), which

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

is some application specific to the Partner EPCglobal Subscriber 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. In general, there are several ways it may do so:

- 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 party's 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 B's service as part of a business agreement, and in obtaining information from B it learns how to reach A's service (which B knows as part of its business agreement with A). This is sometimes referred to as "following the chain."
- The EPCIS Accessing Application may use the Object Name Service (ONS) to locate the EPCIS service of the EPCglobal Subscriber who commissioned the EPC of the object in question.
- The EPCIS Accessing Application may use EPCIS Discovery Services to locate the EPCIS services of all EPCglobal Subscribers that have information about the object in question, including EPCglobal Subscribers 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 "follow the chain." (EPCIS Discovery Services are TBD at the time of this writing, so the precise architecture of roles and interfaces involved in EPCIS Discovery Services is not yet known – 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 EPCglobal Subscriber 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 subscriber. Two EPCglobal Standards govern this interaction. The EPCIS Query Interface defines how data is requested and delivered from an EPCIS service. The EPCIS Data Specification define the format and meaning of this data. The EPCIS Query Interface is designed to support both on-demand or "pull" modes of data transfer, as well as asynchronous or "push" 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 EPCglobal Subscriber accesses the EPCIS service of the first EPCglobal Subscriber, the first EPCglobal Subscriber will usually want to authenticate the identity of the Partner EPCglobal Subscriber in order to determine what data the latter is authorized to receive. The Subscriber Authentication role in the diagram refers to an EPCglobal Core Service that assists in this authentication, making it possible for any EPCglobal Subscriber to authenticate the identity of any other EPCglobal Subscriber without any prior arrangement between the two parties. The

EPCglobal architecture 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 within the EPCglobal network. If X.509 certificates are used, they should comply with the specifications defined in the EPCglobal X.509 Certificate Profile [Cert1.0], which provides a minimum level of cryptographic security and defines and standardizes identification parameters for users, services/servers and devices. In some situations, an EPCglobal Subscriber 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 EPCglobal Subscriber to make.

7.2 Object Exchange Interactions

The lower part of the diagram illustrates how the first EPCglobal Subscriber interacts with physical objects it receives from other subscribers. A physical object is received by the EPCglobal Subscriber, bearing an RFID tag that contains an EPC code. The EPCglobal Subscriber 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 carried through a radio signal to the RFID Reader. The EPC Tag Data Specification defines the format and meaning of this data, namely the EPC code.

Within the EPCglobal Subscriber'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 EPCglobal Subscriber 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.

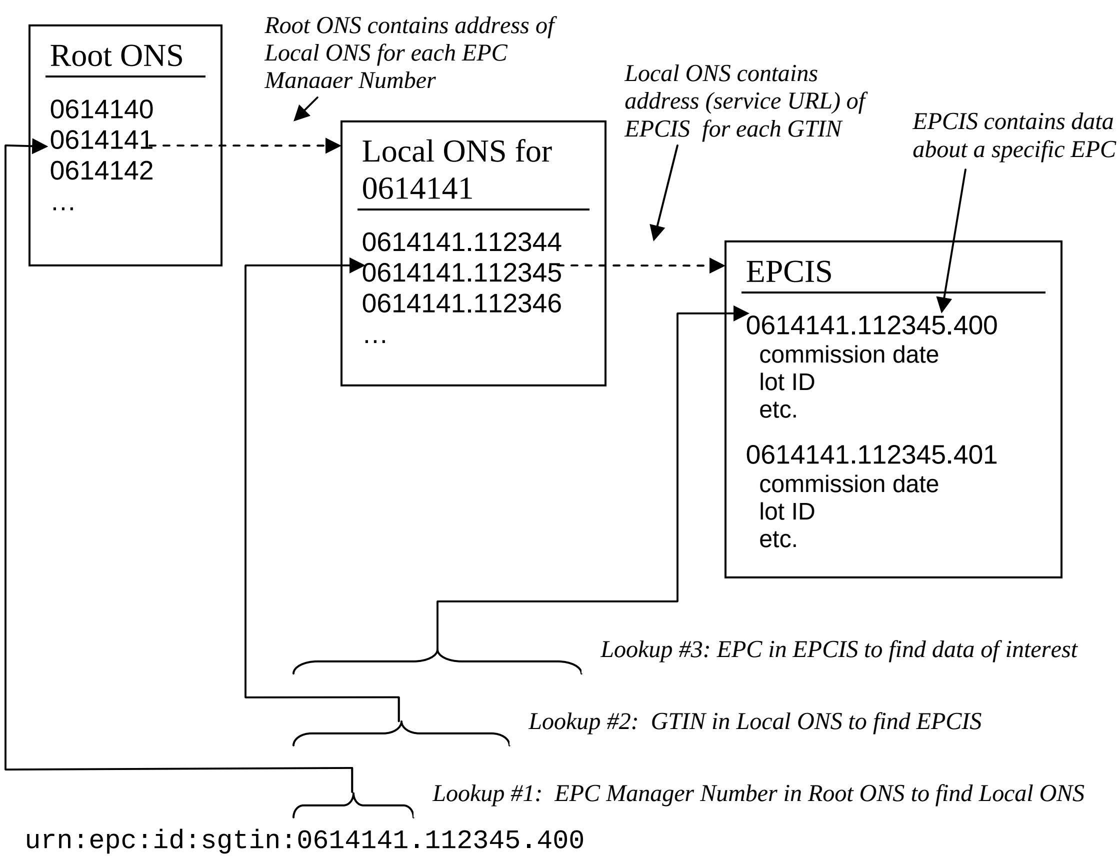
7.3 ONS Interactions

In Section 7.1, it was mentioned that one EPCglobal Subscriber 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 EPCglobal Core Service and a service provided by an individual subscriber.

