



1
2
3
4
5
6
7

GS1 Trusted Source of Data (TSD) 1.0 GS1 Standard

Ratified Standard, 20 December 2012

8 Document Summary

Document Item	Current Value
Document Title	GS1 Trusted Source of Data (TSD) 1.0 GS1 Standard
Date Last Modified	21 December 2012
Current Document Issue	Ratified Standard
Status	Ratified Standard
Document Description	

9 Contributors

Name	Organization
Marc Benhaim, working group co-chair	GS1 France
Carol Edison, working group co-chair	General Mills, Inc.
Steven Robba, working group co-chair	1WorldSync Holdings, Inc.
Dipan Anarkat, B2C working group facilitator	GS1 Global Office
Mark Frey, GSMP working group facilitator	GS1 Global Office
Coen Janssen, UML editor	GS1 Global Office
Ewa Iwicka, XML editor	GS1 Global Office
Ken Traub, editor-in-chief	Ken Traub Consulting LLC (for GS1 Global Office)
Kerry Angelo	GS1 Global Office
Craig Arseneau	Commport Communications Int'l Inc.
Andrea Ausili	GS1 Italy
Sandoche Balakrichenan	AFNIC
Martin Beno	GS1 Slovakia
Bob Bersani	GS1 Global Office
Robert Besford	GS1 UK
Joe Bohning	Nestle Purina
Rebone Boikanyo	GS1 South Africa
Malcolm Bowden	GS1 Global Office
Stefano Brentegani	GS1 Italy
Scott Brown	GS1 US
Le Caignec	About Goods Company
Sergi Cardona	GS1 Spain
Jean-Luc Champion	GS1 Global Office
Gilles Cherix	Icare Research Institute
Janet Chin	GS1 Canada
Richard Chresta	GS1 Switzerland

Name	Organization
Stephen Cole	Gladson Interactive
Edward Collins	Brandbank
Liz Crawford	GS1 Global Office
Janos Cyuris	GS1 Hungary
Henk Dannenberg	NXP Semiconductors
Jay Davies	Unilever UK
Kevin Dean	GS1 Canada
Ted Dickinson	1WorldSync Holdings, Inc.
Arne Dicks	GS1 Germany
Thomas Donzelle	Square SA
Marcel Ducceschi	Migros-Genossenschafts-Bund
Mark van Eegham	GS1 Global Office
Jonas Ekestam	GS1 Sweden
Ahmed el Kalla	GS1 Egypt
Daniel Eumana	GS1 Mexico
Sergey Fedoseev	GS1 Russia
Véra Feuerstein	Nestle
Stefan Gathmann	GS1 Ireland
Alexander Gerasimenko	Mars, Inc.
Johannes Gollowitzer	Mars, Inc.
Cameron Green	GS1 Global Office
Sudu Gupta	ITradeNetwork.com, Inc.
Pertti Hakala	GS1 Finland
Andrew Hearn	GS1 Global Office
Sally Herbert	GS1 Global Office
Tany Hui	GS1 Hong Kong
Alan Hyler	GS1 Global Office
Hideki Ichihara	GS1 Japan
Yoshihiko Iwasaki	GS1 Japan
Yohan Jeon	GS1 Korea
Peter Jönsson	GS1 Sweden
Kevia Kail	EPC Solutions
Sascha Kasper	1WorldSync Holdings, Inc.
Eric Kauz	GS1 Global Office
Robin Kidd	Nestle
Per Kiilsholm	GS1 Denmark
Lorraine Knight	1WorldSync Holdings, Inc.

Name	Organization
Bojan Kovacic	GS1 Slovenia
Jean-Luc Leblond	GS1 France
Sean Lockhead	GS1 Global Office
Andy Martin	GHX
Andrea McEntee	Nestle
Susie Mcintosh-Hinson	GS1 Global Office
Jeremy Morton	GS1 Sweden
Don Mowery	Nestle Purina PetCare
Daniel Mueller	GS1 Switzerland
Barbara Munro	Kraft Foods, Inc.
Doug Naal	Kraft Foods, Inc.
Ekaterina Nasonova	GS1 Russia
Katrín Nunez	GS1 US
Georgy Ogandzhanov	GS1 Russia
Staffan Olsson	GS1 Sweden
Joke Op den Acker	GS1 Belgium & Luxembourg
Michel Ottiker	GS1 Switzerland
Carlos Panos	GS1 Mexico
Nicolas Pauvre	GS1 France
Wim Peeters	GS1 Belgium & Luxembourg
Joao Picoito	GS1 Portugal
Xavier Pujol	GS1 Spain
Barry Pyle	Images in Space Ltd
Craig Alan Repec	GS1 Global Office
Rich Richardson	GS1 US
John Ryu	GS1 Global Office
Michael Sarachman	GS1 Global Office
Sjoerd Schaper	GS1 Netherlands
Armand Schins	Ahold (Europe)
Sue Schmid	GS1 Australia
Chris Schneider	GS1 Switzerland
Rene Schweinzger	GS1 Austria
Felipe Serrano Sanchez	GS1 Colombia
Andrey Sinitsyn	ItRuStore Ltd.
Joe Sloan	1WorldSync Holdings, Inc.
Olga Soboleva	GS1 Russia
Gabriel Sobrino	GS1 Netherlands

Name	Organization
Mike Spindler	ShelfSnap LLC
Sylvia Stein	GS1 Netherlands
Andrey Svistunov	ItRuStore Ltd.
Gina Tomassi	PepsiCo, Inc.
Betty Toth	ipiit
Vyacheslav Turchaninov	ItRuStore Ltd.
Milan Vacval	Gladson Interactive
Grigore Vlad	Xyxle AG
Mark Widman	EPC Solutions
Mary Wilson	GS1 US
Tony Zhang	FSE, Inc.

10 **Log of Changes in Ratified Standard**

Issue No.	Date of Change	Changed By	Summary of Change

11 **Disclaimer**

12 Whilst every effort has been made to ensure that the guidelines to use the GS1 standards contained in the
 13 document are correct, GS1 and any other party involved in the creation of the document HEREBY STATE that
 14 the document is provided without warranty, either expressed or implied, of accuracy or fitness for purpose, AND
 15 HEREBY DISCLAIM any liability, direct or indirect, for damages or loss relating to the use of the document. The
 16 document may be modified, subject to developments in technology, changes to the standards, or new legal
 17 requirements. Several products and company names mentioned herein may be trademarks and/or registered
 18 trademarks of their respective companies.

19

20

21

22

23

Table of Contents

24 **Introduction 8**

25 **1. Scope 8**

26 **2. References 8**

27 **3. Terms and definitions 9**

28 3.1. General Business Terms..... 9

29 3.2. Architecture Terms for This Standard 10

30 **4. Business Background (non-normative) 11**

31 4.1. Business Intention 11

32 4.2. Business Justification 11

33 **5. Architecture 11**

34 5.1. Components and Interfaces 12

35 5.2. Interaction Scenarios 13

36 5.2.1. Scenario 1: Data Delivered Directly from Local Data Aggregator 13

37 5.2.2. Scenario 2: Data Delivered from Peer Data Aggregator 14

38 5.3. Specification Layers 15

39 5.4. Versioning 16

40 5.4.1. Versioning of Product Data Structure 16

41 5.4.2. Versioning of REST Interfaces 17

42 5.5. Conformance 17

43 **6. Product Data – Abstract Definition 17**

44 6.1. Modular Structure 17

45 6.2. Common Data Types 18

46 6.2.1. GTIN 18

47 6.2.2. CountryCode 18

48 6.2.3. LanguageCode 18

49 6.2.4. ServiceReference 18

50 6.2.5. GLN 18

51 6.2.6. Description70, Description80, Description2500 19

52 6.2.7. Measurement 19

53 6.2.8. AttributeValuePairList 19

54 6.3. Product Data 20

55 6.4. Product Data Module Types 22

56 6.4.1. Basic Product Information 22

57 6.4.2. Nutritional Product Information 26

58 **7. Interfaces – Abstract Definition 30**

59	7.1.	Aggregator-Aggregator Query Interface (AAQI) – Abstract Definition	30
60	7.1.1.	AAQI queryByGtin Operation	32
61	7.2.	Aggregator-Index Query Interface (AIQI) – Abstract Definition	34
62	7.2.1.	AIQI queryByGtin	36
63	7.3.	Aggregator-Index Maintenance Interface (AIMI) – Abstract Definition	37
64	7.3.1.	AIMI Index Maintenance Request	39
65	7.3.2.	AIMI Index Maintenance Response	39
66	8.	XML Schemas	41
67	8.1.	Common Data Types XML Schema	42
68	8.1.1.	Common Data Types XML Schema – Shared	42
69	8.1.2.	Common Data Types XML Schema – TSD-specific	45
70	8.2.	Product Data XML Schema.....	47
71	8.2.1.	Product Data XML Schema – Top Level.....	47
72	8.2.2.	Product Data XML Schema – Basic Product Information Module	48
73	8.2.3.	Product Data XML Schema – Nutrition Product Information Module.....	49
74	8.3.	Interface Messages XML Schema	50
75	8.3.1.	Aggregator-Aggregator Query Interface (AAQI) XML Schema.....	50
76	9.	Transport Bindings	52
77	9.1.	Web Service Bindings	52
78	9.1.1.	REST Web Service Bindings.....	52
79	9.2.	ONS 2.0 Binding for Aggregator-Index Query Interface (AIQI).....	56
80	9.2.1.	AIQI Query Via ONS	56
81	9.2.2.	NAPTR Records for TSD	57
82			
83			

List of Figures

84			
85	Figure 1.	Trusted Source of Data Framework Architecture	12
86	Figure 2.	Interaction Scenarios in TSD Framework.....	13
87	Figure 3.	TSD Framework Standard Layers	15
88			
89			
90			

91

Introduction

92
93
94
95
96
97
98

This document is a GS1 Standard which is the normative specification for standard data and interfaces in the Trusted Source of Data (TSD) framework. The aim of the TSD framework to support the communication of authentic and accurate product data by brand owners to consumers/shoppers, retailers, internet applications, and government using internet and mobile devices. This standard defines the data and interfaces by which one or more Data Aggregators (as defined herein) may federate to provide seamless access to product data, for the benefit of Internet Applications across the globe.

99

1. Scope

100

This standard defines the following:

101
102

- The architectural framework within which the roles of Internet Application, Data Aggregator, and Global Index are defined. (See Sections 3 and 4.)

103
104

- Abstract data definitions for product data delivered by Data Aggregators to Internet Applications.

105

- Abstract interface definitions for message exchange, including:

106
107

- Aggregator-Aggregator Query Interface (AAQI) One Data Aggregator querying another for data about a specified product (or products)

108
109
110

- Aggregator-Index Query Interface (AIQI) A Data Aggregator querying the Global Index to obtain references to other Data Aggregators who may have data about a specified product (or products)

111
112
113

- Aggregator-Index Maintenance Interface (AIMI) A Data Aggregator interacting with the Global Index to add, change, correct, or delete index entries that reference the Data Aggregator

114
115

- XML schemas that implement the abstract data definitions and the messages implied by the abstract interface definitions

116
117

- Bindings of the abstract interface definitions to specific transport protocols and associated security mechanisms, specifically:

118
119

- A web services binding for the Aggregator-Aggregator Query Interface (AAQI). This binding makes use of the XML schemas for its data payloads.

120

- A binding for the Aggregator-Index Query Interface (AIQI) based on ONS 2.0 [ONS_20]

121

This version of the standard does not define bindings for the AIMI.

122
123

Specifically out of scope of this standard is the interface between Internet Applications and Data Aggregators.

124

2. References

125

Normative references:

126

- [GS1GS] GS1, "GS1 General Specifications Version 12," January 2012.

- 127
128
129
- [INFOODS] Klensin et al, "Identification of Food Components for INFOODS Data Interchange," Tokyo: United Nations University, 1989, with updates through June 2007. http://www.fao.org/infoods/tagnames_en.stm
- 130
131
- [ISO3166] "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes," ISO 3166-1:2006.
- 132
133
- [ISO639] "Codes for the representation of names of languages -- Part 1: Alpha-2 code," ISO 639-1:2002.
- 134
135
- [ISODir2] ISO/IEC Directives part 2; Rules for the structure and drafting of International Standards – 6th edition, 2011
- 136
- [ONS_20] GS1 Object Naming Service 2.0, draft standard (not yet published).
- 137
138
- [RFC2246] T. Dierks, C. Allen, "The TLS Protocol, Version 1.0," RFC2246, January 1999, <http://www.ietf.org/rfc/rfc2246>.
- 139
140
- [RFC2818] E. Escorla, "HTTP Over TLS," RFC2818, May 2000, <http://www.ietf.org/rfc/rfc2818>.
- 141
142
- [RFC3268] P. Chown, "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)," RFC3268, June 2002, <http://www.ietf.org/rfc/rfc3268>.
- 143
144
- [RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC3986, January 2005, <http://www.ietf.org/rfc/rfc3986>.
- 145
146
147
- [UNECE20] United Nations Economic Commission for Europe, "Recommendation No. 20: Codes for Units of Measure Used in International Trade, Revision 7," September 2010, http://www.unece.org/fileadmin/DAM/cefact/recommendations/rec20/rec20_Rev7e_2010.zip.

148 3. Terms and definitions

149 Within this specification, the terms SHALL, SHALL NOT, SHOULD, SHOULD NOT, MAY, NEED NOT,
150 CAN, and CANNOT are to be interpreted as specified in Annex G of the ISO/IEC Directives, Part 2,
151 2001, 4th edition [ISODir2]. When used in this way, these terms will always be shown in ALL CAPS;
152 when these words appear in ordinary typeface they are intended to have their ordinary English
153 meaning.

154 All sections of this document, with the exception of the introduction, are normative, except where
155 explicitly noted as non-normative.

156 The following typographical conventions are used throughout the document:

- 157
- ALL CAPS type is used for the special terms from [ISODir2] enumerated above.
- 158
- `Monospace` type is used to denote programming language, UML, and XML identifiers, as well
159 as for the text of XML documents.
- 160
- Placeholders for changes that need to be made to this document prior to its reaching the final
161 stage of approved GS1 specification are prefixed by a rightward-facing arrowhead, as this
162 paragraph is.

163 For the purposes of this document, the following terms and definitions apply.

164 3.1. General Business Terms

165 Trade Item

166 In the context of this standard a Trade Item is defined as a product available for sale at a retail outlet
167 or at an online store.

168 3.2. Architecture Terms for This Standard

169 **Aggregator-Index Maintenance Interface (AIMI)**

170 The interface through which a Data Aggregator interacts with the Global Index to add, change, correct,
171 or delete index entries that reference the Data Aggregator.

172 **Aggregator-Aggregator Query Interface (AAQI)**

173 The interface through which one Data Aggregator queries another for data about a specified product
174 (or products).

175 **Aggregator-Index Query Interface (AIQI)**

176 The interface through which a Data Aggregator queries the Global Index to obtain references to other
177 Data Aggregators who may have data about a specified product (or products).

178 **Bindings**

179 The part of this standard that specifies concrete realizations of the Product Data Layer and the Service
180 Interface Layer.

181 **Data Aggregator**

182 A service that gathers or maintains data regarding products, and makes this data available to Internet
183 Applications. Each Data Aggregator is not expected to have data for all products, but instead interacts
184 with peer Data Aggregators as necessary to create a federated system.

185 **Data Pool**

186 A data pool in the GS1 Global Data Synchronization Network. A Data Pool is one source of data that a
187 Data Aggregator draws from.

188 **Global Index**

189 A service that a Data Aggregator uses to identify peer Data Aggregators who have data about a given
190 product, in the case where the first Data Aggregator does not itself have data about that product.

191 **Internet Application**

192 An application or service that is available to consumers via an Internet-based platform (desktop or
193 mobile)

194 **Product Data Layer**

195 The part of this standard that specifies the data used to describe a Trade Item in the context of the
196 TSD framework.

197 **Service Interface Layer**

198 The part of this standard that defines the three service interfaces: the Aggregator-Aggregator Query
199 Interface (AAQI), the Aggregator-Index Query Interface (AIQI), and the Aggregator-Index Maintenance
200 Interface (AIMI).

201 **4. Business Background (non-normative)**

202 The aim of the GS1 Business-to-Consumer Trusted Source of Data (TSD) framework is to support the
203 communication of authentic and accurate product data by brand owners to consumers/shoppers,
204 retailers, internet applications, and government using internet and mobile devices.

205 The vision of the TSD framework is that:

- 206 ■ Brand owners can share relevant product information easily, thus building trust with
207 consumers.
- 208 ■ Internet Applications can ensure they are delivering authentic data.
- 209 ■ Consumers can feel confident that the digital product information they access is accurate, no
210 matter how or where they shop.

211 In the envisioned solution, Internet Applications will interface with Data Aggregators to obtain trusted
212 data regarding products identified with GS1 Identification Keys (namely, the Global Trade Item
213 Number). Each Data Aggregator will draw upon a variety of sources for this data, including the GS1
214 Global Data Synchronization Network (GDSN). Brand owners supply trusted data to the Data
215 Aggregators through GDSN or other means.

216 **4.1. Business Intention**

217 The objective is to facilitate the emergence of a system of federated Data Aggregators, each capable
218 of delivering trusted data about products to Internet Applications. Each Data Aggregator will draw
219 upon a variety of sources for this data, including the GS1 Global Data Synchronization Network
220 (GDSN). Brand owners supply trusted data to the Data Aggregators through GDSN or other means.

