



The Global Language of Business

GS1 Web URI Structure Standard

enabling consistent representation of GS1 identification keys within web addresses to link to online information and services

Release 1.0, Ratified, Aug 2018

Document Summary

Document Item	Current Value
Document Name	GS1 Web URI Structure Standard
Document Date	Aug 2018
Document Version	1.0
Document Issue	
Document Status	Ratified
Document Description	enabling consistent representation of GS1 identification keys within web addresses to link to online information and services

Contributors

Name	Organisation
Dominique Guinard (Chair)	EVERYTHING
Seán Lightholder (Chair)	Wal-Mart Stores, Inc.
Laurent Tonnelier (Chair)	mobiLead / serviceTag
Hirokazu Nagai	Japan Pallet Rental Corporation
Deborah Arcoleo	Hershey Company (The)
Stephen Brown	Mead Westvaco
Henk Dannenberg	NXP Semiconductors
Vera Feuerstein	Nestle
Brian Newby	Unilever UK
Mandeep Sodhi	Nestle
John Terwilliger	Abbott Laboratories Inc.
Julie Vargas	Avery Dennison RFID
Evan Bacchus	Costco Wholesale
Yolande Diaz	Carrefour
Jonas Elander	Axfood Sverige AB
Max Engström	H&M
Sylvia Rubio Alegren	ICA AB
Hans Peter Scheidt	C & A SCS
Joachim Wilkens	C & A SCS
Marc Blanchet	Viagenie
Philip Allgaier	bpcompass GmbH
Dalibor Biscevic	Business Technologies Ltd
Patrick Chanez	INEXTO SA
Roland Donzelle	SQUARE / TINTAMAR
Anthony Graignic	YESITIS
Mark Harrison	Milecastle Media Limited
John Herzig	Barcode Graphics Inc Canada

Name	Organisation
Andr, Machado	TrustaTAG
Zbigniew Sagan	Advanced Track and Trace
Kim Simonalle	Qliktag Software (formally Zeebric LLC)
Elizabeth Waldorf	TraceLink
George Wright IV	Product Identification & Processing Systems
Ilteris Oney	ecomcs
Koji Asano	GS1 Japan
Andrea Ausili	GS1 Italy
Xavier Barras	GS1 France
Mats Bjorkqvist	GS1 Sweden
Jonas Buskenfried	GS1 Sweden
Emanuela Casalini	GS1 Italy
Karolin Catela	GS1 Sweden
Madalina Cernat	GS1 Romania
Anthony Chan	GS1 Hong Kong
Nicolas Collignon	GS1 Belgium & Luxembourg
Luiz Costa	GS1 Brazil
Benjamin Couty	GS1 France
Leonardo de Viveiros	GS1 Brazil
Kevin Dean	GS1 Canada
Ferran Domenech Fuste	GS1 Spain
Xiaowen Dong	GS1 China
Dorothee Doswald-Kuhlmann	GS1 Switzerland
Vanessa Giulieri	GS1 Italy
Heinz Graf	GS1 Switzerland
Rami Habbal	GS1 UAE
Gary Hartley	GS1 New Zealand
Hideki Ichihara	GS1 Japan
Yoshihiko Iwasaki	GS1 Japan
Ricky Jones	GS1 UK
Sang Ik Jung	GS1 Korea
Kimmo Keravuori	GS1 Finland
Kazuna Kimura	GS1 Japan
Boris Krcelic	GS1 Argentina
Ilka Macherer	GS1 Germany
Noriyuki Mama	GS1 Japan
Roberto Matsubayashi	GS1 Brazil
Daniel Mueller-Sauter	GS1 Switzerland
Prince Namane	GS1 South Africa
Mori Naoko	GS1 Japan



