

BS ISO/IEC 19987:2015



BSI Standards Publication

Information technology — EPC Information services — Specification

bsi.

...making excellence a habit.
AVA **Documents**
Document Sharing Hub

National foreword

This British Standard is the UK implementation of ISO/IEC 19987:2015.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

ISBN 978 0 580 87128 3

ICS 35.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 September 2015.

Amendments issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

BS ISO/IEC 19987:2015
**ISO/IEC
19987**

First edition
2015-09-15

Information technology — EPC Information services — Specification

*Technologies de l'information — Services d'information sur les codes
de produit électronique — Spécification*



Reference number
ISO/IEC 19987:2015(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: [Foreword — Supplementary information](#).

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*.



EPC Information Services (EPCIS) Version 1.1 Specification

GS1 Standard

Version 1.1, May 2014



Documents

Document Sharing Hub



© 2007–2014 GS1 AISBL

All rights reserved.

GS1 Global Office

Avenue Louise 326, bte 10

B-1050 Brussels, Belgium

Disclaimer

GS1 AISBL (GS1) is providing this document as a free service to interested industries.

This document was developed through a consensus process of interested parties in developing the Standard. Although efforts have been made to assure that the document is correct, reliable, and technically accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS DOCUMENT IS CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGY DICTATE, OR WILL BE SUITABLE FOR ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of this document is with the understanding that GS1 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF NON-INFRINGEMENT OF PATENTS OR COPYRIGHTS, MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE, THAT THE INFORMATION IS ERROR FREE, NOR SHALL GS1 BE LIABLE FOR DAMAGES OF ANY KIND, INCLUDING DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES, ARISING OUT OF USE OR THE INABILITY TO USE INFORMATION CONTAINED HEREIN OR FROM ERRORS CONTAINED HEREIN.

Abstract

This document is a GS1 Standard that defines Version 1.1 of EPC Information Services (EPCIS). The goal of EPCIS is to enable disparate applications to create and share visibility event data, both within and across enterprises. Ultimately, this sharing is aimed at enabling users to gain a shared view of physical or digital objects within a relevant business context.

Status of this document

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

This version of the GS1 EPCIS 1.1 Standard is the ratified version and has completed all GSMP steps.

Comments on this document should be sent to gsmf@gs1.org.

Differences from EPCIS 1.0.1

EPCIS 1.1 is fully backward compatible with EPCIS 1.0.1.

EPCIS 1.1 includes these new or enhanced features:

- Support for class-level identification is added to `ObjectEvent`, `AggregationEvent`, and `TransformationEvent` through the addition of quantity lists.
- A new event type, `TransformationEvent`, provides for the description of events in which inputs are consumed and outputs are produced.
- The “why” dimension of all event types are enhanced so that information about the sources and destinations of business transfers may be included.
- The “why” dimension of certain event types are enhanced so that item/lot master data may be included.
- The `SimpleEventQuery` is enhanced to encompass the above changes to event types.
- The introductory material is revised to align with the GS1 System Architecture.
- The XML extension mechanism is explained more fully.
- The `QuantityEvent` is deprecated, as its functionality is fully subsumed by `ObjectEvent` with the addition of quantity lists.



Table of Contents

32	1	Introduction.....	6
33	2	Relationship to the GS1 System Architecture	7
34	2.1	Overview of GS1 Standards	7
35	2.2	EPCIS in Relation to the “Capture” and “Share” Layers	8
36	2.3	EPCIS in Relation to Trading Partners	10
37	2.4	EPCIS in Relation to other GS1 System Architecture Components.....	12
38	3	EPCIS Specification Principles.....	15
39	4	Terminology and Typographical Conventions.....	16
40	5	EPCIS Specification Framework	16
41	5.1	Layers	16
42	5.2	Extensibility	18
43	5.3	Modularity.....	18
44	6	Abstract Data Model Layer	19
45	6.1	Event Data and Master Data.....	19
46	6.2	Vocabulary Kinds	22
47	6.3	Extension Mechanisms	23
48	6.4	Identifier Representation	25
49	6.5	Hierarchical Vocabularies	26
50	7	Data Definition Layer	26
51	7.1	General Rules for Specifying Data Definition Layer Modules.....	27
52	7.1.1	Content.....	27
53	7.1.2	Notation	28
54	7.1.3	Semantics	29
55	7.2	Core Event Types Module – Overview	29
56	7.3	Core Event Types Module – Building Blocks.....	33
57	7.3.1	Primitive Types.....	33
58	7.3.2	Action Type	34
59	7.3.3	The “What” Dimension	34
60	7.3.4	The “Where” Dimension – Read Point and Business Location	40
61	7.3.5	The “Why” Dimension	44
62	7.3.6	Instance/Lot Master Data (ILMD).....	47
63	7.4	Core Event Types Module – Events	48
64	7.4.1	EPCISEvent	48
65	7.4.2	ObjectEvent (subclass of EPCISEvent).....	50
66	7.4.3	AggregationEvent (subclass of EPCISEvent).....	54
67	7.4.4	QuantityEvent (subclass of EPCISEvent) – DEPRECATED	60

68	7.4.5	TransactionEvent (subclass of EPCISEvent)	61
69	7.4.6	TransformationEvent (subclass of EPCISEvent)	67
70	8	Service Layer	71
71	8.1	Core Capture Operations Module	73
72	8.1.1	Authentication and Authorization	73
73	8.1.2	Capture Service	73
74	8.2	Core Query Operations Module	75
75	8.2.1	Authentication	75
76	8.2.2	Authorization	75
77	8.2.3	Queries for Large Amounts of Data	76
78	8.2.4	Overly Complex Queries	76
79	8.2.5	Query Framework (EPCIS Query Control Interface)	77
80	8.2.6	Error Conditions	87
81	8.2.7	Predefined Queries for EPCIS	90
82	8.2.8	Query Callback Interface	109
83	9	XML Bindings for Data Definition Modules	109
84	9.1	Extensibility Mechanism	109
85	9.2	Standard Business Document Header	112
86	9.3	EPCglobal Base Schema	114
87	9.4	Additional Information in Location Fields	114
88	9.5	Schema for Core Event Types	115
89	9.6	Core Event Types – Examples (non-normative)	123
90	9.6.1	Example 1 – Object Events with Instance-Level Identification	123
91	9.6.2	Example 2 – Object Event with Class-Level Identification	124
92	9.6.3	Example 3 – Aggregation Event with Mixed Identification	125
93	9.6.4	Example 4 – Transformation Event	126
94	9.7	Schema for Master Data	127
95	9.8	Master Data – Example (non-normative)	130
96	10	Bindings for Core Capture Operations Module	131
97	10.1	Message Queue Binding	131
98	10.2	HTTP Binding	132
99	11	Bindings for Core Query Operations Module	133
100	11.1	XML Schema for Core Query Operations Module	133
101	11.2	SOAP/HTTP Binding for the Query Control Interface	141
102	11.3	AS2 Binding for the Query Control Interface	148
103	11.4	Bindings for Query Callback Interface	153
104	11.4.1	General Considerations for all XML-based Bindings	153
105	11.4.2	HTTP Binding of the Query Callback Interface	154
106	11.4.3	HTTPS Binding of the Query Callback Interface	154
107	11.4.4	AS2 Binding of the Query Callback Interface	155
108	12	References	156
109	1		

Introduction

This document is a GS1 Standard that defines Version 1.1 of EPC Information Services (EPCIS). The goal of EPCIS is to enable disparate applications to create and share visibility event data, both within and across enterprises. Ultimately, this sharing is aimed at enabling users to gain a shared view of physical or digital objects within a relevant business context.

“Objects” in the context of EPCIS typically refers to physical objects that are identified either at a class or instance level and which are handled in physical handling steps of an overall business process involving one or more organizations. Examples of such physical objects include trade items (products), logistic units, returnable assets, fixed assets, physical documents, etc. “Objects” may also refer to digital objects, also identified at either a class or instance level, which participate in comparable business process steps. Examples of such digital objects include digital trade items (music downloads, electronic books, etc.), digital documents (electronic coupons, etc), and so forth. Throughout this document the word “object” is used to denote a physical or digital object, identified at a class or instance level, that is the subject of a business process step. EPCIS data consist of “visibility events,” each of which is the record of the completion of a specific business process step acting upon one or more objects.

The EPCIS standard was originally conceived as part of a broader effort to enhance collaboration between trading partners by sharing of detailed information about physical or digital objects. The name EPCIS reflects the origins of this effort in the development of the Electronic Product Code (EPC). It should be noted, however, that EPCIS does not require the use of Electronic Product Codes, nor of Radio-Frequency Identification (RFID) data carriers, and as of EPCIS 1.1 does not even require instance-level identification (for which the Electronic Product Code was originally designed). The EPCIS standard applies to all situations in which visibility event data is to be captured and shared, and the presence of “EPC” within the name is of historical significance only.

EPCIS provides open, standardised interfaces that allow for seamless integration of well-defined services in inter-company environments as well as within companies. Standard interfaces are defined in the EPCIS standard to enable visibility event data to be captured and queried using a defined set of service operations and associated data standards, all combined with appropriate security mechanisms that satisfy the needs of user companies. In many or most cases, this will involve the use of one or more persistent databases of visibility event data, though elements of the Services approach could be used for direct application-to-application sharing without persistent databases.

With or without persistent databases, the EPCIS specification specifies only a standard data sharing interface between applications that capture visibility event data and those that need access to it. *It does not specify how the service operations or databases themselves should be implemented.* This includes not defining how the EPCIS services should acquire and/or compute the data they need, except to the extent the data is captured using the standard EPCIS capture operations. The interfaces are needed for interoperability, while the implementations allow for competition among those providing the technology and implementing the standard.

EPCIS is intended to be used in conjunction with the GS1 Core Business Vocabulary (CBV) standard [CBV1.1]. The CBV standard provides definitions of data values that may be used to populate the data structures defined in the EPCIS standard. The use of the standardized vocabulary provided by the CBV standard is critical to interoperability and critical to provide for

154 querying of data by reducing the variation in how different businesses express common intent.
155 Therefore, applications should use the CBV standard to the greatest extent possible in
156 constructing EPCIS data.

157 2 Relationship to the GS1 System Architecture

158 This section is largely quoted from [EPCAF] and [GS1Arch], and shows the relationship of
159 EPCIS to other GS1 Standards.

160 2.1 Overview of GS1 Standards

161 GS1 Standards support the information needs of end users interacting with each other in supply
162 chains, specifically the information required to support the business processes through which
163 supply chain participants interact. The subjects of such information are the real-world entities
164 that are part of those business processes. Real-world entities include things traded between
165 companies, such as products, parts, raw materials, packaging, and so on. Other real-world
166 entities of relevance to trading partners include the equipment and material needed to carry out
167 the business processes surrounding trade such as containers, transport, machinery; entities
168 corresponding to physical locations in which the business processes are carried out; legal entities
169 such as companies, divisions; service relationships; business transactions and documents; and
170 others. Real-world entities may exist in the tangible world, or may be digital or conceptual.
171 Examples of physical objects include a consumer electronics product, a transport container, and a
172 manufacturing site (location entity). Examples of digital objects include an electronic music
173 download, an eBook, and an electronic coupon. Examples of conceptual entities include a trade
174 item class, a product category, and a legal entity.

