



2 **The EPCglobal Architecture Framework**

3 EPCglobal Final Version of 1 July 2005

4

5 **Authors:**

6 Ken Traub (ConnecTerra, Inc.) kt@connecterra.com, Editor

7 Greg Allgair (formerly of EPCglobal)

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

9 Leo Burstein (Gillette) leo_burstein@gillette.com

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

11 Bernie Hogan (GS1 US) bhogan@gs1us.org

12 Bryan Rodrigues (CVS) brodrigues@cv.com

13 Sanjay Sarma (MIT) sarma@mit.edu

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

15 Chuck Schramek (EPCglobal) cschramek@epcglobalus.org

16 Roger Stewart (AWID) roger@awid.com

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

18

19 Copyright ©2005, 2004 [EPCglobal](#)[®], All Rights Reserved.

20 **Abstract**

21 This document defines and describes the EPCglobal Architecture Framework. The
22 EPCglobal Architecture Framework is a collection of interrelated standards for hardware,
23 software, and data interfaces, together with core services that are operated by EPCglobal
24 and its delegates, all in service of a common goal of enhancing the supply chain through
25 the use of Electronic Product Codes (EPCs). This document has several aims:

- 26 • To enumerate, at a high level, each of the hardware, software, and data standards that
27 are part of the EPCglobal Architecture Framework and show how they are related.
- 28 • To define the top level architecture of core services that are operated by EPCglobal
29 and its delegates.
- 30 • To explain the underlying principles that have guided the design of individual
31 standards and core service components within the EPCglobal Architecture
32 Framework.
- 33 • To provide architectural guidance to end users and technology vendors seeking to
34 implement EPCglobal standards and to use EPCglobal core services.

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

39 **Audience for this document**

40 The audience for this document includes:

- 41 • Hardware developers working in the areas of developing EPC tags and EPC-enabled
42 systems and appliances, including devices to read and write tag data.
- 43 • Software developers working in the areas of developing EPC middleware and
44 business applications that use, create, store and/or exchange EPC-related information.
- 45 • Enterprise architects and systems integrators that integrate EPC-related processes and
46 applications into enterprise architectures.
- 47 • Participants of EPCglobal Working Groups (including Software Action Group,
48 Hardware Action Group and all Business Action Groups) working on defining
49 requirements and developing EPCglobal standards.
- 50 • Industry groups, governing organizations, and companies that are developing or
51 overseeing business processes that rely on EPC technology.
- 52 • Members of the general public who are interested in understanding the principles and
53 terminology of the EPCglobal Architecture Framework

54 **Status of this document**

55 This section describes the status of this document at the time of its publication. Other
56 documents may supersede this document. The latest status of this document series is
57 maintained at EPCglobal. See www.epcglobalinc.org for more information.

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

59 Comments on this document should be sent to the EPCglobal Architecture Review
60 Committee mailing list arc_request@epclinklist.epcglobalinc.org.

61 **Table of Contents**

62	1	Introduction	6
63	2	Architecture Framework Overview	8
64	2.1	Architecture Framework Activities	8
65	2.2	Architecture Framework Standards	9
66	3	Goals for the EPCglobal Architecture Framework	10
67	3.1	The Role of Standards	10
68	3.2	Global Standards.....	11
69	3.3	Open System.....	11
70	3.4	Platform Independence	11
71	3.5	Scalability and Extensibility	12
72	3.6	Security	12
73	3.7	Privacy	12
74	3.8	Industry Architectures and Standards.....	12
75	3.9	Open, Community Process	13
76	4	Underlying Technical Principles	13
77	4.1	Unique Identity	13
78	4.2	Decentralized Implementation.....	15
79	4.3	Layering of Data Standards – Verticalization	16
80	4.4	Layering of Software Specifications—Technology Agnosticism	16
81	4.5	Extensibility.....	17
82	5	Architectural Foundations	17
83	5.1	Electronic Product Code.....	17
84	5.2	EPC Manager.....	17

85 5.3 EPC Manager Number..... 18

86 5.4 Embedding of Existing Codes 19

87 5.5 Class Level Data versus Instance Level Data..... 19

88 5.6 EPC Information Services (EPCIS)..... 20

89 6 Roles and Interfaces – General Considerations..... 21

90 6.1 Architecture Framework vs. System Architecture 22

91 6.2 Cross-Enterprise versus Intra-Enterprise..... 23

92 7 Data Flow Relationships – Cross-Enterprise 23

93 7.1 Data Exchange Interactions 25

94 7.2 Object Exchange Interactions..... 27

95 7.3 ONS Interactions 27

96 7.4 Number Assignment..... 28

97 8 Data Flow Relationships – Intra-Enterprise 28

98 9 Roles and Interfaces – Reference 32

99 9.1 Roles and Interfaces – Responsibilities and Collaborations..... 35

100 9.1.1 RFID Tag (Role)..... 35

101 9.1.2 EPC Tag Data Specification (Interface) 35

102 9.1.3 Tag Protocol (Interface)..... 35

103 9.1.4 RFID Reader (Role)..... 36

104 9.1.5 Reader Protocol (Interface)..... 36

105 9.1.6 Reader Management Interface (Interface) 37

106 9.1.7 Reader Management (Role) 38

107 9.1.8 Filtering & Collection (Role)..... 38

108 9.1.9 Filtering & Collection (ALE) Interface (Interface) 39

109 9.1.10 EPCIS Capturing Application (Role) 40

110 9.1.11 EPCIS Capture Interface (Interface)..... 40

111 9.1.12 EPCIS Query Interface (Interface) 41

112 9.1.13 EPCIS Accessing Application (Role)..... 41

113 9.1.14 EPCIS Repository (Role)..... 41

114 9.1.15 Object Name Service (ONS) Interface (Interface) 41

115 9.1.16 Local ONS (Role)..... 41

116 9.1.17 ONS Root (Core Service)..... 42

117	9.1.18	Manager Number Assignment (Core Service)	42
118	9.1.19	Tag Data Translation Schema (Core Service)	42
119	9.1.20	Tag Data Translation Interface (Interface)	42
120	9.1.21	EPCIS Discovery (Core Service – TBD)	42
121	9.1.22	Subscriber Authentication (Core Service – TBD).....	43
122	9.1.23	Filtering & Collection Management Interface (Interface – TBD).....	43
123	10	Summary of Unaddressed Issues	44
124	10.1	EPCIS “Discovery”	44
125	10.2	Subscriber Authentication	44
126	10.3	RFID Reader Coordination	45
127	10.4	RFID Tag-level Security and Privacy	45
128	10.5	“User Data” in RFID Tags	45
129	10.6	Tag Writing, Killing, Locking above the Reader Protocol Layer	46
130	10.7	Master Data for RFID Tag Manufacture Data	46
131	11	References	46
132	12	Glossary	47
133	13	Questions Raised During Review	49
134	13.1	Questions about the Electronic Product Code and its use	49
135	13.2	Questions about EPCIS and ONS	51
136	13.3	Questions about Infrastructure Standards	52
137	13.4	Questions about Security and Privacy.....	52
138	14	Acknowledgements.....	52
139			

140 **1 Introduction**

141 This document defines and describes the EPCglobal Architecture Framework. The
142 EPCglobal Architecture Framework is a collection of interrelated standards for hardware,
143 software, and data interfaces (“EPCglobal Standards”), together with core services that
144 are operated by EPCglobal and its delegates (“EPCglobal Core Services”), all in service
145 of a common goal of enhancing the supply chain through the use of Electronic Product
146 Codes (EPCs).

147 The primary beneficiaries of the EPCglobal Architecture Framework are EPCglobal
148 Subscribers and other Solution Providers. An EPCglobal Subscriber is any organization
149 that uses EPCglobal Core Services, or participates in the EPCglobal Standards
150 Development Process to develop EPCglobal Standards. EPCglobal Subscribers may be
151 further classified as End-users or Solution Providers (or both). An End-user is an
152 EPCglobal Subscriber that employs EPCglobal Standards and EPCglobal Core Services
153 as a part of its business operations. A Solution Provider is an organization that
154 implements for End-users systems that use EPCglobal Standards and EPCglobal Core
155 Services. (A Solution Provider may or may not itself be an EPCglobal Subscriber.)
156 Informally, the synergistic effect of EPCglobal Subscribers interacting with EPCglobal
157 and with each other using elements of the EPCglobal Architecture Framework is called
158 the “EPCglobal Network.”

159 This document has several aims:

- 160 • To enumerate, at a high level, each of the hardware, software, and data standards that
161 are part of the EPCglobal Architecture Framework and show how they are related.
162 These standards are implemented by hardware and software systems in the EPCglobal
163 Network, including components deployed by individual EPCglobal subscribers as
164 well as EPCglobal Core Services deployed by EPCglobal and its delegates.
- 165 • To define the top level architecture of EPCglobal Core Services, which provide
166 common services to all subscribers of the EPCglobal Network, through interfaces
167 defined as part of the EPCglobal Architecture Framework.
- 168 • To explain the underlying principles that have guided the design of individual
169 standards and Core Service components within the EPCglobal Network. These
170 underlying principles provide unity across all elements of the EPCglobal Architecture
171 Framework, and provide guidance for the development of future standards and new
172 Core Services.
- 173 • To provide architectural guidance to end users and technology vendors seeking to
174 implement EPCglobal Standards and to use EPCglobal Core Services, and to set
175 expectations as to how these elements will function.

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

180 • Individual hardware, software, and data interfaces are defined normatively by
181 EPCglobal specifications, or by standards produced by other standards bodies.
182 EPCglobal specifications are developed by EPCglobal membership through the
183 EPCglobal Standards Development Process (SDP) [SDP1.1]. EPCglobal
184 specifications are normative, and implementations are subject to conformance and
185 certification requirements.

186 An example of an interface is the UHF Class 1 Gen 2 Tag Protocol, that specifies a
187 radio-frequency communications protocol by which a Radio Frequency Identification
188 (RFID) tag and an RFID reader device may interact. This interface is defined
189 normatively by the UHF Class 1 Gen 2 Tag Protocol Specification.

190 • The design of hardware and software components that implement EPCglobal
191 specifications are proprietary to the vendors and end users that create such
192 components. While EPCglobal specifications provide normative guidance as to the
193 behavior of interfaces between components, implementers are free to innovate in the
194 design of components so long as they correctly implement the interface
195 specifications.

196 An example of a component is an RFID tag that is the product of a specific tag
197 manufacturer. This tag may comply with the UHF Class 1 Gen 2 Tag Protocol
198 Specification.

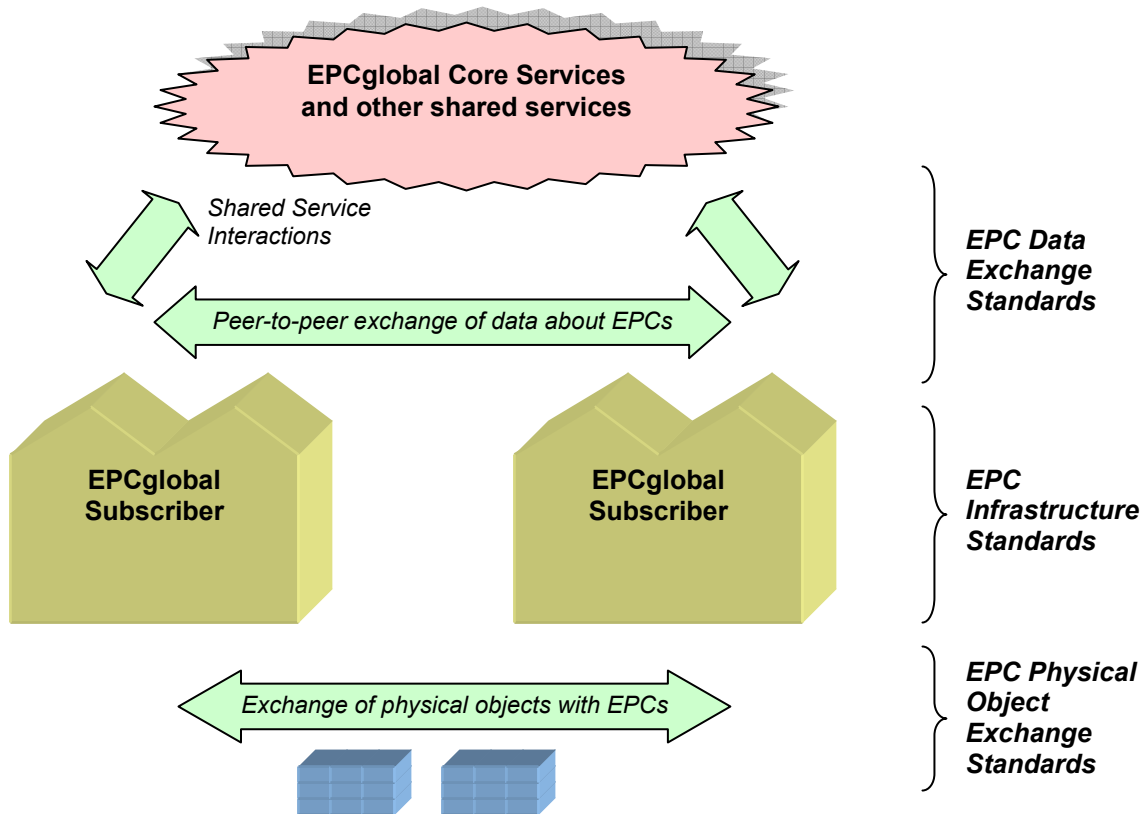
199 • A special case of components that implement EPCglobal specifications are
200 components that are operated and deployed by EPCglobal itself (or by other
201 organizations to whom EPCglobal delegates responsibility). These components are
202 referred to as EPCglobal Core Services, and provide services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

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

211 At the time of this writing, there are many parts of the EPCglobal Architecture
212 Framework that are well understood, and for which EPCglobal standards already exist or
213 are currently in development. There are other parts of the EPCglobal Architecture
214 Framework that are less well understood, but where a need is believed to exist based on
215 the analysis of known use cases. In these cases, the architectural approach has not yet
216 been finalized, and therefore work on developing standards or designing additional Core
217 Services has not yet begun (though architectural analysis is underway within the
218 Architecture Review Committee). This document clearly identifies which parts of the
219 EPCglobal Architecture Framework are understood architecturally and which parts need
220 further work. This document will be the basis for working through and ultimately
221 documenting the architectural decisions around the latter parts as work continues.