Name	Organisation
Daisuke Negishi	GS1 Japan
Staffan Olsson	GS1 Sweden
Michel Ottiker	GS1 Switzerland
Sebastián Perazzo	GS1 Argentina
Bijoy Peter	GS1 India
Sarina Pielaat	GS1 Netherlands
Neil Piper	GS1 UK
Paul Reid	GS1 UK
Alexandre Rieucou	GS1 France
Nick Rusman	GS1 Netherlands
Roxana Saravia Bulmini	GS1 Argentina
Sue Schmid	GS1 Australia
Eugen Sehorz	GS1 Austria GmbH
Xiaojing Shao	GS1 China
Sean Sloan	GS1 Australia
Andrew Steele	GS1 Australia
Xia Tao	GS1 China
Ralph Troeger	GS1 Germany
Frits van den Bos	GS1 Netherlands
Rocio Vizcarra	GS1 Argentina
Amber Walls	GS1 US
Qian Wang	GS1 France
Shu Wang	GS1 China
Yi Wang	GS1 China
Stephan Wijnker	GS1 Australia
Carrie Wilkie	GS1 US
Ruoyun Yan	GS1 China
Yuwen Zhang	GS1 China
Peter Alvarez	GS1 Global Office
Dipan Anarkat	GS1 Global Office
Phil Archer	GS1 Global Office
Henri Barthel	GS1 Global Office
Robert Beideman	GS1 Global Office
Chuck Biss	GS1 Global Office
Scott Gray	GS1 Global Office
Eric Kauz	GS1 Global Office
Sean Lockhead	GS1 Global Office
Jeanette McVeigh	GS1 Global Office
Markus Mueller	GS1 Global Office
Greg Rowe	GS1 Global Office

Name	Organisation
Kevin Stark	GS1 Global Office
Garry Thaniel	GS1 Global Office
Elena Tomanovich	GS1 Global Office

Log of Changes

Release	Date of Change	Changed By	Summary of Change
1.0	Aug 2018	Phil Archer, Dominique Guinard, Mark Harrison, Marie Petre & Greg Rowe	Initial release developed on WR 17-000343

Disclaimer

GS1[®], under its IP Policy, seeks to avoid uncertainty regarding intellectual property claims by requiring the participants in the Work Group that developed this GS1 Web URI Standard to agree to grant to GS1 members a royalty-free licence or a RAND licence to Necessary Claims, as that term is defined in the GS1 IP Policy. Furthermore, attention is drawn to the possibility that an implementation of one or more features of this Standard may be the subject of a patent or other intellectual property right that does not involve a Necessary Claim. Any such patent or other intellectual property right is not subject to the licencing obligations of GS1. Moreover, the agreement to grant licences provided under the GS1 IP Policy does not include IP rights and any claims of third parties who were not participants in the Work Group.

Accordingly, GS1 recommends that any organisation developing an implementation designed to be in conformance with this Standard should determine whether there are any patents that may encompass a specific implementation that the organisation is developing in compliance with the Standard and whether a licence under a patent or other intellectual property right is needed. Such a determination of a need for licencing should be made in view of the details of the specific system designed by the organisation in consultation with their own patent counsel.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims all liability for any damages arising from use or misuse of this document, whether special, indirect, consequential, or compensatory damages, and including liability for infringement of any intellectual property rights, relating to use of information in or reliance upon this document.

GS1 retains the right to make changes to this document at any time, without notice. GS1 makes no warranty for the use of this document and assumes no responsibility for any errors which may appear in the document, nor does it make a commitment to update the information contained herein.

GS1 and the GS1 logo are registered trademarks of GS1 AISBL.