221 It is anticipated that no one Data Aggregator will house data for all products. Therefore Data
222 Aggregators will be federated by means of a Global Index, allowing each Data Aggregator to efficiently
223 fetch missing product records from peer Data Aggregators.

224 **4.2. Business Justification**

225 The following business benefits are expected to be realized from the TSD framework:

- 226 ■ Consumers have accurate information (up-to-date, right product, consumer trust/confidence).
- 227 ■ Consumers are prevented from getting no information, inaccurate information or maliciously
228 falsified information on a product.
- 229 ■ Consumers have relevant information, tailored to their preferences (profile) if needed.
- 230 ■ Brands have an easy way to provide content, either by themselves or in cooperation with
231 certified third parties.
- 232 ■ Solution providers have an easy way to access content and show that it has been brand-
233 authorised.

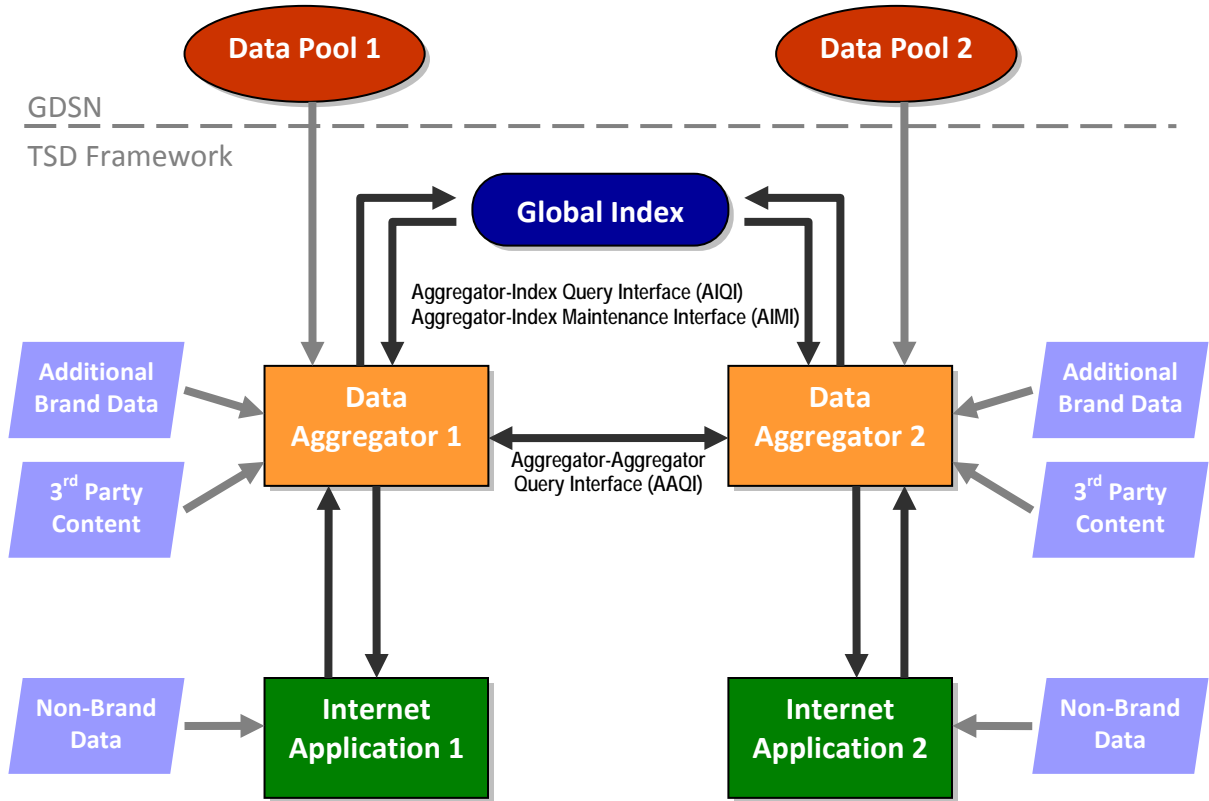
234 **5. Architecture**

235 This section defines the architecture of the Trusted Source of Data framework, providing the
236 foundation for the normative definitions of data and interfaces specified in later sections.

237 **5.1. Components and Interfaces**

238 The Trusted Source of Data framework is based upon an architecture having the following
 239 relationships between networked computer systems:

240 **Figure 1. Trusted Source of Data Framework Architecture**



241 This architecture includes the following components:
 242

- 243 ■ *Internet Application (one or more)* An application or service that is available to consumers
 244 via an Internet-based platform (desktop or mobile), and which queries a Data Aggregator to
 245 obtain product information
- 246 ■ *Data Aggregator (one or more)* A service that gathers or maintains data regarding products,
 247 and makes this data available to Internet Applications. Each Data Aggregator is not expected
 248 to have data for all products, but instead interacts with peer Data Aggregators as necessary to
 249 create a federated system.
- 250 ■ *Global Index (one)* A service that a Data Aggregator uses to identify peer Data Aggregators
 251 who have data about a given product, in the case where the first Data Aggregator does not
 252 itself have data about that product. Logically, there is one Global Index that serves all
 253 aggregators in the TSD framework; physically, this may be implemented by multiple services
 254 that are federated.
- 255 ■ *Data Pool (one or more)* A data pool in the GS1 Global Data Synchronization Network. A
 256 Data Pool is the primary source of data that a Data Aggregator draws from.

257 The implementation of any given component is out of scope for this standard. This standard only
 258 defines the interfaces to which certain components must conform in order to participate in the TSD
 259 framework.

260 The scope of this standard is to define the following interfaces:

261
262

- *Aggregator-Aggregator Query Interface (AAQI)* The interface through which one Data Aggregator queries another for data about a specified product (or products).

263
264
265

- *Aggregator-Index Query Interface (AIQI)* The interface through which a Data Aggregator queries the Global Index to obtain references to other Data Aggregators who may have data about a specified product (or products).

266
267
268

- *Aggregator-Index Maintenance Interface (AIMI)* The interface through which a Data Aggregator interacts with the Global Index to add, change, correct, or delete index entries that reference the Data Aggregator.

269
270

Out of scope of this standard are the interactions between Data Aggregators and Internet Applications, and the interactions between Data Aggregators and GDSN Data Pools.

271

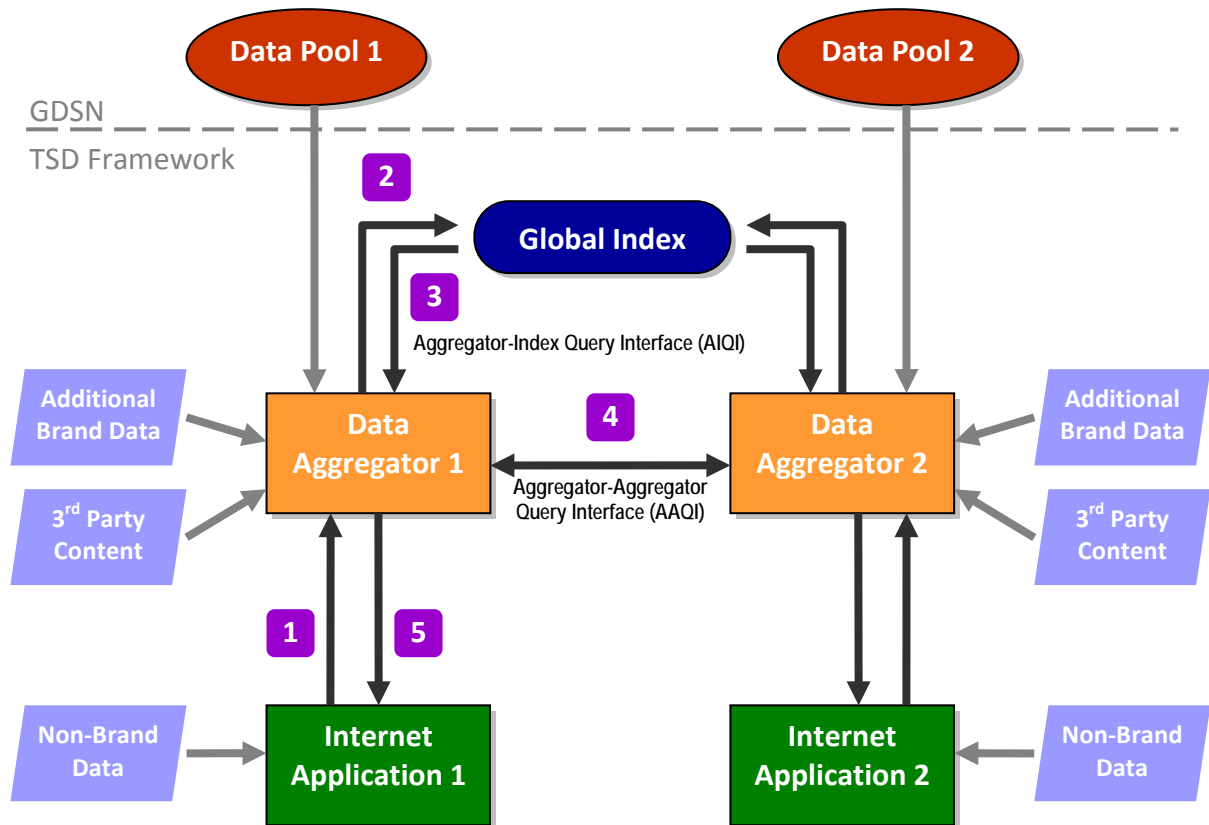
5.2. Interaction Scenarios

272
273
274

This section outlines two typical interactions in the TSD framework, to illustrate the collaboration between system components using the interfaces defined in this standard. Both scenarios make reference to the following figure:

275

Figure 2. Interaction Scenarios in TSD Framework



276

277

5.2.1. Scenario 1: Data Delivered Directly from Local Data Aggregator

278
279

In this scenario, an Internet Application queries a Data Aggregator for product data for a specified product, and the Data Aggregator is able to satisfy the request using data it has locally.

280

Goal: To supply an Internet Application with product data for a specified product

281 Pre-condition: Internet Application and Data Aggregator have established a trust relationship so that
 282 they can mutually authenticate each other. Internet Application is authorized to query the Data
 283 Aggregator. Data Aggregator has the data for the specified product.

284 Post-condition: Internet Application has the requested data.

285 Main flow:

- 286 1. (Figure 2, label “1”) Internet Application issues a query to the Data Aggregator, including a
 287 specification of what product information is desired (e.g., specifying the GTIN and Target Market in
 288 case of a query by GTIN).
- 289 2. Data Aggregator retrieves the requested data from its local sources of data.
- 290 3. (Figure 2, label “5”) Data Aggregator responds to the Internet Application, providing the requested
 291 product data.

292 As this scenario only involves interaction between the Internet Application and the Data Aggregator,
 293 and the interactions between Internet Applications and Data Aggregators are out of scope for this
 294 standard (see Sections 1 and 5.1), this standard is not used at all in the carrying out of this scenario.

295 5.2.2. Scenario 2: Data Delivered from Peer Data Aggregator

296 In this scenario, an Internet Application queries a Data Aggregator for product data for a specified
 297 product, but the Data Aggregator is unable to satisfy the request using data it has locally, so it must
 298 interact with the Global Index and one or more peer Data Aggregators.

299 Goal: To supply an Internet Application with product data for a specified product

300 Pre-condition: Internet Application and Data Aggregator have established a trust relationship so that
 301 they can mutually authenticate each other. Internet Application is authorized to query the Data
 302 Aggregator. Data Aggregator has established trust relationships with the Global Index and peer Data
 303 Aggregators. Brand Owners have previously registered index entries with the Global Index for
 304 relevant products (a Brand Owner may authorize a Data Aggregator to perform index maintenance on
 305 its behalf)..

306 Post-condition: Internet Application has the requested data.

307 Main flow:

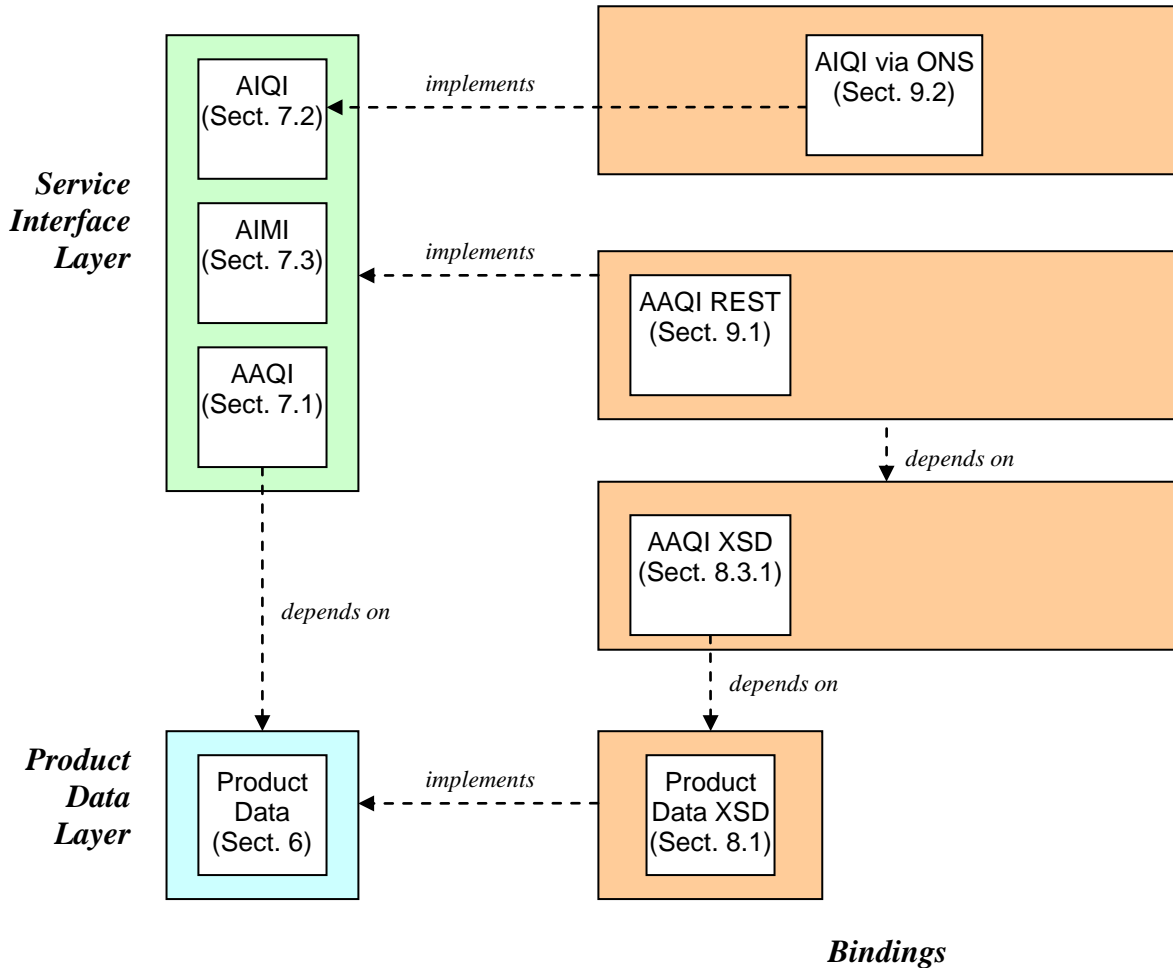
- 308 1. (Figure 2, label “1”) Internet Application issues a query to the Data Aggregator, including a
 309 specification of what product information is desired (e.g., specifying the GTIN and Target Market in
 310 case of a query by GTIN).
- 311 2. Data Aggregator finds that it does not have local data to satisfy the query.
- 312 3. (Figure 2, label “2”) Data Aggregator queries the Global Index to identify peer Data Aggregators
 313 that have relevant data.
- 314 4. (Figure 2, label “3”) Global Index responds with references to one or more Data Aggregators that
 315 have relevant data.
- 316 5. (Figure 2, label “4”) Data Aggregator queries peer Data Aggregator(s) to obtain relevant data.
- 317 6. (Figure 2, label “4”) Peer Data Aggregator(s) respond with relevant data.
- 318 7. (Figure 2, label “5”) Data Aggregator responds to the Internet Application, providing the requested
 319 product data assembled from data received from peer Data Aggregators.

320 Steps 3 and 4 of this flow (Figure 2, labels “2” and “3”) are governed by the Aggregator-Index Query
 321 Interface (AIQI), specified in Section 7.2. Steps 5 and 6 of this flow (Figure 2, label “4”) are governed
 322 by the Aggregator-Aggregator Query interface (AAQI), specified in Section 7.1.

323 **5.3. Specification Layers**

324 The TSD framework standard is organized into several layers, illustrated below.

325 **Figure 3. TSD Framework Standard Layers**



326
327 These layers are described below.