222 **2 Architecture Framework Overview**

223 The diagram below illustrates the activities carried out by EPCglobal Subscribers and the
224 role that components of EPCglobal Architecture Framework play in facilitating those
225 activities.



226

227 **2.1 Architecture Framework Activities**

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

- 230 • *EPC Physical Object Exchange* Subscribers exchange physical objects that are
231 identified with Electronic Product Codes (EPCs). For many end users of the
232 EPCglobal Network, the physical objects are trade goods, the subscribers are parties
233 in a supply chain for those goods, and physical object exchange consists of such
234 operations as shipping, receiving, and so on. There are many other uses that are quite
235 different than this trade goods model, but those other uses still involve the tagging of
236 some object or objects. The EPCglobal Architecture Framework defines EPC
237 physical object exchange standards, designed to ensure that when one subscriber
238 delivers a physical object to another subscriber, the latter will be able to determine the
239 EPC of the physical object and interpret it properly.
- 240 • *EPC Data Exchange* Subscribers benefit from the EPCglobal Network by
241 exchanging data with each other, increasing the visibility they have with respect to
242 the movement of physical objects outside their four walls. The EPCglobal

243 Architecture Framework defines EPC data exchange standards, which provide a
 244 means for one subscriber to share data about EPCs with another through direct peer-
 245 to-peer interaction, and which also provide access to EPCglobal Core Services and
 246 other shared services that facilitate these exchanges.

- 247 • *EPC Infrastructure* In order to have EPC data to share, each subscriber carries out
 248 operations within its four walls that create EPCs for new objects, follow the
 249 movements of objects by sensing their EPC codes, and gather that information into
 250 systems of record within the organization. The EPCglobal Architecture Framework
 251 defines interface standards for the major infrastructure components required to gather
 252 and record EPC data, thus allowing subscribers to build their internal systems using
 253 interoperable components.

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

265 **2.2 Architecture Framework Standards**

266 The following table summarizes all standards within the EPCglobal Architecture
 267 Framework in terms of the three activities described in the preceding section. A fuller
 268 description of each standard is given in Section 5. This table is intended mainly as an
 269 index of all current components of the EPCglobal Architecture Framework, not a
 270 roadmap for future work.

Activity	Standard	Status	Reference
Object Exchange	UHF Class 0 Gen 1 RF Protocol	(Note 4, below)	[UHFC0]
	UHF Class 1 Gen 1 RF Protocol	(Note 4, below)	[UHFC1G1]
	HF Class 1 Gen 1 Tag Protocol	(Note 5, below)	[HFC1]
	UHF Class 1 Gen 2 Tag Protocol	Ratified	[UHFC1G2]
	EPC Tag Data Specification	Ratified	[TDS1.1]
Infrastructure	Reader Protocol	In development	[RP1.0]
	Reader Management	In development	[RM1.0]
	Tag Data Translation	In development	[TDT1.0]

Activity	Standard	Status	Reference
Data Exchange	Application Level Events (ALE)	In development	[ALE1.0]
	EPCIS Capture Interface	In development	[EPCIS1.0]
	EPCIS Data Specification	In development	[EPCIS1.0]
	EPCIS Query Interface	In development	[EPCIS1.0]
	ONS	In development	[ONS1.0]
	EPCIS Discovery	TBD (Note 3)	(none)
	Subscriber Authentication	TBD (Note 3)	(none)

271 Notes for the “Status” column of the table above:

- 272 1. “Ratified” indicates a ratified EPCglobal specification.
- 273 2. “In development” indicates a specification whose development has been chartered
274 and is underway within the EPCglobal standards development process
- 275 3. “TBD” indicates a technical area that is expected to be addressed within the
276 EPCglobal Architecture Framework but where requirements are still under study
277 within the Business Action Groups or the Architecture Review Committee.
- 278 4. Prior to the launch of EPCglobal in November 2003, the former Auto-ID Center
279 published two UHF tag protocol specifications, referred to herein as UHF Class 0
280 Gen 1 and UHF Class 1 Gen 1. These specifications, which are not EPCglobal
281 standards, are superceded by the UHF Class 1 Gen 2 protocol which was ratified by
282 EPCglobal in December 2004.
- 283 5. Prior to the launch of EPCglobal in November 2003, the former Auto-ID Center also
284 published an HF tag protocol specification referred to herein as HF Class 1. The
285 business requirement for EPCglobal standardization of an HF tag protocol is being
286 assessed at the time of this writing.

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

291 **3 Goals for the EPCglobal Architecture Framework**

292 This section outlines high-level goals for the EPCglobal Architecture Framework in
293 terms of the benefits provided to EPCglobal Subscribers.

294 **3.1 The Role of Standards**

295 EPCglobal standards are created to further the following objectives:

- 296 • *To facilitate the exchange of information and physical objects between trading*
297 *partners.*
- 298 For trading partners to exchange information, they must have prior agreement as to
299 the structure and meaning of data to be exchanged, and the mechanisms by which
300 exchange will be carried out. EPCglobal standards include data standards and
301 information exchange standards that form the basis of cross-enterprise exchange.
302 Likewise, for trading partners to exchange physical objects, they must have prior
303 agreement as to how physical objects will carry Electronic Product Codes in a
304 mutually understandable way. EPCglobal standards include specifications for RFID
305 devices and data standards governing the encoding of EPCs on those devices.
- 306 • *To foster the existence of a competitive marketplace for system components.*
- 307 EPCglobal standards define interfaces between system components that facilitate
308 interoperability from components produced by different vendors (or in house). This
309 in turn provides choice to end users, both in implementing systems that will exchange
310 information between trading partners, and systems that are used entirely within four
311 walls.
- 312 • *To encourage innovation*
- 313 EPCglobal standards define *interfaces*, not *implementations*. Implementers are
314 encouraged to innovate in the products and systems they create, while interface
315 standards ensure interoperability between competing systems.

316 **3.2 Global Standards**

317 EPCglobal is committed to the creation and use of global standards. This approach
318 ensures that the EPCglobal Architecture Framework will work anywhere in the world and
319 provides incentives for Solution Providers to support the framework. EPCglobal
320 standards are developed for global use. EPCglobal is committed to making use of
321 existing global standards when appropriate, and EPCglobal works with recognized global
322 standards organizations to ratify standards created within EPCglobal.

323 **3.3 Open System**

324 The EPCglobal Architecture Framework is described in an open and vendor neutral
325 manner. All interfaces between architectural components are specified in open standards,
326 developed by the community through the EPCglobal Standards Development Process or
327 an equivalent process within another standards organization. The Intellectual Property
328 policy of EPCglobal is designed to secure free and open rights to implement EPCglobal
329 Standards in the context of conforming systems, to the extent possible.

330 **3.4 Platform Independence**

331 The EPCglobal Architecture Framework can be implemented on heterogeneous software
332 and hardware platforms. The specifications are platform independent meaning that the
333 structure and semantics of data in an abstract sense is specified separately from the
334 concrete details of data access services and bindings to particular interface protocols.

335 Where possible, interfaces are specified using platform and programming language
336 neutral technology (e.g., SOAP messaging [SOAP1.2]).

337 **3.5 Scalability and Extensibility**

338 The EPCglobal Architecture Framework is designed to scale to meet the needs of each
339 End-user, from a minimal pilot implementation conducted entirely within an End-user's
340 four walls, to a global implementation across entire supply chains. The specifications
341 provide a core set of data types and operations, but also provide several means whereby
342 the core set may be extended for purposes specific to a given industry or application area.
343 Extensions not only provide for proprietary requirements to be addressed in a way that
344 leverages as much of the standard framework as possible, but also provides a natural path
345 for the standards to evolve and grow over time.

346 **3.6 Security**

347 The EPCglobal Architecture Framework is designed to promote a secure environment for
348 operations inside and outside a company's four walls. Security features are either built
349 into the specifications, or best security practice is recommended.

350 **3.7 Privacy**

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

358 As the EPCglobal Privacy Policy evolves, it is anticipated that some elements of the
359 Privacy Policy will rely on features of the EPCglobal Architecture Framework for their
360 implementation. The UHF Class 1 Gen 2 Tag Protocol provides a 32-bit "Kill" code to
361 facilitate rapid destruction of tags at point-of-sale or other points where ownership of a
362 tagged item transitions to a member of the general public. Other architectural aspects are
363 expected to emerge as the Privacy Policy evolves, and will be described in future
364 versions of this document.

365 **3.8 Industry Architectures and Standards**

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

371 A specific example is the significant investment that FMCG companies have made in
372 data synchronization, which will continue for the foreseeable future. Depending on the

373 industry, participation in these and other enablers of e-commerce may be viewed as a
374 prerequisite for or as complementary to use of the EPCglobal Network in its goals for
375 improved supply chain operations.

376 **3.9 Open, Community Process**

377 The EPCglobal standards development process is designed to yield standards that are
378 relevant and beneficial to end users. Important aspects of the process include:

- 379 • End user involvement in developing requirements through the Business Action
380 Groups.
- 381 • Open process in which all EPCglobal subscribers having relevant expertise are
382 encouraged to join working groups that create new standards.
- 383 • Several review milestones in which new standards are vetted by a wide community
384 before final adoption.

385 **4 Underlying Technical Principles**

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

389 **4.1 Unique Identity**

390 A fundamental principle of the EPCglobal Network Architecture is the assignment of a
391 unique identity to physical objects, loads, locations, assets, and other entities whose use is
392 to be tracked. By “unique identity” is simply meant a name, such that the name assigned
393 to one entity is different than the name assigned to another entity. In the EPCglobal
394 Network Architecture, the unique identity is the Electronic Product Code, defined by the
395 EPCglobal Tag Data Specification [TDS1.1].

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

- 398 • *Uniqueness* The EPC assigned to one entity is different than the EPC assigned to
399 another (but see below for exceptions).
 - 400 • *Federation* The EPC is not a single naming structure, but a federation of several
401 naming structures. This allows existing naming structures to be incorporated into the
402 EPC system, so that the EPC is a universal identifier. This attribute is extremely
403 important to ensure wide adoption of the EPC, which would be significantly more
404 difficult if adoption required adoption of a single naming structure.
- 405 For example, both GS1 SSCC codes and GS1 GRAI codes [EANUCC] are also valid
406 EPCs. The various concrete representations of the EPC use a system of headers
407 (textual or binary according to the representation) to distinguish one identity scheme
408 from another; when one EPC is compared to another, the header is always included so

409 that EPCs drawn from different schemes will always be considered distinct. The
410 header is always considered to be a part of the EPC, not something separate.

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

416 • *Representation independence* EPCs are defined in terms of abstract structure, which
417 has several concrete realizations. Especially important are the binary realization that
418 is used on RFID tags and the Universal Resource Identifier (URI) realization that is
419 used for data exchange. Formal conversion rules exist, and work is underway (within
420 the Tag Data Translation Working Group) to create a machine-readable form of these
421 rules.

422 • *Decentralized assignment* EPCs are designed so that independent organizations can
423 assign new EPCs without the possibility of collision. This is done through a
424 hierarchical scheme, not unlike the Internet Domain Name System though somewhat
425 more structured. EPCglobal acts as the Registration Authority for the overall EPC
426 namespace. Each naming structure that is federated within the EPC namespace has a
427 space of codes managed by an Issuing Agency. For the EPC naming structures based
428 on the GS1 family of codes (SGTIN, SSCC, etc), for example, GS1 is the Issuing
429 Agency. An Issuing Agency allocates a portion of the EPC space to a subscriber
430 organization, who then becomes the “EPC Manager” for that block of codes. For
431 GS1 codes, for example, this is done by assigning a Company Prefix to a subscriber.
432 The subscriber is then free to assign EPCs within its allocated portion without any
433 further coordination with any outside agency. (Since there are several EPC naming
434 structures based on GS1 codes, assigning a single Company Prefix has the effect of
435 allocating several blocks of codes to a subscriber, one block within each GS1 coding
436 scheme.)

437 • *Structure* EPCs are not purely random strings, but rather have a certain amount of
438 internal structure in the form of designated fields. This plays a role in
439 decentralization, as described above. More significantly, the EPC’s internal structure
440 is essential to the scalability of lookup services such as the Object Name Service
441 which exploit the structure of EPCs to distribute lookup processing across a scalable
442 network of services.

443 • *Light Weight* EPCs have just enough structure and information to accomplish the
444 goals above, and no more. Other information associated with EPC-bearing entities is
445 not encoded into the EPC itself, but rather associated with the EPC through other
446 means.