Table of Contents

- 1 Executive overview 8**
- 2 Foreword 8**
- 3 Introduction 9**
 - 3.1 What is a URI? 9
 - 3.2 Why Linked Data? 12
 - 3.3 Use of GS1 Web URIs with various kinds of data carriers..... 12
- 4 Working assumptions and technical design principles 14**
 - 4.1 Do no harm to the existing GS1 System Architecture and existing identification practices 14
 - 4.2 Use of barcodes containing GS1 Web URI and GS1 barcodes 15
 - 4.3 A note on terminology 16
 - 4.4 Security and privacy of information 17
 - 4.5 Contextual information 17
 - 4.6 Access to traceability data by various supply chain stakeholders 17
 - 4.7 Supporting machine-to-machine communication and expressing facts and relationships between things and about things..... 18
 - 4.8 Explicit link to the GS1 identification key(s) 20
 - 4.9 Validation principles for GS1 Web URIs 20
- 5 GS1 Web URI syntax..... 20**
 - 5.1 Formal specification..... 20
 - 5.1.1 Character sets 20
 - 5.1.2 Primary identification keys 22
 - 5.1.3 Key qualifiers 22
 - 5.1.4 Primary key formats..... 23
 - 5.1.5 Key qualifier formats..... 24
 - 5.1.6 Primary identifier and value concatenation 24
 - 5.1.7 Key qualifier concatenation 24
 - 5.1.8 Path element order 25
 - 5.1.9 Data attributes 25
 - 5.1.10 Constructing the URI 33
 - 5.2 Canonical GS1 Web URIs..... 34
 - 5.3 An explanation of Web URI syntax and options for extensions..... 35
 - 5.4 Examples of GS1 Web URIs 36
 - 5.4.1 GTIN 36
 - 5.4.2 GTIN + CPV 36
 - 5.4.3 GTIN + Batch/Lot 36
 - 5.4.4 GTIN + Serial Number (also known as SGTIN)..... 37
 - 5.4.5 GTIN + Batch/Lot + Serial Number + Expiry Date..... 37
 - 5.4.6 GTIN + Net Weight 37
 - 5.4.7 GTIN + Typed Link..... 38
 - 5.4.8 GTIN + Batch/Lot + Serial Number + Expiry Date + Typed Link 38
 - 5.4.9 SSCC 39
 - 5.4.10 SSCC with specified Content, Count and Batch/Lot..... 39
 - 5.4.11 Physical location represented by a GLN or GLN + GLN Extension 39



6 Resolving GS1 Web URIs 40

- 6.1 Resolving Protocol 40
- 6.2 Extraction step 41
- 6.3 Validation step..... 41
 - 6.3.1 Check the length 42
 - 6.3.2 Check the character set 42
 - 6.3.3 Check the GS1 check digit (where appropriate) 42
 - 6.3.4 Check the internal database of identifiers supported by this resolver 43
- 6.4 Resolution step 44
 - 6.4.1 A note on HTTP response codes: 44
 - 6.4.2 When a resolver has only partial information..... 44
- 6.5 Examples of use..... 44
 - 6.5.1 Using the provided domain to call the resolver 44
 - 6.5.2 Replacing the domain to call an alternative resolver 45
 - 6.5.3 Retailer operates their own resolver 46
 - 6.5.4 Smartphone app for nutrition / dietary / allergen information 47
 - 6.5.5 Product comparison apps 48

7 Glossary..... 48

8 References..... 52

A Intellectual property..... 54

1 Executive overview

GS1 has been strong in product identification for many years - but GS1 identification keys have been designed to serve use cases that are primarily driven by the needs of supply chains (and traditionally are most relevant from manufacturer to checkout). In the 21st century, that's not enough. As a result of this standard, it will be possible to represent GS1 identification keys consistently within Web addresses (Web URIs) as well as within the barcodes containing GS1 Web URI, such that a single identification approach will be able to support both product identification for supply chain applications and a link to online material. It's this dual functionality and enormous flexibility that is currently not possible when, for example, Brand Owners embed an unstructured Web page address in a QR Code®.

The use of 2D barcodes like ISO/IEC 18004 QR Code® or ISO/IEC 16022 DataMatrix ECC 200, in addition to 1D barcodes on consumer packs, is a core driver of this work. Consumers routinely seek product information on the Web, especially via mobile, and manufacturers may respond to this demand by adding a second machine readable barcode to the product. For retail products, this is usually a QR Code®¹ that encodes the address of a Web page from which product and marketing information is available. SmartLabel™ is a high-profile example of this, where Web addresses (Web URIs) are encoded into QR Codes® today with little or no standardised structure. However, we also need publicly available standard references or services that could support the transformation of the GS1 data encoded in the GS1 1D/2D barcode into this standard Web URI syntax. GS1 DataMatrix without a URI/URL encoded in the barcode in regulated healthcare is the principle data carrier and is another driver for this standard, in order to support the GS1 Digital Bridge Service concept.