175 GS1 Standards may be divided into the following groups according to their role in supporting
176 information needs related to real-world entities in supply chain business processes:

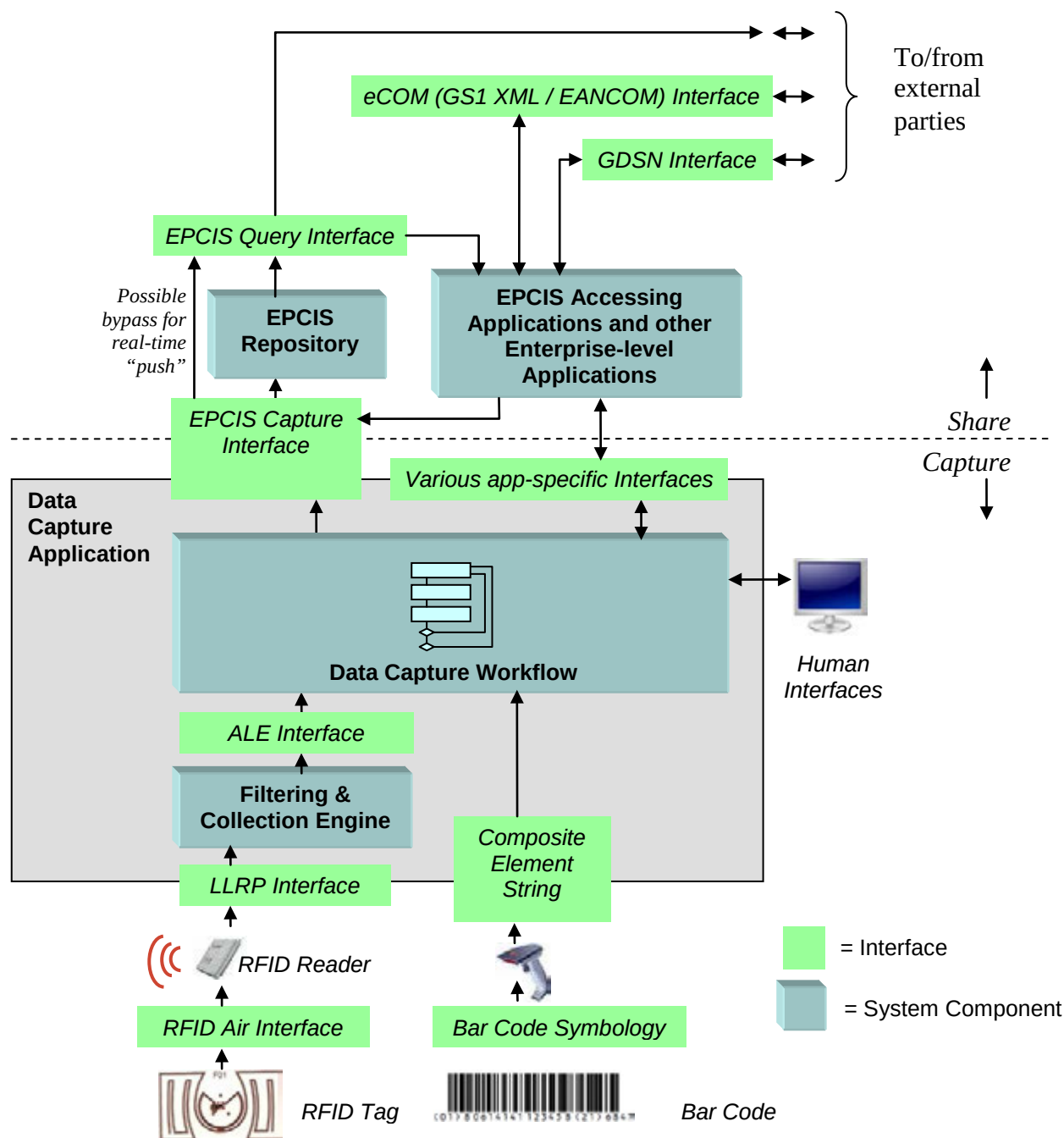
- 177 • Standards which provide the means to **Identify** real-world entities so that they may be the
178 subject of electronic information that is stored and/or communicated by end users. GS1
179 identification standards include standards that define unique identification codes (called GS1
180 Identification Keys).
- 181 • Standards which provide the means to automatically **Capture** data that is carried directly on
182 physical objects, bridging the world of physical things and the world of electronic
183 information. GS1 data capture standards include definitions of bar code and radio-frequency
184 identification (RFID) data carriers which allow identifiers to be affixed directly to a physical
185 object, and standards that specify consistent interfaces to readers, printers, and other
186 hardware and software components that connect the data carriers to business applications.
- 187 • Standards which provide the means to **Share** information, both between trading partners and
188 internally, providing the foundation for electronic business transactions, electronic visibility
189 of the physical or digital world, and other information applications. GS1 standards for
190 information sharing include this EPCIS Standard which is a standard for visibility event data.
191 Other standards in the “Share” group are standards for master data and for business
192 transaction data, as well as discovery standards that help locate where relevant data resides
193 across a supply chain and trust standards that help establish the conditions for sharing data
194 with adequate security.



195 The EPCIS Standard fits into the “Share” group, providing the data standard for visibility event
196 data and the interface standards for capturing such information from data capture infrastructure
197 (which employs standards from the “Capture” group) and for sharing such information with
198 business applications and with trading partners.

199 **2.2 EPCIS in Relation to the “Capture” and “Share” Layers**

200 The following diagram shows the relationship between EPCIS and other GS1 Standards in the
201 “Capture” and “Share” groups. (The “Identify” group of standards pervades the data at all levels
202 of this architecture, and so is not explicitly shown.)



203
204 As depicted in the diagram above, the EPCIS Capture Interface exists as a bridge between the
205 "Capture" and "Share" standards. The EPCIS Query Interface provides visibility event data both
206 to internal applications and for sharing with trading partners.

207 At the centre of a data capture application is the data capture workflow that supervises the
208 business process step within which data capture takes place. This is typically custom logic that is
209 specific to the application. Beneath the data capture workflow in the diagram is the data path
210 between the workflow and GS1 data carriers: bar codes and RFID. The green bars in the diagram
211 denote GS1 Standards that may be used as interfaces to the data carriers. At the top of the

diagram are the interfaces between the data capture workflow and larger-scale enterprise applications. Many of these interfaces are application- or enterprise-specific, though using GS1 data as building blocks; however, the EPCIS interface is a GS1 Standard. Note that the interfaces at the top of the diagram, including EPCIS, are independent of the data carrier used at the bottom of the diagram.

The purpose of the interfaces and the reason for a multi-layer data capture architecture is to provide isolation between different levels of abstraction. Viewed from the perspective of an enterprise application (i.e., from the uppermost blue box in the figure), the entire data capture application shields the enterprise application from the details of exactly how data capture takes place. Through the application-level interfaces (uppermost green bars), an enterprise application interacts with the data capture workflow through data that is data carrier independent and in which all of the interaction between data capture components has been consolidated into that data. At a lower level, the data capture workflow is cognizant of whether it is interacting with bar code scanners, RFID interrogators, human input, etc, but the transfer interfaces (green bars in the middle) shield the data capture workflow from low-level hardware details of exactly how the data carriers work. The lowest level interfaces (green bars on the bottom) embody those internal data carrier details.

EPCIS and the “Share” layer in general differ from elements in the Capture layer in three key respects:

1. EPCIS deals explicitly with historical data (in addition to current data). The Capture layer, in contrast, is oriented exclusively towards real-time processing of captured data.
2. EPCIS often deals not just with raw data captured from data carriers such as bar codes and RFID tags, but also in contexts that imbue those observations with meaning relative to the physical or digital world and to specific steps in operational or analytical business processes. The Capture layers are more purely observational in nature. An EPCIS event, while containing much of the same “Identify” data as a Filtering & Collection event or a bar code scan, is at a semantically higher level because it incorporates an understanding of the business context in which the identifier data were obtained. Moreover, there is no requirement that an EPCIS event be directly related to a specific physical data carrier observation. For example, an EPCIS event may indicate that a perishable trade item has just crossed its expiration date; such an event may be generated purely by software.
3. EPCIS operates within enterprise IT environments at a level that is much more diverse and multi-purpose than exists at the Capture layer, where typically systems are self-contained and exist to serve a single business purpose. In part, and most importantly, this is due to the desire to share EPCIS data between enterprises which are likely to have different solutions deployed to perform similar tasks. In part, it is also due to the persistent nature of EPCIS data. And lastly, it is due to EPCIS being at the highest level of the overall architecture, and hence the natural point of entry into other enterprise systems, which vary widely from one enterprise to the next (or even within parts of the same enterprise).

2.3 EPCIS in Relation to Trading Partners

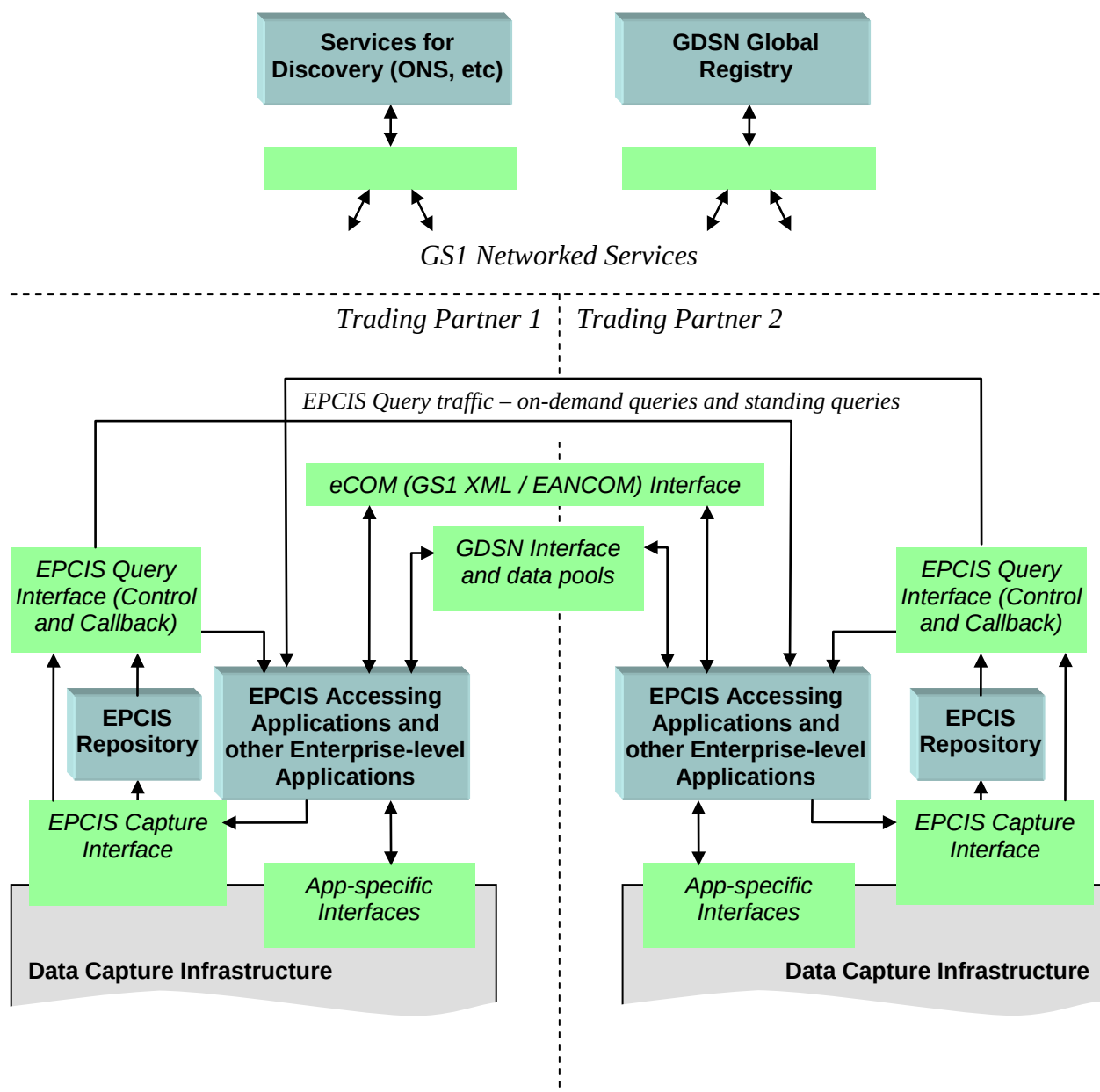
GS1 Standards in the “Share” layer pertain to three categories of data that are shared between end users:

Data	Description	GS1 Standards
Master Data	Data, shared by one trading partner to many trading partners, that provide descriptive attributes of real-world entities identified by GS1 Identification Keys, including trade items, parties, and physical locations.	GDSN
Transaction Data	Trade transactions triggering or confirming the execution of a function within a business process as defined by an explicit business agreement (e.g., a supply contract) or an implicit one (e.g., customs processing), from the start of the business process (e.g., ordering the product) to the end of it (e.g., final settlement), also making use of GS1 Identification Keys.	GS1 eCOM XML EANCOM
Visibility Event Data	Details about physical or digital activity in the supply chain of products and other assets, identified by keys, detailing where these objects are in time, and why; not just within one organization's four walls, but across organizations.	EPCIS

254

255 Transaction Data and Visibility Event Data have the characteristic that new documents of those
256 types are continually created as more business is transacted in a supply chain in steady state,
257 even if no new real-world entities are being created. Master Data, in contrast, is more static: the
258 Master Data for a given entity changes very slowly (if at all), and the quantity of Master Data
259 only increases as new entities are created, not merely because existing entities participate in
260 business processes. For example, as a given trade item instance moves through the supply chain,
261 new transaction data and visibility event data are generated as that instance undergoes business
262 transactions (such as purchase and sale) and physical handling processes (packing, picking,
263 stocking, etc). But new Master Data is only created when a new trade item or location is added to
264 the supply chain.

265 The following figure illustrates the flow of data between trading partners, emphasizing the parts
266 of the EPCIS standard involved in the flow of visibility event data.



267

268 2.4 EPCIS in Relation to other GS1 System Architecture Components

269 The following outlines the responsibilities of each element of the GS1 System Architecture as
 270 illustrated in the figures in the preceding sections. Further information may be found in
 271 [GS1Arch], from which the above diagram and much of the above text is quoted, and [EPCAF],
 272 from which much of the following text is quoted.

- 273 • *RFID and Bar Code Readers* Make observations of RFID tags while they are in the read
 274 zone, and observations of bar codes when reading is triggered.