447 While EPCs are intended to be globally unique in most situations, there are some
448 varieties of EPCs that are not. In particular, a portion of EPC space may be derived from
449 an existing coding scheme for which global uniqueness is not guaranteed. In that
450 situation, the EPCs from that space have uniqueness guarantees which are no stronger
451 than the original scheme. For example, GS1 SSCC codes are not unique over all time

452 and space, but due to the limited size of the SSCC namespace they are recycled
453 periodically. Good practice dictates that SSCCs be recycled no more frequently than the
454 lifetime of loads within the supply chain to which the SSCCs are affixed (plus a
455 reasonable data retention period). This eliminates the possibility that two identical
456 SSCCs would be present on two different loads at the same time, but it might still be
457 possible to find identical SSCCs for different loads in a long-term historical database.
458 Applications that rely on uniqueness properties of EPCs must understand the properties
459 of the various EPC namespaces that they might encounter, and act accordingly.

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

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

478 **4.2 Decentralized Implementation**

479 The EPCglobal Network seeks to link all enterprises together in a single global network.
480 Logically, the EPCglobal Network is a common resource shared by all EPCglobal
481 Subscribers. Considerations of scalability, of course, imply that it is not feasible to
482 literally implement this common resource as a single physical instance of a computer
483 system operated by a central authority. The EPCglobal Architecture Framework is
484 therefore decentralized, meaning that logically centralized functions are distributed
485 among one or more facilities serving individual EPCglobal Subscribers. In some cases,
486 certain of these facilities are operated by EPCglobal Subscribers themselves.

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

490 **4.3 Layering of Data Standards – Verticalization**

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

494 Consequently, EPCglobal standards that govern data are designed in a layered fashion.
495 Within each data standard, there is a framework layer that applies equally to all industries
496 that use the EPCglobal Network. Layered on top of this are several vertical data
497 standards that populate the general framework, each serving the needs of particular
498 industry groups. Vertical data standards may be broad or narrow in their applicability: in
499 many cases a vertical standard will serve several industries that share common business
500 processes, while in other cases a vertical standard will be particular to one industry. It is
501 even possible for a private group of trading partners to develop their own specifications
502 atop the framework similar to a vertical standard. The framework layers tend to be
503 developed by EPCglobal technical action groups, while the requirements for vertical
504 standards tend to be developed by appropriate industry groups.

505 The two important data standards are the EPC Tag Data Specification, and the EPCIS
506 Data Specification. Within the EPC Tag Data Specification, the framework elements
507 include the structure of the “header bits” in the binary EPC representations and the
508 general URI structure of of the text-based EPC representations. Both of these features
509 serve to distinguish one coding scheme from another. The vertical layer of the EPC Tag
510 Data Specification are the specific coding schemes defined for particular industry groups.

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

515 **4.4 Layering of Software Specifications—Technology** 516 **Agnosticism**

517 The EPCglobal Architecture Framework is primarily concerned with the exploitation of
518 new data derived from the use of Electronic Product Codes and RFID technology within
519 business processes. Most of the content of EPCglobal standards does not rely on specific
520 implementation technology such as web services, XML, AS2, EDI, and so on. Each
521 enterprise has its own requirements and preferences for underlying technologies such as
522 these, and these inevitably change over time.

523 To foster the broadest possible applicability for EPCglobal standards, EPCglobal
524 software standards are, whenever possible, defined using a layered approach. In this
525 approach, the abstract content of data and/or services is defined using a technology-
526 neutral description language such as UML. Separately, the abstract specifications are
527 given one or more bindings to specific implementation technology such as XML, web
528 services, and so forth. As most of the technical “meat” of EPCglobal specifications exists
529 in the abstract content, this approach helps ensure that even when different
530 implementation technologies are used in different deployments there is a strong
531 commonality in what the systems do.

532 **4.5 Extensibility**

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

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

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

553 **5 Architectural Foundations**

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

557 **5.1 Electronic Product Code**

558 As previously described in Section 4.1, the Electronic Product Code is the embodiment of
559 the underlying principle of unique identity. Electronic Product Codes are assigned to
560 physical objects, loads, locations, assets, and other entities which are to be tracked
561 through the EPCglobal Network in service of a given industry's business goals. The
562 Electronic Product Code is the thread that ties together all data that flows within the
563 EPCglobal Network, and plays a central part in every role and interface within the
564 EPCglobal Architecture Framework.

565 **5.2 EPC Manager**

566 As noted in Section 4.1, a key characteristic of identity as used in the EPCglobal
567 Architecture Framework is decentralization. Decentralization is achieved through the
568 notion of an EPC Manager. Within this document, the term "EPC Manager" refers to an
569 EPCglobal Subscriber who has been granted rights to use a portion of the EPC
570 namespace by an Issuing Agency. That is, the Issuing Agency has effectively issued the
571 EPC Manager one or more blocks of Electronic Product Codes within designated coding

572 schemes that the EPC Manager can independently assign to physical objects and other
573 entities without further involvement of the Issuing Agency. The EPC Manager is said to
574 be the “managing authority” for the EPCs in this block.

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

- 578 • The EPC Manager is responsible for allocating a new EPC from its assigned block,
579 and associating it with a physical object or other entity. In so doing, the EPC
580 Manager must ensure that the appropriate uniqueness properties are maintained (see
581 Section 4.1). The act of allocating a new EPC is called “commissioning.”
- 582 • The EPC Manager is responsible for maintaining the Object Name Service (ONS)
583 records associated with blocks of EPCs it manages. ONS records are the point of
584 entry for certain types of global lookup operations as described in later sections.
585 (This responsibility is limited to those blocks of EPCs that are allocated by the EPC
586 Manager for objects that are exchanged with other Subscribers; any EPC blocks
587 reserved for internal use by the EPC Manager need not be reflected in ONS. Also,
588 the EPC Manager may elect not to provide ONS lookup for any or all of its EPCs, in
589 which case there is obviously no requirement to maintain ONS records for those
590 EPCs.)

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

595 **5.3 EPC Manager Number**

596 The way that an Issuing Agency grants a block of EPCs to an EPC Manager is by issuing
597 the EPC Manager a single number, called the EPC Manager Number. An End-User
598 Subscriber may hold multiple Manager Numbers, and therefore be in control of multiple
599 blocks of EPCs. The structure of all coding schemes within the Electronic Product Code
600 definition is such that the EPC Manager Number appears as a distinct field within any
601 given representation.

602 Having the EPC Manager Number as a distinct field within any given representation,
603 allows any system to instantly identify the EPC Manager associated with a given EPC.
604 This property is very important to insure the scalability of the overall system, as it allows
605 services that would otherwise be centralized to be delegated to each EPC Manager as
606 appropriate. For example, an ONS lookup is conceptually a lookup in a single large table
607 that maps any EPC to the location of an EPCIS service, but having the EPC Manager
608 Number as a distinct field allows ONS to be implemented as a collection of tables, each
609 maintained by the EPC Manager for a given block of EPCs (see Section 7.3 for more
610 information on ONS specifically).

611 The allocation of a block of EPC codes to an EPC Manager is actually implicit in the act
612 of assigning an EPC Manager Number. The EPC Manager is simply free to commission
613 any EPC code so long as the EPC Manager Number field within the code contains the

614 assigned EPC Manager Number, following the EPC Tag Data Specification. The “block”
615 of codes, therefore, simply consists of all EPCs that contain the assigned EPC Manager
616 Number in the EPC Manager Number field. (This a slight simplification; see Section 5.4
617 for more information.)

618 **5.4 Embedding of Existing Codes**

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

625 In general, this kind of embedding is possible for any existing coding scheme that has a
626 manager-like structure; that is, when the existing coding scheme is based on delegating
627 assignment through the central allocation of a unique prefix or field. The US Department
628 of Defense, for example, has defined an EPC coding scheme based on its own CAGE and
629 DoDAAC codes, which are issued uniquely to DoD suppliers and thus serve as EPC
630 Manager Numbers when used to construct EPCs using the “DoD construct” coding
631 scheme.

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

643 **5.5 Class Level Data versus Instance Level Data**

644 EPCs are assigned uniquely to physical objects and other entities, allowing data to be
645 associated with individual objects. For example, one can associate data with a specific
646 24-count case of Cherry Hydro Soda by referring to its unique EPC.

647 In some cases, it is necessary to associate data with a class of object rather than a specific
648 object itself. In the case of consumer goods, an object class refers to all instances of a
649 specific product (Stock Keeping Unit, or SKU); for example, the class representing all
650 24-count cases of Cherry Hydro Soda. For Electronic Product Codes having a three-part
651 structure of EPC Manager Number, Object Class ID, and Serial Number, a product class
652 is uniquely identified by the first two numbers, disregarding the Serial Number. The
653 Serialized Global Trade Item Number (SGTIN) coding scheme used in the consumer
654 packaged goods industry is an example of an EPC having this structure. In this particular

655 example, the EPC Manager Number and Object Class ID are in fact convertible to the
656 GTIN code that is used outside of the EPC arena to represent product classes. This is
657 another example of how existing codes are embedded within the Electronic Product Code
658 framework.

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

664 **5.6 EPC Information Services (EPCIS)**

665 The primary vehicle for data exchange between EPCglobal Subscribers in the EPCglobal
666 Architecture Framework is EPC Information Services (EPCIS). As explained below,
667 EPCIS encompasses both interfaces for data exchange and specifications of the data
668 itself.

669 EPCIS data is information that trading partners share to gain more insight into what is
670 happening to physical objects in locations not under their direct control. (EPCIS data
671 may, of course, also be used within a company's four walls.) For most industries using
672 the EPCglobal Network, EPCIS data can be divided into five categories, as follows:

- 673 • *Static Data*, which does not change over the life of a physical object. This includes:
 - 674 • *Class-level Static Data*; that is, data which is the same for all objects of a given
675 object class (see Section 0). For consumer products, for example, the “class” is
676 the product, or SKU, as opposed to distinct instances of a given product. In many
677 industries, class-level static data may be the subject of existing data
678 synchronization mechanisms such as the Global Data Synchronization Network
679 (GDSN); in such instances, EPCIS may not be the primary means of exchange.
 - 680 • *Instance-level Static Data*, which may differ from one instance to the next within
681 a given object class. Examples of instance-level static data include such things as
682 date of manufacture, lot number, expiration date, and so forth. Instance-level
683 static data generally takes the form of attributes associated with specific EPCs.
- 684 • *Transactional Data*, which does grow and change over the life of a physical object.
685 This includes:
 - 686 • *Instance Observations*, which record events that occur in the life of one or more
687 specific EPCs. Examples of instance observations include “EPC X was shipped
688 at 12:03pm 15 March 2004 from Acme Distribution Center #2,” and “At 3:45pm
689 22 Jan 2005 the case EPCs (list here) were aggregated to the pallet EPC X at ABC
690 Corp's Boston factory.” Most instance observations have four dimensions: time,
691 location, one or more EPCs, and business process step.
 - 692 • *Quantity Observations*, which record events concerned with measuring the
693 quantity of objects within a particular object class. An example of a quantity
694 observation is “There were 4,100 instances of object class C observed at 2:00am

695 16 Jan 2003 in RetailMart Store #23.” Most quantity observations have five
696 dimensions: time, location, object class, quantity, and business process step.

- 697 • *Business Transaction Observations*, which record an association between one or
698 more EPCs and a business transaction. An example of a business transaction
699 observation is “The pallet with EPC X was shipped in fulfillment of Acme Corp
700 purchase order #23 at 2:20pm.” Most business transaction observations have four
701 dimensions: time, one or more EPCs, a business process step, and a business
702 transaction identifier.

703 The EPCIS Data Specifications provide a precise definition of all the types of EPCIS
704 data, as well as the meaning of “event” as used above.

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

713 A key challenge faced by the EPCglobal Network is to allow any participant in the
714 supply chain to discover all transactional data to which it is authorized, from any other
715 participant. Section 7.1 discusses how the EPCglobal Architecture Framework addresses
716 this challenge.

717 **6 Roles and Interfaces – General Considerations**

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

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

733 **6.1 Architecture Framework vs. System Architecture**

734 The EPCglobal Architecture Framework is a collection of interrelated standards for
735 hardware, software, and data interfaces (EPCglobal Standards), together with core
736 services that are operated by EPCglobal and its delegates (EPCglobal Core Services).
737 End users deploy systems that make use of these elements of the EPCglobal Architecture
738 Framework. In particular, each end user will have a system architecture for their
739 deployment that includes various hardware and software components, and these
740 components may use EPCglobal Standards to communicate with each other and with
741 external systems, and also make use of the EPCglobal Core Services to carry out certain
742 tasks.

743 The EPCglobal Architecture Framework does not define a system architecture that end
744 users must implement, nor does it dictate particular hardware or software components an
745 end user must deploy. The hardware and software components within any end user's
746 system architecture may be created by the end user or obtained by the end user from
747 vendors, but in any case the definition of these components is outside the scope of the
748 EPCglobal Architecture Framework. The EPCglobal Architecture Framework only
749 defines interfaces that the end user's components may implement. The EPCglobal
750 Architecture Framework explicitly avoids specification of components in order to give
751 end users maximal freedom in designing system architectures according to their own
752 preferences and goals, while defining interface standards to ensure that systems deployed
753 by different end users can interoperate and that end users have a wide marketplace of
754 vendor-provided components from which to choose.

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

- 764 • An end-user system architecture may only need to employ a subset of the EPCglobal
765 Standards and Core Services depicted here. For example, an RFID application using
766 EPC tags that exists entirely within the four walls of a single enterprise may use the
767 UHF Class 1 Gen 2 Tag Protocol and the EPC Tag Data Standards, but have no need
768 for the Object Name Service.
- 769 • The mapping between hardware and software roles depicted here and actual hardware
770 or software components deployed by an end-user may not necessarily be one-to-one.
771 For example, to carry out a business process of shipment verification using EPC-
772 encoded RFID tags, one end user may deploy a system in which there is a separate
773 RFID Reader (a hardware device), Filtering & Collection Middleware (software
774 deployed on a server), and EPCIS Capturing Application (software deployed on a
775 different server). Another end user may deploy an integrated verification portal
776 device that combines into a single package all three of these roles, exposing only the

777 EPCIS Capture Interface. For this reason, this document is careful to refer to *roles*
778 rather than *components* when talking about system elements that make use of
779 standard interfaces.