The scope of the work accommodates all Class 1 and Class 2 GS1 Keys and Key qualifiers (e.g., serial number, batch number, consumer product variant) and other relevant attributes as the same technologies are equally applicable to SSCCs, GLNs, GIAIs, GRAIs, GSRNs etc. While the syntax can support Class 2 Keys, it is up to the Class 2 Issuing Agencies to determine its fit for their use. For Class 3 GS1 Keys, GS1 welcomes bilateral discussions with Issuing Agencies to see where alignment is possible.

This GS1 standard references a number of third-party standards from the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).

2 Foreword

GS1 defines a wide range of identifiers that underpin the supply chain and retail industry across the world. This document assumes the reader is familiar with these and the concept of GS1 Application Identifiers. If not, please see information on [GS1 identification keys] and the [GENSPECS] for further background. Readers who are already very familiar with the terms URI, URN and URL, and the differences between these, can skip section [3.1](#).

This work has been motivated by a number of trends. For example: the desire among retailers to move to 2D barcodes that can carry more information than just the GTIN; the problems of multiple barcodes causing scanning errors through conflicts which suggests a need for a single but multipurpose barcode; the growing expectation among consumers that more information is available online about the products they're considering buying; the brand owner concept of the pack as a media channel linking to multimedia experiences, and more.

Section [3.2](#) provides a high-level overview of why Web technologies are appropriate to meet these needs. However, for this to work fully within the existing GS1 system, it must be possible to translate between GS1 Web URIs and GS1 element strings such that the two are completely interchangeable. Achieving that requires the level of precision in the structure of a GS1 Web URI that this document provides.

¹ Unless otherwise specified, the term 'QR Code®' refers to the widely used [ISO / IEC 18004 QR Code®](#), not GS1 QR Codes. 'QR Code' is a registered trademark of Denso Wave, a subsidiary of Denso Corporation.

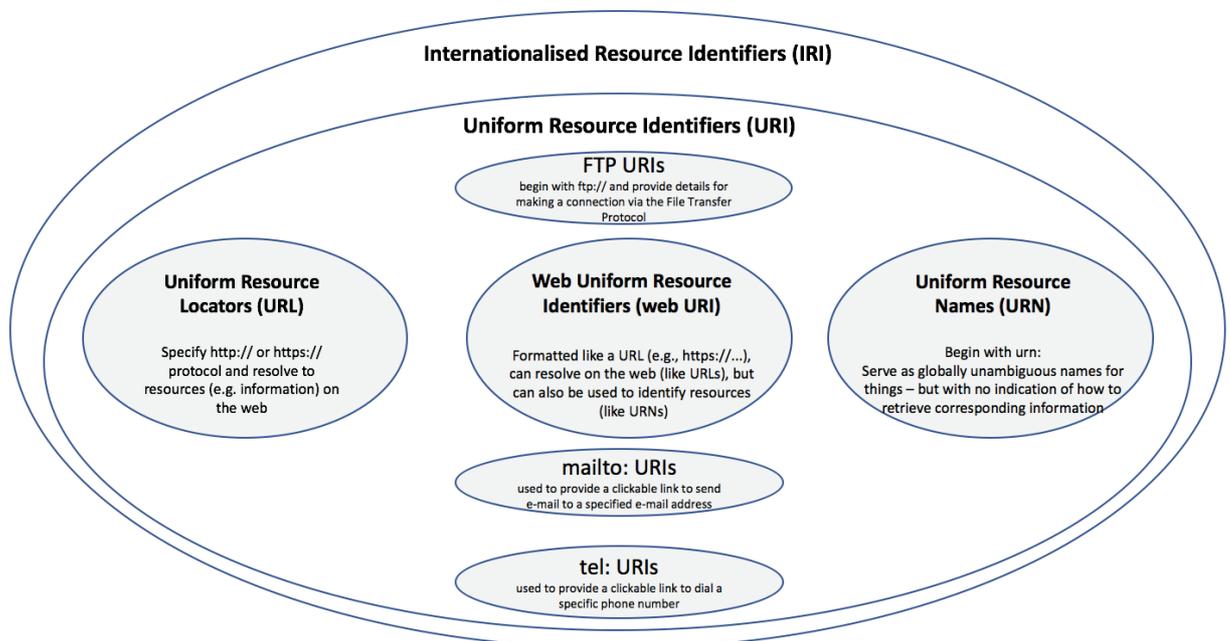
3 Introduction

3.1 What is a URI?

This subsection provides some clarification about what a Uniform Resource Identifier (URI) is, how URIs relate to Uniform Resource Names (URNs) and Uniform Resource Locators (URLs), as well as providing an explanation of the main structural elements within a Web URI.