- 275 • *Low-Level [RFID] Reader Protocol (LLRP) Interface* Defines the control and delivery of raw
276 RFID tag reads from RFID Readers to the Filtering & Collection role. Events at this interface
277 say “Reader A saw EPC X at time T.”
- 278 • *Filtering & Collection* This role filters and collects raw RFID tag reads, over time intervals
279 delimited by events defined by the EPCIS Capturing Application (e.g. tripping a motion
280 detector). No comparable role typically exists for reading bar codes, because bar code readers
281 typically only read a single bar code when triggered.
- 282 • *Filtering & Collection (ALE) Interface* Defines the control and delivery of filtered and
283 collected RFID tag read data from the Filtering & Collection role to the Data Capture
284 Workflow role. Events at this interface say “At Logical Reader L, between time T1 and T2,
285 the following EPCs were observed,” where the list of EPCs has no duplicates and has been
286 filtered by criteria defined by the EPCIS Capturing Application. In the case of bar codes,
287 comparable data is delivered to the Data Capture Workflow role directly from the bar code
288 reader in the form of a GS1 Element String.
- 289 • *Data Capture Workflow* Supervises the operation of the lower-level architectural elements,
290 and provides business context by coordinating with other sources of information involved in
291 executing a particular step of a business process. The Data Capture Workflow may, for
292 example, coordinate a conveyor system with Filtering & Collection events and bar code
293 reads, may check for exceptional conditions and take corrective action (e.g., diverting a bad
294 object into a rework area), may present information to a human operator, and so on. The Data
295 Capture Workflow understands the business process step or steps during which EPCIS event
296 data capture takes place. This role may be complex, involving the association of multiple
297 Filtering & Collection events and/or bar code reads with one or more business events, as in
298 the loading of a shipment. Or it may be straightforward, as in an inventory business process
299 where there may be readers deployed that generate observations about objects that enter or
300 leave the shelf. Here, the Filtering & Collection-level event or bar code read and the EPCIS-
301 level event may be so similar that very little actual processing at the Data Capture Workflow
302 level is necessary, and the Data Capture Workflow merely configures and routes events from
303 the Filtering & Collection interface and/or bar code readers directly through the EPCIS
304 Capture Interface to an EPCIS-enabled Repository or a business application. A Data Capture
305 Workflow whose primary output consists of EPCIS events is called an “EPCIS Capturing
306 Application” within this standard.
- 307 • *EPCIS Interfaces* The interfaces through which EPCIS data is delivered to enterprise-level
308 roles, including EPCIS Repositories, EPCIS Accessing Applications, and data exchange with
309 partners. Events at these interfaces say, for example, “At location X, at time T, the following
310 contained objects (cases) were verified as being aggregated to the following containing
311 object (pallet).” There are actually three EPCIS Interfaces. The EPCIS Capture Interface
312 defines the delivery of EPCIS events from EPCIS Capturing Applications to other roles that
313 consume the data in real time, including EPCIS Repositories, and real-time “push” to EPCIS
314 Accessing Applications and trading partners. The EPCIS Query Control Interface defines a
315 means for EPCIS Accessing Applications and trading partners to obtain EPCIS data
316 subsequent to capture, typically by interacting with an EPCIS Repository. The EPCIS Query
317 Control Interface provides two modes of interaction. In “on-demand” or “synchronous”
318 mode, a client makes a request through the EPCIS Query Control Interface and receives a

response immediately. In “standing request” or “asynchronous” mode, a client establishes a subscription for a periodic query. Each time the periodic query is executed, the results are delivered asynchronously (or “pushed”) to a recipient via the EPCIS Query Callback Interface. The EPCIS Query Callback Interface may also be used to deliver information immediately upon capture; this corresponds to the “possible bypass for real-time push” arrow in the diagram. All three of these EPCIS interfaces are specified normatively in this document.

- *EPCIS Accessing Application* Responsible for carrying out overall enterprise business processes, such as warehouse management, shipping and receiving, historical throughput analysis, and so forth, aided by EPC-related data.

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

- *Partner Application* Trading Partner systems that perform the same role as an EPCIS Accessing Application, though from outside the responding party’s network. Partner Applications may be granted access to a subset of the information that is available from an EPCIS Capturing Application or within an EPCIS Repository.

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

- The *Low-Level [RFID] Reader Protocol (LLRP)* and *GS1 Element String* insulate the higher layers from knowing what RF protocols or bar code symbologies are in use, and what reader makes/models have been chosen. If a different reader is substituted, the information sent through these interfaces remains the same.
- In situations where RFID is used, the Filtering & Collection Interface insulates the higher layers from the physical design choices made regarding how RFID tags are sensed and accumulated, and how the time boundaries of events are triggered. If a single four-antenna RFID reader is replaced by a constellation of five single-antenna “smart antenna” readers, the events at the Filtering & Collection level remain the same. Likewise, if a different triggering mechanism is used to mark the start and end of the time interval over which reads are accumulated, the Filtering & Collection event remains the same.
- EPCIS insulates enterprise applications from understanding the details of how individual steps in a business process are carried out at a detailed level. For example, a typical EPCIS event is “At location X, at time T, the following cases were verified as being on the following pallet.” In a conveyor-based business implementation, this may correspond to a single Filtering & Collection event, in which reads are accumulated during a time interval whose start and end is triggered by the case crossing electric eyes surrounding a reader mounted on the conveyor. But another implementation could involve three strong people who move around the cases and use hand-held readers to read the tags. At the Filtering & Collection level, this looks very different (each triggering of the hand-held reader is likely a distinct Filtering & Collection event), and the processing done by the EPCIS Capturing Application is quite different (perhaps involving an interactive console that the people use to verify their work). But the EPCIS event is still the same for all these implementations.

In summary, EPCIS-level data differs from data employed at the Capture level in the GS1 System Architecture by incorporating semantic information about the business process in which data is collected, and providing historical observations. In doing so, EPCIS insulates applications that consume this information from knowing the low-level details of exactly how a given business process step is carried out.

3 EPCIS Specification Principles

The considerations in the previous two sections reveal that the requirements for standards at the EPCIS layer are considerably more complex than in the Capture layer of the GS1 System Architecture. The historical nature of EPCIS data implies that EPCIS interfaces need a richer set of access techniques than ALE or RFID and bar code reader interfaces. The incorporation of operational or business process context into EPCIS implies that EPCIS traffics in a richer set of data types, and moreover needs to be much more open to extension in order to accommodate the wide variety of business processes in the world. Finally, the diverse environment in which EPCIS operates implies that the EPCIS Standard be layered carefully so that even when EPCIS is used between external systems that differ widely in their details of operation, there is consistency and interoperability at the level of what the abstract structure of the data is and what the data means.

In response to these requirements, EPCIS is described by a framework specification and narrower, more detailed specifications that populate that framework. The framework is designed to be:

- *Layered* In particular, the structure and meaning of data in an abstract sense is specified separately from the concrete details of data access services and bindings to particular interface protocols. This allows for variation in the concrete details over time and across enterprises while preserving a common meaning of the data itself. It also permits EPCIS data specifications to be reused in approaches other than the service-oriented approach of the present specification. For example, data definitions could be reused in an EDI framework.
- *Extensible* The core specifications provide a core set of data types and operations, but also provide several means whereby the core set may be extended for purposes specific to a given industry or application area. Extensions not only provide for proprietary requirements to be addressed in a way that leverages as much of the standard framework as possible, but also provides a natural path for the standards to evolve and grow over time.
- *Modular* The layering and extensibility mechanisms allow different parts of the complete EPCIS framework to be specified by different documents, while promoting coherence across the entire framework. This allows the process of standardization (as well as of implementation) to scale.

The remainder of this document specifies the EPCIS framework. It also populates that framework with a core set of data types and data interfaces. The companion standard, the GS1 Core Business Vocabulary (CBV), provides additional data definitions that layer on top of what is provided by the EPCIS standard.

4 Terminology and Typographical Conventions

Within this specification, the terms SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, NEED NOT, CAN, and CANNOT are to be interpreted as specified in Annex G of the ISO/IEC Directives, Part 2, 2001, 4th edition [ISODir2]. When used in this way, these terms will always be shown in ALL CAPS; when these words appear in ordinary typeface they are intended to have their ordinary English meaning.

All sections of this document, with the exception of Sections 1, 2, and 3, are normative, except where explicitly noted as non-normative.

The following typographical conventions are used throughout the document:

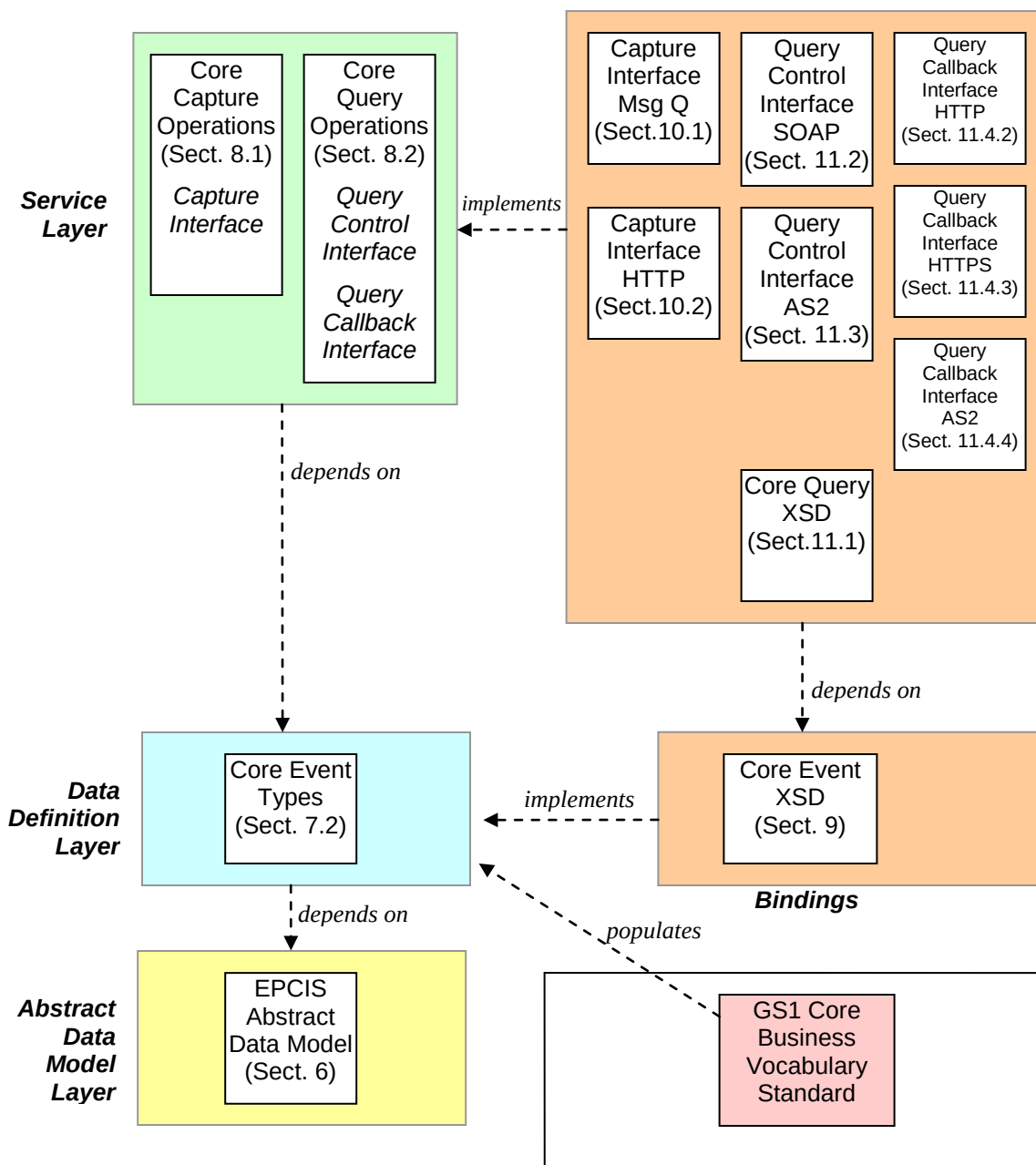
- ALL CAPS type is used for the special terms from [ISODir2] enumerated above.
- Monospace type is used to denote programming language, UML, and XML identifiers, as well as for the text of XML documents.
- Placeholders for changes that need to be made to this document prior to its reaching the final stage of approved GS1 Standard are prefixed by a rightward-facing arrowhead, as this paragraph is.

5 EPCIS Specification Framework

The EPCIS specification is designed to be layered, extensible, and modular.

5.1 Layers

The EPCIS specification framework is organized into several layers, as illustrated below:



These layers are described below.

- Abstract Data Model Layer** The Abstract Data Model Layer specifies the generic structure of EPCIS data. This is the only layer that is not extensible by mechanisms other than a revision to the EPCIS specification itself. The Abstract Data Model Layer specifies the general requirements for creating data definitions within the Data Definition Layer.
- Data Definition Layer** The Data Definition Layer specifies what data is exchanged through EPCIS, what its abstract structure is, and what it means. One data definition module is defined within the present specification, called the Core Event Types Module. Data definitions in the Data Definition Layer are specified abstractly, following rules defined by the Abstract Data Model Layer.