780 • In the same vein, roles depicted here may be carried out by an end user’s legacy
781 system components that may have additional responsibilities outside the scope of the
782 EPCglobal Architecture Framework. For example, it is common to have enterprise
783 applications such as Warehouse Management Systems that simultaneously play the
784 role of EPCIS Capturing Application (e.g., detecting EPCs during product movement
785 during truck loading), an EPCIS-enabled Repository (e.g., recording case-to-pallet
786 associations), and an EPCIS Accessing Application (e.g., carrying out business
787 decisions based on EPCIS-level data).

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

790 **6.2 Cross-Enterprise versus Intra-Enterprise**

791 As discussed in Section 2, elements of the EPCglobal Architecture Framework can be
792 categorized as pertaining to EPC Data Exchange between enterprises, EPC Object
793 Exchange between enterprises, or EPC Infrastructure deployed within a single enterprise.
794 Clearly, all EPCglobal Subscribers will find relevance in the first two categories, as use
795 of these standards is necessary to interact with other subscribers. A subscriber has much
796 more latitude, however, in its decisions surrounding adoption of the EPC Infrastructure
797 standards, as those standards do not affect parties outside the subscriber’s own four walls.

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

806 **7 Data Flow Relationships – Cross-Enterprise**

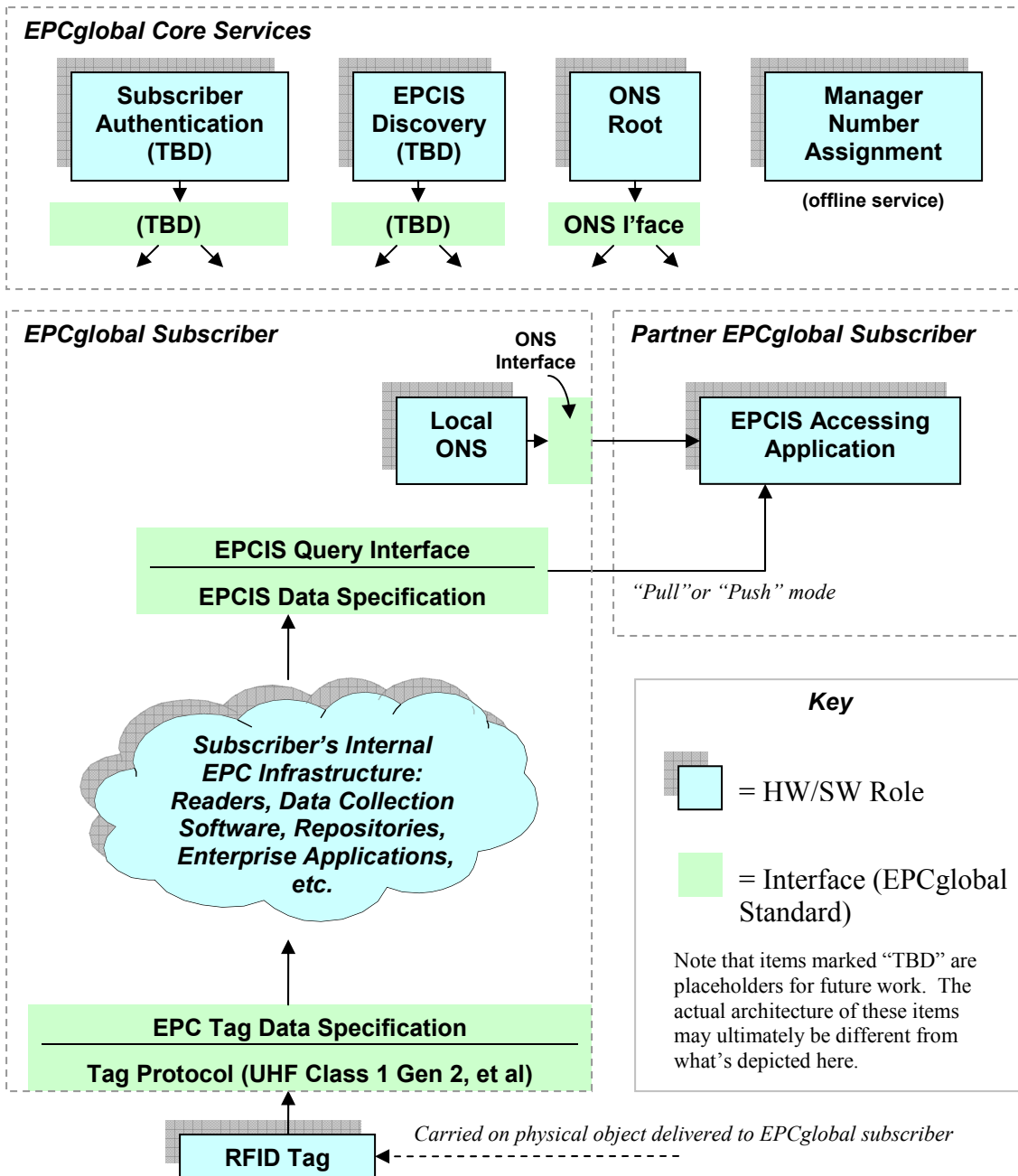
807 This section provides a diagram showing the relationships between EPCglobal Standards,
808 from a data flow perspective. This section shows only the EPCglobal Standards that are
809 typically used between subscribers, namely those categorized as “EPC Object Exchange
810 Standards” or “EPC Data Exchange Standards” in Section 2. EPCglobal Standards that
811 are primarily used within the four walls of a single subscriber (“EPC Infrastructure
812 Standards” from Section 2) are described in Section 8. Most EPCglobal Subscribers will
813 implement the architecture given in this section in order to fully participate in the
814 EPCglobal Network.

815 In the following diagram, the plain green bars denote interfaces governed by EPCglobal
816 standards, while the blue “shadowed” boxes denote roles played by hardware and
817 software components of a typical system architecture. As emphasized in Section 6.1, in

818 any given end user's deployment the mapping of roles in this diagram to actual hardware
819 and software components may not be one-to-one, nor will every end user's deployment
820 contain every role shown here.

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

829



830

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

834 **7.1 Data Exchange Interactions**

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

837 is some application specific to the Partner EPCglobal Subscriber that is interested in
838 information about a particular EPC.

839 The first thing the EPCIS Accessing Application needs to do is to determine where it can
840 go to obtain data of interest. In general, there are several ways it may do so:

- 841 • The EPCIS Accessing Application may know in advance exactly where to find the
842 information. This often arises in simple two-party supply chain scenarios, where one
843 party is given the network address of the other party's EPCIS service as part of a
844 business agreement.
- 845 • The EPCIS Accessing Application may know where to find the information it seeks
846 based on information obtained previously. For example, in a three-party supply chain
847 consisting of parties A, B, and C, party C may know how to reach B's service as part
848 of a business agreement, and in obtaining information from B it learns how to reach
849 A's service (which B knows as part of its business agreement with A). This is
850 sometimes referred to as "following the chain."
- 851 • The EPCIS Accessing Application may use the Object Name Service (ONS) to locate
852 the EPCIS service of the EPCglobal Subscriber who is the EPC Manager of the object
853 in question.
- 854 • The EPCIS Accessing Application may use EPCIS Discovery Services to locate the
855 EPCIS services of all EPCglobal Subscribers that have information about the object
856 in question, including EPCglobal Subscribers other than the EPC Manager of the
857 object. This method is required in the general case of multi-party supply chain, when
858 the participants are not known to the EPCIS Accessing Application in advance and
859 when it is not possible or impractical to "follow the chain." (EPCIS Discovery
860 Services are TBD at the time of this writing, so the precise architecture of roles and
861 interfaces involved in EPCIS Discovery Services is not yet known – the box in the
862 diagram is just a placeholder.)

863 Whatever method is used, the net result is that the EPCIS Accessing Application has
864 located the EPCIS service of the EPCglobal Subscriber from whom it will obtain data to
865 which the EPCIS Accessing Application is authorized. The EPCIS Accessing
866 Application then requests information directly from the EPCIS service of the other
867 subscriber. Two EPCglobal Standards govern this interaction. The EPCIS Query
868 Interface defines how data is requested and delivered from an EPCIS service. The EPCIS
869 Data Specification define the format and meaning of this data. The EPCIS Query
870 Interface is designed to support both on-demand or "pull" modes of data transfer, as well
871 as asynchronous or "push" modes. Several transport bindings are provided, including on-
872 line transport as well as disconnected (store and forward) transport.

873 When an EPCIS Accessing Application of the Partner EPCglobal Subscriber accesses the
874 EPCIS service of the first EPCglobal Subscriber, the first EPCglobal Subscriber will
875 usually want to authenticate the identity of the Partner EPCglobal Subscriber in order to
876 determine what data the latter is authorized to receive. The Subscriber Authentication
877 role in the diagram refers to an EPCglobal Core Service that assists in this authentication,
878 making it possible for any EPCglobal Subscriber to authenticate the identity of any other
879 EPCglobal Subscriber without any prior arrangement between the two parties.

880 (Subscriber Authentication is TBD at the time of this writing, so the precise architecture
881 of roles and interfaces involved in Subscriber Authentication is not yet known – the box
882 in the diagram is just a placeholder.) In some situations, an EPCglobal Subscriber may
883 grant EPCIS access to another party whose identity is not authenticated or authenticated
884 by means other than those facilitated by EPCglobal. This is a policy decision that is up to
885 each EPCglobal Subscriber to make.

886 **7.2 Object Exchange Interactions**

887 The lower part of the diagram illustrates how the first EPCglobal Subscriber interacts
888 with physical objects it receives from other subscribers. A physical object is received by
889 the EPCglobal Subscriber, bearing an RFID tag that contains an EPC code. The
890 EPCglobal Subscriber reads the tag using RFID Readers deployed as part of its internal
891 EPC infrastructure. Two EPCglobal Standards govern this interaction. A tag protocol
892 defines how data is carried through a radio signal to the RFID Reader. The EPC Tag
893 Data Specification defines the format and meaning of this data, namely the EPC code.

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

903 **7.3 ONS Interactions**

904 In Section 7.1, it was mentioned that one EPCglobal Subscriber may locate the EPCIS
905 service of the EPC Manager associated with a given EPC by using the Object Name
906 Service, or ONS. This section describes in somewhat more detail how this takes place as
907 a collaboration between an EPCglobal Core Service and a service provided by an
908 individual subscriber.

909 The Object Name Service can be thought of as a simple lookup service that takes an EPC
910 as input, and produces as output the address (in the form of a Uniform Resource Locator,
911 or URL) of an EPCIS service designated by the EPC Manager of the EPC in question.
912 (An EPC Manager may actually use ONS to associate several different services, not just
913 an EPCIS service, with an EPC. All of the following discussion applies equally
914 regardless of which type of service is looked up.) In general, there may be many
915 different object classes that fall under the authority of a single EPC Manager, and it may
916 not be the case that all object classes of a given EPC Manager will have information
917 provided by the same EPCIS service. Therefore, ONS requires a separate entry for each
918 object class. (The current design of ONS does not, however, permit different entries for
919 different serial numbers of the *same* object class. The current ONS specification also
920 does not address coding schemes which do not have a field corresponding to object class,
921 such as the SSCC and GIAI codes.)

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

928 ONS works as follows. When an EPCglobal Subscriber wishes to locate an EPCIS
929 service, it first consults the Root ONS service controlled by EPCglobal. The Root ONS
930 service identifies the Local ONS service of the EPC Manager organization for that EPC.
931 The EPCglobal Subscriber then completes the lookup by consulting the Local ONS
932 service, which provides the pointer to the EPCIS service in question.

933 ONS is implemented as an application of DNS, simply by specifying a convention
934 whereby an EPC is converted to an Internet Domain Name in the `onsepccom.com` domain.
935 This has several implications:

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

945 More information may be found in the DNS specifications [RFC1034, RFC1035], and in
946 the ONS Specification [ONS1.0].