[Figure 3-1](#) shows a Venn diagram in which we see that Uniform Resource Identifier is the broad term that includes Uniform Resource Names (URNs) and Uniform Resource Locators (URLs) as well as URIs with various protocols including http or https, ftp, mailto, tel etc. This means that every URL and every URN is also a URI, since URI is the broader umbrella term. Furthermore, Internationalized Resource Identifiers (IRIs) are an even broader category that support characters from the Universal Character Set / Unicode, whereas URIs only support the ASCII character set. IRIs are defined in [IRIs].

Figure 3-1 URNs and URLs are also URIs



[Figure 3-2](#) shows another Venn diagram. This time, it shows two capabilities:

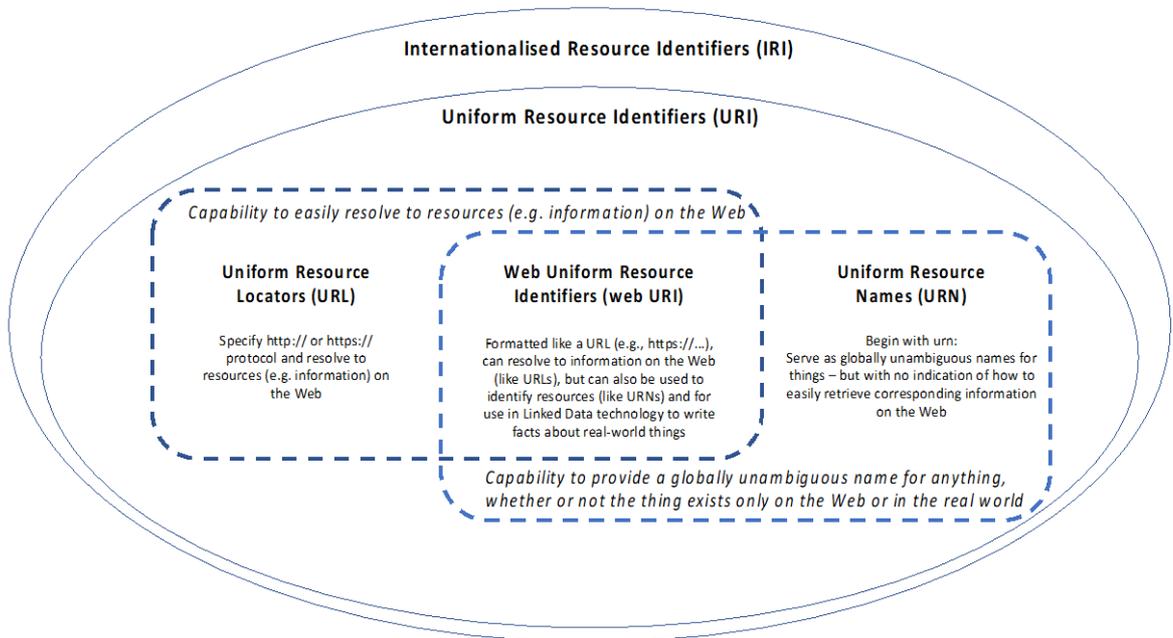
1. The capability to easily resolve to resources (e.g. information) on the Web.
2. The capability to provide a globally unambiguous name for anything, whether or not the thing exists only on the Web or in the real world.

The first capability is usually associated with URLs and Web addresses.

The second capability is usually associated with URNs.