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 EPC Manager of the EPC in question. (An EPC Manager 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 EPC Manager, and it may not be the case that all object classes of a given EPC Manager will have information provided by the same EPCIS service. This is especially true when the EPC Manager 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. The current ONS specification also does not address coding schemes which do not have a field corresponding to object class, such as the SSCC and GIAI codes.)

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 EPC Manager 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 Service (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 EPCglobal Subscriber wishes to locate an EPCIS service, it first consults the Root ONS service controlled by EPCglobal. The Root ONS service identifies the Local ONS service of the EPC Manager organization for that EPC. The EPCglobal Subscriber 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 holds the EPC manager number 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 A's EPC manager number, but for contract-manufactured products it returns pointers to Company B's 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	EPCglobal	Address of Local ONS for given EPC Manager Number (GS1 Company Prefix)
2	Local ONS for given EPC Manager Number	Holder of EPC Manager Number (GS1 Company Prefix)	Address of EPCIS Service for given EPC Class (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 Service (DNS), simply by specifying a convention whereby an EPC is converted to an Internet Domain Name in the `onsepc.com` domain. For example, given an EPC:

```
urn:epc:id:sgtin:0614141.112345.400
```

an ONS lookup is performed by transforming the EPC into the following Internet Domain Name (essentially, by dropping the serial number, dropping the `urn:epc:id` prefix, reversing what remains, and adding `onsepc.com`):

```
112345.0614141.sgtin.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 "NAPTR" records). An "ONS service," therefore is nothing more than an ordinary DNS nameserver that happens to be part of the domain name tree rooted at `onsepc`.com. This has several implications:

- The "Root ONS service" and "Local ONS service" as used above may each be implemented by multiple independent 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.

- EPCglobal's Root ONS service is actually itself two levels down in a hierarchy of lookups, which has its true root in the worldwide DNS root service.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

- 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 Specification [ONS1.0].

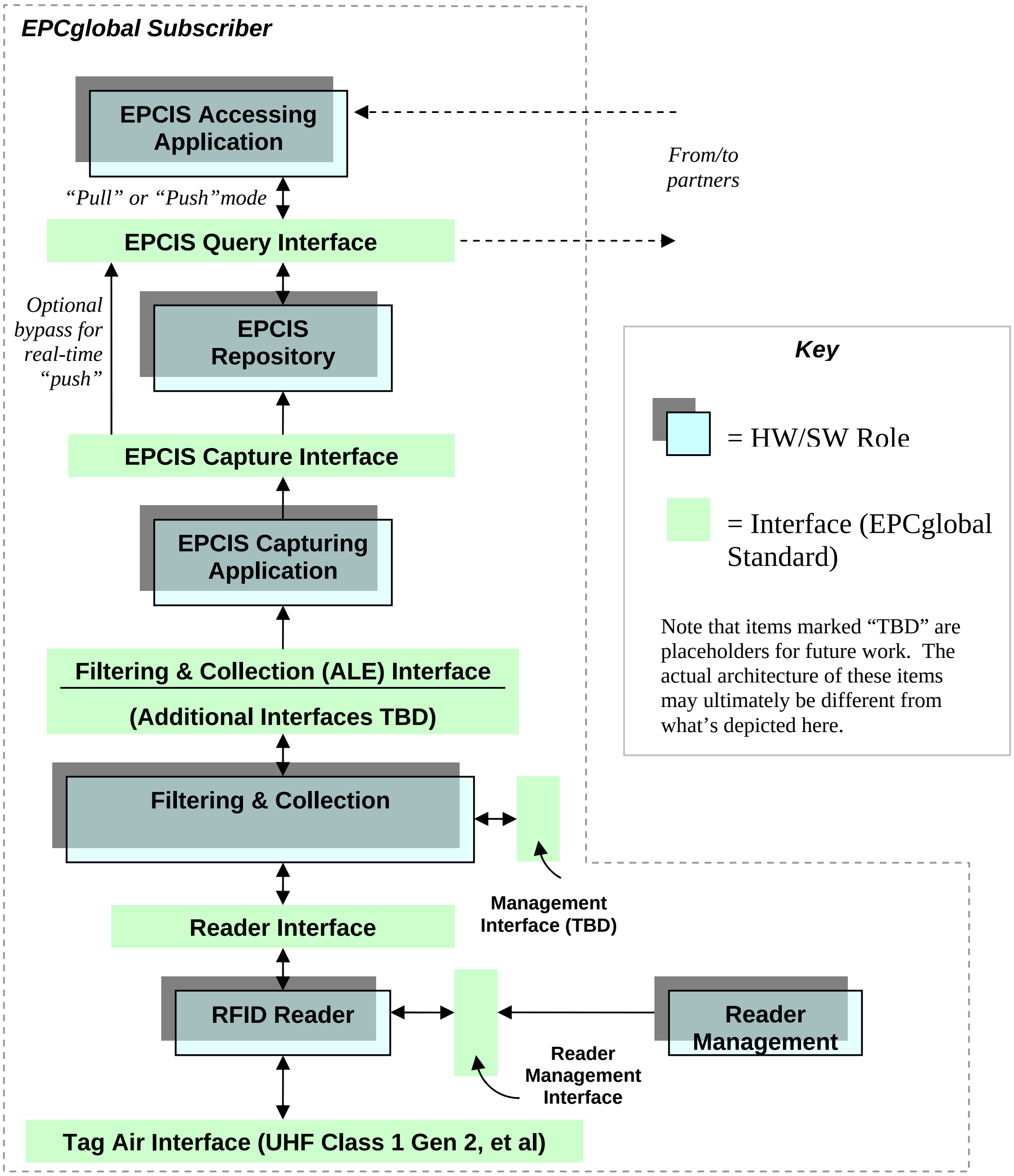
7.4 Number Assignment

The foregoing text has described every role and interface in the diagram at the beginning of this Section 7, except for Manager Number Assignment. This role simply refers to EPCglobal's service of issuing unique EPC Manager Numbers to each EPC Manager organization that requests one, in its capacity as the Issuing Agency for the GS1 family of codes (see Section 4.1). By insuring that every EPC Manager Number that is issued is unique, the uniqueness of EPCs assigned by individual EPCglobal Subscribers within the GS1 family of codes is ensured. (Number assignment for other coding schemes is carried out by Issuing Agencies other than EPCglobal, and so EPCglobal's Manager Number Assignment Core 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 subscriber, namely those categorized as "EPC Infrastructure Standards" in Section 2. This section expands the "cloud" in the diagram from Section 7. Because this cloud is completely internal to a given enterprise, a subscriber has much more latitude to deviate from this picture when appropriate to that subscriber's unique business conditions. EPCglobal sets standards in this area, however, to encourage solution providers to create interoperable system components from which subscribers may choose.