947 **7.4 Number Assignment**

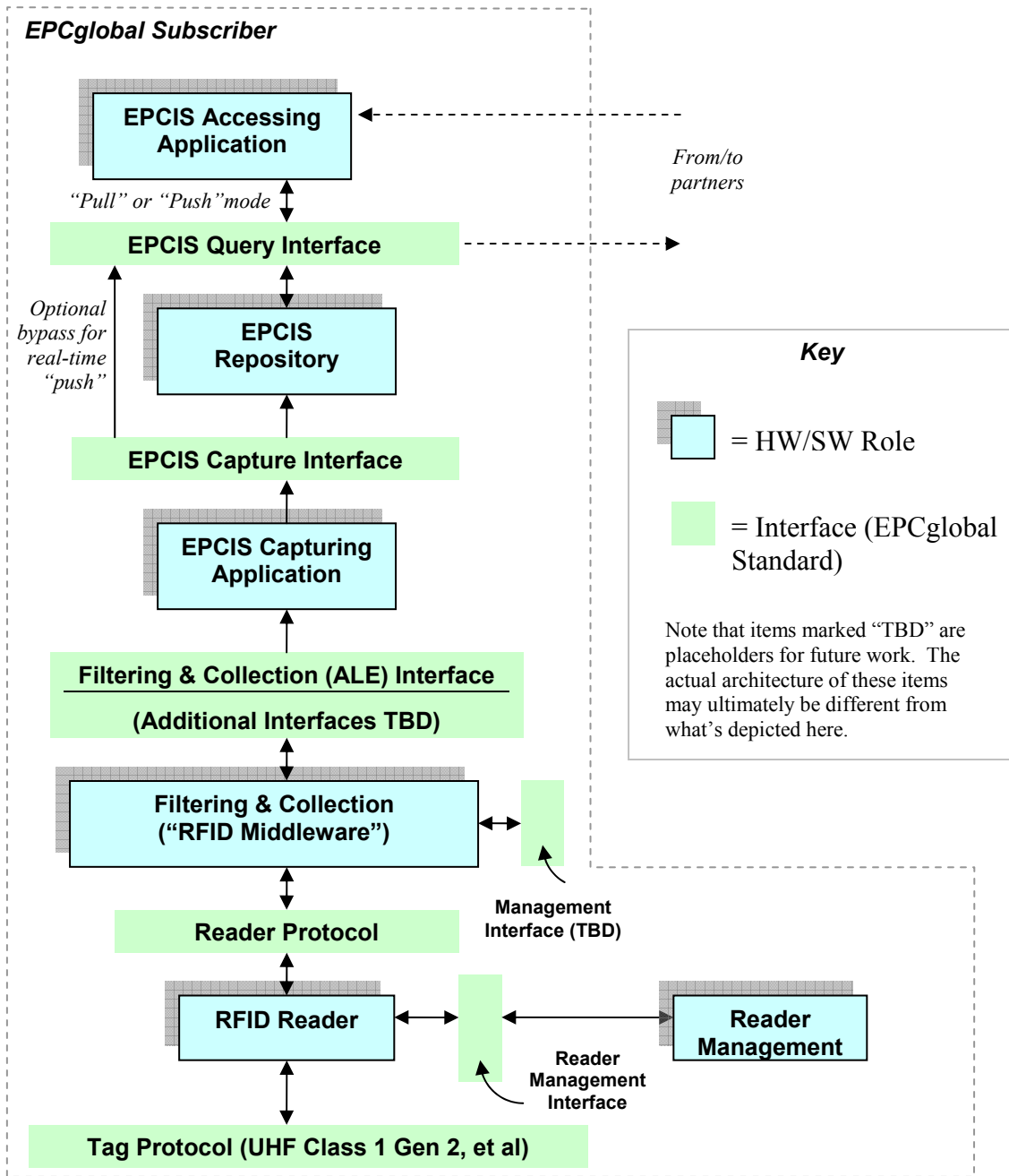
948 The foregoing text has described every role and interface in the diagram at the beginning
949 of this Section 7, except for Manager Number Assignment. This role simply refers to
950 EPCglobal's service of issuing unique EPC Manager Numbers to each EPC Manager
951 organization that requests one, in its capacity as the Issuing Agency for the GS1 family of
952 codes (see Section 4.1). By insuring that every EPC Manager Number that is issued is
953 unique, the uniqueness of EPCs assigned by individual EPCglobal Subscribers within the
954 GS1 family of codes is ensured. (Number assignment for other coding schemes is carried
955 out by Issuing Agencies other than EPCglobal, and so EPCglobal's Manager Number
956 Assignment Core Service does not apply in those cases.)

957 **8 Data Flow Relationships – Intra-Enterprise**

958 This section provides a diagram showing the relationships between EPCglobal Standards,
959 from a data flow perspective. In contrast to Section 7, this section shows only the
960 EPCglobal Standards that are typically used within the four walls of a single subscriber,
961 namely those categorized as "EPC Infrastructure Standards" in Section 2. This section

962 expands the “cloud” in the diagram from Section 7. Because this cloud is completely
963 internal to a given enterprise, a subscriber has much more latitude to deviate from this
964 picture when appropriate to that subscriber’s unique business conditions. EPCglobal sets
965 standards in this area, however, to encourage solution providers to create interoperable
966 system components from which subscribers may choose.

967 As in Section 7, the plain green bars in the diagram below denote interfaces governed by
968 EPCglobal standards, while the blue “shadowed” boxes denote roles played by hardware
969 and software components of a typical system architecture. As emphasized in Section 6.1,
970 in any given end user’s deployment the mapping of roles in this diagram to actual
971 hardware and software components may not be one-to-one, nor will every end user’s
972 deployment contain every role shown here.



973

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

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

- 983 • *Readers* Make multiple observations of RFID tags while they are in the read zone.
- 984 • *Reader Protocol Interface* Defines the control and delivery of raw tag reads from
985 Readers to the Filtering & Collection role. Events at this interface say “Reader A saw
986 EPC X at time T.”
- 987 • *Filtering & Collection* This role filters and collects raw tag reads, over time intervals
988 delimited by events defined by the EPCIS Capturing Application (e.g. tripping a
989 motion detector).
- 990 • *Filtering & Collection (ALE) Interface* Defines the control and delivery of filtered
991 and collected tag read data from Filtering & Collection role to the EPCIS Capturing
992 Application role. Events at this interface say “At Location L, between time T1 and
993 T2, the following EPCs were observed,” where the list of EPCs has no duplicates and
994 has been filtered by criteria defined by the EPCIS Capturing Application.
- 995 • *EPCIS Capturing Application* Supervises the operation of the lower EPC elements,
996 and provides business context by coordinating with other sources of information
997 involved in executing a particular step of a business process. The EPCIS Capturing
998 Application may, for example, coordinate a conveyor system with Filtering &
999 Collection events, may check for exceptional conditions and take corrective action
1000 (e.g., diverting a bad case into a rework area), may present information to a human
1001 operator, and so on. The EPCIS Capturing Application understands the business
1002 process step or steps during which EPCIS data capture takes place. This role may be
1003 complex, involving the association of multiple Filtering & Collection events with one
1004 or more business events, as in the loading of a shipment. Or it may be
1005 straightforward, as in an inventory business process where there may be “smart
1006 shelves” deployed that generate periodic observations about objects that enter or
1007 leave the shelf. Here, the Filtering & Collection-level event and the EPCIS-level
1008 event may be so similar that no actual processing at the EPCIS Capturing Application
1009 level is necessary, and the EPCIS Capturing Application merely configures and routes
1010 events from the Filtering & Collection interface directly to an EPCIS-enabled
1011 Repository.
- 1012 • *EPCIS Capture Interface* The interface through which EPCIS data is delivered to
1013 enterprise-level roles, including EPCIS Repositories, EPCIS Accessing Applications,
1014 and data exchange with partners. Events at this interface say, for example, “At
1015 location X, at time T, the following contained objects (cases) were verified as being
1016 aggregated to the following containing object (pallet).”
- 1017 • *EPCIS Accessing Application* Responsible for carrying out overall enterprise
1018 business processes, such as warehouse management, shipping and receiving,
1019 historical throughput analysis, and so forth, aided by EPC-related data.
- 1020 • *EPCIS Repository* Software that records EPCIS-level events generated by one or
1021 more EPCIS Capturing Applications, and makes them available for later query by
1022 EPCIS Accessing Applications.

1023 The interfaces within this stack are designed to insulate the higher levels of the stack
1024 from unnecessary details of how the lower levels are implemented. One way to
1025 understand this is to consider what happens if certain changes are made:

- 1026 • The Reader Protocol Interface insulates the higher layers from knowing what reader
1027 makes/models have been chosen. If a different reader is substituted, the information
1028 at the Reader Protocol Interface remains the same. The Reader Protocol Interface
1029 may, to some extent, also provide insulation from knowing what tag protocols are in
1030 use, though obviously not when one tag type or tag protocol provides fundamentally
1031 different functionality from another.
- 1032 • The Filtering & Collection Interface insulates the higher layers from the physical
1033 design choices made regarding how tags are sensed and accumulated, and how the
1034 time boundaries of events are triggered. If a single four-antenna reader is replaced by
1035 a constellation of five single-antenna “smart antenna” readers, the events at the
1036 Filtering & Collection level remain the same. Likewise, if a different triggering
1037 mechanism is used to mark the start and end of the time interval over which reads are
1038 accumulated, the Filtering & Collection event remains the same.
- 1039 • The EPCIS interfaces insulate enterprise applications from understanding the details
1040 of how individual steps in a business process are carried out at a detailed level. For
1041 example, a typical EPCIS event is “At location X, at time T, the following cases were
1042 verified as being on the following pallet.” In a conveyor-based business
1043 implementation, this likely corresponds to a single Filtering & Collection event, in
1044 which reads are accumulated during a time interval whose start and end is triggered
1045 by the case crossing electric eyes surrounding a reader mounted on the conveyor. But
1046 another implementation could involve three strong people who move around the cases
1047 and use hand-held readers to read the EPC codes. At the Filtering & Collection level,
1048 this looks very different (each triggering of the hand-held reader is likely a distinct
1049 Filtering & Collection event), and the processing done by the EPCIS Capturing
1050 Application is quite different (perhaps involving an interactive console that the people
1051 use to verify their work). But the EPCIS event is still the same.

1052 In summary, the different steps in the data path correspond to different semantic levels,
1053 and serve to insulate different concerns from one another as data moves up from raw tag
1054 reads towards EPCIS.

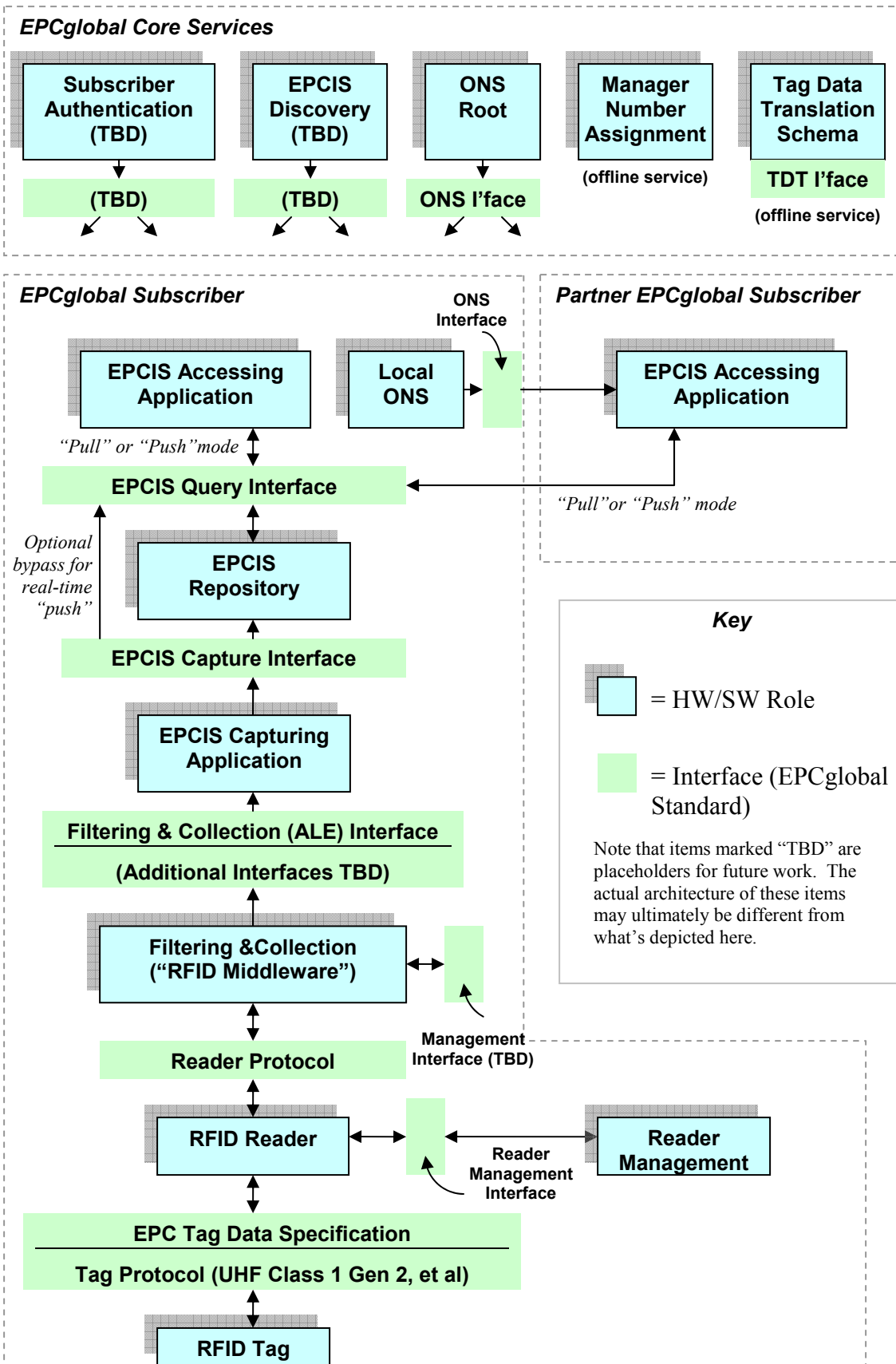
1055 Besides the data path described above, there is also a control path responsible for
1056 managing and monitoring of the infrastructure. At present, the only EPCglobal standard
1057 involved is the Reader Management standard.

1058 **9 Roles and Interfaces – Reference**

1059 This section provides a complete reference to all roles and interfaces described in
1060 Sections 7 and 8, describing each in more formal terms. For convenience, the following
1061 diagram combines the figures from the two previous sections into a single figure. As in
1062 Sections 7 and 8, the plain green bars in the diagram below denote interfaces governed by
1063 EPCglobal standards, while the blue “shadowed” boxes denote roles played by hardware
1064 and software components of a typical system architecture. As emphasized in Section 6.1,

1065 in any given end user's deployment the mapping of roles in this diagram to actual
1066 hardware and software components may not be one-to-one, nor will every end user's
1067 deployment contain every role shown here.