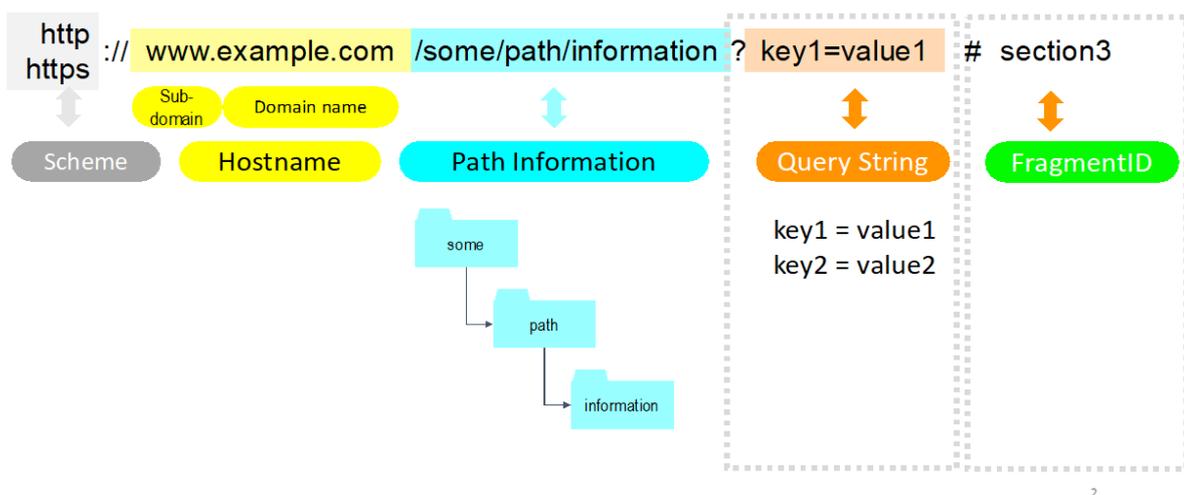
Web URIs exist at the intersection of these two capabilities; in terms of their syntax, they look like URLs because they specify http or https as their protocol - and they can be configured to behave like URLs in terms of supporting Web requests via the http / https Web protocol. However, they are also a perfectly valid way of assigning a globally unambiguous name for anything, whether in the real world or online. Note that 'globally unambiguous' does not mean globally unique; two different things should have distinct URIs in any situation where we want to be able to distinguish between them. However, there may be many URIs that all refer to the same thing, even within the same URI namespace or domain name. It is also possible to use Linked Data [Linked Data] to make an assertion between two URIs to formally express that they both refer to the same thing, even if the URIs are different strings.

Figure 3-2 A Web URI can act both as a globally unambiguous name for something, as well as providing an easy way to retrieve Web resources (e.g. information) relating to the identified thing



[Figure 3-3](#) provides a brief overview of the internal structural elements of a Web URI:

Figure 3-3 Internal structure of a Web URI



[Figure 3-3](#) shows the structural elements of a Web URI. The scheme indicates the protocol and (at the time of writing) is always http:// or https:// (use of HTTPS is recommended as best practice). The hostname is typically a registered Internet domain name or a subdomain of such a registered domain name. Following the domain name, the remainder of the Web URI is case sensitive. The URI path information consists of a number of strings separated by the forward slash character. Although

this is just a string, it is often used by the Linked Data community and in REST interfaces² to represent a collection of resources organised in a conceptually hierarchical way, with the broadest (most general, least specific) category appearing towards the left of the URI path information and with the narrowest (most specific) category appearing towards the right of the URI path information.

This design pattern provides a hint to humans that related Web URIs may exist and can be formed by successively truncating the Web URI path information from right to left, removing each successive segment preceded by its forward slash ("/") character. These related Web URIs may provide information about an object at a broader, more general, less specific granularity.

However, this is only a legible hint to humans. Computer software would typically treat the entire URI (at least up to the fragment identifier) as an opaque indivisible string and would not attempt such truncation. Instead, they will look for explicit links to related URIs, ideally expressed with semantic annotation, using Linked Data properties.