As in Section 7, the plain green bars in the diagram below denote interfaces governed by EPCglobal standards, while the blue "shadowed" 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.



Between the EPC Object Exchange interfaces and the EPC Data Exchange interfaces in the figure from Section 7 is a "cloud" 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 "Reader A saw EPC X at time T."
- *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 "At Location L, between time T1 and T2, the following EPCs were observed," 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 "smart shelves" deployed that generate periodic observations about objects that enter or leave the shelf. Here, 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 exchange with partners. Events at this interface say, for example, "At location X, at time T, the following contained objects (cases) were verified as being aggregated to the following containing object (pallet)."
- *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 "smart antenna" 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 "At location X, at time T, the following cases were verified as being on the following pallet." 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 EPC codes. 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. At present, the only EPCglobal standard involved is the Reader Management standard.

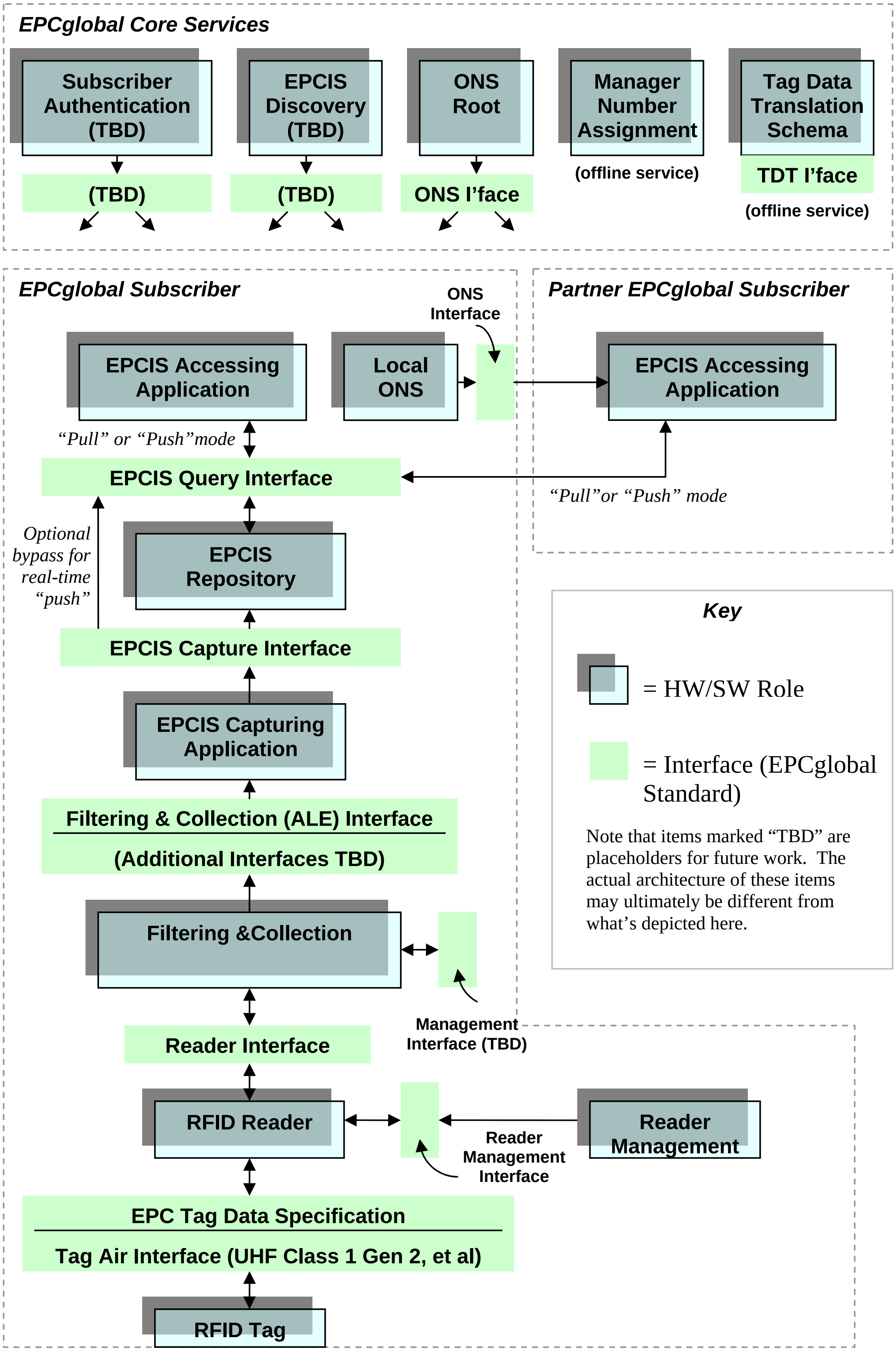
9 Roles and Interfaces – Reference

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 "shadowed" boxes denote roles played by hardware and software components of a typical system architecture. As emphasized in Section 6.1,

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

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.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.