1068



1070 The next section explains the roles and interfaces in this diagram in more detail.

1071 **9.1 Roles and Interfaces – Responsibilities and Collaborations**

1072 This section defines each of the roles and interfaces shown in the diagram above.

1073 **9.1.1 RFID Tag (Role)**

1074 *Responsibilities:*

- 1075 • Holds an EPC code
- 1076 • May allow the EPC code to be changed post-manufacture.
- 1077 • May hold an immutable code that gives manufacture information, including the
1078 manufacturer identity, unique manufacture serial number, etc.
- 1079 • May have additional user data apart from the EPC code.
- 1080 • May have additional features such as lock, kill, access control, etc. These features
1081 may involve additional data stored on the tag such as a lock code, lock status, kill
1082 code, access password, etc.

1083 **9.1.2 EPC Tag Data Specification (Interface)**

1084 *Normative references:*

- 1085 • Ratified EPCglobal Standard: [TDS1.0]

1086 *Responsibilities:*

- 1087 • Defines the overall structure of the Electronic Product Code, including the
1088 mechanism for federating different coding schemes.
- 1089 • Defines specific EPCglobal coding schemes.
- 1090 • For each EPCglobal coding scheme, defines binary representations for use on RFID
1091 tags, text representations for use within information systems (in particular, at the ALE
1092 level and higher in the EPCglobal Architecture Framework), and rules for converting
1093 between one representation and another.

1094 **9.1.3 Tag Protocol (Interface)**

1095 As explained in the notes to the table in Section 2, there are several Tag Protocols that
1096 have been defined for use within the EPCglobal Network: one that is a ratified EPCglobal
1097 standard (the UHF Class 1 Gen 2 tag protocol), and three others that were published by
1098 the Auto-ID Center prior to the creation of EPCglobal. The notes to the table in Section 2
1099 give a full description of the status of each of these tag protocols. At the level of this
1100 document, the various tag protocols are essentially equivalent; they differ primarily in
1101 technical detail as to how commands and data are exchanged between reader and tag and
1102 what the specific command set is.

1103 *Normative references:*

- 1104 • EPCglobal Specifications (from Auto-ID Center): [UHFC0], [UHFC1G1], [HFC1]
1105 • Ratified EPCglobal Standard: [UHFC1G2]
1106 *Responsibilities:*
1107 • Communicates a command to a tag from an RFID Reader.
1108 • Communicates a response from a tag to the RFID Reader that issued the command.
1109 • Provides means for a reader to singulate individual tags when more than one is within
1110 range of the RFID Reader.
1111 • Provides means for readers and tags to minimize interference with each other.

1112 **9.1.4 RFID Reader (Role)**

1113 *Responsibilities:*

- 1114 • Reads the EPCs of RFID Tags within range of one or more antennas (via a Tag
1115 Protocol) and reports the EPCs to a host application (via the Reader Protocol).
1116 • When an RFID Tag allows the EPC code to be written post-manufacture, writes the
1117 EPC to a tag (via a Tag Protocol) as commanded by a host application (via the Reader
1118 Protocol).
1119 • When an RFID Tag provides additional user data apart from the EPC code, reads and
1120 writes user data (via a Tag Protocol) as directed by a host application (via the Reader
1121 Protocol).
1122 • When an RFID Tag provides additional features such as kill, lock, etc, operates those
1123 features (via a Tag Protocol) as directed by a host application (via the Reader
1124 Protocol).
1125 • May provide additional processing such as filtering of EPCs, aggregation of reads,
1126 and so forth. See also the Filtering & Collection Role, Section 9.1.8.

1127 **9.1.5 Reader Protocol (Interface)**

1128 The Reader Protocol provides the means for software to control all aspects of RFID
1129 Reader operation, including all capabilities implied by features of the Tag Protocols. In
1130 particular, the EPCglobal Reader Protocol is intended to provide complete access to all
1131 capabilities of the UHF Class 1 Gen 2 Tag Protocol including modulation formats, data
1132 rates sessions, and passwords, as well as reading, writing, locking, and killing tags. As of
1133 this writing, the latest working draft of the Reader Protocol specification does not fully
1134 realize this goal, but it is the intent of EPCglobal to address this issue in a timely fashion.

1135 *Normative references:*

- 1136 • EPCglobal Working Draft: [RP1.0]

1137 *Responsibilities¹:*

- 1138 • Provides means to command an RFID Reader to inventory tags (that is, to read the
1139 EPC codes carried on tags), read tags (that is, to read other data on the tags apart from
1140 the EPC code), write tags, manipulate tag user and tag-identification data, and access
1141 other features such as kill, lock, etc.
- 1142 • May provide means to access RFID Reader management functions including
1143 discovery, firmware/software configuration and updates, health monitoring,
1144 connectivity monitoring, statistics gathering, antenna connectivity, transmit power
1145 level, and managing reader power consumption.
- 1146 • May provide means to control RF aspects of RFID Reader operation including control
1147 of RF spectrum utilization, interference detection and measurement, modulation
1148 format, data rates, etc.
- 1149 • May provide means to control aspects of Tag Protocol operation, including protocol
1150 parameters and singulation parameters.
- 1151 • May provide access to processing features such as filtering of EPCs, aggregation of
1152 reads, and so forth. For features that require converting between different
1153 representations of EPCs, may use the Tag Data Translation Interface (Section 9.1.20)
1154 to obtain machine-readable rules for doing so.

1155 Some of the responsibilities enumerated above are not yet addressed in the latest working
1156 draft of the Reader Protocol specification, but will be addressed as the specification
1157 evolves to fully exploit features of the UHF Class 1 Gen 2 Tag Protocol. Responsibilities
1158 expected to be addressed include providing means to manage readers to prevent reader-
1159 to-reader collisions and facilitate “scouring” to find tags. This includes management of
1160 power levels, carrier frequencies, “sessions” (as that term is defined in the UHF Class 1
1161 Gen 2 Tag protocol), and protocol parameters. How these responsibilities are divided (or
1162 duplicated) between the Reader Protocol Interface and the Reader Management Interface
1163 is TBD.

1164 **9.1.6 Reader Management Interface (Interface)**

1165 *Normative references:*

- 1166 • EPCglobal Working Draft: [RM1.0]

1167 *Responsibilities:*

- 1168 • Provides means to query the configuration of an RFID Reader, such as its identity,
1169 number of antennas, and so forth.
- 1170 • Provides means to monitor the operational status of an RFID Reader, such as the
1171 number of tags read, status of communication channels, health monitoring, antenna
1172 connectivity, transmit power levels, and so forth.

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

- 1173 • Provides means to control configuration of an RFID Reader, such as
1174 enabling/disabling specific antennas or features, and so forth.
- 1175 • May provide means to access RFID Reader management functions including
1176 discovery, firmware/software configuration and updates, and managing reader power
1177 consumption.

1178 As the specification of this interface evolves to fully exploit features of the UHF Class 1
1179 Gen 2 Tag Protocol, it is expected that it will gain additional responsibilities including
1180 providing means to manage readers to prevent reader-to-reader collisions and facilitate
1181 “scouring” to find tags. This includes management of power levels, carrier frequencies,
1182 “sessions” (as that term is defined in the UHF Class 1 Gen 2 Tag protocol), and protocol
1183 parameters. How these responsibilities are divided (or duplicated) between the Reader
1184 Protocol Interface and the Reader Management Interface is TBD.

1185 Management of roles above the RFID Reader role is not currently addressed by
1186 EPCglobal standards, but may be considered in the future as warranted.

1187 **9.1.7 Reader Management (Role)**

1188 *Responsibilities:*

- 1189 • Monitors the operational status of one or more RFID Readers within a deployed
1190 infrastructure.
- 1191 • Manages the configuration of one or more RFID Readers.
- 1192 • Carries out other RFID Reader management functions including discovery,
1193 firmware/software configuration and updates, and managing reader power
1194 consumption.

1195 **9.1.8 Filtering & Collection (Role)**

1196 The Filtering & Collection role coordinates the activities of one or more RFID Readers
1197 that occupy the same physical space and which therefore have the possibility of radio-
1198 frequency interference.

1199 *Responsibilities:*

- 1200 • Receives raw tag reads from one or more RFID Readers.
- 1201 • Carries out processing to reduce the volume of EPC data, transforming raw tag reads
1202 into streams of events more suitable for application logic than raw tag reads.
1203 Examples of such processing include filtering (eliminating some EPCs according to
1204 their identities, such as eliminating all but EPCs for a specific object class),
1205 aggregating over time intervals (eliminating duplicate reads within that interval),
1206 grouping (e.g., summarizing EPCs within a specific object class), counting (reporting
1207 the number of EPCs rather than the EPC values themselves), and differential analysis
1208 (reporting which EPCs have been added or removed rather than all EPCs read).
- 1209 • Determines which processing operations as described above may be delegated to the
1210 RFID Reader, and which must be performed by the Filtering & Collection role itself.

- 1211 Implicit in this responsibility is that the Filtering & Collection role knows the
1212 capabilities of associated RFID Readers.
- 1213 • Decodes raw tag values into URI representations defined by the Tag Data
1214 Specification. May use the Tag Data Translation Interface (Section 9.1.20) to obtain
1215 machine-readable rules for doing so.
 - 1216 • Maps between “logical reader names” and physical resources such as reader devices
1217 and/or specific antennas.
 - 1218 • When the Filtering & Collection role is accessed by more than one client application,
1219 mediates between multiple client application requests for data when those requests
1220 involve the same set or overlapping subsets of RFID Readers.
 - 1221 • Sets and controls the strategy for finding tags employed by RFID Readers.
 - 1222 • May coordinate the operation of many readers and antennas within a local region in
1223 which RFID Readers may affect each other's operation; e.g., to minimize interference.
1224 For example, this role may control when specific readers are activated so that
1225 physically adjacent readers are not activated simultaneously. In another example, this
1226 role may make use of reader- or tag protocol-specific features, such as the “sessions”
1227 feature of the UHF Class 1 Gen 2 Tag Protocol, to minimize interference.
- 1228 The Filtering & Collection role has many responsibilities. The EPCglobal Architecture
1229 Framework currently provides standard interfaces to access some, but not all, of these
1230 responsibilities. Specifically:
- 1231 • The Filtering & Collection (ALE) Interface (Section 9.1.9) provides a standard
1232 interface that applies to a large collection of use cases in which RFID Tags are
1233 inventoried (i.e., where the EPCs carried on the tags are read).
 - 1234 • The ALE Interface does not currently support use cases in which tags are written or
1235 killed, in which the kill or lock passwords are maintained, or in which “user data” or
1236 TID memory on the tags is read or written. These are expected to be addressed by
1237 future specifications, possibly new interfaces adjacent to the ALE Interface, or
1238 possibly by extending the ALE interface itself.
 - 1239 • Management of the Filtering & Collection role is not yet addressed by any EPCglobal
1240 specification. This includes controlling aspects of coordination the activities of
1241 multiple readers to minimize interference, setting parameters that govern inventorying
1242 strategies, control over Tag Protocol-specific features, and so on.

1243 **9.1.9 Filtering & Collection (ALE) Interface (Interface)**

1244 The Filtering & Collection (ALE) Interface provides a standard interface to the Filtering
1245 & Collection role that applies to a large collection of use cases in which RFID Tags are
1246 inventoried (i.e., where the EPCs carried on the tags are read). It does not apply to other
1247 use cases, as explained in the previous section.

1248 *Normative references:*

- 1249 • EPCglobal Proposed Standard: [ALE1.0]

1250 *Responsibilities:*

- 1251 • Provides means for one or more client applications to request EPC data from one or
1252 more data sources.
- 1253 • Insulates client applications from knowing how many readers/antennas, and what
1254 makes and models of readers are deployed to constitute a single, logical data source.
- 1255 • Provides declarative means for client applications to specify what processing to
1256 perform on EPC data, including filtering, aggregation, grouping, counting, and
1257 differential analysis, as described in Section 9.1.8.
- 1258 • Provides a means for client applications to request data on demand (synchronous
1259 delivery) or as a standing request (asynchronous delivery).
- 1260 • Provides means for multiple client applications to share data from the same reader or
1261 readers, without prior coordination between the applications.
- 1262 • Provides a standardized representation for client requests for EPC data, and a
1263 standardized representation for reporting filtered, collected EPC data.

1264 **9.1.10 EPCIS Capturing Application (Role)**

1265 *Responsibilities:*

- 1266 • Recognizes the occurrence of EPC-related business events, and delivers these as
1267 EPCIS data.
- 1268 • May coordinate multiple sources of data in the course of recognizing an individual
1269 EPCIS event. Sources of data may include filtered, collected EPC data obtained
1270 through the Filtering & Collection Interface, other device-generated data such as
1271 barcode data, human input, and data gathered from other software systems.
- 1272 • May control the carrying out of actions in the physical environment, including writing
1273 RFID tags and controlling other devices. (When tag writing and related features are
1274 addressed in a future version of the Filtering & Collection Interface, as noted in
1275 Section 9.1.8, the EPCIS Capturing Application may use the Filtering & Collection
1276 Interface to carry out some of these responsibilities.)

1277 **9.1.11 EPCIS Capture Interface (Interface)**

1278 *Normative references:*

- 1279 • EPCglobal Working Draft: [EPCIS1.0]

1280 *Responsibilities:*

- 1281 • Provides a path for communicating EPCIS events generated by EPCIS Capturing
1282 Applications to other roles that require them, including EPCIS Repositories, internal
1283 EPCIS Accessing Applications, and Partner EPCIS Accessing Applications.