- 328 ■ **Product Data Layer** The Product Data Layer specifies the data exchanged through the
329 Aggregator-Aggregator Query Interface: its abstract structure, and what it means.
- 330 ■ **Service Interface Layer** The Service Interface Layer defines the three service interfaces
331 defined in Section 5.1: the Aggregator-Aggregator Query Interface (AAQI), the Aggregator-
332 Index Query Interface (AIQI), and the Aggregator-Index Maintenance Interface (AIMI). This
333 layer defines the abstract structure of these interfaces (what operations are contained within
334 each interface), the abstract content of messages communicated in these interfaces to carry
335 out the operations, and the semantics of each operation.
- 336 ■ **Bindings** Bindings specify concrete realizations of the Product Data Layer and the Service
337 Interface Layer. In principle, many bindings may be defined for any given data definition or
338 service interface. In this standard, an XML binding is given for the Product Data Layer; that
339 is, an XML schema that concretely realizes the abstract content of product data specified in
340 the Product Data Layer. This standard also defines an XML binding for the messages in the
341 AAQI, and this XML binding is used in turn to define a binding of the AAQI to a REST-style

342 web service over HTTP. A binding for the AIQI is given in terms of ONS 2.0, which does not
343 use XML. In this version of the standard, no bindings of the AIMI are specified. The abstract
344 definition is provided as a guide for Data Aggregators and Global Index implementations that
345 wish to provide a proprietary (non-standard) implementation of the AIMI. Concrete bindings of
346 AIMI may be specified in a future version of this standard.

347 Taken together, these layers are a complete specification of the data and interfaces within scope of
348 this standard.

349 5.4. Versioning

350 The following mechanisms are provided to allow for evolution of this standard.

351 5.4.1. Versioning of Product Data Structure

352 The response to an AAQI query for product data is a `ProductData` structure as specified in
353 Sections 6 and 8.1. Versioning of `ProductData` in this standard is designed to align with GS1 Global
354 Data Synchronization Network (GSDN) standards, and provides for the following types of changes:

- 355 ■ *Major Version* A new `ProductData` major version introduces a new XML schema which is
356 not necessarily forward nor backward compatible with previous versions.
- 357 ■ *Minor Version* A new `ProductData` minor version introduces a new XML schema which is
358 backwards compatible with earlier minor versions within the same major version, though not
359 necessarily forward compatible. Backwards compatibility means that any data that is valid
360 according to an older minor version is still valid in the new minor version and has the same
361 meaning. Data that is valid in the new minor version may include new data elements that are
362 not present in older minor versions.
- 363 ■ *Temporary Attributes* Between minor versions, it is possible to include additional attributes
364 not defined in the XML schema for `ProductData`. Such attributes are included using a
365 generic attribute-value-pair mechanism provided for in the XML schema. Attributes introduced
366 in this way are resolved in the next version. When a temporary attribute is introduced, any
367 Data Aggregator implementing the temporary attribute SHALL include it in `ProductData` it
368 sends to other Data Aggregators.

369 A major or minor version is indicated by a version number. An AAQI query indicates the desired
370 version number for data in the response. Normally, a Data Aggregator SHALL respond affirmatively to
371 a query that specifies the current version and MAY respond affirmatively or reject a request specifying
372 a different version. During a defined period of transition between an old version and a new version, a
373 Data Aggregator SHALL respond affirmatively to a request specifying the old version and MAY
374 respond affirmatively to a request specifying the new version or any other version.

375 During the defined period of transition, a querying Data Aggregator behaves as follows:

- 376 ■ If the querying Data Aggregator has not yet been upgraded to support the new version, it
377 always issues a query specifying the old version. Because all Data Aggregators support
378 queries for the old version during the transition period, they can all respond affirmatively.
- 379 ■ If the querying Data Aggregator has been upgraded to support the new version, it first issues
380 a query specifying the new version. If the peer Data Aggregator responds negatively, the first
381 Data Aggregator reissues the query specifying the old version. In this way, the upgraded
382 Data Aggregator can interoperate with peers regardless of whether they have been upgraded
383 yet.

384 5.4.2. Versioning of REST Interfaces

385 The REST interfaces for AAQI and AIMI use the following mechanisms to provide versioning for the
386 request URL:

- 387 ■ *Compatible Change* New operations and new query parameters may be added to a REST
388 interface by a new version of TSD. The base Service URL and service URL version number
389 remains the same.
- 390 ■ *Incompatible Change* A REST interface may be changed incompatibly by updating the
391 version string embedded in the request URL (see Section 9.1.1). During a period of transition,
392 both the old and new URLs may be implemented simultaneously.

393 5.5. Conformance

394 The following types of implementation artefacts may conform to this standard.

- 395 ■ *Data Aggregator* A Data Aggregator implementation is in conformance to this standard if all
396 of the following are true:
 - 397 □ The implementation conforms to all normative statements in this standard that pertain to
398 the Aggregator-Aggregator Query Interface (AAQI), both the querying side and the
399 responding side, with respect to at least one binding of that interface.
 - 400 □ The implementation conforms to all normative statements in this standard that pertain to
401 the client side of the Aggregator-Index Query Interface (AIQI), with respect to at least one
402 binding of that interface.
- 403 ■ *Global Index* A Global Index implementation is in conformance to this standard if the
404 following is true:
 - 405 □ The implementation conforms to all normative statements in this standard that pertain to
406 the service side of the Aggregator-Index Query Interface (AIQI), with respect to at least
407 one binding of that interface.

408 In addition to conformance to this standard, Data Aggregator implementations and Global Index
409 implementations may be subject to service level agreements (SLAs) in order to participate in the GS1
410 TSD framework. Such SLAs are defined elsewhere.

411 Note that because the scope of this standard does not include any interfaces to which an Internet
412 Application is a party, it is not valid to assert that an Internet Application is either in conformance or out
413 of conformance to this standard. Individual Data Aggregator implementations may, however, choose
414 to reuse the Product Data Layer of this standard in defining their own proprietary interfaces to Internet
415 Applications, and in such cases a Data Aggregator may reference this standard in defining its own
416 proprietary conformance criteria for its Internet Applications.

417 6. Product Data – Abstract Definition

418 This section defines product data that is communicated from one Data Aggregator to another in the
419 Aggregator-Aggregator Query Interface (AAQI). This section defines the data content and meaning at
420 an abstract level; see Section 8.1 for a concrete realization of the data content as an XML schema.

421 6.1. Modular Structure

422 Product data is defined in several modules. This standard defines two product data modules:

- 423 ■ *Basic Product Information* This module contains common identifying information for a
424 product. This data is intended to be suitable to assist an end user in understanding to which
425 product a given data set pertains.

426 ■ *Nutritional Product Information* This module contains data pertaining to the nutritional
427 content of a food product, including those data element that are of interest to consumers, are
428 commonly available on the product label, and are defined in the Global Data Synchronization
429 Network (GDSN).

430 Future versions of this standard may add additional modules.

431 A given instance of product data provided by a Data Aggregator describes a single product (single
432 GTIN) as sold in a single target market. The product data for a given GTIN + target market
433 combination SHALL conform to the following:

- 434 ■ The product data SHALL include Basic Product Information as specified in Section 6.4.1.
- 435 ■ The product data MAY include Nutritional Product Information. If it does, it SHALL be as
436 specified in Section 6.4.2.

437 The UML diagram in Section 6.3 expresses the high level structure of product data.

438 **6.2. Common Data Types**

439 This section defines data types used by all product data modules and interfaces.

440 **6.2.1. GTIN**

441 The `GTIN` type represents a Global Trade Item Number (GTIN). The `GTIN` type is a 14-character
442 string which may contain a GTIN-8, GTIN-12, GTIN-13, or GTIN-14 as defined in the GS1 General
443 Specifications [GS1GS]. When the `GTIN` type holds a GTIN-8, GTIN-12, or GTIN-13, the GTIN value
444 is padded on the left with zeros to make 14 characters total, as illustrated in [GS1GS, Section 3.3.2].

445 **6.2.2. CountryCode**

446 The `CountryCode` type is a 3-character numeric string that identifies a country, using the 3-digit
447 country codes defined by ISO 3166-1 numeric [ISO3166].

448 The `CountryCode` type is used in the TSD standard to denote a target market for a GTIN.

449 **6.2.3. LanguageCode**

450 The `LanguageCode` type is a 2-character string that identifies a human language, using the 2-
451 character language codes defined by ISO 639-1 [ISO639]. Note that both characters of an ISO 639-1
452 code are lowercase.

453 **6.2.4. ServiceReference**

454 A `ServiceReference` is used in the response to a Global Index query to refer to a specific Data
455 Aggregator service. A `ServiceReference` is an absolute HTTP URL that is used as the base URL
456 in an AAQI query (see Section 9.1.1).

457 **6.2.5. GLN**

458 The `GLN` type represents a Global Location Number (GLN). The `GLN` type is a 13-character string.

459 **6.2.6. Description70, Description80, Description2500**

460 The types `Description70`, `Description80`, and `Description2500` are a pair consisting of a
 461 text string and a `LanguageCode` (Section 6.2.3). The maximum length of the text string is 70, 80, or
 462 2500 characters, respectively.

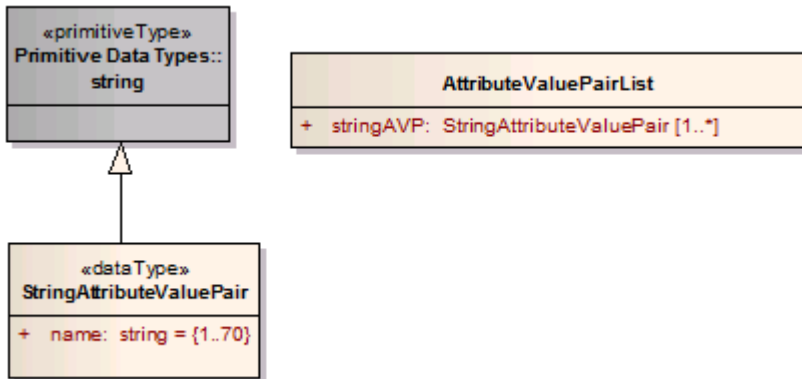
463 These types are used within product data where a human-readable string may be expressed in more
 464 than one language. In each such place more than one element of the chosen type is permitted, to
 465 allow for a description to be provided in multiple languages.

466 **6.2.7. Measurement**

467 The `Measurement` type is a pair consisting of a value of type `Float` and a unit of measure of type
 468 `String`. The value of the unit of measure SHALL be one of the units of measure specified in UN/ECE
 469 Recommendation 20 [UNECE20], specifically the string defined in the “Common Code” column of
 470 [UNECE20] for the selected unit.

471 **6.2.8. AttributeValuePairList**

472 Attribute-value pair lists are used to provide a means to introduce temporary attributes (see
 473 Section 5.4.1) into product data between minor versions. The following UML diagram specifies the
 474 structure of an attribute-value pair list:

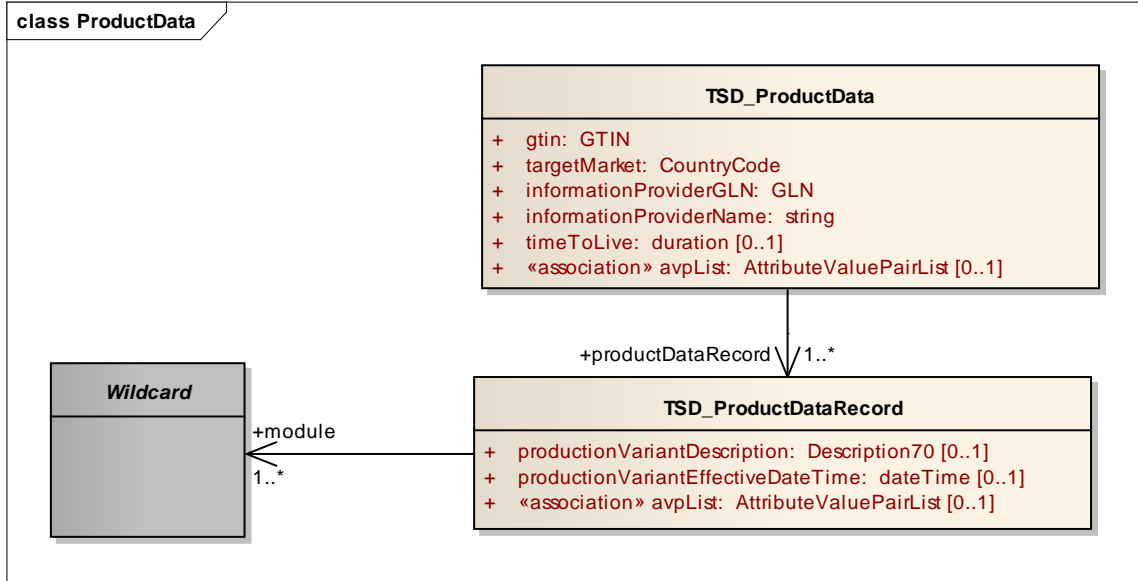


475
 476 An attribute value pair list consists of one or more attribute-value pairs, each of which contains a name
 477 and a value.

478 To avoid clutter in UML diagrams in this standard, references to attribute value pair lists are shown as
 479 UML attributes rather than associations; that is, a class containing an attribute value pair list includes a
 480 UML attribute of type `AttributeValuePairList` rather than showing an arrow to the
 481 `AttributeValuePairList` class. The UML stereotype `<<association>>` is used to mark this
 482 deviation from normal GS1 UML design rules. It has no effect on the XML schemas.

483 **6.3. Product Data**

484 The following UML diagram expresses the data content of Product Data. This is the structure that one
 485 Data Aggregator sends to another in response to a query issued via the Aggregator-Aggregator Query
 486 Interface (AAQI). See Section 7.1.



487
 488 The data content of a TSD_ProductData structure SHALL be as follows:

Data Element	Type	Cardinality	Description
gtin	GTIN (Section 6.2.1)	1	The GTIN of the product described by this TSD_ProductData instance.
targetMarket	CountryCode (Section 6.2.2)	1	Target market to which this TSD_ProductData instance applies, expressed as a CountryCode (Section 6.2.2).
informationProviderGLN	GLN (Section 6.2.5)	1	Party Global Location Number (GLN) of the party providing this data
informationProviderName	String	1	Human-readable name of the party providing this data.

Data Element	Type	Cardinality	Description
timeToLive	Duration	0..1	If present, indicates that the data is only valid within the specified duration after receipt by the requesting Data Aggregator.
productDataRecord	TSD_ProductDataRecord (below)	1..*	Product data records describing the product identified by the specified GTIN for the specified target market. If more than one product data record is included, each describes a different variant of the product.
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

489

The data content of a TSD_ProductDataRecord SHALL be as follows:

Data Element	Type	Cardinality	Description
productionVariant Description	Description70 (Section 6.2.6)	0..1	Free text assigned by the manufacturer to describe the production variant. Examples are: package series X, package series Y.
productionVariant EffectiveDateTime	DateTime	0..1	The start date of a production variant.
module	Product data module type; one or more of the types specified in Section 6.4	1..*	Product data modules. The first module SHALL be the basic product information module. Other modules are optional. This list SHALL contain at most one module of each module type.
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

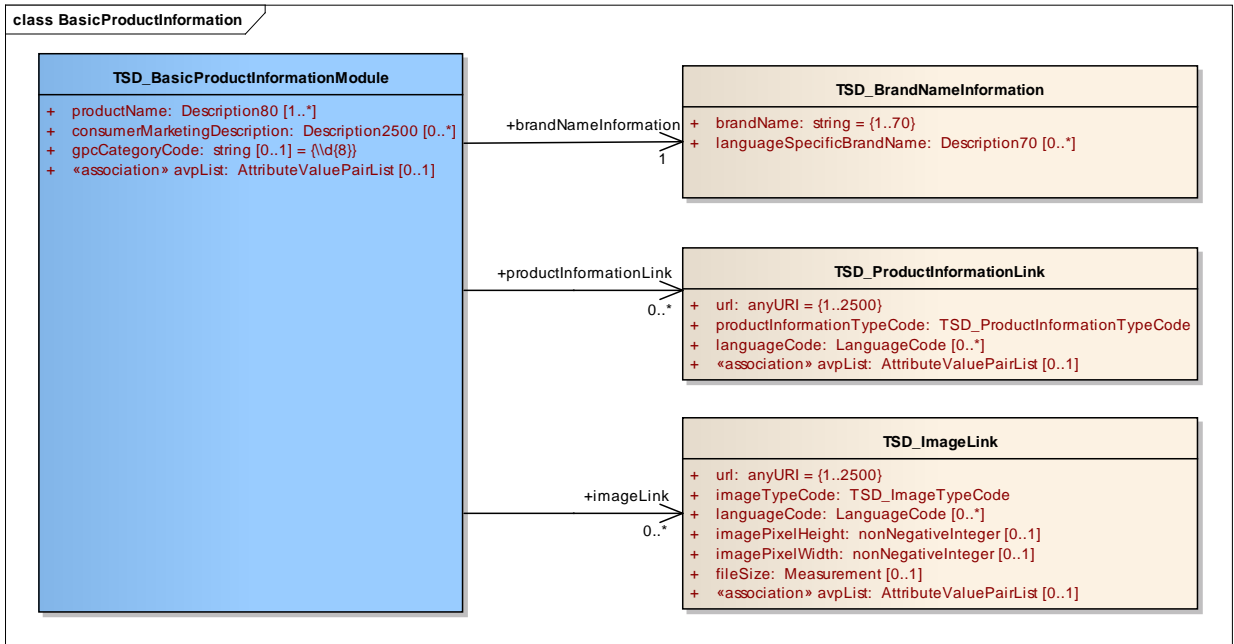
490

491 **6.4. Product Data Module Types**

492 This section specifies product data module types that may occur in the modules list of a ProductData
 493 structure (Section 6.3).