EPCglobal Core Services
Subscriber Authentication (TBD)
EPCIS Discovery (TBD)
ONS Root
Manager Number Assignment
Tag Data Translation Schema
(TBD)
(TBD)
ONS I'face
(offline service)
TDT I'face
(offline service)
EPCglobal Subscriber
ONS Interface
EPCIS Accessing Application
Local ONS
"Pull" or "Push"mode
EPCIS Query Interface
Optional bypass for real-time "push"
EPCIS Repository
EPCIS Capture Interface
EPCIS Capturing Application
Filtering & Collection (ALE) Interface
(Additional Interfaces TBD)
Filtering &Collection
Management Interface (TBD)
Reader Interface
RFID Reader
Reader Management Interface
Reader Management
EPC Tag Data Specification
Tag Air Interface (UHF Class 1 Gen 2, et al)
RFID Tag
Partner EPCglobal Subscriber
EPCIS Accessing Application
"Pull"or "Push" mode
Key
= HW/SW Role
= Interface (EPCglobal Standard)
Note that items marked "TBD" are placeholders for future work. The actual architecture of these items may ultimately be different from what's depicted here.

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)

EPCglobal has defined a tag classification system to describe tag functionality. The responsibilities of the RFID Tag role based on classification are shown below. EPCglobal is still evaluating responsibilities and roles for the tag classifications beyond Class1.

Class-1: Identity Tags: Passive-backscatter Tags.

Responsibilities:

- Holds an electronic product code (EPC) identifier. May allow the EPC code to be changed post-manufacture.
- May hold an immutable code that gives manufacture information, including the manufacturer identity, unique manufacture serial number, etc.
- May provide a means to permanently disable the Tag. This feature may involve additional data stored on the tag such as a kill code.
- May have additional features such as lock, access control, etc. These features may involve additional data stored on the tag such as a lock code, lock status, access password, etc.
- May have additional user data apart from the EPC code.

Class-2: Higher-Functionality Tags: Passive Tags with the following anticipated features above and beyond those of Class-1 Tags:

Responsibilities:

- An extended Tag ID
- Extended user memory
- Authenticated access control, and additional features as will be defined in the Class-2 specification.

Class-3: Semi-Passive Tags: Semi-passive Tags with the following anticipated features above and beyond those of Class-2 Tags:

Responsibilities:

- An integral power source
- Integrated sensing circuitry

Class-4: Active Tags: Active Tags with the following anticipated features above and beyond those of Class-3 Tags:

Responsibilities:

- Tag-to-Tag communications
- Active communications
- Ad-hoc networking capabilities.

9.1.2 EPC Tag Data Specification (Interface)

Normative references:

- Ratified EPCglobal Standard: [TDS1.3]

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

9.1.3 Tag Air Interface (Interface)

As explained in the notes to the table in Section 2, there are several Tag Air Interfaces that have been defined for use within the EPCglobal Network and several under development: one that is a ratified EPCglobal standard (the UHF Class 1 Gen 2 Tag Air Interface), and three others that were published by the Auto-ID Center prior to the creation of EPCglobal. The notes to the table in Section 2 give a full description of the status of each of these Tag Air Interfaces. At the level of this document, the various Tag Air Interfaces differ only with respect to the class of functionality that they provide [CLASS1]. They also differ in technical detail as to how commands and data are exchanged between reader and tag and what the specific command set is.

Normative references:

- EPCglobal Specifications (from Auto-ID Center): [UHFC0], [UHFC1G1], [HFC1]
- Ratified EPCglobal Standard: [UHFC1G21.0.9]
- Standards in development: [UHFC1G21.1.0], [UHFC1G21.2.0], [HFC1V2]

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 code 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 code, 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. All EPCglobal Reader Interface standards are 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.

At the time of this writing, there are different Reader Interface standards, at different stages of completion. They are:

- *Reader Protocol (RP) 1.1* This is the first Reader Interface standard developed by EPCglobal, and is now a ratified specification.
- *Low-Level Reader Protocol (LLRP) 1.0* This is a newer Reader Interface that provides greater control to clients over the use of the RF channel and protocol-specific tag features such as Gen2 inventory sessions. It is now a ratified EPCglobal standard.
- *High-Level Reader Protocol (HLRP)* This is the name given in the Software Action Group Reader Operations Working Group charter for work intended to continue the development of Reader Protocol 1.1. At the time of this writing, this activity has not yet been initiated.

Normative references:

- Ratified EPCglobal Standard: [RP1.1]
- Ratified EPCglobal Standard: [LLRP1.0]

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Responsibilities[1]*:*

- Provides means to command an RFID Reader to inventory tags (that is, to read the EPC codes carried on tags), read tags (that is, to read other data on the tags apart from the EPC code), write tags, manipulate tag user and tag-identification data, and access other features such as kill, lock, etc.
- May provide means to access RFID Reader management functions including discovery, firmware/software configuration and updates, health monitoring, connectivity monitoring, statistics gathering, antenna connectivity, transmit power level, and managing reader power consumption. These features are more characteristic of lower-level Reader Interfaces.
- May provide means to control RF aspects of RFID Reader operation including control of RF spectrum utilization, interference detection and measurement, modulation format, data rates, etc. These features are more characteristic of lower-level Reader Interfaces.