- Service Layer** The Service Layer defines service interfaces through which EPCIS clients interact. In the present specification, two service layer modules are defined. The Core Capture Operations Module defines a service interface (the EPCIS Capture Interface) through which EPCIS Capturing Applications use to deliver Core Event Types to interested parties. The Core Query Operations Module defines two service interfaces (the EPCIS Query Control Interface and the EPCIS Query Callback Interface) that EPCIS Accessing Applications use to obtain data previously captured. Interface definitions in the Service Layer are specified abstractly using UML.
- Bindings** Bindings specify concrete realizations of the Data Definition Layer and the Service Layer. There may be many bindings defined for any given Data Definition or Service module. In this specification, a total of nine bindings are specified for the three modules defined in the Data Definition and Service Layers. The data definitions in the Core Event Types data definition module are given a binding to an XML schema. The EPCIS Capture Interface in the Core Capture Operations Module is given bindings for Message Queue and HTTP. The EPCIS Query Control Interface in the Core Query Operations Module is given a binding to SOAP over HTTP via a WSDL web services description, and a second binding for AS2. The EPCIS Query Callback Interface in the Core Query Operations Module is given bindings to HTTP, HTTPS, and AS2.
- GS1 Core Business Vocabulary Standard** The GS1 Core Business Vocabulary standard [CBV1.1] is a companion to the EPCIS standard. It defines specific vocabulary elements that may be used to populate the data definitions specified in the Data Definition Layer of the EPCIS standard. While EPCIS may be used without CBV, by employing only private or proprietary data values, it is far more beneficial for EPCIS applications to make as much use of the CBV Standard as possible.

5.2 Extensibility

The layered technique for specification promotes extensibility, as one layer may be reused by more than one implementation in another layer. For example, while this specification includes an XML binding of the Core Event Types data definition module, another specification may define a binding of the same module to a different syntax, for example a CSV file.

Besides the extensibility inherent in layering, the EPCIS specification includes several specific mechanisms for extensibility:

- Subclassing** Data definitions in the Data Definition Layer are defined using UML, which allows a new data definition to be introduced by creating a subclass of an existing one. A subclass is a new type that includes all of the fields of an existing type, extending it with new fields. An instance of a subclass may be used in any context in which an instance of the parent class is expected.
- Extension Points** Data definitions and service specifications also include extension points, which vendors may use to provide extended functionality without creating subclasses.

5.3 Modularity

The EPCIS specification framework is designed to be modular. That is, it does not consist of a single specification, but rather a collection of individual specifications that are interrelated. This

allows EPCIS to grow and evolve in a distributed fashion. The layered structure and the extension mechanisms provide the essential ingredients to achieving modularity, as does the grouping into modules.

While EPCIS specifications are modular, there is no requirement that the module boundaries of the specifications be visible or explicit within *implementations* of EPCIS. For example, there may be a particular software product that provides a SOAP/HTTP-based implementation of a case-to-pallet association service and a product catalogue service that traffics in data defined in the relevant data definition modules. This product may conform to as many as six different modules from the EPCIS standard: the data definition module that describes product catalogue data, the data definition module that defines case-to-pallet associations, the specifications for the respective services, and the respective SOAP/HTTP bindings. But the source code of the product may have no trace of these boundaries, and indeed the concrete database schema used by the product may denormalize the data so that product catalogue and case-to-pallet association data are inextricably entwined. But as long as the net result conforms to the specifications, this implementation is permitted.

6 Abstract Data Model Layer

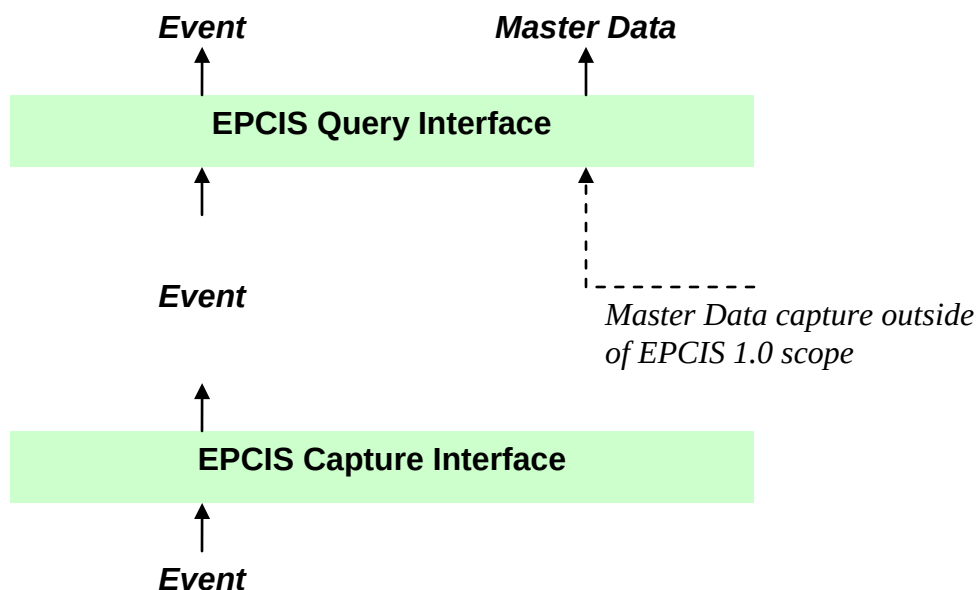
This section gives a normative description of the abstract data model that underlies EPCIS.

6.1 Event Data and Master Data

Generically, EPCIS deals in two kinds of data: event data and master data. Event data arises in the course of carrying out business processes, and is captured through the EPCIS Capture Interface and made available for query through the EPCIS Query Interfaces. Master data is additional data that provides the necessary context for interpreting the event data. It is available for query through the EPCIS Query Control Interface, but the means by which master data enters the system is not specified in the EPCIS 1.1 specification.

Roadmap (non-normative): It is possible that capture of master data will be addressed in a future version of the EPCIS specification.

These relationships are illustrated below:

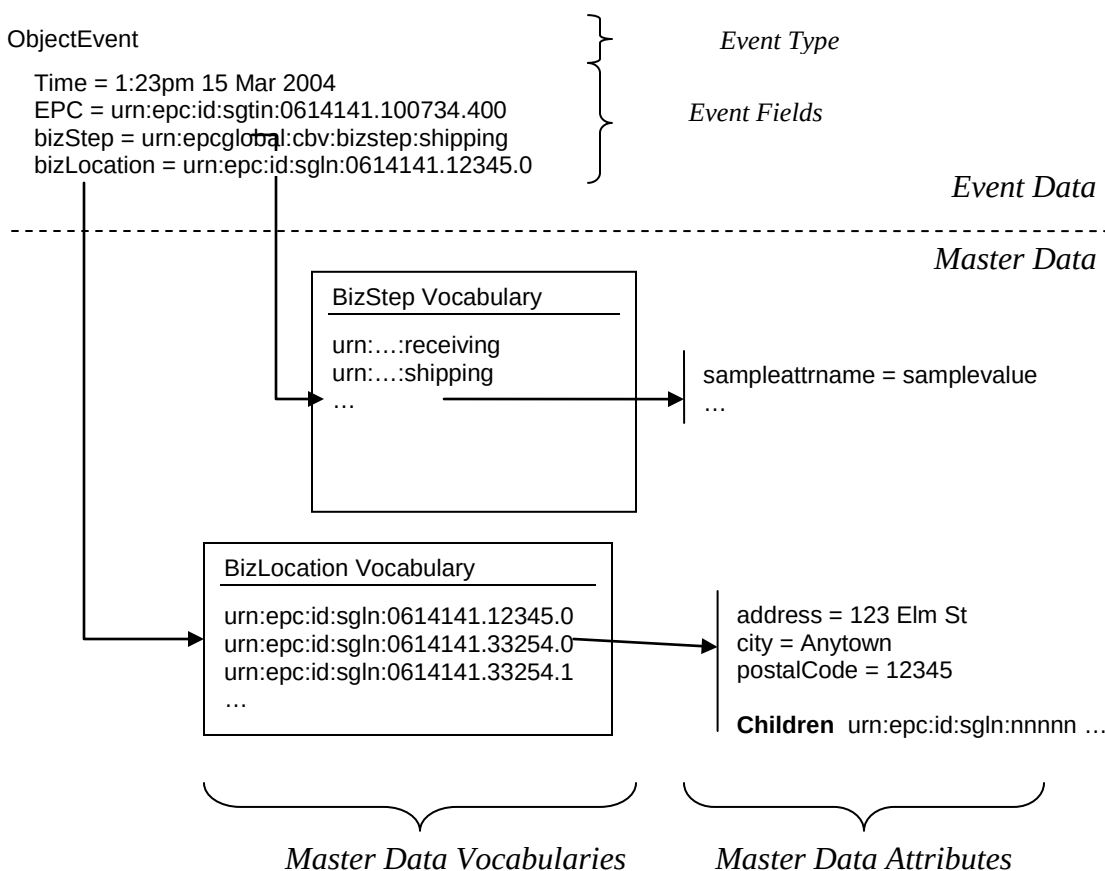


499

500 The Abstract Data Model Layer does not attempt to define the meaning of the terms “event data”
501 or “master data,” other than to provide precise definitions of the structure of the data as used by
502 the EPCIS specification. The modelling of real-world business information as event data and
503 master data is the responsibility of the Data Definition Layer, and of industry and end-user
504 agreements that build on top of this specification.

505 *Explanation (non-normative): While for the purposes of this specification the terms “event data”*
506 *and “master data” mean nothing more than “data that fits the structure provided here,” the*
507 *structures defined in the Abstract Data Model Layer are designed to provide an appropriate*
508 *representation for data commonly requiring exchange through EPCIS. Informally, these two*
509 *types of data may be understood as follows. Event data grows in quantity as more business is*
510 *transacted, and refers to things that happen at specific moments in time. An example of event*
511 *data is “At 1:23pm on 15 March 2004, EPC X was observed at Location L.” Master data does*
512 *not generally grow merely because more business is transacted (though master data does tend to*
513 *grow as organizations grow in size), is not typically tied to specific moments in time (though*
514 *master data may change slowly over time), and provides interpretation for elements of event*
515 *data. An example of master data is “Location L refers to the distribution centre located at*
516 *123 Elm Street, Anytown, US.” All of the data in the set of use cases considered in the creation*
517 *of the EPCIS 1.1 specification can be modelled as a combination of event data and master data*
518 *of this kind.*

519 The structure of event data and master data in EPCIS is illustrated below. (Note that this is an
520 illustration only: the specific vocabulary elements and master data attribute names in this figure
521 are not defined within this specification.)



- 522
- 523 The ingredients of the EPCIS Abstract Data Model are defined below:
- 524 • *Event Data* A set of Events.
- 525 • *Event* A structure consisting of an Event Type and one or more named Event Fields.
- 526 • *Event Type* A namespace-qualified name (qname) that indicates to which of several possible
- 527 Event structures (as defined by the Data Definition Layer) a given event conforms.
- 528 • *Event Field* A named field within an Event. The name of the field is given by a qname,
- 529 referring either to a field name specified by the Data Definition Layer or a field name defined
- 530 as an extension to this specification. The value of the field may be a primitive type (such as
- 531 an integer or timestamp), a Vocabulary Element, or a list of primitive types or Vocabulary
- 532 Elements.
- 533 • *Master Data* A set of Vocabularies, together with Master Data Attributes associated with
- 534 elements of those Vocabularies.
- 535 • *Vocabulary* A named set of identifiers. The name of a Vocabulary is a qname that may be
- 536 used as a type name for an event field. The identifiers within a Vocabulary are called
- 537 Vocabulary Elements. A Vocabulary represents a set of alternative values that may appear as
- 538 the values of specific Event Fields. Vocabularies in EPCIS are used to model sets such as the
- 539 set of available location names, the set of available business process step names, and so on.

• *Vocabulary Element* An identifier that names one of the alternatives modelled by a Vocabulary. The value of an Event Field may be a Vocabulary Element. Vocabulary Elements are represented as Uniform Resource Identifiers (URIs). Each Vocabulary Element may have associated Master Data Attributes.

• *Master Data Attributes* An unordered set of name/value pairs associated with an individual Vocabulary Element. The name part of a pair is a qname. The value part of a pair may be a value of arbitrary type. A special attribute is a (possibly empty) list of children, each child being another vocabulary element from the same vocabulary. See Section 6.5.

New EPCIS Events are generated at the edge and delivered into EPCIS infrastructure through the EPCIS Capture Interface, where they can subsequently be delivered to interested applications through the EPCIS Query Interfaces. There is no mechanism provided in either interface by which an application can delete or modify an EPCIS Event. The only way to “retract” or “correct” an EPCIS Event is to generate a subsequent event whose business meaning is to rescind or amend the effect of a prior event.

While the EPCIS Capture Interface and EPCIS Query Interfaces provide no means for an application to explicitly request the deletion of an event, EPCIS Repositories MAY implement data retention policies that cause old EPCIS events to become inaccessible after some period of time.

Master data, in contrast, may change over time, though such changes are expected to be infrequent relative to the rate at which new event data is generated. The current version of this specification does not specify how master data changes (nor, as noted above, does it specify how master data is entered in the first place).