For example, we might formally express that:

```
https://example.com/gtin/{gtin}/ser/{ser} or  
https://example.com/gtin/{gtin}/lot/{lot}/ser/{ser}  
dcterms:isPartOf https://example.com/gtin/{gtin}/lot/{lot}
```

or equivalently,

```
https://example.com/01/{gtin}/21/{ser} or  
https://example.com/01/{gtin}/10/{lot}/21/{ser}  
dcterms:isPartOf https://example.com/01/{gtin}/10/{lot}
```

(These examples follow the convention used in [RFC 6570]. The places where the values of variables should be inserted are written in braces, so, for example, {gtin} means "insert gtin here") As explained in [RFC 2606] and [RFC 6761], the domains example.com, example.org and example.net are second-level domain names reserved by the Internet Assigned Numbers Authority (IANA) for use in documentation. These are used in a number of examples throughout this standard and should be understood as a placeholder for any registered second-level domain name.

It should be noted that a single script or Web Resource may provide information or services for Web URIs with different numbers of levels of path information and that care should be taken when using relative links for images, scripts, stylesheets etc., because most Web clients will treat such a hierarchical URI path as if the Web resource were literally nested within several parent folders.

The query string enables multiple key=value pairs to be sent to a Web resource. The URI query string appears after the URI path information and consists of everything between the "?" at the end of the path information and the end of the URI or the "#" symbol indicating the start of the fragment identifier. Within the URI query string, key=value pairs may be concatenated using & or ; and it should be noted that each key should not appear more than once; the last declared value for a key within the query string is the final value assigned to that key.

The URI fragment identifier is optional and appears after the query string (if present) and preceded by the "#" character. The URI fragment identifier is typically used to provide a link to an internal subsection of an information resource. The Linked Data community do make use of URIs with fragment identifiers, although the fragment identifier is not useful for passing key=value pairs. Importantly, fragment identifiers are *not* sent to the server but are handled entirely within the client.

Web URIs provide essentially two options for expressing the values of GS1 Application Identifiers - either within the URI path information or within the URI query string. The URI path information is the most appropriate place for expressing a GS1 identification key and an ordered set of optional qualifiers that are used in conjunction with the GS1 identification key to form a compound key that is used to retrieve information about something at a finer level of granularity (e.g. traceability data about an SGTIN, lot-level master data). The query string is appropriate for data attributes of the identified resource such as expiry date, weight etc., as well as being a natural extension point for any additional arbitrary key=value pairs that cannot be expressed using GS1 Application Identifiers; for example, the query string could include a key=value pair to indicate a specific stakeholder role or a specific action or activity or type of service to be accessed. Any key=value pairs used for extension data shall not conflict with keys used for GS1 Application Identifiers either in terms of

² See https://en.wikipedia.org/wiki/Representational_state_transfer

semantics or syntax; a key for extension data shall not use an all-numeric value NOR any of the alphabetic short names (e.g. 'lot', 'exp', 'expdt') that are defined as alternatives to the numeric GS1 Application Identifiers for data attributes appearing in the URI query string, key=value pairs should not be used to express a value (such as a value for net weight) if that value can be expressed using GS1 Application Identifiers as data attributes. It should be noted that no key=value pair should be repeated with the same key in the URI query string. If a key is repeated, the last defined value for that key takes precedence over any previously defined value. Conformance test scripts should be developed for implementations of GS1 Web URI and should test for repeats of key=value pairs for the same key, as well as taking into account conversion of any alphabetic key names (specifically 'lot', 'exp' and 'expdt') into their corresponding numeric keys ('10','17' and '7003' respectively) to check that no value is assigned to a key of a key=value pair more than once and that for example 'exp' and '17' are treated as the same key for the purposes of such tests.

Section [5.4.7](#) and [5.4.8](#) provide some examples of use of additional key=value pairs within the URI query string.

3.2 Why Linked Data?

The previous section indicates that Linked Data [Linked Data] is an appropriate technology to satisfy the kind of industries typically served by GS1. It is reasonable to ask why.