- May provide means to control aspects of Tag Air Interface operation, including protocol parameters and singulation parameters. These features are more characteristic of lower-level Reader Interfaces.
- May provide 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. These features are more characteristic of higher-level Reader Interfaces.

9.1.6 Reader Management Interface (Interface)

Normative references:

- Ratified EPCglobal Standard: [RM1.0]
- Standard in development: [DCI]

Responsibilities:

- 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.

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

- 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 specification [RM 1.0] addresses only reader monitoring functions.

As the specification of this interface evolves to fully exploit features of the UHF Class 1 Gen 2 Tag Air Interface, it is expected that it will gain additional responsibilities including providing means to manage readers to prevent reader-to-reader collisions and facilitate “scouring” to find tags. This includes management of power levels, carrier frequencies, “sessions” (as that term is defined in the UHF Class 1 Gen 2 Tag Air Interface), and protocol parameters. In its current [RM 1.0] version, the Reader Management specification supports counters and statistics for all UHF Class 1 Gen 2 operations (singulate, memory read, lock, kill, etc.), but is not fully aware of “sessions” and other Tag Air Interface concepts.

There are currently two EPCglobal specifications defining different aspects if the Reader Management Interface. The Reader Management specification [RM 1.0] 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 specification focuses on reader’s identification, configuration and network connectivity management. How these responsibilities are divided between the different reader-level interfaces in the future is TBD.

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 application's 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 Specification, 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 "logical reader names" 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.
- Sets and controls 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

"sessions" 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) 1.0 Interface (Section 9.1.9) provides a standard interface that applies to a large collection of use cases in which RFID Tags are inventoried (i.e., where the EPCs carried on the tags are read).
- The Filtering & Collection (ALE) 1.1 Interface (Section 9.1.9), currently under development, provides additional interfaces that support use cases in which tags are written or killed, in which the kill or lock passwords are maintained, and in which "user data" or TID memory on the tags is read or written.
- Management of the Filtering & Collection role is not yet addressed by any EPCglobal specification. This includes controlling aspects of coordination the activities of multiple readers to minimize interference, setting parameters that govern inventorying strategies, control over Tag Air Interface-specific features, and so on.

9.1.9 Filtering & Collection (ALE) Interface (Interface)

The Filtering & Collection (ALE) 1.0 Interface provides a standard interface to the Filtering & Collection role that applies to a large collection of use cases in which RFID Tags are inventoried (i.e., where the EPCs carried on the tags are read). The Filtering & Collection (ALE) 1.1 Interface (Section 9.1.9), currently under development, provides additional interfaces that support use cases in which tags are written or killed, in which the kill or lock passwords are maintained, and in which "user data" or TID memory on the tags is read or written.

Normative references:

- Ratified EPCglobal Standard: [ALE1.0]
- Standard in development: [ALE 1.1]

Responsibilities:

- 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 readers' 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.

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 barcode 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. (When tag writing and related features are addressed in a future version of the Filtering & Collection Interface, as noted in Section 9.1.8, 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]

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]

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 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 "pedigree" for drug products. Drug Pedigree Messaging is a data exchange interface intended to standardize the exchange 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 Specification that can be adopted by members of the pharmaceutical supply chain.

9.1.16 Object Name Service (ONS) Interface (Interface)

Normative references:

- Ratified EPCglobal Standard: [ONS1.0]

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 EPC Manager for that EPC, and typically includes services operated by the organization that commissioned the EPC.

9.1.17 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 EPC Manager.

See also the discussion of ONS in Section 7.3.

9.1.18 ONS Root (Core Service)

Responsibilities:

- Provides the initial point of contact for ONS lookups.
- In most cases, delegates the remainder of the lookup operation to a Local ONS operated by the EPC Manager for the requested EPC.
- May completely fulfill ONS requests in cases where there is no local ONS to which to delegate a lookup operation.
- Provides a lookup service for 64-bit Manager Index values as required by the EPC Tag Data Specification.

See also the discussion of ONS in Section 7.3.

9.1.19 Manager Number Assignment (Core Service)

Responsibilities:

- Ensures global uniqueness of EPCs by maintaining uniqueness of EPC Manager Numbers assigned to EPCglobal Subscribers
- Assigns new EPC Manager Numbers as required by EPCglobal Subscribers.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

9.1.20 Tag Data Translation Schema (Core Service)

Responsibilities:

- Provides a machine-readable file that defines how to translate between EPC encodings defined by the EPC Tag Data Specification (Section 9.1.2). EPCglobal provides this file for use by End-users, so that components of their infrastructure may automatically become aware of new EPC formats as they are defined.

9.1.21 Tag Data Translation Interface (Interface)

Normative references:

- Ratified EPCglobal Standard: [TDT1.0]

Responsibilities:

- Encodes in machine-readable form all of the rules that define how to translate between EPC encodings defined by the EPC Tag Data Specification (Section 9.1.2).

9.1.22 EPCIS Discovery (Core Service – TBD)

Note that "EPCIS Discovery" is not yet a defined part of the EPCglobal Architecture Framework, but rather a placeholder for functionality that is envisioned for the EPCglobal Network but not yet architected. The responsibilities enumerated below are an envisioned set of responsibilities, but it is not yet known if this list is complete or accurate, nor how many distinct roles and interfaces will ultimately be required to carry out these responsibilities. Moreover, while "EPCIS Discovery" is labeled an EPCglobal Core Service, this is also just a placeholder, and the final set of responsibilities may be addressed by a combination of EPCglobal Core Services and services operated by EPCglobal Subscribers.

Responsibilities:

- Provides a means to locate all EPCIS services that may have information about a specific EPC.
- May provide a cache for selected EPCIS data.