6.2 Vocabulary Kinds

Vocabularies are used extensively within EPCIS to model physical, digital, and conceptual entities that exist in the real world. Examples of vocabularies defined in the core EPCIS Data Definition Layer are location names, object class names (an object class name is something like “Acme Deluxe Widget,” as opposed to an EPC which names a specific instance of an Acme Deluxe Widget), and business step names. In each case, a vocabulary represents a finite (though open-ended) set of alternatives that may appear in specific fields of events.

It is useful to distinguish two kinds of vocabularies, which follow different patterns in the way they are defined and extended over time:

• *Standard Vocabulary* A Standard Vocabulary represents a set of Vocabulary Elements whose definition and meaning must be agreed to in advance by trading partners who will exchange events using the vocabulary. For example, the EPCIS Core Data Definition Layer defines a vocabulary called “business step,” whose elements are identifiers denoting such things as “shipping,” “receiving,” and so on. One trading partner may generate an event having a business step of “shipping,” and another partner receiving that event through a query can interpret it because of a prior agreement as to what “shipping” means.

Standard Vocabulary elements tend to be defined by organizations of multiple end users, such as GS1, industry consortia outside GS1, private trading partner groups, and so on. The master data associated with Standard Vocabulary elements are defined by those same organizations, and tend to be distributed to users as part of a specification or by some similar

means. New vocabulary elements within a given Standard Vocabulary tend to be introduced through a very deliberate and occasional process, such as the ratification of a new version of a standard or through a vote of an industry group. While an individual end user organization acting alone may introduce a new Standard Vocabulary element, such an element would have limited use in a data exchange setting, and would probably only be used within an organization's four walls.

- *User Vocabulary* A User Vocabulary represents a set of Vocabulary Elements whose definition and meaning are under the control of a single organization. For example, the EPCIS Core Data Definition Layer defines a vocabulary called "business location," whose elements are identifiers denoting such things as "Acme Corp. Distribution Centre #3." Acme Corp may generate an event having a business location of "Acme Corp. Distribution Centre #3," and another partner receiving that event through a query can interpret it either because it correlates it with other events naming the same location, or by looking at master data attributes associated with the location, or both.

User Vocabulary elements are primarily defined by individual end user organizations acting independently. The master data associated with User Vocabulary elements are defined by those same organizations, and are usually distributed to trading partners through the EPCIS Query Control Interface or other data exchange / data synchronization mechanisms. New vocabulary elements within a given User Vocabulary are introduced at the sole discretion of an end user, and trading partners must be prepared to respond accordingly. Usually, however, the rules for constructing new User Vocabulary Elements are established by organizations of multiple end users, and in any case must follow the rules defined in Section 6.4 below.

The lines between these two kinds of vocabularies are somewhat subjective. However, the mechanisms defined in the EPCIS specification make absolutely no distinction between the two vocabulary types, and so it is never necessary to identify a particular vocabulary as belonging to one type or the other. The terms "Standard Vocabulary" and "User Vocabulary" are introduced only because they are useful as a hint as to the way a given vocabulary is expected to be defined and extended.

The GS1 Core Business Vocabulary (CBV) standard [CBV1.1] provides standardized vocabulary elements for many of the vocabulary types used in EPCIS event types. In particular, the CBV defines vocabulary elements for the following EPCIS Standard Vocabulary types: Business Step, Disposition, Business Transaction Type, and Source/Destination Type. The CBV also defines templates for constructing vocabulary elements for the following EPCIS User Vocabulary types: Object (EPC), Object Class (EPCClass), Location (Read Point and Business Location), Business Transaction ID, Source/Destination ID, and Transformation ID.

6.3 Extension Mechanisms

A key feature of EPCIS is its ability to be extended by different organizations to adapt to particular business situations. In all, the Abstract Data Model Layer provides five methods by which the data processed by EPCIS may be extended (the Service Layer, in addition, provides mechanisms for adding additional services), enumerated here from the most invasive type of extension to the least invasive:

- 623 • *New Event Type* A new Event Type may be added in the Data Definition Layer. Adding a
624 new Event Type requires each of the Data Definition Bindings to be extended, and may also
625 require extension to the Capture and Query Interfaces and their Bindings.
- 626 • *New Event Field* A new field may be added to an existing Event Type in the Data Definition
627 Layer. The bindings, capture interface, and query interfaces defined in this specification are
628 designed to permit this type of extension without requiring changes to the specification itself.
629 (The same may not be true of other bindings or query languages defined outside this
630 specification.)
- 631 • *New Vocabulary Type* A new Vocabulary Type may be added to the repertoire of available
632 Vocabulary Types. No change to bindings or interfaces are required.
- 633 • *New Master Data Attribute* A new attribute name may be defined for an existing Vocabulary.
634 No change to bindings or interfaces are required.
- 635 • *New Instance/Lot Master Data (ILMD) Attribute* A new attribute name may be defined for
636 use in Instance/Lot Master Data (ILMD); see Section 7.3.6. No change to bindings or
637 interfaces are required.
- 638 • *New Vocabulary Element* A new element may be added to an existing Vocabulary.
- 639 The Abstract Data Model Layer has been designed so that most extensions arising from adoption
640 by different industries or increased understanding within a given industry can be accommodated
641 by the latter methods in the above list, which do not require revision to the specification itself.
642 The more invasive methods at the head of the list are available, however, in case a situation
643 arises that cannot be accommodated by the latter methods.
- 644 It is expected that there will be several different ways to extend the EPCIS specification, as
645 summarized below:

How Extension is Disseminated	Responsible Organization	Extension Method				
		New Event Type	New Event Field	New Vocabulary Type	New Master Data or ILMD (Section 7.3.6) Attribute	New Vocabulary Element
New Version of EPCIS standard	GS1 EPCIS Working Group	Yes	Yes	Yes	Occasionally	Rarely
New Version of CBV standard	GS1 Core Business Vocabulary Working Group	No	No	No	Occasionally	Yes (Standard Vocabulary, User Vocabulary template)
GS1 Application Standard for a specific industry	GS1 Application Standard Working Group for a specific industry	Rarely	Rarely	Occasionally	Yes	Yes (Standard Vocabulary)

How Extension is Disseminated	Responsible Organization	Extension Method				
		New Event Type	New Event Field	New Vocabulary Type	New Master Data or ILMD (Section 7.3.6) Attribute	New Vocabulary Element
GS1 Member Organisation Local Recommendation Document for a specific industry within a specific geography	GS1 Member Organization	Rarely	Rarely	Occasionally	Yes	Yes (Standard Vocabulary)
Private Group Interoperability Specification	Industry Consortium or Private End User Group outside GS1	Rarely	Rarely	Occasionally	Yes	Yes (Standard Vocabulary)
Updated Master Data via EPCIS Query or other data sync	Individual End User	Rarely	Rarely	Rarely	Rarely	Yes (User vocabulary)

646

647 6.4 Identifier Representation

648 The Abstract Data Model Layer introduces several kinds of identifiers, including Event Type
649 names, Event Field names, Vocabulary names, Vocabulary Elements, and Master Data Attribute
650 Names. Because all of these namespaces are open to extension, this specification imposes some
651 rules on the construction of these names so that independent organizations may create extensions
652 without fear of name collision.

653 Vocabulary Elements are subject to the following rules. In all cases, a Vocabulary Element is
654 represented as Uniform Resource Identifier (URI) whose general syntax is defined in
655 [RFC2396]. The types of URIs admissible as Vocabulary Elements are those URIs for which
656 there is an owning authority. This includes:

- 657 • URI representations for EPC codes [TDS1.9, Section 7]. The owning authority for a
658 particular EPC URI is the organization to whom the EPC manager number was assigned.
- 659 • Absolute Uniform Resource Locators (URLs) [RFC1738]. The owning authority for a
660 particular URL is the organization that owns the Internet domain name in the authority
661 portion of the URL.

- Uniform Resource Names (URNs) [RFC2141] in the `oid` namespace that begin with a Private Enterprise Number (PEN). The owning authority for an OID-URN is the organization to which the PEN was issued.
 - Uniform Resource Names (URNs) [RFC2141] in the `epc` or `epcglobal` namespace, other than URIs used to represent EPCs [TDS1.9]. The owning authority for these URNs is GS1.
- Event Type names and Event Field names are represented as namespace-qualified names (qnames), consisting of a namespace URI and a name. This has a straightforward representation in XML bindings that is convenient for extension.

6.5 Hierarchical Vocabularies

Some Vocabularies have a hierarchical or multi-hierarchical structure. For example, a vocabulary of location names may have an element that means “Acme Corp. Retail Store #3” as well others that mean “Acme Corp. Retail Store #3 Backroom” and “Acme Corp. Retail Store #3 Sales Floor.” In this example, there is a natural hierarchical relationship in which the first identifier is the parent and the latter two identifiers are children.

Hierarchical relationships between vocabulary elements are represented through master data. Specifically, a parent identifier carries, in addition to its master data attributes, a list of its children identifiers. Each child identifier SHALL belong to the same Vocabulary as the parent. In the example above, the element meaning “Acme Corp. Distribution Centre #3” would have a children list including the element that means “Acme Corp. Distribution Centre #3 Door #5.”

Elsewhere in this specification, the term “direct or indirect descendant” is used to refer to the set of vocabulary elements including the children of a given vocabulary element, the children of those children, etc. That is, the “direct or indirect descendants” of a vocabulary element are the set of vocabulary elements obtained by taking the transitive closure of the “children” relation starting with the given vocabulary element.

A given element MAY be the child of more than one parent. This allows for more than one way of grouping vocabulary elements; for example, locations could be grouped both by geography and by function. An element SHALL NOT, however, be a child of itself, either directly or indirectly.

Explanation (non-normative): In the present version of this specification, only one hierarchical relationship is provided for, namely the relationship encoded in the special “children” list. Future versions of this specification may generalize this to allow more than one relationship, perhaps encoding each relationship via a different master data attribute.

Hierarchical relationships are given special treatment in queries (Section 8.2), and may play a role in carrying out authorization policies (Section 8.2.2), but do not otherwise add any additional complexity or mechanism to the Abstract Data Model Layer.

7 Data Definition Layer

This section includes normative specifications of modules in the Data Definition Layer.

7.1 General Rules for Specifying Data Definition Layer Modules

The general rules for specifying modules in the Data Definition Layer are given here. These rules are then applied in Section 7.2 to define the Core Event Types Module. These rules can also be applied by organizations wishing to layer a specification on top of this specification.

7.1.1 Content

In general, a Data Definition Module specification has these components, which populate the Abstract Data Model framework specified in Section 6:

- *Value Types* Definitions of data types that are used to describe the values of Event Fields and of Master Data Attributes. The Core Event Types Module defines the primitive types that are available for use by all Data Definition Modules. Each Vocabulary that is defined is also implicitly a Value Type.
- *Event Types* Definitions of Event Types, each definition giving the name of the Event Type (which must be unique across all Event Types) and a list of standard Event Fields for that type. An Event Type may be defined as a subclass of an existing Event Type, meaning that the new Event Type includes all Event Fields of the existing Event Type plus any additional Event Fields provided as part of its specification.
- *Event Fields* Definitions of Event Fields within Event Types. Each Event Field definition specifies a name for the field (which must be unique across all fields of the enclosing Event Type) and the data type for values in that field. Event Field definitions within a Data Definition Module may be part of new Event Types introduced by that Module, or may extend Event Types defined in other Modules.
- *Vocabulary Types* Definitions of Vocabulary Types, each definition giving the name of the Vocabulary (which must be unique across all Vocabularies), a list of standard Master Data Attributes for elements of that Vocabulary, and rules for constructing new Vocabulary Elements for that Vocabulary. (Any rules specified for constructing Vocabulary Elements in a Vocabulary Type must be consistent with the general rules given in Section 6.4.)
- *Master Data Attributes* Definitions of Master Data Attributes for Vocabulary Types. Each Master Data Attribute definition specifies a name for the Attribute (which must be unique across all attributes of the enclosing Vocabulary Type) and the data type for values of that attribute. Master Data definitions within a Data Definition Module may belong to new Vocabulary Types introduced by that Module, or may extend Vocabulary Types defined in other Modules.
- *Vocabulary Elements* Definitions of Vocabulary Elements, each definition specifying a name (which must be unique across all elements within the Vocabulary, and conform to the general rules for Vocabulary Elements given in Section 6.4 as well as any specific rules specified in the definition of the Vocabulary Type), and optionally specifying master data (specific attribute values) for that element.