494 **6.4.1. Basic Product Information**

495 The following UML diagram expresses the data content of the Basic Product Information module.



496 The data content of TSD_BasicProductInformationModule SHALL be as specified in the
 497 following table.
 498

Data Element	Type	Cardinality	Description
productName	Description80 (Section 6.2.6)	1..*	Consumer friendly short description of the product suitable for compact presentation. Each Description80 structure SHALL have a different language code, and SHALL represent presentations of the same value in different languages.
consumerMarketingDescription	Description2500 (Section 6.2.6)	0..*	Consumer-friendly marketing detailed description of the product. Each Description2500 structure SHALL have a different language code, and SHALL represent presentations of the same value in different languages.

Data Element	Type	Cardinality	Description
gpcCategoryCode	String	0..1	8-digit code specifying a product category according to the GS1 Global Product Classification (GPC) standard
brandNameInformation	TSD_BrandName Information (below)	1	The brand name of the product.
productInformation Link	TSD_ProductInformation Link (below)	0..*	URLs linking to product information provided by the brand owner.
imageLink	TSD_ImageLink (below)	0..*	Images provided by the brand owner
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

499

The data content of a TSD_BrandNameInformation structure SHALL be as follows:

Data Element	Type	Cardinality	Description
brandName	String	1	The brand name of the product that appears on the consumer package.
languageSpecific BrandName	Description70 (Section 6.2.6)	0..*	An alternate form of the brand name for a specified language.

500

The data content of a TSD_ProductInformationLink structure SHALL be as follows:

Data Element	Type	Cardinality	Description
url	AnyURI (up to 2500 characters)	1	Uniform Resource Locator (URL) that references a World-Wide Web resource providing information about the product. The value of url SHALL be an absolute URL with or without a fragment identifier, as specified in [RFC3986]; relative URL references SHALL NOT be used.
productInformation TypeCode	TSD_ProductInformation TypeCode	1	Code specifying the type of product information available at the specified url.

Data Element	Type	Cardinality	Description
languageCode	LanguageCode	0..*	Language codes specifying the language(s) for which this URL applies, or omitted if this URL applies to all languages
avpList	AttributeValuePairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

501

The value of `productInformationTypeCode` SHALL be one of the following:

Value	Description
AUDIO	Link to a file containing an audio clip which is relevant to the product. Examples are commercials, or instructional/how to use audio files.
CONSUMER_HANDLING_AND_STORAGE	Link to website, file, or image containing the Manufacturer's recommendations for how the consumer or end user should store and handle the product.
DOCUMENT	Link to a document or text file containing product information. Examples of this type could be an instruction manual, assembly guide, or warranty document.
IFU	Link to a file containing the Instructions For Use (IFU). This type of file is primarily used in conjunction with ingestible products, regardless of species.
MARKETING_INFORMATION	Link to a file with product information associated with selling a product or service.
OTHER_EXTERNAL_INFORMATION	Link to a file containing product information of an unspecified type.
VIDEO	Link to a file containing a video clip which is relevant to the product. Examples are commercials, trailers, or instructional/how to use video files.
WARRANTY_INFORMATION	Link to a file with information associated with any guarantee given by a company stating that a product is reliable and free from known defects and that the seller will, without charge, repair or replace defective parts within a given time limit and under certain conditions.
WEBSITE	Link to a website containing product or manufacturer information.

502

503

The data content of an TSD_ImageLink structure SHALL be as follows:

Data Element	Type	Cardinality	Description
url	AnyURI (up to 2500 characters)	1	Uniform Resource Locator (URL) that references a World-Wide Web resource providing a product image. The value of url SHALL be an absolute URL without a fragment identifier, as specified in [RFC3986]; relative URL references SHALL NOT be used.
imageTypeCode	TSD_ImageTypeCode	1	Code specifying the type of images available at the specified url.
languageCode	LanguageCode	0..*	Language codes specifying the language(s) for which this image applies, or omitted if this image applies to all languages.
imagePixelHeight	nonNegativeInteger	0..1	The number of pixels along the vertical axis of the image.
imagePixelWidth	nonNegativeInteger	0..1	The number of pixels along the horizontal axis of the image.
fileSize	Measurement (Section 6.2.7)	0..1	Measure of the size of the file, expressed as value and unit of measure code. Example: 2 megabytes.
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

504

The value of imageTypeCode SHALL be one of the following:

Value	Description
LOGO	Link to a file containing the Manufacturer or Brand Logo(s) associated with the product.
OUT_OF_PACKAGE_IMAGE	Link to an image of an item out of its packaging and, if necessary, assembled ready for use by the end user. This type of file is subject to the GDSN Product Image Specification Standard.
PRODUCT_IMAGE	Link to a file containing a visual representation of the product.
PRODUCT_LABEL_IMAGE	Link to a file containing a visual representation of the product label.

505 **6.4.2. Nutritional Product Information**

506 The following UML diagram expresses the data content of the Nutritional Product Information Module.



507
 508 The data content of a TSD_NutritionalProductInformationModule structure SHALL be as
 509 follows:

Data Element	Type	Cardinality	Description
nutrientHeader	TSD_NutrientHeader (below)	1..*	The nutritional information for a specified serving size and state of preparation.
avpList	AttributeValuePairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

The data content of a TSD_NutrientHeader structure SHALL be as follows:

Data Element	Type	Cardinality	Description
preparationStateCode	TSD_Preparation StateCode (below)	1	Code specifying the preparation state of the product for which the nutrient information is valid.
dailyValue IntakeReference	Description70 (Section 6.2.6)	0..*	Free text field specifying the daily value intake base on which the daily value intake per nutrient has been based. Examples: "Based on a 2000 calorie diet.", "based on one cup skimmed milk", "diet for 4 year old". If more than one Description70 structure is present each SHALL have a different language code, and SHALL represent representations of the same value in different languages.
servingSizeDescription	Description70 (Section 6.2.6)	0..*	A free text field specifying the serving size for which the nutrient information has been stated; for example: per 1/3 cup (42 g). If more than one Description70 structure is present each SHALL have a different language code, and SHALL represent representations of the same value in different languages.

Data Element	Type	Cardinality	Description
servingSize	Measurement (Section 6.2.7)	0..*	Measurement value specifying the actual serving size. Example: 40 grams. When specified, <code>servingSize</code> establishes the basis for all contained <code>NutrientDetail</code> records. A <code>NutrientHeader</code> SHALL contain <code>servingSize</code> or <code>nutrientBasisQuantity</code> (but not both). If more than one Measurement structure is present each SHALL have a different <code>unitOfMeasure</code> and SHALL represent the same value expressed in different units.
nutrientBasisQuantity	Measurement (Section 6.2.7)	0..*	Quantity on which the nutrient information has been based; for example, per 100 grams. When specified, <code>basisQuantity</code> establishes the basis for all contained <code>NutrientDetail</code> records. A <code>NutrientHeader</code> SHALL contain <code>servingSize</code> or <code>nutrientBasisQuantity</code> (but not both). If more than one Measurement structure is present each SHALL have a different <code>unitOfMeasure</code> and SHALL represent the same value expressed in different units.
numberOfServings	Decimal	0..1	Number of servings. If both this and <code>servingSize</code> are specified, the value SHALL be the number of servings based on <code>servingSize</code> . Otherwise, the value is informational and not related to any particular serving sized expressed within this product data.

Data Element	Type	Cardinality	Description
nutrientDetail	TSD_NutrientDetail (below)	1..*	Detailed information about specific nutrients for this serving size and preparation state. Each member of this list SHALL have a distinct value of nutrientTypeCode.
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

511 The value of preparationStateCode SHALL be one of the following:

Value	Description
PREPARED	The state of the product after preparation (e.g. after adding milk or water)
UNPREPARED	The initial state of the product

512 The data content of a TSD_NutrientDetail structure SHALL be as follows:

Data Element	Type	Cardinality	Description
nutrientTypeCode	TSD_NutrientTypeCode (below)	1	Code that specifies which nutrient this NutrientDetail structure describes.
dailyValue IntakePercent	Float	0..1	The percentage of the recommended daily intake of a nutrient provided by the quantity of product defined in the nutrient header (by servingSize or nutrientBasisQuantity), as recommended by authorities of the target market. For example, a value of 50 denotes 50% of daily value. If the NutrientHeader structure includes a dailyValue IntakeReference, this value SHALL be computed with reference to the diet specified by dailyValueIntakeReference. Otherwise, this value SHALL be computed with reference to a diet not expressed in this product data, but used consistently for all nutrient detail records associated with the same nutrient header.
measurement Precision	Measurement Precision (below)	0..1	Code indicating the exactness of the method used to analyse the nutritional value.

Data Element	Type	Cardinality	Description
quantityContained	Measurement (Section 6.2.7)	0..*	Measurement value indicating the amount of nutrient contained in the quantity of product defined in the nutrient header (by <code>servingSize</code> or <code>nutrientBasisQuantity</code>). If more than one Measurement structure is present each SHALL have a different <code>unitOfMeasure</code> and SHALL represent the same value expressed in different units.
avpList	AttributeValue PairList (Section 6.2.8)	0..1	Temporary attributes introduced between minor versions; see Section 5.4.1.

513 The value of `nutrientTypeCode` SHALL be a “tagname” defined in the INFOODS Food Component
514 Tagname list [INFOODS].

515 The value of `measurementPrecision` SHALL be one of the following values:

Value	Meaning
APPROXIMATELY	The method used to analyse the products resulted in approximate value of the nutritional content.
EXACT	The method used to analyse the products resulted in exact value of the nutritional content.
LESS_THAN	To indicate presence when the measurement value is too small to be measured precisely (rule states less than 0.5).

516

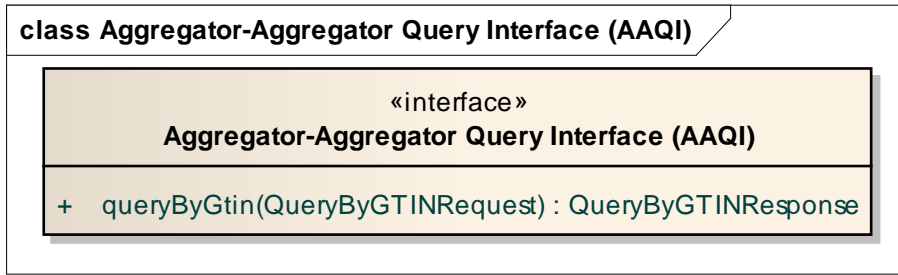
517 7. Interfaces – Abstract Definition

518 This section defines the three service interfaces that constitute this standard: the Aggregator-
519 Aggregator Query Interface (AAQI), the Aggregator-Index Query Interface (AIQI), and the Aggregator-
520 Index Maintenance Interface (AIMI). This section defines the operations within each interface and
521 their semantics at an abstract level. See Section 8.3 for concrete realizations as an XML schema of
522 the request and response messages implied by the interface definitions, Section 9.1 for a concrete
523 realization as web services of these interfaces based on the XML schemas in Section 8.3, and
524 Section 9.2 for a (non-XML) implementation of the AIQI based on ONS 2.0.

525 7.1. Aggregator-Aggregator Query Interface (AAQI) – Abstract Definition

526 This section specifies the Aggregator-Aggregator Query Interface (AAQI). A Data Aggregator SHALL
527 implement both the requesting and responding side of this interface.

528 The following UML diagram expresses the operations in the AAQI.



529
530 A Data Aggregator SHALL implement the following interface operations:

Operation	Section	Description
queryByGtin	7.1.1	Retrieve product data for a specified GTIN and target market, subject to specified language preferences.

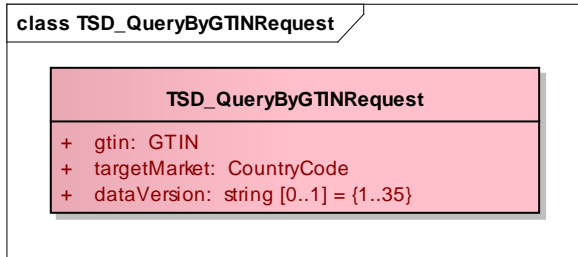
531 A Data Aggregator SHALL implement the following interface exceptions:

Exception	Operations	Description
NoDataException	queryByGtin	No data is available for the GTIN + target market combination specified in the request
InvalidGtinException	queryByGtin	The GTIN specified in the request is not properly formatted, or has an incorrect check digit.
InvalidTargetMarketException	queryByGtin	The target market specified in the request is not properly formatted
UnsupportedVersionException	queryByGtin	The ProductData version specified in the request is not supported (see Section 5.4.1)
InvalidRequestException	queryByGtin	The request was not formatted correctly, contained an unknown parameter, or specified an unknown operation.
SecurityException	queryByGtin	The operation was not permitted due to an access control violation or other security concern. This includes the case where the service wishes to deny authorization to execute a particular operation based on the authenticated client identity.
ImplementationException	queryByGtin	A generic exception thrown by the implementation for reasons that are implementation-specific, such as a software error.

532 If more than one exception applies to a given request, the Data Aggregator SHALL respond with the
533 applicable exception that occurs closer to the bottom of the above list; however the Data Aggregator
534 MAY choose to respond with InvalidRequestException in the case that
535 InvalidRequestException and SecurityException both apply.

536 **7.1.1. AAQI queryByGtin Operation**

537 The following UML diagram expresses the data content of the request for the `queryByGtin` operation
 538 of the AAQI.

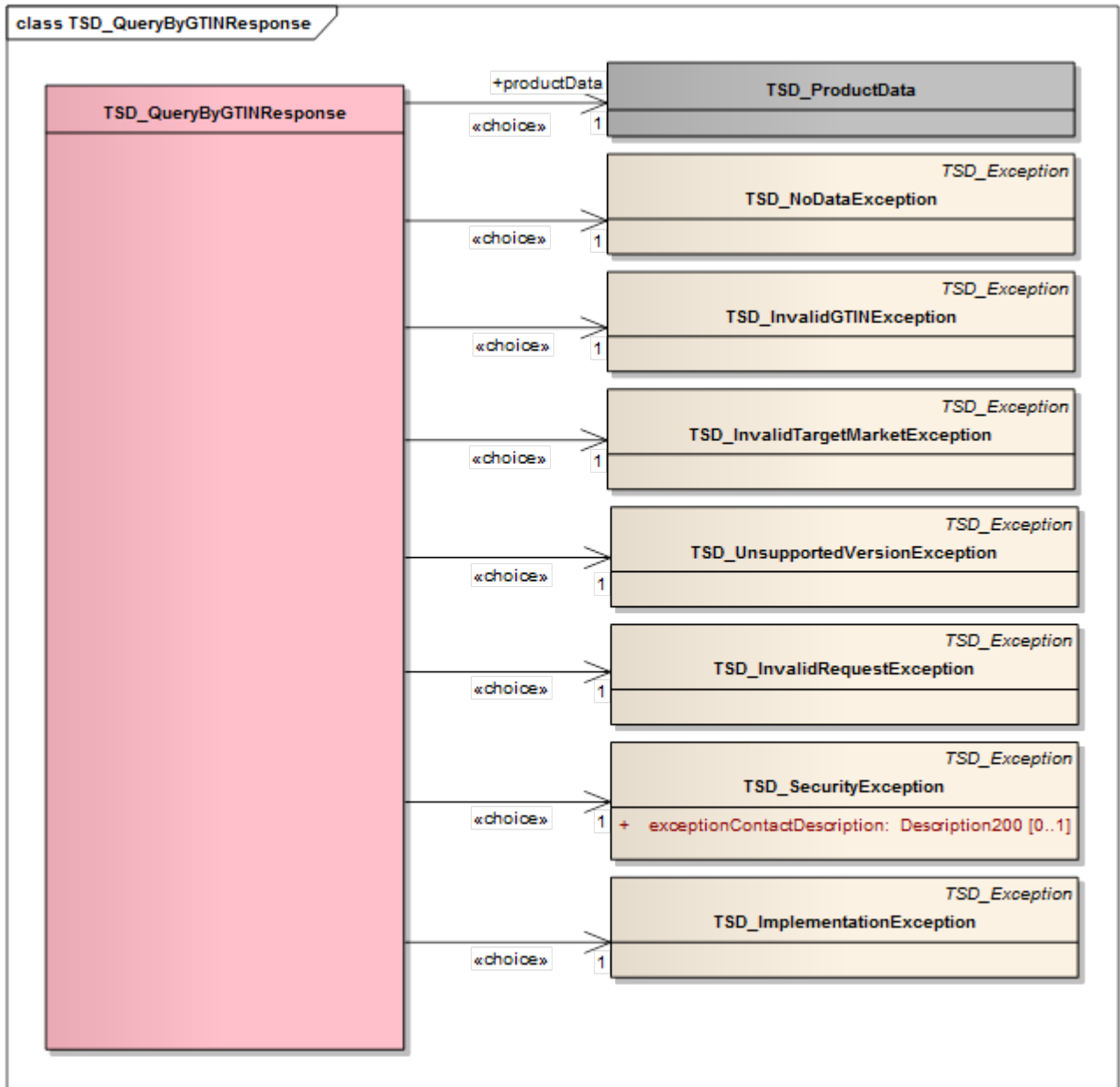


539
 540 The data content of the `TSD_QueryByGtinRequest` SHALL be as follows:

Data Element	Type	Cardinality	Description
<code>gtin</code>	GTIN (Section 6.2.1)	1	The GTIN of the product being requested.
<code>targetMarket</code>	CountryCode (Section 6.2.2)	1	Target market for the data being requested, expressed as a <code>CountryCode</code> (Section 6.2.2).
<code>dataVersion</code>	String	1	The version number of the desired version of <code>ProductData</code>

541
542

The following UML diagram expresses the data content of the response for the QueryByGtin operation of the AAQI.