- Enforces authorization policies with respect to access of the aforementioned data.

9.1.23 Subscriber Authentication (Core Service – TBD)

The EPCglobal architecture 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 within the EPCglobal network. The responsibilities enumerated below are an envisioned set of responsibilities, but it is not yet known if this list is complete or accurate, nor how many distinct roles and interfaces will ultimately be required to carry out these responsibilities.

Responsibilities:

- Authenticates the identity of an EPCglobal Subscriber.

- Provides credentials that one EPCglobal Subscriber may use to authenticate itself to another EPCglobal Subscriber, without prior arrangement between the two Subscribers.
- Authenticates participation in network services through validation of active EPCglobal Subscription.

9.1.24 Filtering & Collection Management Interface (Interface – TBD)

In Section 9.1.6 it is noted that management of roles above the RFID Reader role is not currently addressed by EPCglobal standards, but may be considered in the future as warranted. The Filtering & Collection Management Interface shown in the diagram at the beginning of this section is a placeholder for future work that may arise in this area. The responsibilities enumerated below are an envisioned set of responsibilities, but it is not yet known if this list is complete or accurate, nor how many distinct roles and interfaces will ultimately be required to carry out these responsibilities.

Responsibilities:

- Provides means to query the configuration of systems that carry out Filtering & Collection responsibilities.
- Provides means to monitor the operational status of systems that carry out Filtering & Collection responsibilities.
- Provides means to control configuration of systems that carry out Filtering & Collection responsibilities.

10 Summary of Unaddressed Issues

As noted in Section 1 and throughout the document, there are technical needs that are believed to exist based on the analysis of known use cases, where those needs are not yet fully addressed by the EPCglobal Architecture Framework. In these cases, the architectural approach has not yet been finalized, and therefore work on developing standards or designing additional Core Services has not yet begun, though architectural analysis is underway within the Architecture Review Committee. This section summarizes the known unaddressed issues, and will serve as a starting point for continued refinement of the EPCglobal Architecture Framework.

The following list of issues is *not* intended to suggest the relative importance or priority of any issue.

10.1 EPCIS "Discovery"

The EPCIS Interface provides the means for one Subscriber to query another for EPC-related information. As discussed in Section 7.1, there are several ways a Subscriber might locate the relevant EPCIS Services in a given situation.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

The EPCglobal Architecture Framework does not currently provide a means to locate EPCIS Services in the most general situations arising from multi-party supply chains, in which several different organizations may have relevant data about an EPC but the identities of those organizations are not known in advance. Sections 7.1 and 9.1.21 discuss some of the thinking that has gone on in this area, but the EPCglobal Architecture Framework does not yet address these requirements.

10.2 Subscriber Authentication

Section 7.1 also points out the need for subscribers to mutually authenticate each other when they are involved in EPCIS exchanges. It is desirable for this authentication to be as easy as possible for a subscriber to implement. In particular, it is undesirable if each subscriber has to make prior arrangements with every other subscriber that might be involved in a future EPCIS exchange; instead, it is better if each subscriber need only register once with a central authority and thereafter be able to mutually authenticate with any other subscriber.

To achieve this goal, the X.509 authentication framework could be widely employed within the EPCglobal network. The EPCglobal Certificate Profile specification for X.509 certificates [Cert1.0] has been developed to ensure that existing Internet standards for X.509 certificates can be deployed to authenticate Users, Services/Servers, Readers and Devices within the network.

10.3 RFID Reader Coordination

The UHF Class 1 Gen 2 Tag Air Interface provides a number of features designed to improve the performance of RFID Readers, especially when many readers are deployed in close physical proximity. These features serve to minimize reader-to-reader collisions, and facilitate “scouring” algorithms to find tags. Among the features provided for these ends are control over power management, carrier frequencies, “sessions” that help insure one reader does not interfere with another reader’s conversation with the same tag, and other protocol parameters.

The Reader Protocol and Reader Management specifications do not specifically address these new features, nor does the EPCglobal Architecture Framework specify how these features would be exploited at an architectural level (e.g., by giving some responsibility to the Filtering & Collection role, or possibly to higher-level roles or new roles). The LLRP specification currently under development, however, does provide access to these features.

10.4 RFID Tag-level Security and Privacy

Sections 3.6 and 3.7 discuss EPCglobal Network goals of security and privacy. The UHF Class 1 Generation 2 Tag Air Interface supports specific RFID Tag features designed to further security and privacy goals. These features include a “kill” feature with an associated kill password, a “lock” feature, and an access control password.

The EPCglobal Architecture Framework does not currently discuss how these features affect the architecture above the level of the Reader Interface, nor is there any architectural discussion of how the goals of security and privacy are addressed through these or other features. In particular, it is not clear how the passwords required to operate the “kill” and “lock” features are to be distributed through the network to reach the places where they are required.

It should be noted that the “kill” and “lock” features are only components of a comprehensive privacy policy, not a complete solution to privacy issues facing the EPCglobal Network. 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.

10.5 “User Data” in RFID Tags

The EPCglobal Architecture Framework discusses the use of RFID Tags that are used to hold an EPC code associated with an object to which the tag is affixed. The UHF Class 1 Generation 2 Tag Air Interface supports RFID Tags that contain additional “user data” besides the EPC code.

The EPCglobal Architecture Framework does not currently discuss how RFID Tag “user data” is to be exploited at any level of the architecture. The ratified Reader Protocol specification and the currently in-development ALE 1.1 specification do, however, provide access to user memory.