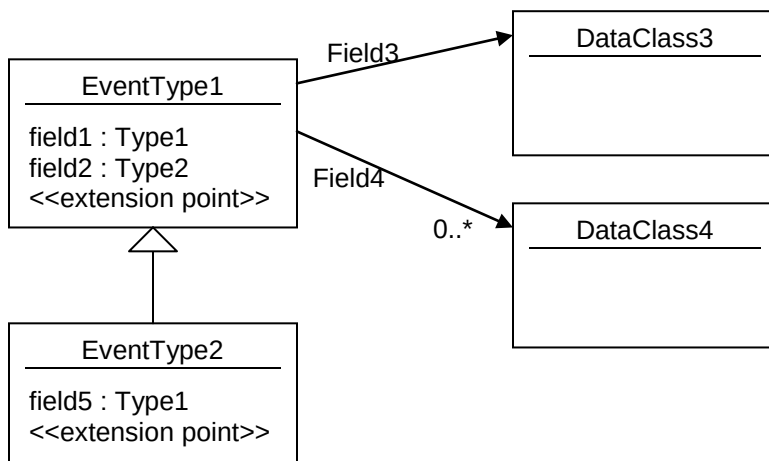
Amplification (non-normative): As explained in Section 6.3, Data Definition Modules defined in this specification and by companion specifications developed by the EPCIS Working Group will tend to include definitions of Value Types, Event Types, Event Fields, and Vocabulary Types, while modules defined by other groups will tend to include definitions of Event Fields that extend

existing Event Types, Master Data Attributes that extend existing Vocabulary Types, and Vocabulary Elements that populate existing Vocabularies. Other groups may also occasionally define Vocabulary Types.

The word “Vocabulary” is used informally to refer to a Vocabulary Type and the set of all Vocabulary Elements that populate it.

7.1.2 Notation

In the sections below, Event Types and Event fields are specified using a restricted form of UML class diagram notation. UML class diagrams used for this purpose may contain classes that have attributes (fields) and associations, but not operations. Here is an example:



This diagram shows a data definition for two Event Types, **EventType1** and **EventType2**. These event types make use of four Value Types: **Type1**, **Type2**, **DataClass3**, and **DataClass4**. **Type1** and **Type2** are primitive types, while **DataClass3** and **DataClass4** are complex types whose structure is also specified in UML.

The Event Type **EventType1** in this example has four fields. **Field1** and **Field2** are of primitive type **Type1** and **Type2** respectively. **EventType1** has another field **Field3** whose type is **DataClass3**. Finally, **EventType1** has another field **Field4** that contains a list of zero or more instances of type **DataClass4** (the “0..*” notation indicates “zero or more”).

This diagram also shows a data definition for **EventType2**. The arrow with the open-triangle arrowhead indicates that **EventType2** is a subclass of **EventType1**. This means that **EventType2** actually has five fields: four fields inherited from **EventType1** plus a fifth field **field5** of type **Type1**.

Within the UML descriptions, the notation **<<extension point>>** identifies a place where implementations SHALL provide for extensibility through the addition of new data members. (When one type has an extension point, and another type is defined as a subclass of the first type and also has an extension point, it does not mean the second type has two extension points; rather, it merely emphasizes that the second type is also open to extension.) Extensibility

mechanisms SHALL provide for both proprietary extensions by vendors of EPCIS-compliant products, and for extensions defined by GS1 through future versions of this specification or through new specifications.

In the case of the standard XML bindings, the extension points are implemented within the XML schema following the methodology described in Section 9.1.

All definitions of Event Types SHALL include an extension point, to provide for the extensibility defined in Section 6.3 (“New Event Fields”). Value Types MAY include an extension point.

7.1.3 Semantics

Each event (an instance of an Event Type) encodes several assertions which collectively define the semantics of the event. Some of these assertions say what was true at the time the event was captured. Other assertions say what is expected to be true following the event, until invalidated by a subsequent event. These are called, respectively, the *retrospective semantics* and the *prospective semantics* of the event. For example, if widget #23 enters building #5 through door #6 at 11:23pm, then one retrospective assertion is that “widget #23 was observed at door #6 at 11:23pm,” while a prospective assertion is that “widget #23 is in building #5.” The key difference is that the retrospective assertion refers to a specific time in the past (“widget #23 was observed...”), while the prospective assertion is a statement about the present condition of the object (“widget #23 is in...”). The prospective assertion presumes that if widget #23 ever leaves building #5, another EPCIS capture event will be recorded to supersede the prior one.

In general, retrospective semantics are things that were incontrovertibly known to be true at the time of event capture, and can usually be relied upon by EPCIS Accessing Applications as accurate statements of historical fact. Prospective semantics, since they attempt to say what is true after an event has taken place, must be considered at best to be statements of “what ought to be” rather than of “what is.” A prospective assertion may turn out not to be true if the capturing apparatus does not function perfectly, or if the business process or system architecture were not designed to capture EPCIS events in all circumstances. Moreover, in order to make use of a prospective assertion implicit in an event, an EPCIS Accessing Application must be sure that it has access to any subsequent event that might supersede the event in question.

The retrospective/prospective dichotomy plays an important role in EPCIS’s definition of location, in Section 7.3.4.

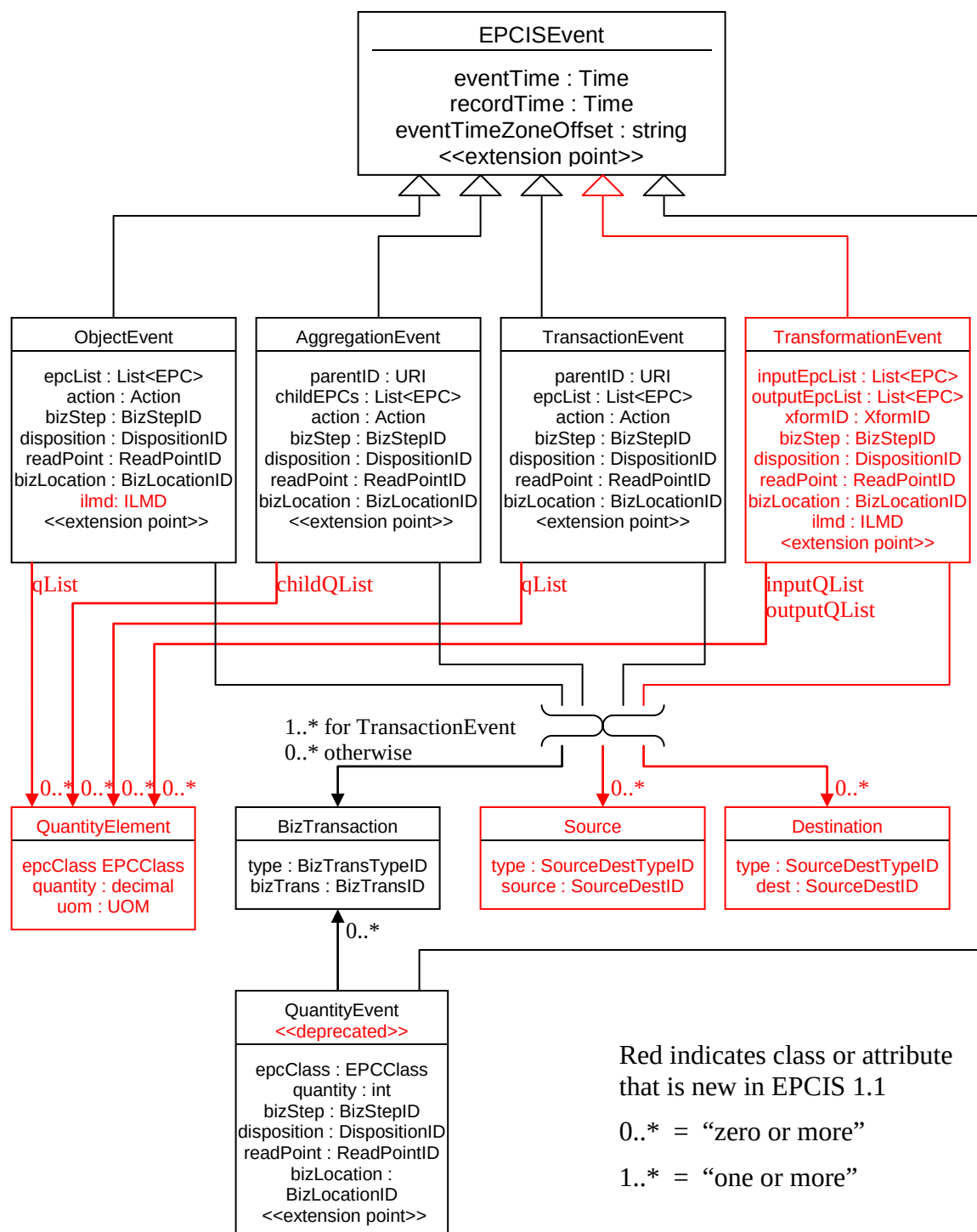
7.2 Core Event Types Module – Overview

The Core Event Types data definition module specifies the Event Types that represent EPCIS data capture events. These events are typically generated by an EPCIS Capturing Application and provided to EPCIS infrastructure using the data capture operations defined in Section 8.1. These events are also returned in response to query operations that retrieve events according to query criteria.

The components of this module, following the outline given in Section 7.1.1, are as follows:

- *Value Types* Primitive types defined in Sections 7.3.1 and 7.3.2.

- 806 • *Event Types* Event types as shown in the UML diagram below, and defined in Sections 7.4.1
807 through 7.4.6.
- 808 • *Event Fields* Included as part of the Event Types definitions.
- 809 • *Vocabulary Types* Types defined in Sections 7.3.3 through 7.3.5, and summarized in
810 Section 7.2.
- 811 • *Master Data Attributes* Included as part of Vocabulary Types definitions. It is expected that
812 industry vertical working groups will define additional master data attributes for the
813 vocabularies defined here.
- 814 • *Vocabulary Elements* None provided as part of this specification. It is expected that industry
815 vertical working groups will define vocabulary elements for the **BusinessStep**
816 vocabulary (Section 7.3.5), the **Disposition** vocabulary (Section 7.3.5.2), and the
817 **BusinessTransactionType** vocabulary (Section 7.3.5.3.1).
- 818 This module defines six event types, one very generic event and five subclasses (one of which is
819 deprecated as of EPCIS 1.1) that can represent events arising from supply chain activity across a
820 wide variety of industries:
- 821 • **EPCISEvent** (Section 7.4.1) is a generic base class for all event types in this module as
822 well as others.
- 823 • **ObjectEvent** (Section 7.4.2) represents an event that happened to one or more physical or
824 digital objects.
- 825 • **AggregationEvent** (Section 7.4.3) represents an event that happened to one or more
826 objects that are physically aggregated together (physically constrained to be in the same
827 place at the same time, as when cases are aggregated to a pallet).
- 828 • **QuantityEvent** (Section 7.4.4) represents an event concerned with a specific quantity of
829 objects sharing a common EPC class, but where the individual identities of the entities are
830 not specified. As of EPCIS 1.1, this event is deprecated; an **ObjectEvent** (Section 7.4.2)
831 with one or more **QuantityElements** (Section 7.3.3.3) should be used instead.
- 832 • **TransactionEvent** (Section 7.4.5) represents an event in which one or more objects
833 become associated or disassociated with one or more identified business transactions.
- 834 • **TransformationEvent** (Section 7.4.6) represents an event in which input objects are
835 fully or partially consumed and output objects are produced, such that any of the input
836 objects may have contributed to all of the output objects.
- 837 A UML diagram showing these Event Types is as follows:
838



Note: in this diagram, certain names have been abbreviated owing to space constraints; e.g., *BizLocationID* is used in the diagram, whereas the actual type is called *BusinessLocationID*. See the text of the specification for the normative names of fields and their types

Each of the core event types (not counting the generic **EPCISEvent**) has fields that represent four key dimensions of any EPCIS event. These four dimensions are: (1) the object(s) or other entities that are the subject of the event; (2) the date and time; (3) the location at which the event occurred; (4) the business context. These four dimensions may be conveniently remembered as “what, when, where, and why” (respectively). The “what” dimension varies depending on the event type (e.g., for an **ObjectEvent** the “what” dimension is one or more EPCs; for an **AggregationEvent** the “what” dimension is a parent ID and list of child EPCs). The “where” and “why” dimensions have both a retrospective aspect and a prospective aspect (see Section 7.1.3), represented by different fields.

The following table summarizes the fields of the event types that pertain to the four key dimensions:

	Retrospective (at the time of the event)	Prospective (true until contradicted by subsequent event)
What	EPC EPCClass + quantity	
When	Time	
Where	ReadPointID	BusinessLocationID
Why (business context)	BusinessStepID	DispositionID
	BusinessTransactionList	
	Source/Destination ILMD	

In addition to the fields belonging to the four key dimensions, events may carry additional descriptive information in other fields. It is expected that the majority of additional descriptive information fields will be defined by industry-specific specifications layered on top of this one.

The following table summarizes the vocabulary types defined in this module. The URI column gives the formal name for the vocabulary used when the vocabulary must be referred to by name across the EPCIS interface.