543
544
545

The data content of the TSD_QueryByGtinResponse SHALL be either a TSD_ProductData structure (Section 6) or one of the exceptions specified above.

546

The data content of each exception *except* TSD_SecurityException SHALL be as follows:

Data Element	Type	Cardinality	Description
exceptionReason (provided in base class TSD_Exception)	Description200 (Section 6.2.6)	1	A human readable string providing additional information about the exception. This is not intended for presentation to end users. The content is at the discretion of the responding Data Aggregator but SHOULD always be in English

547

The data content of TSD_SecurityException SHALL be as follows:

Data Element	Type	Cardinality	Description
exceptionReason (provided in base class TSD_Exception)	Description200 (Section 6.2.6)	1	A human readable string providing additional information about the exception. This is not intended for presentation to end users. The content is at the discretion of the responding Data Aggregator but SHOULD always be in English
exceptionContactDescription	Description200 (Section 6.2.6)	0..1	A human readable string providing contact information for establishing a trust relationship with a Data Aggregator not previously encountered.

548

7.2. Aggregator-Index Query Interface (AIQI) – Abstract Definition

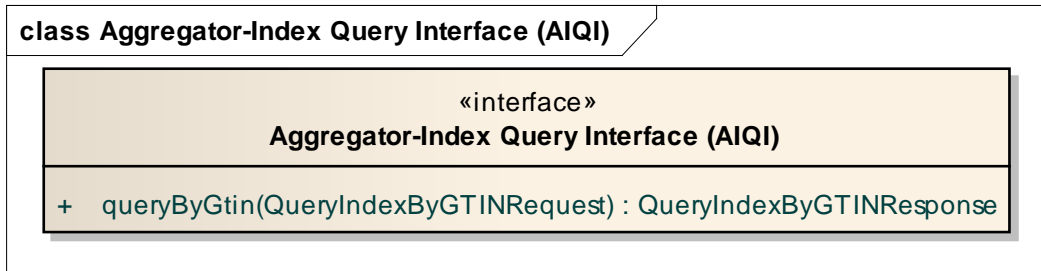
549

550

551

This section specifies the Aggregator-Index Query Interface (AAQI). A Data Aggregator SHALL implement the client side of this interface. A Global Index SHALL implement the server side of this interface.

552 The following UML diagram expresses the operations in the AIQI.



553
554 A Data Aggregator (client side) or Global Index (server side) SHALL implement the following interface
555 operations:

Operation	Section	Description
queryByGtin	7.2.1	Retrieve index data for a specified GTIN and target market.

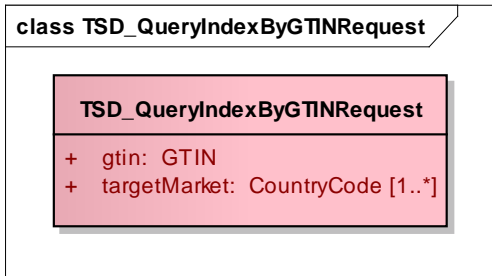
556 A Data Aggregator (client side) or Global Index (server side) SHALL implement the following interface
557 exceptions:

Exception	Operations	Description
InvalidGtinException	queryByGtin	The GTIN specified in the request is not properly formatted, or has an incorrect check digit.
InvalidTargetMarketException	queryByGtin	The target market specified in the request is not properly formatted
InvalidRequestException	queryByGtin	The request was not formatted correctly, contained an unknown parameter, or specified an unknown operation.
SecurityException	queryByGtin	The operation was not permitted due to an access control violation or other security concern. This includes the case where the service wishes to deny authorization to execute a particular operation based on the authenticated client identity.
ImplementationException	queryByGtin	A generic exception thrown by the implementation for reasons that are implementation-specific, such as a software error.

558 If more than one exception applies to a given request, the Global Index SHALL respond with the
559 applicable exception that occurs closer to the bottom of the above list; ; however the Global Index
560 MAY choose to respond with InvalidRequestException in the case that
561 InvalidRequestException and SecurityException both apply..

562 **7.2.1. AIQI queryByGtin**

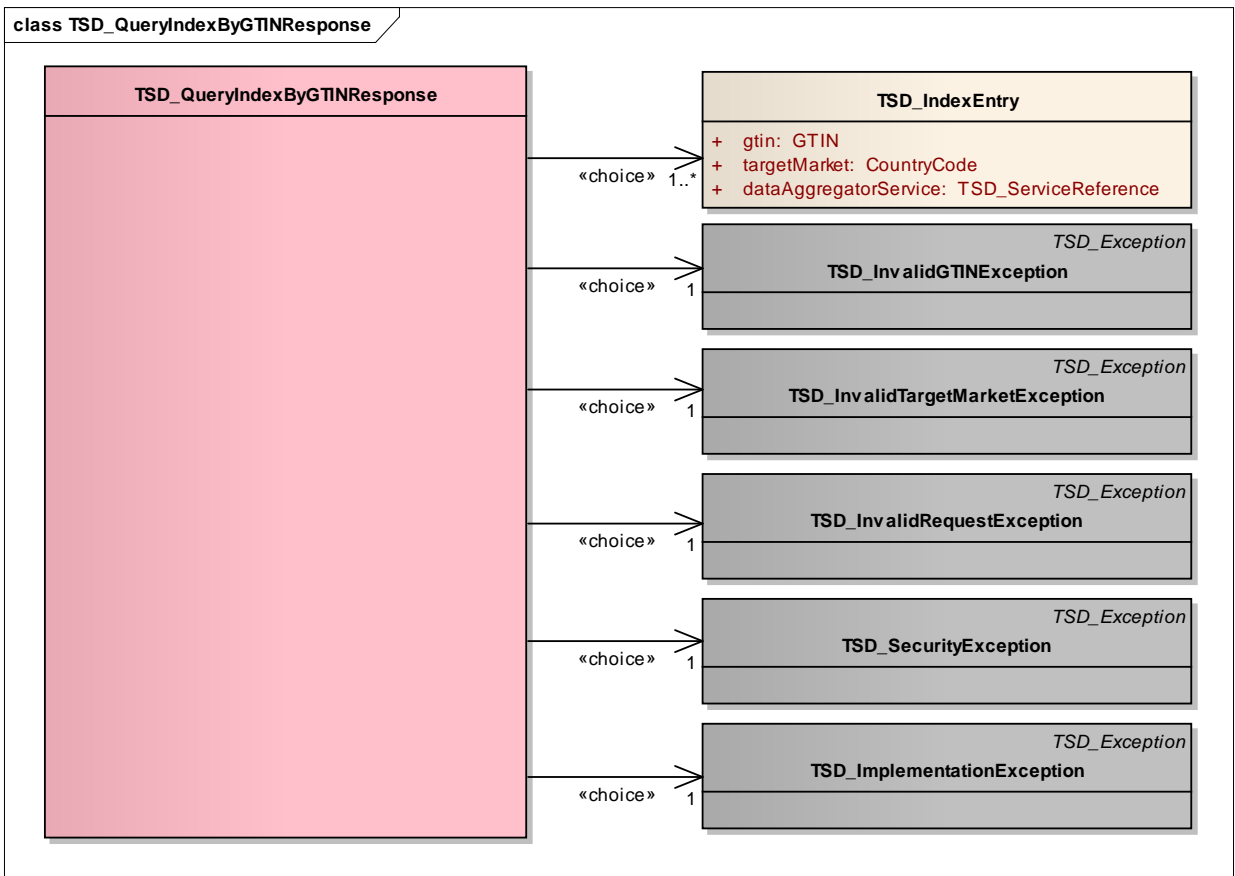
563 The following UML diagram expresses the data content of the request for the `queryByGtin` operation
 564 of the AIQI.



565
 566 The data content of the `TSD_QueryIndexByGtinRequest` SHALL be as follows:

Data Element	Type	Cardinality	Description
gtin	GTIN (Section 6.2.1)	1	The GTIN of the product being requested.
targetMarket	CountryCode (Section 6.2.2)	1..*	Target markets for the data being requested, each expressed as a <code>CountryCode</code> (Section 6.2.2).

567 The following UML diagram expresses the data content of the response for the `queryByGtin`
 568 operation of the AIQI.



569

570
571
572
573

The data content of the `TSD_QueryIndexByGtinResponse` SHALL be either a list of zero or more `IndexEntry` structures that match the GTIN and target market(s) specified in the request, or be one of the exceptions defined above.

The data content of an `IndexEntry` SHALL be as follows:

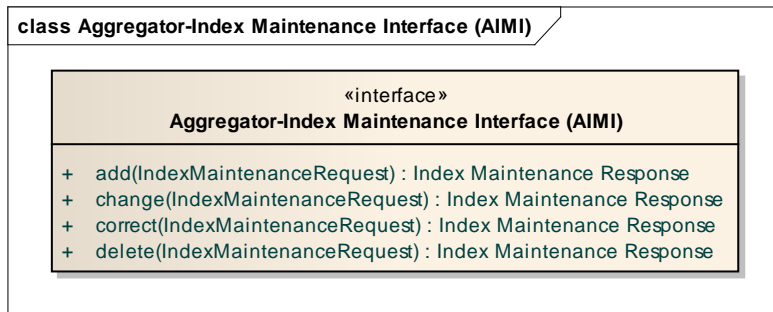
Data Element	Type	Cardinality	Description
<code>gtin</code>	GTIN (Section 6.2.1)	1	The GTIN of the product referenced by this index entry.
<code>targetMarket</code>	CountryCode (Section 6.2.2)	1	Target market for the data referenced by this index entry, expressed as a <code>CountryCode</code> (Section 6.2.2).
<code>dataAggregatorService</code>	ServiceReference (Section 6.2.4)	1	Reference to a Data Aggregator service that has data for the GTIN, target market combination specified in this index entry.

574
575
576
577
578
579
580
581

7.3. Aggregator-Index Maintenance Interface (AIMI) – Abstract Definition

This section specifies the Aggregator-Index Maintenance Interface (AIMI). A Data Aggregator MAY implement the client side of this interface. A Global Index MAY implement the server side of this interface. In this version of the standard, no binding of this abstract interface to a concrete service is specified. The abstract definition is provided as a guide for Data Aggregators and Global Index implementations that wish to provide a proprietary (non-standard) implementation of this interface. Concrete bindings of AIMI interface may be specified in a future version of this standard.

The following UML diagram expresses the operations in the AIMI.



582
583
584

A Data Aggregator (client side) or Global Index (server side) SHALL implement the following interface operations:

Operation	Description
<code>add</code>	Add a new index entry for a specified GTIN and target market.
<code>change</code>	Change an existing index entry for a specified GTIN and target market.
<code>correct</code>	Correct an existing index entry for a specified GTIN and target market. The effect of this operation is identical to the <code>change</code> operation, but expresses a different intent on the part of the client.

Operation	Description
delete	Delete an index entry for a specified GTIN and target market.

585
586
587
588
589
590
591
592

At most one index entry exists for each unique combination of gtin, target market, and data aggregator service. Note that the AIMI is designed to support maintenance of index entries that may contain additional searchable attributes besides gtin, target market, and data aggregator. The `change` and `correct` operations provide a means to alter the additional searchable attributes for an entry that already exists for a given gtin, target market, and data aggregator. As specified below, however, there are no such additional searchable attributes and the AIQI only supports query by gtin and target market. Without additional searchable attributes, the `change` and `correct` operations have no effect.

593
594

A Data Aggregator (client side) or Global Index (server side) SHALL implement the following interface exceptions:

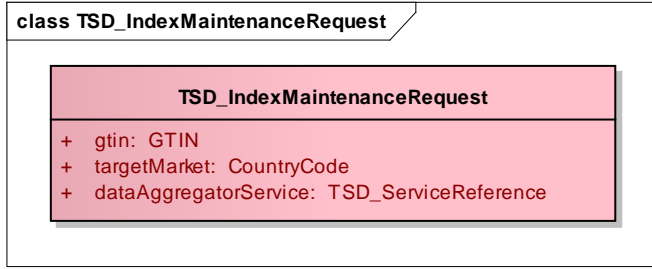
Exception	Operations	Description
<code>DuplicateEntryException</code>	add	There already exists an entry for the specified GTIN, target market, and data aggregator service.
<code>NoSuchEntryException</code>	change correct delete	No entry exists for the specified GTIN, target market, and data aggregator service.
<code>InvalidGtinException</code>	(all)	The GTIN specified in the request is not properly formatted, or has an incorrect check digit.
<code>InvalidTargetMarketException</code>	(all)	The target market specified in the request is not properly formatted.
<code>InvalidServiceURLException</code>	(all)	The data aggregator service URL specified in the request is not an absolute URL as specified in [RFC3986].
<code>InvalidRequestException</code>	(all)	The request was not formatted correctly, contained an unknown parameter, or specified an unknown operation.
<code>SecurityException</code>	(all)	The operation was not permitted due to an access control violation or other security concern. This includes the case where the service wishes to deny authorization to execute a particular operation based on the authenticated client identity.
<code>ImplementationException</code>	(all)	A generic exception thrown by the implementation for reasons that are implementation-specific, such as a software error.

595
596
597
598

If more than one exception applies to a given request, the Global Index SHALL respond with the applicable exception that occurs closer to the bottom of the above list; however the Global Index MAY choose to respond with `InvalidRequestException` in the case that `InvalidRequestException` and `SecurityException` both apply..

599 **7.3.1. AIMI Index Maintenance Request**

600 The following UML diagram expresses the data content of the request for all AIMI operations.



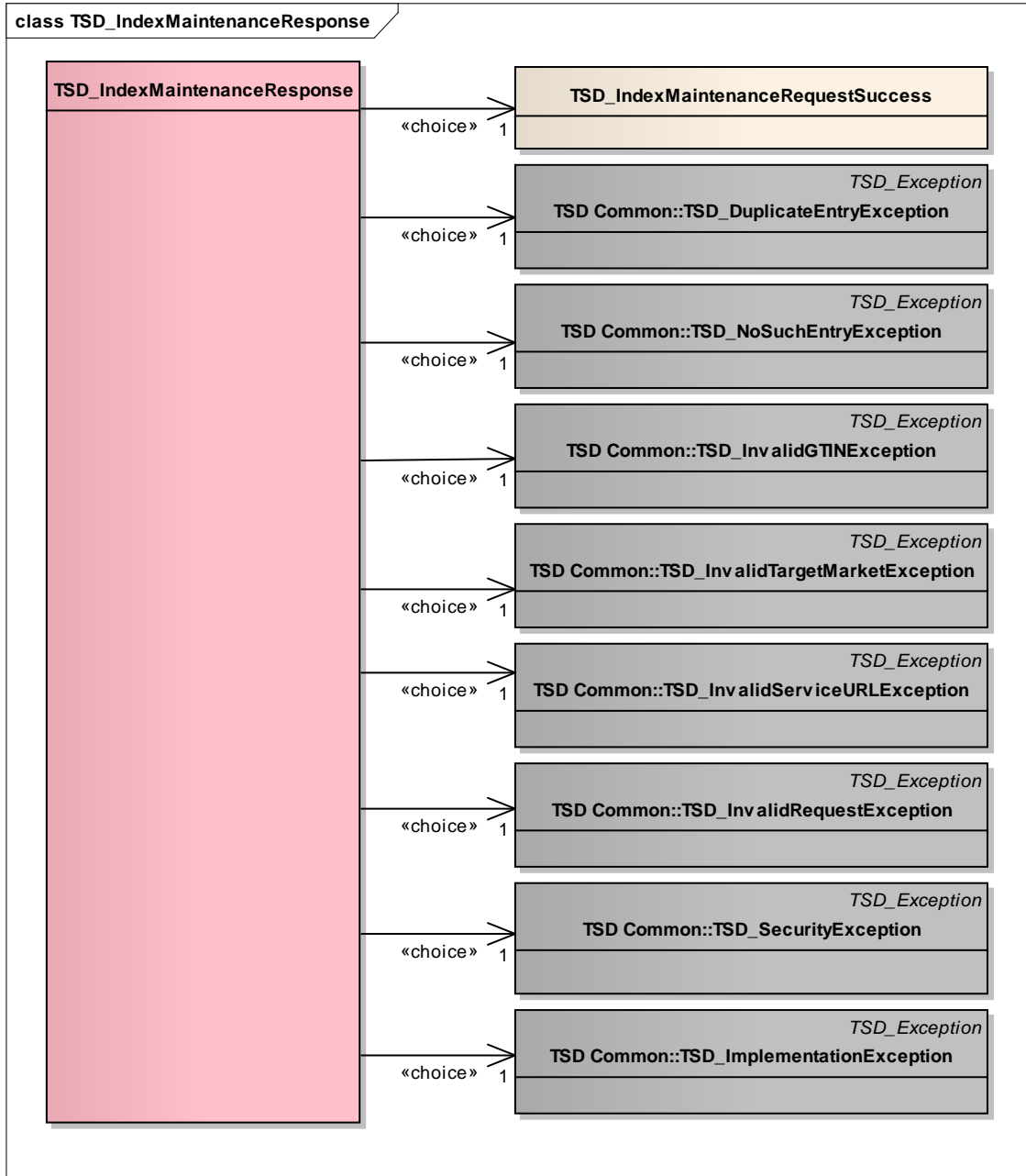
601

602 The data content of an TSD_IndexMaintenanceRequest SHALL be as follows:

Data Element	Type	Cardinality	Description
gtin	GTIN (Section 6.2.1)	1	The GTIN of the product referenced by this index entry.
targetMarket	CountryCode (Section 6.2.2)	1	Target market for the data referenced by this index entry, expressed as a CountryCode (Section 6.2.2).
dataAggregatorService	ServiceReference (Section 6.2.4)	1	Reference to a Data Aggregator service that has data for the GTIN, target market combination specified in this index entry.