1284 **9.1.12 EPCIS Query Interface (Interface)**

1285 *Normative references:*

- 1286 • EPCglobal Working Draft: [EPCIS1.0]

1287 *Responsibilities:*

- 1288 • Provides means whereby an EPCIS Accessing Application can request EPCIS data
1289 from an EPCIS Repository or an EPCIS Capturing Application, and the means by
1290 which the result is returned.
- 1291 • Provides a means for mutual authentication of the two parties.
- 1292 • Reflects the result of authorization decisions taken by the providing party, which may
1293 include denying a request made by the requesting party, or limiting the scope of data
1294 that is delivered in response.

1295 **9.1.13 EPCIS Accessing Application (Role)**

1296 *Responsibilities:*

- 1297 • Carries out overall enterprise business processes, such as warehouse management,
1298 shipping and receiving, historical throughput analysis, and so forth, aided by EPC-
1299 related data.

1300 **9.1.14 EPCIS Repository (Role)**

1301 *Responsibilities:*

- 1302 • Records EPCIS-level events generated by one or more EPCIS Capturing
1303 Applications, and makes them available for later query by EPCIS Accessing
1304 Applications.

1305 **9.1.15 Object Name Service (ONS) Interface (Interface)**

1306 *Normative references:*

- 1307 • EPCglobal Working Draft: [ONS1.0]

1308 *Responsibilities:*

- 1309 • Provides a means for looking up a reference to an EPCIS service or other service that
1310 is provided by the EPC Manager of a specific EPC.

1311 **9.1.16 Local ONS (Role)**

1312 *Responsibilities:*

- 1313 • Fulfills ONS lookup requests for EPCs within the control of the enterprise that
1314 operates the Local ONS; that is, EPCs for which the enterprise is the EPC Manager.

1315 See also the discussion of ONS in Section 7.3.

1316 **9.1.17 ONS Root (Core Service)**

1317 *Responsibilities:*

- 1318 • Provides the initial point of contact for ONS lookups.
- 1319 • In most cases, delegates the remainder of the lookup operation to a Local ONS
1320 operated by the EPC Manager for the requested EPC.
- 1321 • May completely fulfill ONS requests in cases where there is no local ONS to which
1322 to delegate a lookup operation.
- 1323 • Provides a lookup service for 64-bit Manager Index values as required by the EPC
1324 Tag Data Specification.

1325 See also the discussion of ONS in Section 7.3.

1326 **9.1.18 Manager Number Assignment (Core Service)**

1327 *Responsibilities:*

- 1328 • Ensures global uniqueness of EPCs by maintaining uniqueness of EPC Manager
1329 Numbers assigned to EPCglobal Subscribers
- 1330 • Assigns new EPC Manager Numbers as required by EPCglobal Subscribers.

1331 **9.1.19 Tag Data Translation Schema (Core Service)**

1332 *Normative references:*

- 1333 • EPCglobal Working Draft: [TDT1.0]

1334 *Responsibilities:*

- 1335 • Provides a machine-readable file that defines how to translate between EPC
1336 encodings defined by the EPC Tag Data Specification (Section 9.1.2). EPCglobal
1337 provides this file for use by End-users, so that components of their infrastructure may
1338 automatically become aware of new EPC formats as they are defined.

1339 **9.1.20 Tag Data Translation Interface (Interface)**

1340 *Normative references:*

- 1341 • EPCglobal Working Draft: [TDT1.0]

1342 *Responsibilities:*

- 1343 • Encodes in machine-readable form all of the rules that define how to translate
1344 between EPC encodings defined by the EPC Tag Data Specification (Section 9.1.2).

1345 **9.1.21 EPCIS Discovery (Core Service – TBD)**

1346 Note that “EPCIS Discovery” is not yet a defined part of the EPCglobal Architecture
1347 Framework, but rather a placeholder for functionality that is envisioned for the
1348 EPCglobal Network but not yet architected. The responsibilities enumerated below are

1349 an envisioned set of responsibilities, but it is not yet known if this list is complete or
1350 accurate, nor how many distinct roles and interfaces will ultimately be required to carry
1351 out these responsibilities. Moreover, while “EPCIS Discovery” is labeled an EPCglobal
1352 Core Service, this is also just a placeholder, and the final set of responsibilities may be
1353 addressed by a combination of EPCglobal Core Services and services operated by
1354 EPCglobal Subscribers.

1355 *Responsibilities:*

- 1356 • Provides a means to locate all EPCIS services that may have information about a
1357 specific EPC.
- 1358 • May provide a cache for selected EPCIS data.
- 1359 • Enforces authorization policies with respect to access of the aforementioned data.

1360 **9.1.22 Subscriber Authentication (Core Service – TBD)**

1361 Note that “Subscriber Authentication” is not yet a defined part of the EPCglobal
1362 Architecture Framework, but rather a placeholder for functionality that is envisioned for
1363 the EPCglobal Network but not yet architected. The responsibilities enumerated below
1364 are an envisioned set of responsibilities, but it is not yet known if this list is complete or
1365 accurate, nor how many distinct roles and interfaces will ultimately be required to carry
1366 out these responsibilities.

1367 *Responsibilities:*

- 1368 • Authenticates the identity of an EPCglobal Subscriber.
- 1369 • Provides credentials that one EPCglobal Subscriber may use to authenticate itself to
1370 another EPCglobal Subscriber, without prior arrangement between the two
1371 Subscribers.
- 1372 • Authenticates participation in network services through validation of active
1373 EPCglobal Subscription.

1374 **9.1.23 Filtering & Collection Management Interface (Interface** 1375 **– TBD)**

1376 In Section 9.1.6 it is noted that management of roles above the RFID Reader role is not
1377 currently addressed by EPCglobal standards, but may be considered in the future as
1378 warranted. The Filtering & Collection Management Interface shown in the diagram at
1379 the beginning of this section is a placeholder for future work that may arise in this area.
1380 The responsibilities enumerated below are an envisioned set of responsibilities, but it is
1381 not yet known if this list is complete or accurate, nor how many distinct roles and
1382 interfaces will ultimately be required to carry out these responsibilities.

1383 *Responsibilities:*

- 1384 • Provides means to query the configuration of systems that carry out Filtering &
1385 Collection responsibilities.

1386 • Provides means to monitor the operational status of systems that carry out Filtering &
1387 Collection responsibilities.

1388 • Provides means to control configuration of systems that carry out Filtering &
1389 Collection responsibilities.

1390 **10 Summary of Unaddressed Issues**

1391 As noted in Section 1 and throughout the document, there are technical needs that are
1392 believed to exist based on the analysis of known use cases, where those needs are not yet
1393 fully addressed by the EPCglobal Architecture Framework. In these cases, the
1394 architectural approach has not yet been finalized, and therefore work on developing
1395 standards or designing additional Core Services has not yet begun, though architectural
1396 analysis is underway within the Architecture Review Committee. This section
1397 summarizes the known unaddressed issues, and will serve as a starting point for
1398 continued refinement of the EPCglobal Architecture Framework.

1399 The following list of issues is *not* intended to suggest the relative importance or priority
1400 of any issue.

1401 **10.1 EPCIS “Discovery”**

1402 The EPCIS Interface provides the means for one Subscriber to query another for EPC-
1403 related information. As discussed in Section 7.1, there are several ways a Subscriber
1404 might locate the relevant EPCIS Services in a given situation.

1405 The EPCglobal Architecture Framework does not currently provide a means to locate
1406 EPCIS Services in the most general situations arising from multi-party supply chains, in
1407 which several different organizations may have relevant data about an EPC but the
1408 identities of those organizations are not known in advance. Sections 7.1 and 9.1.20
1409 discuss some of the thinking that has gone on in this area, but the EPCglobal Architecture
1410 Framework does not yet address these requirements.

1411 **10.2 Subscriber Authentication**

1412 Section 7.1 also points out the need for subscribers to mutually authenticate each other
1413 when they are involved in EPCIS exchanges. It is desirable for this authentication to be
1414 as easy as possible for a subscriber to implement. In particular, it is undesirable if each
1415 subscriber has to make prior arrangements with every other subscriber that might be
1416 involved in a future EPCIS exchange; instead, it is better if each subscriber need only
1417 register once with a central authority and thereafter be able to mutually authenticate with
1418 any other subscriber.

1419 The EPCglobal Architecture Framework does not yet address this requirement, but some
1420 of the thinking is captured in Sections 7.1 and 9.1.22.

1421 **10.3 RFID Reader Coordination**

1422 The UHF Class 1 Gen 2 Tag Protocol provides a number of features designed to improve
1423 the performance of RFID Readers, especially when many readers are deployed in close
1424 physical proximity. These features serve to minimize reader-to-reader collisions, and
1425 facilitate “scouring” algorithms to find tags. Among the features provided for these ends
1426 are control over power management, carrier frequencies, “sessions” that help insure one
1427 reader does not interfere with another reader’s conversation with the same tag, and other
1428 protocol parameters.

1429 The Reader Protocol and Reader Management specifications currently under
1430 development do not specifically address these new features, nor does the EPCglobal
1431 Architecture Framework specify how these features would be exploited at an
1432 architectural level (e.g., by giving some responsibility to the Filtering & Collection role,
1433 or possibly to higher-level roles or new roles).

1434 **10.4 RFID Tag-level Security and Privacy**

1435 Sections 3.6 and 3.7 discuss EPCglobal Network goals of security and privacy. The UHF
1436 Class 1 Generation 2 Tag Protocol supports specific RFID Tag features designed to
1437 further security and privacy goals. These features include a “kill” feature with an
1438 associated kill password, a “lock” feature, and an access control password.

1439 The EPCglobal Architecture Framework does not currently discuss how these features
1440 affect the architecture above the level of the Reader Protocol, nor is there any
1441 architectural discussion of how the goals of security and privacy are address through
1442 these or other features. In particular, it is not clear how the passwords required to operate
1443 the “kill” and “lock” features are to be distributed through the network to reach the places
1444 where they are required.

1445 It should be noted that the “kill” and “lock” features are only components of a
1446 comprehensive privacy policy, not a complete solution to privacy issues facing the
1447 EPCglobal Network. The EPCglobal Public Policy Steering Committee (PPSC) is
1448 responsible for creating and maintaining the EPCglobal Privacy Policy; readers should
1449 refer to PPSC documents for more information.

1450 **10.5 “User Data” in RFID Tags**

1451 The EPCglobal Architecture Framework discusses the use of RFID Tags that are used to
1452 hold an EPC code associated with an object to which the tag is affixed. The UHF Class 1
1453 Generation 2 Tag Protocol supports RFID Tags that contain additional “user data”
1454 besides the EPC code.

1455 The EPCglobal Architecture Framework does not currently discuss how RFID Tag “user
1456 data” is to be exploited at any level of the architecture.

1457 **10.6 Tag Writing, Killing, Locking above the Reader Protocol**
1458 **Layer**

1459 Reading (apart from reading EPCs), writing, locking, and killing of RFID Tags, as well
1460 as maintenance of the kill and access passwords, are currently addressed by the Tag
1461 Protocol and Reader Protocol, but not yet at higher layers of the architecture framework.
1462 See Section 9.1.8 for further discussion.

1463 **10.7 Master Data for RFID Tag Manufacture Data**

1464 The UHF Class 1 Generation 2 Tag Protocol provides for a read-only “tag ID” (TID)
1465 field that is written at RFID Tag manufacture time. The TID is intended to provide
1466 information about the manufacture of the tag, including the identity of the tag
1467 manufacturer and other information. This information would be associated with the TID
1468 in an external database, maintained by EPCglobal or some other authority.

1469 The EPCglobal Architecture Framework does not currently provide a specification for the
1470 TID or associated information. Existing architecture components (e.g., ONS) might be
1471 useful for this purpose.

1472

1473 **11 References**

1474 [ALE1.0] K. Traub, S. Bent, T. Osinski, S. N. Peretz, S. Rehling, S. Rosenthal, B.
1475 Tracey, “The Application Level Events (ALE) Specification, Version 1.0,” EPCglobal
1476 Proposed Standard, February 2005.

1477 [EANUCC] GS1, “General EAN.UCC Specifications v6.0,” 2005,
1478 <http://www.gs1uk.org/EANUCC/>.

1479 [EPCIS1.0] EPCglobal, “EPC Information Services (EPCIS) Version 1.0 Specification,”
1480 EPCglobal Working Draft, June 2005.

1481 [HFC1] MIT Auto-ID Center, “13.56 MHz ISM Band Class 1 Radio Frequency
1482 Identification Tag Interface Specification: Candidate Recommendation, Version 1.0.0,”
1483 February 2003, [http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-
1484 Class1.pdf](http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-Class1.pdf).

1485 [ISODir2] ISO, “Rules for the structure and drafting of International Standards
1486 (ISO/IEC Directives, Part 2, 2001, 4th edition),” July 2002.

1487 [ONS1.0] M. Mealling, “EPCglobal Object Name Service (ONS) 1.0,” EPCglobal
1488 Working Draft, December 2004.

1489 [RFC1034] P. V. Mockapetris, “Domain names – concepts and facilities.” RFC1034,
1490 November 1987, <http://www.ietf.org/rfc/rfc1034>.

1491 [RFC1035] P. V. Mockapetris, “Domain names – implementation and specification.”
1492 RFC1035, November 1987, <http://www.ietf.org/rfc/rfc1035>.

1493 [RM1.0] F.A. Ahmed, C. Birger, G. Gangl, L-E. Helander, M. Jackson, P. Krishna, R.
1494 Labiaga, C. Sayers, S. Shafer, M. Ulrich, “Reader Management 1.0,” EPCglobal Working
1495 Draft, May 2005.

1496 [RP1.0] EPCglobal, “Reader Protocol 1.0,” EPCglobal Working Draft, March 2005.