10.6 Tag Writing, Killing, Locking above the Reader Interface Layer

Reading (apart from reading EPCs), writing, locking, and killing of RFID Tags, as well as maintenance of the kill and access passwords, are currently addressed by the UHF Class 1 Gen 2 Tag Air Interface and the Reader Protocol 1.1 specification, but not yet at higher layers of the architecture framework. The ALE 1.1 specification currently under development will provide support for these functions. See Section 9.1.8 for further discussion.

10.7 Master Data for RFID Tag Manufacture Data

The UHF Class 1 Generation 2 Tag Air Interface provides for a read-only “tag ID” (TID) field that is written at RFID Tag manufacture time. The TID is intended to provide information about the manufacture of the tag, including the identity of the tag manufacturer and other information. This information would be associated with the TID in an external database, maintained by EPCglobal or some other authority.

The EPCglobal Architecture Framework does not currently provide a specification for the TID or associated information. Existing architecture components (e.g., ONS) might be useful for this purpose.

11 Data Protection in the EPCglobal Network

11.1 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 network and its related standards.

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 network.

Section 11.2 introduces security concepts. Section 11.3 describes the data protection mechanisms defined within the existing EPCglobal ratified standards. Section 0 introduces the data protection methods that are being developed in evolving EPCglobal standards.

11.2 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, specification 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 “security” 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 the EPCglobal network and relate these mechanisms to the system attributes described above.

11.3 Existing Data Protection Mechanisms

This section summarizes the existing data protection mechanism within the standards and specifications forming the EPCglobal network.

11.3.1 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 specifications.

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 exchange. 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 network interface 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 “security lock” shown on the task bar indicate 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.

11.3.1.1 Application Level Events 1.0 (ALE)

The ALE 1.0 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 1.0 specification, see [ALE1.0].

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 1.0 specification 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 exchange 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 allows clients to subscribe to events that are delivered asynchronously. ALE implementations deliver these notifications by posting or sending XML data to a specified URI. The notification channel URIs specified by the standard are based on protocols that do not protect data via encryption or authentication, but allows vendors to provide additional notification mechanisms that may provide these protections.

11.3.1.2 Reader Protocol 1.1 (RP)

The current RP 1.1 specification 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 specification provides information and guidance for those desiring secure communication links when using other defined transports; see the RP specification for more details.

11.3.1.3 Reader Management 1.0 (RM)

The reader management specification 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].

RM divides its specification 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 specification 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 11.3.1.2. 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.

11.3.1.4 EPC Information Services 1.0 (EPC-IS)

EPCIS provides EPC data sharing services between disparate applications both within and across enterprises. For a complete description of EPCIS, see [EPCIS1.0]

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 parties' 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 specification 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 specification 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 specification also includes an HTTPS binding for the EPCIS query callback interface.

11.3.2 EPCglobal Core Services

EPCglobal provides core services as part of the EPCglobal network. The following section describes the data protection methods employed by these services.

11.3.2.1 Object Name Service 1.0 (ONS)

The EPCglobal ONS core service is based on the current internet DNS. ONS provides authoritative lookup of information about an electronic identifier. See [ONS1.0] 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 objectID within 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 host, and the domain server returns IP address information for the host in question. The system is a hierarchical set of DNS servers, culminating that 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].

11.3.2.2 Discovery

Discovery has not been addressed in the existing architecture.

11.3.2.3 Number Assignment

Manager ID number assignment is provided by EPCglobal core services. These documents are provided as standard text files on the EPCglobal public web site. Currently, these files contain only a list of the assigned manager numbers, and do not contain any information on the assignee of each ID.

11.3.3 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.

11.3.3.1 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 and 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 [UHFC1G21.0.9].

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.

11.3.3.1.1 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 10's 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 tag's 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.

11.3.3.1.2 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 “guessing” 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.

11.3.3.1.3 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 furnishes 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.

11.3.3.1.4 Kill Command

Gen2 contains a command to “kill” 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.

11.3.4 Data Format

11.3.4.1 Tag Data Standard (TDS)

The Tag Data Standard, currently version 1.3, 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 specification 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 specification, see [TDS1.3]

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 ID information (e.g. SGTIN) and serial number. TDS also specifies encoding of formats that contain company prefix and serial number, but do not contain object identification information.

The TDS specification 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.

11.3.5 Security

Several standards within the EPCglobal network were created specifically to address security issues of shared data.

11.3.6 EPCglobal X.509 Certificate Profile

The authentication of entities (subscribers, services, physical devices) operating within the EPCglobal network serves as the foundation of any security function incorporated into the network. The EPCglobal architecture 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 within the EPCglobal network. 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 [Cert1.0]

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

11.3.7 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 specification.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

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.

12 References

[ALE1.0] EPCglobal, "The Application Level Events (ALE) Specification, Version 1.0," EPCglobal Ratified Standard, September 2005, http://www.epcglobalinc.org/standards/ale/ale_1_0-standard-20050915.pdf.

[ALE1.1] EPCglobal, "The Application Level Events (ALE) Specification, Version 1.1," EPCglobal Last Call Working Draft, August 2007.

[Cert1.0] EPCglobal, "EPCglobal Certificate Profile 1.0," EPCglobal Ratified Standard, March, 2006, http://www.epcglobalinc.org/standards/cert/cert_1_0-standard-20060308.pdf.

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

[EPCIS1.0] EPCglobal, "EPC Information Services (EPCIS) Version 1.0 Specification," EPCglobal Ratified Standard, April 2007, http://www.epcglobalinc.org/standards/epcis/epcis_1_0-standard-20070412.pdf.

[GS1GS] GS1, "General Specifications v7.1," January 2007, http://www.gs1uk.org/EANUCC/

[HFC1] MIT Auto-ID Center, "13.56 MHz ISM Band Class 1Radio Frequency Identification Tag Interface Specification: Candidate Recommendation, Version 1.0.0," February 2003, http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-Class1.pdf.