603 **7.3.2. AIMI Index Maintenance Response**

604 The data content of the IndexMaintenanceResponse SHALL be either an
 605 IndexMaintenanceSuccess structure or one of the exceptions defined above.



607

The data content of each exception *except* TSD_SecurityException SHALL be as follows:

Data Element	Type	Cardinality	Description
exceptionReason (provided in base class TSD_Exception)	Description200 (Section 6.2.6)	1	A human readable string providing additional information about the exception. This is not intended for presentation to end users. The content is at the discretion of the responding Data Aggregator but SHOULD always be in English

608

The data content of TSD_SecurityException SHALL be as follows:

Data Element	Type	Cardinality	Description
exceptionReason (provided in base class TSD_Exception)	Description200 (Section 6.2.6)	1	A human readable string providing additional information about the exception. This is not intended for presentation to end users. The content is at the discretion of the responding Data Aggregator but SHOULD always be in English
exceptionContactDescription	Description200 (Section 6.2.6)	0..1	A human readable string providing contact information for establishing a trust relationship with a Data Aggregator not previously encountered.

609

8. XML Schemas

610

This section specifies XML schemas that implement the data definitions specified in previous sections. These schemas are referenced normatively by the definitions of XML-based service interface bindings specified in Section 9.1.

611

612

613

The XML data returned in the REST binding of AAQI as specified in Section 7.1 SHALL conform to the schemas in this section.

614

615 8.1. Common Data Types XML Schema

616 This section specifies two XML schemas that implements the common data types specified in
617 Section 6.2.

618 8.1.1. Common Data Types XML Schema – Shared

619 This section specifies an XML schema that implements common data types specified in Section 6.2
620 that are also used in other GS1 standards. Only those common data types used by this TSD standard
621 are included in the schema below; the published XSD file accompanying this standard may include
622 additional type definitions not used by TSD.

```
623 <?xml version="1.0" encoding="UTF-8"?>
624 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
625           xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
626           attributeFormDefault="unqualified"
627           elementFormDefault="unqualified"
628           targetNamespace="urn:gs1:shared:shared_common:xsd:3"
629           version="3.0">
630   <xsd:annotation>
631     <xsd:documentation><![CDATA[-----
632   © Copyright GS1, 2012
```

633
634 GS1 is providing this XML Schema Definition file and resultant XML file as a service to
635 interested industries.
636 This XML Schema Definition file and resultant XML file were developed through a consensus
637 process of interested parties.

638
639 Although efforts have been made to ensure that the XML Schema Definition file and resultant
640 XML file are correct, reliable, and technically
641 accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
642 resultant XML file ARE
643 CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
644 WILL BE SUITABLE FOR
645 ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
646 file and resultant XML
647 file are with the understanding that GS1 has no liability for any claim to the contrary, or
648 for any damage or loss of any kind or nature.

```
649
650 Version Information:
651 Version Number: 3.0
652 Date of creation: January 2012
```

653
654 The schema and subsequent updates will be provided on the GS1 websites.

```
655 -----
656 ]]></xsd:documentation>
657 </xsd:annotation>
658 <xsd:complexType name="CountryCodeType">
659   <xsd:simpleContent>
660     <xsd:extension base="shared_common:GS1CodeType"/>
661   </xsd:simpleContent>
662 </xsd:complexType>
663 <xsd:complexType name="Description200Type">
664   <xsd:simpleContent>
665     <xsd:extension base="shared_common:String200Type">
666       <xsd:attribute name="languageCode" use="required">
667         <xsd:simpleType>
668           <xsd:restriction base="xsd:string">
669             <xsd:maxLength value="80"/>
670             <xsd:minLength value="1"/>
671           </xsd:restriction>
672         </xsd:simpleType>
673       </xsd:attribute>
674       <xsd:attribute name="codeListVersion">
675         <xsd:simpleType>
676           <xsd:restriction base="xsd:string">
677             <xsd:maxLength value="35"/>
678             <xsd:minLength value="1"/>
```

```

679         </xsd:restriction>
680     </xsd:simpleType>
681     </xsd:attribute>
682 </xsd:extension>
683 </xsd:simpleContent>
684 </xsd:complexType>
685 <xsd:complexType name="Description2500Type">
686     <xsd:simpleContent>
687         <xsd:extension base="shared_common:String2500Type">
688             <xsd:attribute name="languageCode" use="required">
689                 <xsd:simpleType>
690                     <xsd:restriction base="xsd:string">
691                         <xsd:maxLength value="80"/>
692                         <xsd:minLength value="1"/>
693                     </xsd:restriction>
694                 </xsd:simpleType>
695             </xsd:attribute>
696             <xsd:attribute name="codeListVersion">
697                 <xsd:simpleType>
698                     <xsd:restriction base="xsd:string">
699                         <xsd:maxLength value="35"/>
700                         <xsd:minLength value="1"/>
701                     </xsd:restriction>
702                 </xsd:simpleType>
703             </xsd:attribute>
704         </xsd:extension>
705     </xsd:simpleContent>
706 </xsd:complexType>
707 <xsd:complexType name="Description70Type">
708     <xsd:simpleContent>
709         <xsd:extension base="shared_common:String70Type">
710             <xsd:attribute name="languageCode" use="required">
711                 <xsd:simpleType>
712                     <xsd:restriction base="xsd:string">
713                         <xsd:maxLength value="80"/>
714                         <xsd:minLength value="1"/>
715                     </xsd:restriction>
716                 </xsd:simpleType>
717             </xsd:attribute>
718             <xsd:attribute name="codeListVersion">
719                 <xsd:simpleType>
720                     <xsd:restriction base="xsd:string">
721                         <xsd:maxLength value="35"/>
722                         <xsd:minLength value="1"/>
723                     </xsd:restriction>
724                 </xsd:simpleType>
725             </xsd:attribute>
726         </xsd:extension>
727     </xsd:simpleContent>
728 </xsd:complexType>
729 <xsd:complexType name="Description80Type">
730     <xsd:simpleContent>
731         <xsd:extension base="shared_common:String80Type">
732             <xsd:attribute name="languageCode" use="required">
733                 <xsd:simpleType>
734                     <xsd:restriction base="xsd:string">
735                         <xsd:maxLength value="80"/>
736                         <xsd:minLength value="1"/>
737                     </xsd:restriction>
738                 </xsd:simpleType>
739             </xsd:attribute>
740             <xsd:attribute name="codeListVersion">
741                 <xsd:simpleType>
742                     <xsd:restriction base="xsd:string">
743                         <xsd:maxLength value="35"/>
744                         <xsd:minLength value="1"/>
745                     </xsd:restriction>
746                 </xsd:simpleType>
747             </xsd:attribute>
748         </xsd:extension>
749     </xsd:simpleContent>

```

```

750     </xsd:complexType>
751     <xsd:complexType name="GS1CodeType">
752         <xsd:simpleContent>
753             <xsd:extension base="shared_common:String80Type">
754                 <xsd:attribute name="codeListVersion">
755                     <xsd:simpleType>
756                         <xsd:restriction base="xsd:string">
757                             <xsd:maxLength value="35"/>
758                             <xsd:minLength value="1"/>
759                         </xsd:restriction>
760                     </xsd:simpleType>
761                 </xsd:attribute>
762             </xsd:extension>
763         </xsd:simpleContent>
764     </xsd:complexType>
765     <xsd:complexType name="LanguageCodeType">
766         <xsd:simpleContent>
767             <xsd:extension base="shared_common:GS1CodeType"/>
768         </xsd:simpleContent>
769     </xsd:complexType>
770     <xsd:complexType name="MeasurementType">
771         <xsd:simpleContent>
772             <xsd:extension base="xsd:decimal">
773                 <xsd:attribute name="measurementUnitCode" use="required">
774                     <xsd:simpleType>
775                         <xsd:restriction base="xsd:string">
776                             <xsd:maxLength value="80"/>
777                             <xsd:minLength value="1"/>
778                         </xsd:restriction>
779                     </xsd:simpleType>
780                 </xsd:attribute>
781                 <xsd:attribute name="codeListVersion">
782                     <xsd:simpleType>
783                         <xsd:restriction base="xsd:string">
784                             <xsd:maxLength value="35"/>
785                             <xsd:minLength value="1"/>
786                         </xsd:restriction>
787                     </xsd:simpleType>
788                 </xsd:attribute>
789             </xsd:extension>
790         </xsd:simpleContent>
791     </xsd:complexType>
792     <xsd:simpleType name="GTINType">
793         <xsd:restriction base="xsd:string">
794             <xsd:pattern value="\d{14}"/>
795         </xsd:restriction>
796     </xsd:simpleType>
797     <xsd:simpleType name="GLNType">
798         <xsd:restriction base="xsd:string">
799             <xsd:pattern value="\d{13}"/>
800         </xsd:restriction>
801     </xsd:simpleType>
802     <xsd:simpleType name="String200Type">
803         <xsd:restriction base="xsd:string">
804             <xsd:maxLength value="200"/>
805             <xsd:minLength value="1"/>
806         </xsd:restriction>
807     </xsd:simpleType>
808     <xsd:simpleType name="String2500Type">
809         <xsd:restriction base="xsd:string">
810             <xsd:maxLength value="2500"/>
811             <xsd:minLength value="1"/>
812         </xsd:restriction>
813     </xsd:simpleType>
814     <xsd:simpleType name="String70Type">
815         <xsd:restriction base="xsd:string">
816             <xsd:maxLength value="70"/>
817             <xsd:minLength value="1"/>
818         </xsd:restriction>
819     </xsd:simpleType>
820     <xsd:simpleType name="String80Type">

```

```

821         <xsd:restriction base="xsd:string">
822             <xsd:maxLength value="80"/>
823             <xsd:minLength value="1"/>
824         </xsd:restriction>
825     </xsd:simpleType>
826 </xsd:schema>

```

827 8.1.2. Common Data Types XML Schema – TSD-specific

828 This section specifies an XML schema that implements common data types specified in Section 6.2
829 that are only used in this TSD standard (and not any other GS1 standard).

```

830 <?xml version="1.0" encoding="UTF-8"?>
831 <xsd:schema xmlns:tsd_common="urn:gs1:tsd:tsd_common:xsd:1"
832   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
833   xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
834   targetNamespace="urn:gs1:tsd:tsd_common:xsd:1" elementFormDefault="unqualified"
835   attributeFormDefault="unqualified" version="1.0">
836     <xsd:annotation>
837       <xsd:documentation><![CDATA[-----
838       Â© Copyright GS1, 2012
839     </xsd:documentation>

```

840 GS1 is providing this XML Schema Definition file and resultant XML file as a service to
841 interested industries.
842 This XML Schema Definition file and resultant XML file were developed through a consensus
843 process of interested parties.

844
845 Although efforts have been made to ensure that the XML Schema Definition file and resultant
846 XML file are correct, reliable, and technically
847 accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
848 resultant XML file ARE
849 CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
850 WILL BE SUITABLE FOR
851 ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
852 file and resultant XML
853 file are with the understanding that GS1 has no liability for any claim to the contrary, or
854 for any damage or loss of any kind or nature.

855
856 Version Information:
857 Version Number: 1.0
858 Date of creation: August 2012

859
860 The schema and subsequent updates will be provided on the GS1 websites.

```

861 -----
862 ]]></xsd:documentation>
863 </xsd:annotation>
864 <xsd:import namespace="urn:gs1:shared:shared_common:xsd:3"
865   schemaLocation="../shared/SharedCommon.xsd"/>
866 <xsd:complexType name="AttributeValuePairListType">
867   <xsd:sequence>
868     <xsd:element name="stringAVP" type="tsd_common:StringAttributeValuePairType"
869     maxOccurs="unbounded"/>
870   </xsd:sequence>
871 </xsd:complexType>
872 <xsd:complexType name="MeasurementPrecisionCodeType">
873   <xsd:complexContent>
874     <xsd:extension base="shared_common:GS1CodeType"/>
875   </xsd:complexContent>
876 </xsd:complexType>
877 <xsd:complexType name="StringAttributeValuePairType">
878   <xsd:attribute name="name" use="required">
879     <xsd:simpleType>
880       <xsd:restriction base="xsd:string">
881         <xsd:minLength value="1"/>
882         <xsd:maxLength value="70"/>
883       </xsd:restriction>
884     </xsd:simpleType>
885   </xsd:attribute>
886 </xsd:complexType>
887 <xsd:complexType name="TSD_BrandNameInformationType">

```

```

888     <xsd:sequence>
889         <xsd:element name="brandName">
890             <xsd:simpleType>
891                 <xsd:restriction base="xsd:string">
892                     <xsd:minLength value="1"/>
893                     <xsd:maxLength value="70"/>
894                 </xsd:restriction>
895             </xsd:simpleType>
896         </xsd:element>
897         <xsd:element name="languageSpecificBrandName" type="shared_common:Description70Type"
898 minOccurs="0" maxOccurs="unbounded"/>
899     </xsd:sequence>
900 </xsd:complexType>
901 <xsd:complexType name="TSD_ExceptionType">
902     <xsd:sequence>
903         <xsd:element name="exceptionReason" type="shared_common:Description200Type"/>
904     </xsd:sequence>
905 </xsd:complexType>
906 <xsd:complexType name="TSD_ImageLinkType">
907     <xsd:sequence>
908         <xsd:element name="url">
909             <xsd:simpleType>
910                 <xsd:restriction base="xsd:anyURI">
911                     <xsd:minLength value="1"/>
912                     <xsd:maxLength value="2500"/>
913                 </xsd:restriction>
914             </xsd:simpleType>
915         </xsd:element>
916         <xsd:element name="imageTypeCode" type="tsd_common:TSD_ImageTypeCodeType"/>
917         <xsd:element name="languageCode" type="shared_common:LanguageCodeType" minOccurs="0"
918 maxOccurs="unbounded"/>
919         <xsd:element name="imagePixelHeight" type="xsd:nonNegativeInteger" minOccurs="0"/>
920         <xsd:element name="imagePixelWidth" type="xsd:nonNegativeInteger" minOccurs="0"/>
921         <xsd:element name="fileSize" type="shared_common:MeasurementType" minOccurs="0"/>
922         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0"/>
923     </xsd:sequence>
924 </xsd:complexType>
925 <xsd:complexType name="TSD_ImageTypeCodeType">
926     <xsd:complexContent>
927         <xsd:extension base="shared_common:GS1CodeType"/>
928     </xsd:complexContent>
929 </xsd:complexType>
930 <xsd:complexType name="TSD_PreparationStateCodeType">
931     <xsd:complexContent>
932         <xsd:extension base="shared_common:GS1CodeType"/>
933     </xsd:complexContent>
934 </xsd:complexType>
935 <xsd:complexType name="TSD_ProductInformationLinkType">
936     <xsd:sequence>
937         <xsd:element name="url">
938             <xsd:simpleType>
939                 <xsd:restriction base="xsd:anyURI">
940                     <xsd:maxLength value="2500"/>
941                     <xsd:minLength value="1"/>
942                 </xsd:restriction>
943             </xsd:simpleType>
944         </xsd:element>
945         <xsd:element name="productInformationTypeCode"
946 type="tsd_common:TSD_ProductInformationTypeCodeType"/>
947         <xsd:element name="languageCode" type="shared_common:LanguageCodeType" minOccurs="0"
948 maxOccurs="unbounded"/>
949         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0"/>
950     </xsd:sequence>
951 </xsd:complexType>
952 <xsd:complexType name="TSD_ProductInformationTypeCodeType">
953     <xsd:complexContent>
954         <xsd:extension base="shared_common:GS1CodeType"/>
955     </xsd:complexContent>
956 </xsd:complexType>
957 <xsd:complexType name="TSD_ServiceReferenceType">
958     <xsd:sequence>

```

```

959     <xsd:element name="url">
960       <xsd:simpleType>
961         <xsd:restriction base="xsd:anyURI">
962           <xsd:minLength value="1"/>
963           <xsd:maxLength value="2500"/>
964         </xsd:restriction>
965       </xsd:simpleType>
966     </xsd:element>
967   </xsd:sequence>
968 </xsd:complexType>
969 </xsd:schema>

```

970 8.2. Product Data XML Schema

971 This section specifies an XML schema that implements the abstract definition of product data specified
 972 in Section 6.