1497 [SDP1.1] EPCglobal, “Standards Development Process Specification v1.1,” EPCglobal
1498 publication, March 2005,
1499 [http://www.epcglobalinc.org/standards_technology/SDP_March_%2014_%202005_v1.1.](http://www.epcglobalinc.org/standards_technology/SDP_March_%2014_%202005_v1.1.pdf)
1500 [pdf](http://www.epcglobalinc.org/standards_technology/SDP_March_%2014_%202005_v1.1.pdf).

1501 [SLRRP] P. Krishna, D. Husak, “Simple Lightweight RFID Reader Protocol,” IETF
1502 Internet Draft, June 2005, <http://www.ietf.org/internet-drafts/draft-krishna-slrrp-02.txt>.

1503 [SOAP1.2] M. Gudgin, M. Hadley, N. Mendelsohn, J-J. Moreau, H. F. Nielsen, “SOAP
1504 Version 1.2,” W3C Recommendation, June 2003, <http://www.w3.org/TR/soap12>.

1505 [TDS1.1] EPCglobal, “EPC Tag Data Standards Version 1.1 Rev 1.27,” EPCglobal
1506 Standard Specification, May 2005.

1507 [TDT1.0] M. Harrison, V. Sundhar, T. Osinski, “EPCglobal Tag Data Translation (TDT)
1508 1.0,” EPCglobal Working Draft, June 2005.

1509 [UHFC0] MIT Auto-ID Center, “Draft protocol specification for a 900 MHz Class 0
1510 Radio Frequency Identification Tag,” EPCglobal Specification, February 2003,
1511 http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class0.pdf.

1512 [UHFC1G1] MIT Auto-ID Center, “860MHz–930MHz Class I Radio Frequency
1513 Identification Tag Radio Frequency & Logical Communication Interface Specification
1514 Candidate Recommendation, Version 1.0.1,” EPCglobal Specification, November 2002,
1515 http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf.

1516 [UHFC1G2] EPCglobal, “EPC™ Radio-Frequency Identity Protocols Class-1
1517 Generation-2 UHF RFID Protocol for Communications at 860 MHz – 960 MHz Version
1518 1.0.9,” EPCglobal Standard, January 2005,
1519 [http://www.epcglobalinc.org/standards_technology/EPCglobalClass-1Generation-](http://www.epcglobalinc.org/standards_technology/EPCglobalClass-1Generation-2UHFRFIDProtocolV109.pdf)
1520 [2UHFRFIDProtocolV109.pdf](http://www.epcglobalinc.org/standards_technology/EPCglobalClass-1Generation-2UHFRFIDProtocolV109.pdf).

1521 **12 Glossary**

1522 This section provides a summary of terms used within this document. For fuller
1523 definitions of these terms, please consult the relevant sections of the document. See also
1524 the whole of Section 9, which defines all roles and interfaces within the EPCglobal
1525 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.

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.

1526 **13 Questions Raised During Review**

1527 A draft of this document was reviewed by the EPCglobal community between April 15,
1528 2005 and May 18, 2005. Many of the comments received led to clarifications or other
1529 changes in the body of this document. This section enumerates other comments that are
1530 not specifically addressed elsewhere within the document.

1531 **13.1 Questions about the Electronic Product Code and its use**

1532 *In Section 4.1, EPC is described as a federation of several naming structures. Isn't it*
1533 *more accurate to say that EPC functions as a multi-industry naming structure by*
1534 *providing one naming structure standard?*

1535 Saying that EPC provides “one naming structure standard” might easily be misconstrued.
1536 It is important to recognize that EPC does not impose a *single* numbering structure on all
1537 industries, but rather federates existing numbering structures including the GS1 family of
1538 coding schemes, the US Department of Defense’s CAGE/DoDAAC scheme, etc. This is
1539 in contrast to the original conception of the EPC as proposed by the MIT Auto-ID Center,
1540 which *was* a single numbering structure. The evolution from the single numbering
1541 structure model to the federated model was a key factor in the acceptance of EPC for
1542 most of the participating industries.

1543 *In Section 4.1 it is stated that EPCs are globally unique. Yet it also says that in the case*
1544 *of an SSCC, the EPC might not be unique. Will the allocation rules for SSCC or the*
1545 *definition of EPC have to change to resolve this apparent contradiction?*

1546 Because of the End-user community's desire to have an EPC representation that includes
1547 existing coding schemes such as SSCC, some compromise was needed. As stated in
1548 Section 4.1, each coding scheme that is federated within the EPC system may specify its
1549 own rules as regards uniqueness over time. Global uniqueness over time is a desirable
1550 goal for any coding scheme, but when existing schemes are federated into the EPC
1551 structure they may bring along their own restrictions. In the case of SSCC, current
1552 practice together with the size limitations of the SSCC imply that they are recycled
1553 periodically. Since it is not practical to change the SSCC, the EPCglobal Architecture
1554 Framework must recognize that the SSCC variety of EPC is not globally unique over all
1555 time. This does not, however, compromise the uniqueness guarantees of other kinds of
1556 EPCs, such as SGTINs. Since an EPC always carries an indication of coding scheme,
1557 information processing systems can be aware of the restrictions that apply to any
1558 particular EPC.

1559 *Some End-users want to use the same EPC for identical objects, or two EPCs for a single*
1560 *object. Can the EPCglobal Architecture Framework handle such requests?*

1561 Because the EPC is intended to *uniquely* identify an object (see Section 4.1), it is not
1562 permitted to use the same EPC for identical objects. As for assigning two EPCs to a
1563 single object, certain cases of this are discussed in Section 4.1.

1564 In other instances, an End-user may wish to affix two or more RFID tags to the same
1565 physical object, all bearing the same EPC code, in order to improve the chances of
1566 reading the EPC. In this case there is still one EPC for one object, and so this is
1567 permitted by the EPCglobal Architecture Framework. Whether it is technically feasible
1568 depends on the Tag Protocol being used (e.g., this will work using UHF Class 1 Gen 2
1569 tags, but not using the older Class 1 tags).

1570 *The UHF Class 1 Gen 2 Tag Protocol allows an optional memory bank for user data.*
1571 *Doesn't this conflict with the statement in Section 4.1 the EPC is a lightweight identifier*
1572 *with no additional information? How will the optional user data segment be structured?*

1573 The statement in Section 4.1 is that the *EPC* is lightweight. The user data on a UHF
1574 Class 1 Gen 2 tag is not part of the EPC, but rather accompanies the EPC on the tag. This
1575 is an example of "information ... associated with the EPC through other means" as
1576 mentioned in Section 4.1; in this case, the means of association is that the EPC and the
1577 user data are carried on the same RFID tag.

1578 The structure and contents of the user data memory of a UHF Class 1 Gen 2 tag will be
1579 defined in a future version of the EPC Tag Data Standard, or by a new specification
1580 created by that Working Group.

1581 *Section 5.2 enumerates some responsibilities of the EPC Manager with respect to the*
1582 *EPC codes that it manages. Doesn't the EPC Manager have responsibilities, including*
1583 *the providing (via EPCIS) of EPC master data like class level data or 'as built'*
1584 *specifications, as well as establishing initial kill and access passwords?*

1585 While it is true that these are typically things that an EPC Manager does, the details are
1586 specific to a particular industry. Section 5.2 only enumerates the responsibilities that are
1587 necessary in all cases to insure the integrity of the EPCglobal Network.

1588 **13.2 Questions about EPCIS and ONS**

1589 *The document does not contain any reference to the Physical Markup Language (PML).*
1590 *Is PML not considered part of the architectural framework?*

1591 PML was a data exchange standard proposed by the Auto-ID Center prior to the creation
1592 of EPCglobal. Within the EPCglobal Architecture Framework, PML has been
1593 superseded by EPCIS, in particular the EPCIS Data Specification (see Sections
1594 5.6 and 9.1.12).

1595 *How is temporary data, that is not stored on an RFID Tag, maintained within the*
1596 *EPCglobal Network. For example, suppose that a reader in a distribution center (DC) is*
1597 *able to determine that a certain RFID Tag, when detected, is either leaving or entering*
1598 *the DC. How can this information be accessed and used by other EPCglobal*
1599 *Subscribers?*

1600 This is exactly the role of EPCIS. See Sections 5.6, 9.1.12, 9.1.10, 9.1.11, 9.1.13,
1601 and 9.1.14.

1602 *ONS only contains references to EPCIS services of the enterprise that issued a given*
1603 *EPC. When an object having an EPC “travels” in the supply chain, multiple enterprises*
1604 *read the EPC and keep information about it. This information will be stored in multiple*
1605 *EPCIS services which must be referenced by the ONS.*

1606 This is a common misunderstanding about ONS. It is true that ONS only refers to the
1607 EPCIS services of the EPC Manager (that is, the enterprise that issued a given EPC), and
1608 so ONS does *not* provide a means to find EPCIS services of other enterprises that may
1609 have information about a given EPC. This latter responsibility is the subject of “EPCIS
1610 Discovery” services, which are still under development within EPCglobal. See Sections
1611 7.1 and 9.1.20.

1612 *Section 7.1 describes “following the chain” as a way for discovering what EPCIS*
1613 *services contain data about a given EPC. Are the backwards and forwards links*
1614 *required to follow the chain being implemented as part of EPCIS?*

1615 The EPCIS Working Group anticipates that EPCIS event data may contain “business
1616 transaction identifiers” (e.g., a purchase order identifier) that could be used to follow the
1617 chain.

1618 *Is “following the chain” as described in Section 7.1 the standard technique in the*
1619 *EPCglobal Network for locating information other than that in the EPC Manager’s*
1620 *EPCIS? “Following the chain” seems fragile.*

1621 The ARC recognizes that “following the chain” is fragile, which is why effort is
1622 underway to define the more robust “EPCIS Discovery” function (see Sections
1623 7.1 and 9.1.20). “Following the chain” is viable in many situations, however, and so it is
1624 included in this document.

1625 *Who is developing the EPCIS Query Interface? When is it expected to be available?*

1626 The EPCIS Query Interface is under development within the EPCIS Phase 2 Working
1627 Group within the Software Action Group. Please contact the Working Group co-chairs
1628 for information about schedule.

1629 **13.3 Questions about Infrastructure Standards**

1630 *How are operational parameters handled within the EPCglobal Architecture*
1631 *Framework? For example, the UHF Class 1 Gen 2 Tag Protocol requires that each*
1632 *reader select a session in which to singulate tags. Sessions must be managed at the level*
1633 *of a physical site. Which EPCglobal specification(s) governs the choice of session, and*
1634 *how does this information flow down to each and every reader?*

1635 The EPCglobal Architecture Framework provides two interfaces for control of reader
1636 parameters such as these: the Reader Protocol and the Reader Management Interface. At
1637 the time of this writing, the respective Working Groups have not yet addressed normative
1638 specifications for choice of session and how this is to be coordinated across multiple
1639 readers within a site. This is expected to be among the topics addressed as these Working
1640 Groups revise their specifications to encompass more Gen 2 features.

1641 **13.4 Questions about Security and Privacy**

1642 *Where in the diagram of Section 9 are components that provide functions such as*
1643 *authentication, access control, validation, privacy protection of individuals and*
1644 *corporations, and so on?*

1645 In general, security functions such as these are not distinct roles within the EPCglobal
1646 Architecture Framework. Rather, each role includes responsibilities related to security
1647 and privacy. For example, an RFID Tag (Section 9.1.1) has responsibilities related to kill
1648 and lock codes, and the EPCIS Query Interface (Section 9.1.12) has responsibilities
1649 related to authentication and authorization in cross-enterprise data exchange. The
1650 individual EPCglobal Standards define these responsibilities in more detail. In other
1651 cases, it is acknowledged that there are security and privacy issues not yet fully addressed
1652 in the EPCglobal Architecture Framework, and these are noted within this document (see,
1653 for example, Sections 9.1.22, 10.2, and 10.4).

1654 *Section 10.4 states that the network aspects of "kill" and "lock" are open. Retailers might*
1655 *face a privacy issue here, if tags of tagged cases cannot be killed at the time of checkout.*

1656 The ARC agrees with this concern. Section 10.4 merely reflects the reality that this is a
1657 topic of current work, and so it is not possible to say anything more definitive at this
1658 time.

1659 **14 Acknowledgements**

1660 The authors would like to thank the following persons and organizations for their
1661 comments on earlier versions of this document:

1662 John Anderla (Kimberly Clark), Chet Birger (ConnecTerra), Judy Bueg (Eastman
1663 Kodak), Curt Carrender (Alien Technologies), Chris Diorio (Impinj), Andreas F ubler (GS1
1664 Europe), Lim Joo Ghee (Institute for Infocomm Research), Graham Gillen (VeriSign),
1665 Sue Hutchinson (EPCglobal), Osamu Inoue (EPCglobal Japan), P. Krishna (Reva
1666 Systems), Shinichi Nakahara (NTT), Mike O'Shea (Kimberly Clark), Andrew Osborne
1667 (GS1 Technical Steering Team), Hidenori Ota (Fujitsu), Tom Pounds (Alien
1668 Technologies), Steve Rehling (Procter & Gamble), Steve Smith (Alien Technologies),

1669 Suzanne Stuart-Smith (GS1 UK), Hiroyasu Sugano (Fujitsu), Hiroki Tagato (NEC), Neil
1670 Tan (UPS), Joseph Tobolski (Accenture), Nicholas Tsougas (US Defense Logistics
1671 Agency), Mitsuo Tsukada (NTT), Shashi Shekhar Vempati (Infosys), Ulrich Wertz (MGI
1672 METRO Group), Gerd Wolfram (MGI METRO Group), and Ochi Wu (CODEplus).