Vocabulary Type	Section	User / Standard	URI
ReadPointID	7.3.4	User	urn:epcglobal:epcis:vtype:ReadPoint
BusinessLocationID	7.3.4	User	urn:epcglobal:epcis:vtype:BusinessLocation
BusinessStepID	7.3.5	Standard	urn:epcglobal:epcis:vtype:BusinessStep
DispositionID	7.3.5.2	Standard	urn:epcglobal:epcis:vtype:Disposition

Vocabulary Type	Section	User / Standard	URI
BusinessTransaction	7.3.5.3.2	User	urn:epcglobal:epcis:vtype:BusinessTransaction
BusinessTransactionTypeID	7.3.5.3.1	Standard	urn:epcglobal:epcis:vtype:BusinessTransactionType
EPCClass	7.3.5.4	User	urn:epcglobal:epcis:vtype:EPCClass
SourceDestTypeID	7.3.5.4.1	Standard	urn:epcglobal:epcis:vtype:SourceDestType
SourceDestID	7.3.5.4.2	User	urn:epcglobal:epcis:vtype:SourceDest

858

859 7.3 Core Event Types Module – Building Blocks

860 This section specifies the building blocks for the event types defined in Section 7.3.5.4.

861 7.3.1 Primitive Types

862 The following primitive types are used within the Core Event Types Module.

Type	Description
int	An integer. Range restrictions are noted where applicable.
Time	A timestamp, giving the date and time in a time zone-independent manner. For bindings in which fields of this type are represented textually, an ISO-8601 compliant representation SHOULD be used.
EPC	An Electronic Product Code, as defined in [TDS1.9]. Unless otherwise noted, EPCs are represented in “pure identity” URI form as defined in [TDS1.9], Section 7.

863

864 The EPC type is defined as a primitive type for use in events when referring to EPCs that are not
865 part of a Vocabulary Type. For example, an SGTIN EPC used to denote an instance of a trade
866 item in the `epcList` field of an `ObjectEvent` is an instance of the EPC primitive type. But
867 an SGLN EPC used as a read point identifier (Section 7.3.4) in the `ReadPoint` field of an
868 `ObjectEvent` is a Vocabulary Element, not an instance of the EPC primitive type.

869 *Explanation (non-normative): This reflects a design decision not to consider individual trade*
870 *item instances as Vocabulary Elements having Master Data, owing to the fact that trade item*
871 *instances are constantly being created and hence new EPCs representing trade items are*
872 *constantly being commissioned. In part, this design decision reflects consistent treatment of*
873 *Master Data as excluding data that grows as more business is transacted (see comment in*
874 *Section 6.1), and in part reflects the pragmatic reality that data about trade item instances is*

875 likely to be managed more like event data than master data when it comes to aging, database
876 design, etc.

877 7.3.2 Action Type

878 The **Action** type says how an event relates to the lifecycle of the entity being described. For
879 example, **AggregationEvent** (Section 7.4.3) is used to capture events related to
880 aggregations of objects, such as cases aggregated to a pallet. Throughout its life, the pallet load
881 participates in many business process steps, each of which may generate an EPCIS event. The
882 **action** field of each event says how the aggregation itself has changed during the event: have
883 objects been added to the aggregation, have objects been removed from the aggregation, or has
884 the aggregation simply been observed without change to its membership? The **action** is
885 independent of the **bizStep** (of type **BusinessStepID**) which identifies the specific
886 business process step in which the action took place.

887 The **Action** type is an enumerated type having three possible values:

Action value	Meaning
ADD	The entity in question has been created or added to.
OBSERVE	The entity in question has not been changed: it has neither been created, added to, destroyed, or removed from.
DELETE	The entity in question has been removed from or destroyed altogether.

888 The description below for each event type that includes an **Action** value says more precisely
889 what **Action** means in the context of that event.

890 Note that the three values above are the only three values possible for **Action**. Unlike other
891 types defined below, **Action** is *not* a vocabulary type, and SHALL NOT be extended by
892 industry groups.

893 7.3.3 The “What” Dimension

894 This section defines the data types used in the “What” dimension of the event types specified in
895 Section 7.3.5.4.

896 7.3.3.1 Instance-level vs. Class-level Identification

897 The “What” dimension of an EPCIS event specifies what physical or digital objects participated
898 in the event. EPCIS provides for objects to be identified in two ways:

- 899 • *Instance-level* An identifier is said to be an instance-level identifier if such identifiers are
900 assigned so that each is unique to a single object. That is, no two objects are allowed to carry
901 the same instance-level identifier.
- 902 • *Class-level* An identifier is said to be a class-level identifier if multiple objects may carry the
903 same identifier.

904 In general, instance-level identifiers allow EPCIS events to convey more information, because it
905 is possible to correlate multiple EPCIS events whose “what” dimension includes the same
906 instance-level identifiers. For example, if an EPCIS event contains a given instance-level
907 identifier, and a subsequent EPCIS event contains the same identifier, then it is certain that the
908 very same object participated in both events. In contrast, if both events contained class-level
909 identifiers, then it is not certain that the same object participated in both events, because the
910 second event could have been a different instance of the same class (i.e., a different object
911 carrying the same class-level identifier as the first object). Class-level identifiers are typically
912 used only when it is impractical to assign unique instance-level identifiers to each object.

913 *Examples (non-normative): In the GS1 System, examples of instance-level identifiers include*
914 *GTIN+serial, SSCC, GRAI including serial, GIAI, GSRN, and GDTI including serial. Examples*
915 *of class-level identifiers include GTIN, GTIN+lot, GRAI without serial, and GDTI without serial.*

916 7.3.3.2 EPC

917 An Electronic Product Code (EPC) is an instance-level identifier structure defined in the EPC
918 Tag Data Standard [TDS1.9]. In the “what” dimension of an EPCIS event, the value of an **epc**
919 element SHALL be a URI [RFC2396] denoting the unique instance-level identity for an object.
920 When the unique identity is an Electronic Product Code, the list element SHALL be the “pure
921 identity” URI for the EPC as specified in [TDS1.9], Section 6. Implementations MAY accept
922 URI-formatted identifiers other than EPCs as the value of an **epc** element.

923 7.3.3.3 QuantityElement

924 A **QuantityElement** is a structure that identifies objects identified by a specific class-level
925 identifier, either a specific quantity or an unspecified quantity. It has the following structure:

Field	Type	Description
epcClass	EPCClass	A class-level identifier for the class to which the specified quantity of objects belongs.

Field	Type	Description
quantity	Decimal	(Optional) A number that specifies how many or how much of the specified EPCClass is denoted by this QuantityElement. The quantity may be omitted to indicate that the quantity is unknown or not specified. If quantity is omitted, then uom SHALL be omitted as well. Otherwise, if quantity is specified: If the QuantityElement lacks a uom field (below), then the quantity SHALL have a positive integer value, and denotes a count of the number of instances of the specified EPCClass that are denoted by this QuantityElement. If the QuantityElement includes a uom, then the quantity SHALL have a positive value (but not necessarily an integer value), and denotes the magnitude of the physical measure that specifies how much of the specified EPCClass is denoted by this QuantityElement
uom	UOM	(Optional) If present, specifies a unit of measure by which the specified quantity is to be interpreted as a physical measure, specifying how much of the specified EPCClass is denoted by this QuantityElement. The uom SHALL be omitted if quantity is omitted.

926

927 EPCClass is a Vocabulary whose elements denote classes of objects. EPCClass is a User
928 Vocabulary as defined in Section 6.2. Any EPC whose structure incorporates the concept of
929 object class can be referenced as an EPCClass. The standards for SGTIN EPCs are elaborated
930 below.

931 An EPCClass may refer to a class having fixed measure or variable measure. A fixed measure
932 class has instances that may be counted; for example, a GTIN that refers to fixed-size cartons of
933 a product. A variable measure class has instances that cannot be counted and so the quantity is
934 specified as a physical measure; for example, a GTIN that refers to copper wire that is sold by
935 length, carpeting that is sold by area, bulk oil that is sold by volume, or fresh produce that is sold
936 by weight. The following table summarizes how the quantity and uom fields are used in each
937 case:

EPCClass	quantity field	uom field	Meaning
Fixed measure	Positive integer	Omitted	The quantity field specifies the count of the specified class.

EPCClass	quantity field	uom field	Meaning
Variable measure	Positive number, not necessarily an integer	Present	The quantity field specifies the magnitude, and the uom field the physical unit, of a physical measure describing the amount of the specified class.
Fixed or Variable Measure	Omitted	Omitted	The quantity is unknown or not specified.

938

939 Master Data Attributes for the **EPCClass** vocabulary contain whatever master data is defined
940 for the referenced objects independent of EPCIS (for example, product catalogue data);
941 definitions of these are outside the scope of this specification.

942 7.3.3.3.1 UOM

943 As specified above, the **uom** field of a **QuantityElement** is present when the
944 **QuantityElement** uses a physical measure to specify the quantity of the specified
945 **EPCClass**. When a **uom** field is present, its value SHALL be the 2- or 3-character code for a
946 physical unit specified in the “Common Code” column of UN/CEFACT Recommendation 20
947 [CEFACT20]. Moreover, the code SHALL be a code contained in a row of [CEFACT20]
948 meeting all of the following criteria:

- 949 • The “Quantity” column contains one of the following quantities: *length*, *area*, *volume*, or
950 *mass*.
- 951 • The “Status” column does *not* contain “X” (deleted) or “D” (deprecated).

952 For purposes of the first criterion, the quantity must appear as a complete phrase. Example:
953 “metre” (MTR) is allowed, because the quantity includes *length* (among other quantities such as
954 *breadth*, *height*, etc.). But “pound-force per foot” (F17) is *not* allowed, because the quantity is
955 *force divided by length*, not just *length*.

956 7.3.3.3.2 EPCClass Values for GTIN

957 When a Vocabulary Element in **EPCClass** represents the class of SGTIN EPCs denoted by a
958 specific GTIN, it SHALL be a URI in the following form, as defined in Version 1.3 and later of
959 the EPC Tag Data Standards:

960 `urn:epc:idpat:sgtin:CompanyPrefix.ItemRefAndIndicator.*`

961 where *CompanyPrefix* is the GS1 Company Prefix of the GTIN (including leading zeros) and
962 *ItemRefAndIndicator* consists of the indicator digit of the GTIN followed by the digits of
963 the item reference of the GTIN.

964 An **EPCClass** vocabulary element in this form denotes the class of objects whose EPCs are
965 SGTINs (`urn:epc:id:sgtin:...`) having the same *CompanyPrefix* and

966 ItemRefAndIndicator fields, and having any serial number whatsoever (or no serial
967 number at all).

968 7.3.3.3.3 EPCClass Values for GTIN + Batch/Lot

969 When a Vocabulary Element in EPCClass represents the class of SGTIN EPCs denoted by a
970 specific GTIN and batch/lot, it SHALL be a URI in the following form, as defined in [TDS1.9,
971 Section 6]:

972 urn:epc:class:lgtin:CompanyPrefix.ItemRefAndIndicator.Lot

973 where *CompanyPrefix* is the GS1 Company Prefix of the GTIN (including leading zeros),
974 *ItemRefAndIndicator* consists of the indicator digit of the GTIN followed by the digits of
975 the item reference of a GTIN, and *Lot* is the batch/lot number of the specific batch/lot.

976 An EPCClass vocabulary element in this form denotes the class of objects whose EPCs are
977 SGTINs (urn:epc:id:sgtin:...) having the same *CompanyPrefix* and
978 *ItemRefAndIndicator* fields, and belonging to the specified batch/lot, regardless of serial
979 number (if any).

980 7.3.3.4 Summary of Identifier Types (Non-Normative)

981 This section summarizes the identifiers that may be used in the “what” dimension of EPCIS
982 events. The normative specifications of identifiers are in the EPC Tag Data Standard [TDS1.9]
983 and the EPC Core Business Vocabulary [CBV1.1].

Identifier Type	Instance-Level (EPC)	Class-Level (EPCClass)	URI Prefix	Normative Reference
GTIN		✓	urn:epc:idpat:sgtin:	[TDS1.9, Section 8]
GTIN + batch/lot		✓	urn:epc:class:lgtin:	[TDS1.9, Section 6]
GTIN + serial	✓		urn:epc:id:sgtin:	[TDS1.9, Section 6.3.1]
SSCC	✓		urn:epc:id:sscc:	[TDS1.9, Section 6.3.2]
GRAI (no serial)		✓	urn:epc:idpat:grai:	[TDS1.9, Section 8]
GRAI (with serial)	✓		urn:epc:id:grai:	[TDS 1.9, Section 6.3.4]

Identifier Type	Instance-Level (EPC)	Class-Level (EPCClass)	URI Prefix	Normative Reference
GIAI	✓		urn:epc:id:giai:	[TDS1.9, Section 6.3.5]
GDTI (no serial)		✓	urn:epc:idpat:gdti:	[TDS1.9, Section 8]
GDTI (with serial)	✓		urn:epc:id:gdti:	[TDS1.9, Section 6.3.7]
GSRN (Supplier)	✓		urn:epc:id:gsrn:	[TDS1.9, Section 6.3.6]
GSRN (Provider)	✓		urn:epc:id:gsrnp:	[TDS1.9, Section 6.3.6]
GCN (no serial)		✓	urn:epc:idpat:sgcn:	[TDS1.9, Section 8]
GCN (with serial)	✓		urn:epc:id:sgcn:	[TDS1.9, Section 6]
CPI		✓	urn:epc:idpat:cpi:	[TDS1.9, Section 8]
CPI + serial	✓		urn:epc:id:cpi:	[TDS1.9, Section 6.3.11]
GID	✓		urn:epc:id:gid:	[TDS1.9, Section 6.3.8]
USDoD	✓		urn:epc:id:usdod:	[TDS1.9, Section 6.3.9]
ADI	✓		urn:epc:id:adi:	[TDS1.9, Section 6.3.10]
Non-GS1 Identifier	✓	✓	Any URI – see CBV for recommendations	[CBV1.1, Section 8.2]

7.3.4 The “Where” Dimension – Read Point and Business Location

This section defines four types that all relate to the notion of *location* information as used in EPCIS. Two of these types are ways of referring to “readers,” or devices that sense the presence of EPC-tagged objects using RFID or other means. These are not actually considered to be “location” types at all for the purposes of EPCIS. They are included in this specification mainly to contrast them to the true location types (though some applications may want to use them as extension fields on observations, for auditing purposes).