973 8.2.1. Product Data XML Schema – Top Level

974 This section specifies an XML schema for the top level of Product Data as specified in Section 6.3.
 975 The module element, which may be repeated any number of times, is specified as an XSD wildcard,
 976 but in actual instance data each occurrence SHALL be populated by an instance of one of the module
 977 schemas specified in the subsequent subsections.

```

978 <?xml version="1.0" encoding="UTF-8"?>
979 <xsd:schema xmlns:product_data="urn:gs1:tsd:product_data:xsd:1"
980   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
981   xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
982   xmlns:tsd_common="urn:gs1:tsd:tsd_common:xsd:1"
983   targetNamespace="urn:gs1:tsd:product_data:xsd:1" elementFormDefault="unqualified"
984   attributeFormDefault="unqualified" version="1.0">
985   <xsd:annotation>
986     <xsd:documentation><![CDATA[-----
987     Â© Copyright GS1, 2012

```

988
 989 GS1 is providing this XML Schema Definition file and resultant XML file as a service to
 990 interested industries.
 991 This XML Schema Definition file and resultant XML file were developed through a consensus
 992 process of interested parties.

993
 994 Although efforts have been made to ensure that the XML Schema Definition file and resultant
 995 XML file are correct, reliable, and technically
 996 accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
 997 resultant XML file ARE
 998 CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
 999 WILL BE SUITABLE FOR
 1000 ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
 1001 file and resultant XML
 1002 file are with the understanding that GS1 has no liability for any claim to the contrary, or
 1003 for any damage or loss of any kind or nature.

```

1004  

1005 Version Information:
1006 Version Number: 1.0
1007 Date of creation: August 2012

```

1008
 1009 The schema and subsequent updates will be provided on the GS1 websites.

```

1010 -----
1011 ]]></xsd:documentation>
1012 </xsd:annotation>
1013 <xsd:import namespace="urn:gs1:shared:shared_common:xsd:3"
1014   schemaLocation="../shared/SharedCommon.xsd"/>
1015 <xsd:import namespace="urn:gs1:tsd:tsd_common:xsd:1" schemaLocation="TSDCommon.xsd"/>
1016 <xsd:complexType name="TSD_ProductDataRecordType">
1017   <xsd:sequence>
1018     <xsd:element name="module" maxOccurs="unbounded">
1019       <xsd:complexType>
1020         <xsd:sequence>

```

```

1021         <xsd:any namespace="##other" processContents="lax"/>
1022     </xsd:sequence>
1023 </xsd:complexType>
1024 </xsd:element>
1025 <xsd:element name="productionVariantDescription" type="shared_common:Description70Type"
1026 minOccurs="0"/>
1027 <xsd:element name="productionVariantEffectiveDateTime" type="xsd:dateTime"
1028 minOccurs="0"/>
1029 <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0"/>
1030 </xsd:sequence>
1031 </xsd:complexType>
1032 <xsd:complexType name="TSD_ProductDataType">
1033 <xsd:sequence>
1034 <xsd:element name="gtin" type="shared_common:GTINType"/>
1035 <xsd:element name="targetMarket" type="shared_common:CountryCodeType"/>
1036 <xsd:element name="informationProviderGLN" type="shared_common:GLNType"/>
1037 <xsd:element name="informationProviderName" type="xsd:string"/>
1038 <xsd:element name="timeToLive" type="xsd:duration" minOccurs="0"/>
1039 <xsd:element name="productDataRecord" type="product_data:TSD_ProductDataRecordType"
1040 maxOccurs="unbounded"/>
1041 <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0"/>
1042 </xsd:sequence>
1043 </xsd:complexType>
1044 </xsd:schema>

```

1045 8.2.2. Product Data XML Schema – Basic Product Information Module

1046 This section specifies an XML schema for the Basic Product Information module as specified in
 1047 Section 6.4.1.

```

1048 <?xml version="1.0" encoding="UTF-8"?>
1049 <xsd:schema
1050 xmlns:basic_product_information_module="urn:gs1:tsd:basic_product_information_module:xsd:1"
1051 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1052 xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
1053 xmlns:tsd_common="urn:gs1:tsd:tsd_common:xsd:1"
1054 targetNamespace="urn:gs1:tsd:basic_product_information_module:xsd:1"
1055 elementFormDefault="unqualified" attributeFormDefault="unqualified" version="1.0">
1056 <xsd:annotation>
1057 <xsd:documentation><![CDATA[-----
1058 Â© Copyright GS1, 2012
1059
1060 GS1 is providing this XML Schema Definition file and resultant XML file as a service to
1061 interested industries.
1062 This XML Schema Definition file and resultant XML file were developed through a consensus
1063 process of interested parties.
1064
1065 Although efforts have been made to ensure that the XML Schema Definition file and resultant
1066 XML file are correct, reliable, and technically
1067 accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
1068 resultant XML file ARE
1069 CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
1070 WILL BE SUITABLE FOR
1071 ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
1072 file and resultant XML
1073 file are with the understanding that GS1 has no liability for any claim to the contrary, or
1074 for any damage or loss of any kind or nature.
1075
1076 Version Information:
1077 Version Number: 1.0
1078 Date of creation: August 2012
1079
1080 The schema and subsequent updates will be provided on the GS1 websites.
1081 -----
1082 ]]></xsd:documentation>
1083 </xsd:annotation>
1084 <xsd:import namespace="urn:gs1:shared:shared_common:xsd:3"
1085 schemaLocation=" ../shared/SharedCommon.xsd"/>
1086 <xsd:import namespace="urn:gs1:tsd:tsd_common:xsd:1" schemaLocation="TSDCommon.xsd"/>

```



```

1087     <xsd:element name="basicProductInformationModule"
1088 type="basic_product_information_module:TSD_BasicProductInformationModuleType"/>
1089     <xsd:complexType name="TSD_BasicProductInformationModuleType">
1090       <xsd:sequence>
1091         <xsd:element name="productName" type="shared_common:Description80Type"
1092 maxOccurs="unbounded"/>
1093         <xsd:element name="consumerMarketingDescription"
1094 type="shared_common:Description2500Type" minOccurs="0" maxOccurs="unbounded"/>
1095         <xsd:element name="gpcCategoryCode" minOccurs="0">
1096           <xsd:simpleType>
1097             <xsd:restriction base="xsd:string">
1098               <xsd:pattern value="\d{8}"/>
1099             </xsd:restriction>
1100           </xsd:simpleType>
1101         </xsd:element>
1102         <xsd:element name="brandNameInformation"
1103 type="tsd_common:TSD_BrandNameInformationType"/>
1104         <xsd:element name="productInformationLink"
1105 type="tsd_common:TSD_ProductInformationLinkType" minOccurs="0" maxOccurs="unbounded"/>
1106         <xsd:element name="imageLink" type="tsd_common:TSD_ImageLinkType" minOccurs="0"
1107 maxOccurs="unbounded"/>
1108         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0"/>
1109       </xsd:sequence>
1110     </xsd:complexType>
1111 </xsd:schema>

```

1112 8.2.3. Product Data XML Schema – Nutrition Product Information Module

1113 This section specifies an XML schema for the Nutrition Product Information module as specified in
 1114 Section 6.4.2.

```

1115 <?xml version="1.0" encoding="UTF-8"?>
1116 <xsd:schema
1117 xmlns:nutritional_product_information_module="urn:gs1:tsd:nutritional_product_information_modu
1118 le:xsd:1" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1119 xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
1120 xmlns:tsd_common="urn:gs1:tsd:tsd_common:xsd:1"
1121 targetNamespace="urn:gs1:tsd:nutritional_product_information_module:xsd:1"
1122 elementFormDefault="unqualified" attributeFormDefault="unqualified" version="1.0">
1123   <xsd:annotation>
1124     <xsd:documentation><![CDATA[-----
1125     Â© Copyright GS1, 2012
1126
1127     GS1 is providing this XML Schema Definition file and resultant XML file as a service to
1128     interested industries.
1129     This XML Schema Definition file and resultant XML file were developed through a consensus
1130     process of interested parties.
1131
1132     Although efforts have been made to ensure that the XML Schema Definition file and resultant
1133     XML file are correct, reliable, and technically
1134     accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
1135     resultant XML file ARE
1136     CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
1137     WILL BE SUITABLE FOR
1138     ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
1139     file and resultant XML
1140     file are with the understanding that GS1 has no liability for any claim to the contrary, or
1141     for any damage or loss of any kind or nature.
1142
1143     Version Information:
1144     Version Number: 1.0
1145     Date of creation: August 2012
1146
1147     The schema and subsequent updates will be provided on the GS1 websites.
1148     -----
1149   ]]></xsd:documentation>
1150   </xsd:annotation>
1151   <xsd:import namespace="urn:gs1:shared:shared_common:xsd:3"
1152 schemaLocation="../shared/SharedCommon.xsd"/>
1153   <xsd:import namespace="urn:gs1:tsd:tsd_common:xsd:1" schemaLocation="TSDCommon.xsd"/>

```

```

1154     <xsd:element name="nutritionalProductInformationModule"
1155 type="nutritional_product_information_module:TSD_NutritionalProductInformationModuleType" />
1156     <xsd:complexType name="NutrientTypeCodeType">
1157       <xsd:complexContent>
1158         <xsd:extension base="shared_common:GS1CodeType" />
1159       </xsd:complexContent>
1160     </xsd:complexType>
1161     <xsd:complexType name="TSD_NutrientDetailType">
1162       <xsd:sequence>
1163         <xsd:element name="nutrientTypeCode"
1164 type="nutritional_product_information_module:NutrientTypeCodeType" />
1165         <xsd:element name="dailyValueIntakePercent" type="xsd:float" minOccurs="0" />
1166         <xsd:element name="measurementPrecision" type="tsd_common:MeasurementPrecisionCodeType"
1167 minOccurs="0" />
1168         <xsd:element name="quantityContained" type="shared_common:MeasurementType" minOccurs="0"
1169 maxOccurs="unbounded" />
1170         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0" />
1171       </xsd:sequence>
1172     </xsd:complexType>
1173     <xsd:complexType name="TSD_NutrientHeaderType">
1174       <xsd:sequence>
1175         <xsd:element name="preparationStateCode"
1176 type="tsd_common:TSD_PreparationStateCodeType" />
1177         <xsd:element name="dailyValueIntakeReference" type="shared_common:Description70Type"
1178 minOccurs="0" maxOccurs="unbounded" />
1179         <xsd:element name="servingSizeDescription" type="shared_common:Description70Type"
1180 minOccurs="0" maxOccurs="unbounded" />
1181         <xsd:element name="servingSize" type="shared_common:MeasurementType" minOccurs="0"
1182 maxOccurs="unbounded" />
1183         <xsd:element name="nutrientBasisQuantity" type="shared_common:MeasurementType"
1184 minOccurs="0" maxOccurs="unbounded" />
1185         <xsd:element name="numberOfServings" type="xsd:decimal" minOccurs="0" />
1186         <xsd:element name="nutrientDetail"
1187 type="nutritional_product_information_module:TSD_NutrientDetailType" maxOccurs="unbounded" />
1188         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0" />
1189       </xsd:sequence>
1190     </xsd:complexType>
1191     <xsd:complexType name="TSD_NutritionalProductInformationModuleType">
1192       <xsd:sequence>
1193         <xsd:element name="nutrientHeader"
1194 type="nutritional_product_information_module:TSD_NutrientHeaderType" maxOccurs="unbounded" />
1195         <xsd:element name="avpList" type="tsd_common:AttributeValuePairListType" minOccurs="0" />
1196       </xsd:sequence>
1197     </xsd:complexType>
1198 </xsd:schema>

```

1199 8.3. Interface Messages XML Schema

1200 This section specifies XML schemas that implement request and response message payloads implied
1201 by the abstract definition of service interfaces specified in Section 7.

1202 8.3.1. Aggregator-Aggregator Query Interface (AAQI) XML Schema

1203 This section specifies an XML schema that implement the response message payloads implied by the
1204 abstract definition of the Aggregator-Aggregator Query Interface (AAQI) specified in Section 7.1.

```

1205 <?xml version="1.0" encoding="UTF-8"?>
1206 <xsd:schema xmlns:query_by_gtin_response="urn:gs1:tsd:query_by_gtin_response:xsd:1"
1207 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
1208 xmlns:shared_common="urn:gs1:shared:shared_common:xsd:3"
1209 xmlns:product_data="urn:gs1:tsd:product_data:xsd:1"
1210 xmlns:tsd_common="urn:gs1:tsd:tsd_common:xsd:1"
1211 targetNamespace="urn:gs1:tsd:query_by_gtin_response:xsd:1" elementFormDefault="unqualified"
1212 attributeFormDefault="unqualified" version="1.0">
1213   <xsd:annotation>
1214     <xsd:documentation><![CDATA[-----
1215     Â© Copyright GS1, 2012
1216

```

1217 GS1 is providing this XML Schema Definition file and resultant XML file as a service to
1218 interested industries.
1219 This XML Schema Definition file and resultant XML file were developed through a consensus
1220 process of interested parties.
1221

1222 Although efforts have been made to ensure that the XML Schema Definition file and resultant
1223 XML file are correct, reliable, and technically
1224 accurate, GS1 makes NO WARRANTY, EXPRESS OR IMPLIED, THAT THIS XML Schema Definition file and
1225 resultant XML file ARE
1226 CORRECT, WILL NOT REQUIRE MODIFICATION AS EXPERIENCE AND TECHNOLOGICAL ADVANCES DICTATE, OR
1227 WILL BE SUITABLE FOR
1228 ANY PURPOSE OR WORKABLE IN ANY APPLICATION, OR OTHERWISE. Use of the XML Schema Definition
1229 file and resultant XML
1230 file are with the understanding that GS1 has no liability for any claim to the contrary, or
1231 for any damage or loss of any kind or nature.
1232

1233 Version Information:
1234 Version Number: 1.0
1235 Date of creation: August 2012
1236

1237 The schema and subsequent updates will be provided on the GS1 websites.
1238 -----
1239

```
1239 ]]></xsd:documentation>
1240 </xsd:annotation>
1241 <xsd:import namespace="urn:gsl:shared:shared_common:xsd:3"
1242 schemaLocation="../shared/SharedCommon.xsd"/>
1243 <xsd:import namespace="urn:gsl:tsd:tsd_common:xsd:1" schemaLocation="TSDCommon.xsd"/>
1244 <xsd:import namespace="urn:gsl:tsd:product_data:xsd:1" schemaLocation="ProductData.xsd"/>
1245 <xsd:element name="queryByGtinResponse"
1246 type="query_by_gtin_response:TSD_QueryByGtinResponseType"/>
1247 <xsd:complexType name="TSD_ImplementationExceptionType">
1248 <xsd:complexContent>
1249 <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1250 </xsd:complexContent>
1251 </xsd:complexType>
1252 <xsd:complexType name="TSD_InvalidGTINExceptionType">
1253 <xsd:complexContent>
1254 <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1255 </xsd:complexContent>
1256 </xsd:complexType>
1257 <xsd:complexType name="TSD_InvalidRequestExceptionType">
1258 <xsd:complexContent>
1259 <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1260 </xsd:complexContent>
1261 </xsd:complexType>
1262 <xsd:complexType name="TSD_InvalidTargetMarketExceptionType">
1263 <xsd:complexContent>
1264 <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1265 </xsd:complexContent>
1266 </xsd:complexType>
1267 <xsd:complexType name="TSD_NoDataExceptionType">
1268 <xsd:complexContent>
1269 <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1270 </xsd:complexContent>
1271 </xsd:complexType>
1272 <xsd:complexType name="TSD_QueryByGtinResponseCodeType">
1273 <xsd:complexContent>
1274 <xsd:extension base="shared_common:GS1CodeType"/>
1275 </xsd:complexContent>
1276 </xsd:complexType>
1277 <xsd:complexType name="TSD_QueryByGtinResponseType">
1278 <xsd:choice>
1279 <xsd:element name="productData" type="product_data:TSD_ProductDataType"/>
1280 <xsd:element name="noDataException"
1281 type="query_by_gtin_response:TSD_NoDataExceptionType"/>
1282 <xsd:element name="invalidGTINException"
1283 type="query_by_gtin_response:TSD_InvalidGTINExceptionType"/>
1284 <xsd:element name="invalidTargetMarketException"
1285 type="query_by_gtin_response:TSD_InvalidTargetMarketExceptionType"/>
1286 <xsd:element name="unsupportedVersionException"
1287 type="query_by_gtin_response:TSD_UnsupportedVersionExceptionType"/>
```

```

1288     <xsd:element name="invalidRequestException"
1289 type="query_by_gtin_response:TSD_InvalidRequestExceptionType"/>
1290     <xsd:element name="securityException"
1291 type="query_by_gtin_response:TSD_SecurityExceptionType"/>
1292     <xsd:element name="implementationException"
1293 type="query_by_gtin_response:TSD_ImplementationExceptionType"/>
1294     </xsd:choice>
1295   </xsd:complexType>
1296   <xsd:complexType name="TSD_SecurityExceptionType">
1297     <xsd:complexContent>
1298       <xsd:extension base="tsd_common:TSD_ExceptionType">
1299         <xsd:sequence>
1300           <xsd:element name="exceptionContactDescription"
1301 type="shared_common:Description200Type" minOccurs="0"/>
1302         </xsd:sequence>
1303       </xsd:extension>
1304     </xsd:complexContent>
1305   </xsd:complexType>
1306   <xsd:complexType name="TSD_UnsupportedVersionExceptionType">
1307     <xsd:complexContent>
1308       <xsd:extension base="tsd_common:TSD_ExceptionType"/>
1309     </xsd:complexContent>
1310   </xsd:complexType>
1311 </xsd:schema>

```

1312 9. Transport Bindings

1313 This section defines concrete implementations of the service interfaces defined in Section 7 using
 1314 particular transport technologies.