[HFC1V2] EPCglobal, "HF Version 2," EPCglobal Last Call Working Draft, August, 2007.

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

[LLRP1.0] EPCglobal, "EPCglobal Low Level Reader Protocol (LLRP), Version 1.0.1", Ratified EPCglobal Standard, August 2007, http://www.epcglobalinc.org/standards/llrp/llrp_1_0_1-standard-20070813.pdf.

[ONS1.0] EPCglobal, "EPCglobal Object Naming Service (ONS), Version 1.0," EPCglobal Ratified Standard, October 2005, http://www.epcglobalinc.org/standards/ons/ons_1_0-standard-20051004.pdf.

[Pedigree1.0] EPCglobal, "Pedigree Ratified Standard, Version 1.0," EPCglobal Ratified Standard, January, 2007, http://www.epcglobalinc.org/standards/pedigree/pedigree_1_0-standard-20070105.pdf.

[RFC1034] P. V. Mockapetris, "Domain names – concepts and facilities." RFC1034, November 1987, http://www.ietf.org/rfc/rfc1034.

[RFC1035] P. V. Mockapetris, "Domain names – implementation and specification." RFC1035, November 1987, http://www.ietf.org/rfc/rfc1035.

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

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

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

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

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

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

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

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

[RM1.0] "Reader Management 1.0.1," EPCglobal Ratified Standard, May 2007, http://www.epcglobalinc.org/standards/rm/rm_1_0_1-standard-20070531.pdf.

[DCI] EPCglobal, "Discovery, Configuration, and Initialization (DCI) for Reader Operations", EPCglobal Candidate Specification, August 2007.

[RP1.1] EPCglobal, "EPCglobal Reader Protocol Standard, Version 1.1," EPCglobal Ratified Standard, June 2006, http://www.epcglobalinc.org/standards/rp/rp_1_1-standard-20060621.pdf.

[SDP1.3] EPCglobal, "EPCglobal Standards Development Process Version 1.3," EPCglobal publication, February 2007, http://www.epcglobalinc.org/standards/sdp/EPCglobal_SDP_10002.3_Feb_27_2007.pdf.

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

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

[TDS1.3] EPCglobal, "EPCglobal Tag Data Standards Version 1.3," EPCglobal Ratified Standard, March 2006, http://www.epcglobalinc.org/standards/tds/tds_1_3-standard-20060308.pdf.

[TDT1.0] EPCglobal, "EPCglobal Tag Data Translation (TDT) 1.0," EPCglobal Ratified Standard, January 2006, http://www.epcglobalinc.org/standards/tdt/tdt_1_0-standard-20060121.pdf.

[UHFC0] MIT Auto-ID Center, "Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag," EPCglobal Specification, Februrary 2003, http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class0.pdf.

[UHFC1G1] MIT Auto-ID Center, "860MHz–930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Candidate Recommendation, Version 1.0.1," EPCglobal Specification, November 2002, http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf.

[UHFC1G21.0.9] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.0.9," EPCglobal Standard, January 2006, http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_0_9-standard-20050126.pdf.

[UHFC1G21.1.0] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.1.0," EPCglobal Proposed Specification, March 2007.

[UHFC1G21.2.0] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version 1.2.0," EPCglobal Last Call Working Draft, March 2007.

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

13 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 ("EPCglobal Standards"), together with services operated by EPCglobal ("EPCglobal Core Services"), all in service of a common goal of enhancing the supply chain through the use of Electronic Product Codes (EPCs).

Term	Section	Meaning
EPCglobal Standards	1	Specifications for hardware and software interfaces through which components of the EPCglobal Architecture Framework interact. EPCglobal Standards are developed through the EPCglobal Standards Development Process. EPCglobal standards are implemented both by EPCglobal Core Services and by systems deployed by end user companies and their solution providers.
EPCglobal Core Services	1	Network-accessible services, operated by EPCglobal and its delegates, that provide common services to all subscribers of the EPCglobal Network, through interfaces defined as part of the EPCglobal Architecture Framework.
EPCglobal Network	1	An informal marketing term used to refer loosely to EPCglobal Subscribers and their interaction with EPCglobal and with each other, where that interaction takes place directly through the use of EPCglobal Standards and indirectly through EPCglobal Core Services.
EPCglobal Subscriber	1	An organization that participates in the EPCglobal Network through the use of EPCglobal Core Services, or through participation in the EPCglobal Standards Development Process. An EPCglobal Subscriber may be an End-User, a Solution Provider, or both.
End-user	1	An EPCglobal Subscriber that employs EPCglobal Standards and EPCglobal Core Services as a part of its business operations.
Solution Provider	1	An organization that implements systems on behalf of end-users that use EPCglobal Standards and EPCglobal Core Services. A Solution Provider may or may not itself be an EPCglobal Subscriber.
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 exchange among enterprise information systems.

Copyright © 2005 – 2007 EPCglobal®, All Rights Reserved.

Term	Section	Meaning
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-based codes such as SGTIN, SSCC, etc) and the US Department of Defense (for DoD codes). An Issuing Agency issues a block of EPCs to an EPC Manager, who may then commission individual EPCs without further coordination.
EPC Manager	5.2	An EPCglobal Subscriber that has been allocated a block of Electronic Product Codes by an Issuing Agency.
EPC Manager Number	5.3	A number that uniquely identifies one or more blocks of Electronic Product Codes issued to an EPC Manager.
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	"A 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." [ISO19762-3]

14 Acknowledgements

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 O'Shea (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).