The next two concepts are true location types, and are defined as EPCIS Vocabulary Types.

Explanation (non-normative): In the EPC context, the term location has been used to signify many different things and this has led to confusion about the meaning and use of the term, particularly when viewed from a business perspective. This confusion stems from a number of causes:

1. In situations where EPC Readers are stationary, there’s a natural tendency to equate the reader with a location, though that may not always be valid if there is more than one reader in a location;

2. There are situations where stationary Readers are placed between what business people would consider to be different locations (such as at the door between the backroom and sales floor of a retail store) and thus do not inherently determine the location without an indication of the direction in which the tagged object was travelling;

3. A single physical Reader having multiple, independently addressable antennas might be used to detect tagged objects in multiple locations as viewed by the business people;

4. Conversely, more than one Reader might be used to detect tagged objects in what business people would consider a single location;

5. With mobile Readers, a given Reader may read tagged objects in multiple locations, perhaps using “location” tags or other means to determine the specific location associated with a given read event;

6. And finally, locations of interest to one party (trading partner or application) may not be of interest to or authorized for viewing by another party, prompting interest in ways to differentiate locations.

The key to balancing these seemingly conflicting requirements is to define and relate various location types, and then to rely on the EPCIS Capturing Application to properly record them for a given capture event. This is why EPCIS events contain both a ReadPointID and a BusinessLocationID (the two primitive location types).

In addition, there has historically been much confusion around the difference between “location” as needed by EPCIS-level applications and reader identities. This EPCIS specification defines location as something quite distinct from reader identity. To help make this clear, the reader identity types are defined below to provide a contrast to the definitions of the true EPCIS location types. Also, reader identity types may enter into EPCIS as “observational attributes” when an application desires to retain a record of what readers played a role in an observation; e.g., for auditing purposes. (Capture and sharing of “observational attributes” would require use of extension fields not defined in this specification.)

1026 The reader/location types are as follows:

Type	Description
Primitive Reader Types – <i>not</i> location types for EPCIS	
PhysicalReaderID	This is the unique identity or name of the specific information source (e.g., a physical RFID Reader) that reports the results of an EPC read event. Physical Reader ID is further defined in [ALE1.0].
LogicalReaderID	This is the identity or name given to an EPC read event information source independent of the physical device or devices that are used to perform the read event. Logical Reader ID is further defined in [ALE1.0]. There are several reasons for introducing the Logical Reader concept as outlined in [ALE1.0], including allowing physical readers to be replaced without requiring changes to EPCIS Capturing Applications, allowing multiple physical readers to be given a single name when they are always used simultaneously to cover a single location, and (conversely) allowing a single physical reader to map to multiple logical readers when a physical reader has multiple antennas used independently to cover different locations.
True Location Types	
ReadPointID	A Read Point is a discretely recorded location that is meant to identify the most specific place at which an EPCIS event took place. Read Points are determined by the EPCIS Capturing Application, perhaps inferred as a function of logical reader if stationary readers are used, perhaps determined overtly by reading a location tag if the reader is mobile, or in general determined by any other means the EPCIS Capturing Application chooses to use. Conceptually, the Read Point is designed to identify “where objects were at the time of the EPCIS event.”
BusinessLocationID	A Business Location is a uniquely identified and discretely recorded location that is meant to designate the specific place where an object is assumed to be following an EPCIS event until it is reported to be at a different Business Location by a subsequent EPCIS event. As with the Read

Type	Description
	Point, the EPCIS Capturing Application determines the Business Location based on whatever means it chooses. Conceptually, the Business Location is designed to identify “where objects are following the EPCIS event.”

1027

1028 ReadPointID and BusinessLocationID are User Vocabularies as defined in Section 6.2.
1029 Some industries may wish to use EPCs as vocabulary elements, in which case pure identity URIs
1030 as defined in [TDS1.9] SHALL be used.

1031 *Illustration (non-normative): For example, in industries governed by GS1 General*
1032 *Specifications, readPointID, and businessLocationID may be SGLN-URIs [TDS1.9,*
1033 *Section 6.3.3], and physicalReaderID may be an SGTIN-URI [TDS1.9, Section 6.3.1].*

1034 But in all cases, location vocabulary elements are not *required* to be EPCs.

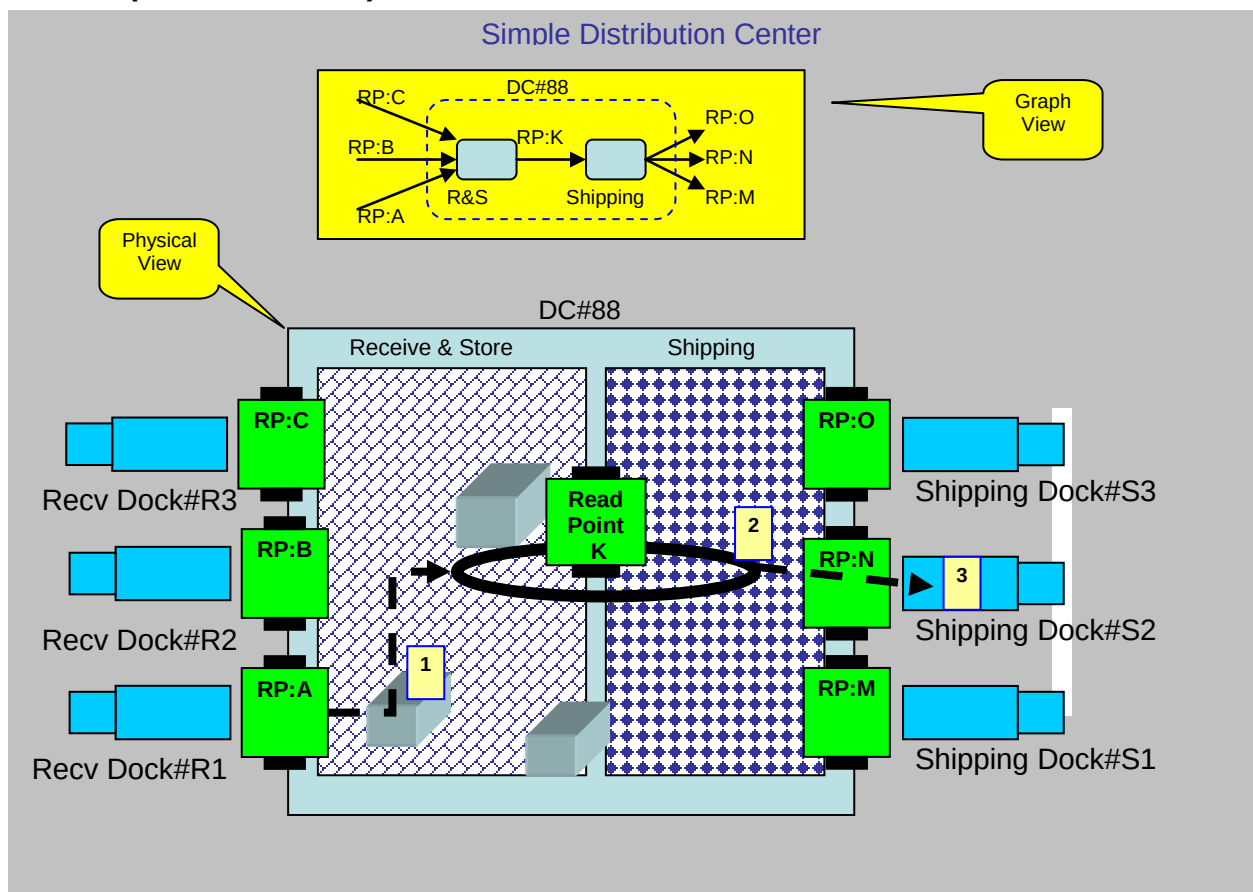
1035 *Explanation (non-normative): Allowing non-EPC URIs for locations gives organizations greater*
1036 *freedom to reuse existing ways of naming locations.*

1037 For all of the EPCIS Event Types defined in this Section 7.2, capture events include separate
1038 fields for Read Point and Business Location. In most cases, both are optional, so that it is still
1039 possible for an EPCIS Capturing Application to include partial information if both are not
1040 known.

1041 *Explanation (non-normative): Logical Reader and Physical Reader are omitted from the*
1042 *definitions of EPCIS events in this specification. Physical Reader is generally not useful*
1043 *information for exchange between partners. For example, if a reader malfunctions and is*
1044 *replaced by another reader of identical make and model, the Physical Reader ID has changed.*
1045 *This information is of little interest to trading partners. Likewise, the Logical Reader ID may*
1046 *change if the capturing organization makes a change in the way a particular business process is*
1047 *executed; again, not often of interest to a partner.*

1048 The distinction between Read Point and Business Location is very much related to the dichotomy
1049 between retrospective semantics and prospective semantics discussed above. In general, Read
1050 Points play a role in retrospective semantics, while Business Locations are involved in
1051 prospective statements. This is made explicit in the way each type of location enters the semantic
1052 descriptions given at the end of each section below that defines an EPCIS capture event.

7.3.4.1 Example of the distinction between a Read Point and a Business Location (Non-Normative)



Tag	Time	Read Point	Business Location	Comment
#123	7:00	"RP-DC#88-A"	DC#88.Receive & Store	Product entered DC via DockDoor#R1
#123	9:00	"RP-DC#88-K"	DC#88.Shipping	Product placed on conveyor for shipping
#123	9:30	"RP-DC#88-N"	[omitted]	Product shipped via dock door#S2

The figure above shows a typical use case consisting of rooms with fixed doorways at the boundaries of the rooms. In such a case, Read Points correspond to the doorways (with RFID instrumentation) and Business Locations correspond to the rooms. Note that the Read Points and Business Locations are not in one-to-one correspondence; the only situation where Read Points and Business Locations could have a 1:1 relationship is the unusual case of a room with a single door, such a small storeroom.

1063 *Still considering the rooms-and-doors example, the Business Location is usually the location*
 1064 *type of most interest to a business application, as it says which room an object is in. Thus it is*
 1065 *meaningful to ask the inventory of a Business Location such as the backroom. In contrast, the*
 1066 *Read Point indicates the doorway through which the object entered the room. It is not*
 1067 *meaningful to ask the inventory of a doorway. While sometimes not as relevant to a business*
 1068 *application, the Read Point is nevertheless of significant interest to higher level software to*
 1069 *understand the business process and the final status of the object, particularly in the presence of*
 1070 *less than 100% read rates. Note that correct designation of the business location requires both*
 1071 *that the tagged object be observed at the Read Point and that the direction of movement be*
 1072 *correctly determined – again reporting the Read Point in the event will be very valuable for*
 1073 *higher level software.*

1074 *A supply chain like the rooms-and-doors example may be represented by a graph in which each*
 1075 *node in the graph represents a room in which objects may be found, and each arc represents a*
 1076 *doorway that connects two rooms. Business Locations, therefore, correspond to nodes of this*
 1077 *graph, and Read Points correspond to the arcs. If the graph were a straight, unidirectional*
 1078 *chain, the arcs traversed by a given object could be reconstructed from knowing the nodes; that*
 1079 *is, Read Point information would be redundant given the Business Location information. In more*
 1080 *real-world situations, however, objects can take multiple paths and move “backwards” in the*
 1081 *supply chain. In these real-world situations, providing Read Point information in addition to*
 1082 *Business Location information is valuable for higher level software.*

1083 7.3.5 The “Why” Dimension

1084 This section defines the data types used in the “Why” dimension of the event types specified in
 1085 Section 7.3.5.4.

1086 7.3.5.1 Business Step

1087 **BusinessStepID** is a vocabulary whose elements denote steps in business processes. An
 1088 example is an identifier that denotes “shipping.” The business step field of an event specifies the
 1089 business context of an event: what business process step was taking place that caused the event
 1090 to be captured? **BusinessStepID** is an example of a Standard Vocabulary as defined in
 1091 Section 6.2.

1092 *Explanation (non-normative): Using an extensible vocabulary for business step identifiers allows*
 1093 *GS1 standards (including and especially the GS1 Core Business Vocabulary) to define some*
 1094 *common terms such as “shipping” or “receiving,” while allowing for industry groups and*
 1095 *individual end-users to define their own terms. Master data provides additional information.*

1096 This specification defines no Master Data Attributes for business step identifiers.

1097 7.3.5.2 Disposition

1098 **DispositionID** is a vocabulary whose elements denote a business state of an object. An
 1099 example is an identifier that denotes “recalled.” The disposition field of an event specifies the
 1100 business condition of the event’s objects, subsequent to the event. The disposition is assumed to
 1101 hold true until another event indicates a change of disposition. Intervening events that do not