1315 9.1. Web Service Bindings

1316 This section defines a concrete implementation of the AAQI service interface (Section 7.1) using
 1317 REST-style web services, where the response payloads are based on the XML schemas defined in
 1318 Section 8. See Section 9.2 for a binding of the AIQI service interface (Section 7.2) based on ONS. In
 1319 this version of the standard, no binding of the AIMI to a concrete service is specified.

1320 9.1.1. REST Web Service Bindings

1321 This section specifies an implementation of the AAQI service interface (Section 7.1) using web
 1322 services defined in a Representation State Transfer (REST) style. This style is characterized by:

- 1323 ■ HTTP is used as the transport protocol.
- 1324 ■ The request is an HTTP GET request, where the request parameters are represented as URL
 1325 query parameters.
- 1326 ■ The response payload is the XML response document defined in Section 8
- 1327 ■ HTTP response code 200 indicates success. HTTP response codes not beginning with “2”
 1328 are used to indicate exceptions.

1329 The specifics are defined in the following sections.

1330 9.1.1.1. Protocol

1331 In the REST web service binding for the AAQI the client of the interface SHALL send a request using
 1332 HTTP, and the service SHALL respond using HTTP. Both the client and service SHALL conform to
 1333 HTTP 1.1.

1334 9.1.1.2. Security

1335 In the REST web service binding for the AAQI the client and service SHALL use Transport Level
 1336 Security (TLS) as defined in [RFC2818]. TLS for this purpose SHALL be implemented as defined in
 1337 [RFC2246] except that the mandatory cipher suite is TLS_RSA_WITH_AES_128_CBC_SHA, as defined
 1338 in [RFC3268] with CompressionMethod.null. Implementations MAY support additional cipher suites
 1339 and compression algorithms as desired.

1340 Mutual authentication of the AAQI client and service is provided through the use of message
 1341 authentication codes as follows.

1342 Prior to interaction via AAQI, data aggregators establish trust in the following manner. Data
 1343 Aggregator A provides to Data Aggregator B a GLN that uniquely identifies Aggregator A as a TSD
 1344 Data Aggregator and a symmetric key (as defined below) that Data Aggregator A will use to
 1345 authenticate its identity to Data Aggregator B in the request and vice versa in the response. This key
 1346 is referred to as $K_{A \rightarrow B}$. Data Aggregator B provides to Data Aggregator A the service URL that Data
 1347 Aggregator B registers in the Global Index for GTINs for which B provides data. This exchange of
 1348 information provides the ability for Data Aggregator A to issue requests to Aggregator B. As B will
 1349 likely wish to issue requests to A the converse set of information is also exchanged, including a
 1350 different key $K_{B \rightarrow A}$. Each symmetric key K SHALL be 256 bits in length.

1351 When a client Data Aggregator A makes an AAQI request to a service Data Aggregator B, it SHALL
 1352 include a Message Authentication Code (MAC) in the request URL calculated in the following manner:

- 1353 1. Data Aggregator A constructs the request URL according to Section 9.1.1.3, including its GLN (as
 1354 provided to Data Aggregator B during trust establishment) as the value of the `clientGln`
 1355 parameter. At this stage, the request URL does not include the `mac` parameter.
- 1356 2. Data Aggregator A computes a message authentication code using the HMAC algorithm
 1357 [RFC2104] with SHA-256 [RFC4634] as the digest algorithm (a combination commonly referred to
 1358 as HMAC-SHA256). The data input to the HMAC algorithm is the path and query portions of the
 1359 request URL including the initial slash character and including all query parameters in the order
 1360 chosen by Data Aggregator A. These URL characters shall be treated as octets in the US-ASCII
 1361 encoding for purposes of this calculation. The key input to the HMAC algorithm is $K_{A \rightarrow B}$
 1362 as previously shared during trust establishment, which Data Aggregator A looks up as a function of
 1363 B's service URL.

1364 For example, if the request URL (prior to adding the `mac` parameter) is the following:

1365 [http://tsd.example.com/service/v1/ProductData/gtin/00614141123452?target](http://tsd.example.com/service/v1/ProductData/gtin/00614141123452?targetMarket=840&dataVersion=1.0&clientGln=0614141000005)
 1366 [Market=840&dataVersion=1.0&clientGln=0614141000005](http://tsd.example.com/service/v1/ProductData/gtin/00614141123452?targetMarket=840&dataVersion=1.0&clientGln=0614141000005)

1367 then the input to the HMAC algorithm are the following characters, encoded as US-ASCII octets:

1368 `/service/v1/ProductData/gtin/00614141123452?targetMarket=840&dataVersion`
 1369 `=1.0&clientGln=0614141000005`

- 1370 3. The resulting message authentication code is converted to a 64-character hexadecimal string
 1371 (using uppercase letters) and appended to the request URL as the value of the `mac` parameter.
- 1372 4. The final URL is used to make the HTTP request.

1373 The server Data Aggregator B SHALL authenticate the identity of the client Data Aggregator A in the
 1374 following manner:

- 1375 1. Data Aggregator B receives the HTTP request.
- 1376 2. Using the value of the `clientGln` parameter Data Aggregator B looks up the corresponding key
 1377 $K_{A \rightarrow B}$. If no such key is known, stop: the client is unknown and a `SecurityException` is raised.
- 1378 3. Data Aggregator B computes a message authentication code using HMAC-SHA256 where the
 1379 data input to the HMAC algorithm is the path and query portions of the request URL including the

1380 initial slash character and including all query parameters except the `mac` parameter and where the
 1381 key is $K_{A \rightarrow B}$.

1382 4. Data Aggregator B compares the MAC computed in the previous step to the MAC included in the
 1383 request URL's `mac` parameter. If the MACs are identical, then Data Aggregator A's identity is
 1384 authenticated. Otherwise, stop: the client is not authenticated and a `SecurityException` is
 1385 raised.

1386 The service Data Aggregator does not authenticate to the client Data Aggregator prior to the request
 1387 being delivered from client to server. Instead, the service Data Aggregator provides a MAC in the
 1388 response so that the client may authenticate the server when it receives the response.

1389 When a service Data Aggregator B responds to an AAQI request made by client Data Aggregator A, it
 1390 SHALL include a Message Authentication Code (MAC) in the following manner:

1391 1. Data Aggregator B computes a message authentication code using the HMAC-SHA256 where the
 1392 data input to the HMAC algorithm is the entire HTTP response payload. The payload shall be
 1393 treated as a sequence of bytes for the purposes of this calculation. The key input is $K_{A \rightarrow B}$
 1394 as previously shared during trust establishment, which Data Aggregator B looks up as a function of
 1395 the `clientGln` parameter included in the request.

1396 2. The resulting message authentication code is converted to a 64-character hexadecimal string
 1397 (using uppercase letters) and included in the HTTP response as the value of an HTTP header
 1398 whose name is `GS1-MAC`.

1399 The client Data Aggregator A SHALL authenticate the identity of server Data Aggregator B in the
 1400 following manner:

- 1401 1. Data Aggregator A receives the HTTP response.
- 1402 2. Data Aggregator A retrieves the key $K_{A \rightarrow B}$ previously used to create the request.
- 1403 3. Data Aggregator A computes a message authentication code using HMAC-SHA256 where the
 1404 data input to the HMAC algorithm is the response payload and where the key is $K_{A \rightarrow B}$.
- 1405 4. Data Aggregator A compares the MAC computed in the previous step to the MAC included in the
 1406 response's `GS1-MAC` HTTP header. If the MACs are identical, then Data Aggregator B's identity is
 1407 authenticated. Otherwise, stop: the server is not authenticated and the response should not be
 1408 trusted.

1409 9.1.1.3. Request

1410 A Data Aggregator acting as an AAQI client using the REST binding SHALL implement each interface
 1411 method by formulating an HTTP request as specified in this section. A Data Aggregator acting as an
 1412 AAQI server using the REST binding SHALL interpret an HTTP request as an invocation of an
 1413 interface method as specified in this section and respond accordingly.

1414 The HTTP URL for the request SHALL have the following form:

1415 *Base/v1/Resource?Parameters*

1416 where:

- 1417 ■ *Base* is an absolute HTTP (or HTTPS) URL, determined as follows:
 - 1418 □ For the AAQI, *Base* is a URL obtained through the AAQI as the
 1419 `dataAggregatorService` data element contained in an AIQI method response.
- 1420 ■ The two characters `v1` denote that the operation being invoked is as specified in this version
 1421 of the TSD standard. These characters are reserved to provide for a staged transition to an
 1422 incompatible version of the TSD standard in the future. See Section 5.4.2.

1423 ■ *Resource* is a URL segment as specified below according to the interface operation to be
 1424 invoked by this request and the subject of that operation

1425 ■ *Parameters* is an HTTP query string encoding parameters for the request beyond those
 1426 encoded into *Resource*.

1427 The HTTP method SHALL be GET and the HTTP request payload SHALL be empty.

1428 The following table specifies the requests for the AAQI:

Interface operation	<i>Resource</i> portion of HTTP URL	HTTP Method	<i>Parameters</i> portion of HTTP URL	Request payload
queryByGtin	As specified below	GET	As specified below	Empty

1429 The *Resource* portion of the HTTP URL for the AAQI queryByGtin operation SHALL have the
 1430 following form:

1431 `ProductData/gtin/gtin`

1432 (Note that the second segment of the *Resource* portion is the four-character string *gtin* while the
 1433 third segment is the 14-digit GTIN of the product.)

1434 The *Parameters* portion of the HTTP URL for the AAQI queryByGtin operation SHALL have the
 1435 following form:

1436 `targetMarket=targetMarket&dataVersion=version&clientGln=clientGln&mac=mac`

1437 where:

- 1438 ■ *gtin* is the GTIN for the request (see Section 7.1.1).
- 1439 ■ *targetMarket* is the target market code for the request (see Section 7.1.1).
- 1440 ■ *dataVersion* is the desired data version for the request (see Section 7.1.1).
- 1441 ■ *clientGln* is a 13-digit GLN that identifies the client; see Section 9.1.1.2.
- 1442 ■ *mac* is a 64-character hexadecimal numeral (using uppercase letters) whose value is the
 1443 message authentication code for the request as computed according to Section 9.1.1.2.

1444 9.1.1.4. Response

1445 A Data Aggregator acting as an AAQI server using the REST binding SHALL respond to an HTTP
 1446 request as specified in this section. A Data Aggregator acting as an AAQI client using the REST
 1447 binding SHALL interpret an HTTP response as specified in this section as a response to an interface
 1448 operation.

1449 A REST implementation of the AAQI SHALL process an incoming HTTP request as follows:

- 1450 ■ If the HTTP URL matches the base URL and resource portion of the URL as specified in
 1451 Section 9.1.1.3, then:
 - 1452 □ If the HTTP method (GET) matches the HTTP method specified in Section 9.1.1.3 for the
 1453 operation named in the URL, then process the request as specified below.
 - 1454 □ Otherwise, the HTTP method is inappropriate: respond with HTTP error code 405, with no
 1455 HTTP response payload
- 1456 ■ Otherwise, the request is outside the scope of this standard, and the response is
 1457 implementation-defined. If the request is considered invalid by the implementation, the
 1458 implementation SHOULD respond with HTTP error code 400 or 404.

1459
1460
1461
1462
1463
1464
1465
1466
1467
1468

Except as noted above, the response payload for the AAQI SHALL be an XML document conforming to the schema specified in Section 8.3.1 whose top level element is `queryByGtinResponse` and where the element contained immediately within the top level element is as specified in the table below. The response SHALL include an HTTP header named `GS1-MAC` whose value is a 64-character hexadecimal numeral (using uppercase letters) whose value is the message authentication code for the response as computed according to Section 9.1.1.2, except that the server MAY omit the MAC in the case of a `securityException` or `implementationException`. The server SHOULD include the message authentication code in the response for those exceptions if the client identity and its key are known to the server. The following table also specifies the HTTP response code for each outcome:

Outcome	HTTP response code	XML element contained within <code>queryByGtinResponse</code>
Success	200	<code>productData</code>
<code>NoDataException</code>	404	<code>noDataException</code>
<code>InvalidGtinException</code>	400	<code>invalidGtinException</code>
<code>InvalidTargetMarketException</code>	400	<code>invalidTargetMarketException</code>
<code>UnsupportedVersionException</code>	400	<code>unsupportedversionException</code>
<code>SecurityException</code>	403	<code>securityException</code>
<code>InvalidRequestException</code>	400	<code>invalidRequestException</code>
<code>ImplementationException</code>	500	<code>implementationException</code>

1469
1470
1471

9.2. ONS 2.0 Binding for Aggregator-Index Query Interface (AIQI)

This section defines a concrete implementation of the Aggregator-Index Query Interface (AIQI) defined in Section 7.2 using ONS 2.0.

1472
1473
1474
1475

ONS 2.0 is an application of the Internet Domain Name System (DNS) which provides a means to look up a GS1 key and obtain a list of one or more services registered for that key. In the ONS binding for the AIQI, the GS1 key is the GTIN in the `queryByGtin` request, and the services registered in ONS are references to Data Aggregators that have data for that GTIN.

1476

9.2.1. AIQI Query Via ONS

1477
1478
1479

A Data Aggregator using the ONS 2.0 binding of the AIQI SHALL formulate a query by constructing the following Application Unique String (AUS), which [ONS_20] Section 6.1 defines as the input to the ONS 2.0 query process:

1480

```
|country|gtin|nnnnnnnnnnnnnnnn
```

1481

where:

1482
1483
1484
1485
1486
1487

- `country` is replaced by the 2-character ISO 3166-1 alpha-2 country code [ISO3166] for the desired target market, converted to lowercase. Note that the 3-digit `CountryCode` as used in the abstract interface definition specified in Section 7.1.1 must be translated to the corresponding 2-character ISO 3166-1 alpha-2 code for the purposes of formulating the AUS. While ISO 3166-1 defines the alpha-2 code as uppercase, it must be converted to lowercase for this purpose.

1488 ■ *nnnnnnnnnnnnnnnn* is replaced by the 14-digit GTIN for which product data is requested. If
 1489 product data for a GTIN-8, GTIN-12, or GTIN-13 is requested, the Data Aggregator SHALL
 1490 prepend as many zero ('0') digits as required to make 14 digits total in the AUS.

1491 The Data Aggregator SHALL make a DNS query based on the AUS as specified in [ONS_20],
 1492 Sections 6.2 and 7.1. If the Data Aggregator desires to include two or more target markets in the
 1493 query, it may make a single DNS query because the country part of the AUS does not enter into the
 1494 fully-qualified domain name (FQDN) computed from the AUS according to [ONS_20]. However, in
 1495 matching the NAPTR records that are returned, the Data Aggregator SHALL formulate separate AUSs
 1496 for each target market and treat each independently to match returned NAPTR records.

1497 The response to the DNS query consists of zero or more DNS NAPTR records. The Data Aggregator
 1498 SHALL apply the procedure specified in [ONS_20] Section 8 to obtain one or more service URLs for
 1499 peer Data Aggregators. In applying this procedure, the Data Aggregator SHALL implement the
 1500 procedure specified in [ONS_20] Section 8 as follows:

1501 ■ In Step 2, only consider those NAPTR records whose Service field contains the following
 1502 service type URL:

1503 `http://ons.gs1.org/tsd/servicetype/aaqi-1.xml`

1504 ■ In Step 4, rather than attempting to find the first NAPTR record that matches the criteria,
 1505 continue processing all NAPTR records and gather all service URLs that match the criteria. In
 1506 this way, multiple peer Data Aggregators may be identified as a result.

1507 **9.2.2. NAPTR Records for TSD**

1508 A Global Index that implements the ONS 2.0 binding of AIQI must maintain DNS NAPTR records that
 1509 will be interpreted properly by the procedures specified in Section 9.2.1.

1510 Each NAPTR record defines the Data Aggregator registered for a particular combination of GTIN,
 1511 target market, and language code. Nominally, each NAPTR record SHALL be associated to a DNS
 1512 domain name resulting from transforming a GTIN according to [ONS_20], Section 6.2. If two or more
 1513 GTINs have digits in common such that their corresponding DNS domain names share a common
 1514 suffix, and those GTINs would have an identical set of NAPTR records, then the set of NAPTR records
 1515 MAY be associated to a higher level DNS domain name using the DNS wildcard mechanism.

1516 Each NAPTR record SHALL contain data as specified in [ONS_20], Section 7.2, and as noted below:

1517 ■ The Flags field SHALL consist of the single character 'u', indicating that the Regexp field
 1518 contains a URI.

1519 ■ The Service field SHALL consist of the following service type URL:

1520 `http://ons.gs1.org/tsd/servicetype/aaqi-1.xml`

1521 ■ The Regexp field SHALL be composed by concatenating the following strings:

1522 □ The characters `!^\\|` (exclamation point, caret, backslash, and vertical bar)

1523 □ The two-letter ISO 3166-1 alpha-2 country code corresponding to the target market to
 1524 which this NAPTR record applies, converted to lowercase. More complex regular
 1525 expressions may be used to allow this NAPTR record to match more than one target
 1526 market; see [ONS_20].

1527 □ The characters `\\. *$!` (backslash, vertical bar, period, asterisk, dollar sign, and
 1528 exclamation point)

1529 □ The service URL for the Data Aggregator to which this NAPTR record refers

1530 □ A `!` character (exclamation point)

1531