Put simply, Linked Data uses the concepts and technologies of the Web to model the real world. It uses the familiar idea of a hyperlink – one thing pointing to another – and adds semantics, that is meaning, to those links. So, for example, an item identified by a GTIN might be linked to its assembly instructions, its nutritional or food safety information; a location identified by a GLN might be linked to a related entry in a geospatial data system, a GSRN-P to contact information about the relevant organisation and so on. These links create a *Network Effect*. The first fax machine only became useful when the second one went online; both became more valuable with the advent of the third and so on. Each node in the network is connected via links that can be understood and processed by computers into a multi-dimensional model, known as a knowledge graph. The best known example of a knowledge graph is the information you often see to the right of search results: tabulated facts about whatever it is you searched for, drawn from multiple sources across the Web. A version of Linked Data is the technology behind that.

Section [4.7](#) has more to say on the topic of Linked Data.

3.3 Use of GS1 Web URIs with various kinds of data carriers

GS1 Web URI provides a new syntax for expressing GS1 Identifier Keys, Key qualifiers and data attributes in a format that can be used on the Web in an intuitive manner (via a straightforward Web request) to enable consumers and others to directly access relevant information and services about products, assets, locations, etc.

GS1 Web URI can be obtained by translation of element strings in existing GS1 data carriers (including 1D and 2D barcodes, EPC RFID tags etc.) and can also be encoded natively in any other data carrier that can support the encoding of a web address or URL. This means that additional data carriers such as QR Codes® and NFC tags will also be able to include GS1 identification keys and function as though they have a URL that links to relevant information.

Many products currently carry an EAN/UPC barcode that simply encodes a GTIN, typically expressed as a GTIN-8, GTIN-12 (UPC) or GTIN-13 (EAN). In such situations, a *reference GS1 Web URI* can always be constructed by simply appending the GTIN value to `https://id.gs1.org/gtin/`, as shown in the examples in section [5.4.1](#).



Note: use of the `id.gs1.org` domain is *not* mandatory. A GS1 Web URI may be constructed under any domain name. Those that do use the `id.gs1.org` domain are referred to as reference GS1 Web URIs. See section [4.3](#) for more.

From identification at GTIN granularity, we can access class-level product master data in human-readable format (e.g. as a product Web page for a specific product) and in a machine-readable format (e.g. using structured data based on `schema.org` and the GS1 Web vocabulary [GS1Voc]), which enables smartphones to access "just the data" and to provide an appropriate display based on

specific data items, such as alerting if a product contains a problematic allergen or has a high fat content.

Moving beyond the current EAN/UPC barcode for a GTIN identifier, some objects carry 2-dimensional GS1 barcodes or matrix codes, such as GS1 Data Matrix or GS1 QR Code or linear barcodes with an optional 2-dimensional component, as is the case for GS1 DataBar. In some sectors, requirements recommend the use of GTIN together with other information such as Batch/Lot or Serial Number. For example, in the healthcare / pharmaceutical sector, it is typical for such GS1 DataMatrix codes to encode four essential elements: GTIN, Batch/Lot, Serial Number and Expiry Date. Without making any changes to current practices for marking identifiers on products, it is possible to translate such information into the GS1 Web URI syntax, to enable access to information and services that use this finer granularity of identification to provide information such as traceability and provenance information or to support services such as warranty registration for a specific instance of a product.

In this scenario, from a GS1 DataMatrix symbol carrying element strings, a reference GS1 Web URI can be constructed by simply inserting the actual values of the GTIN, Batch/Lot, Serial Number and Expiry Date into a URI template that looks like:

```
https://id.gs1.org/gtin/{gtin}/lot/{lot}/ser/{ser}?exp={exp}
```

where {gtin}, {lot}, {ser} and {exp} are actual values for GTIN, Batch/Lot, Serial Number and Expiry Date. Note that some symbol characters need to be percent-encoded when used within a URI - see section [5.1.1](#) for further details.

By translation of the element strings read from GS1 barcodes into a GS1 Web URI syntax, it is possible to access information and services on the Web that are defined at GTIN granularity (as in the previous scenario) and additionally, to access information and services on the Web that are defined at finer granularity, such as traceability / provenance data that is defined at GTIN+batch/lot or GTIN+serial number granularity or warranty registration that is defined at GTIN+serial number granularity.

In retail, some brand owners have already begun adding ISO / IEC 18004 QR codes containing URLs to their products, to enable consumers to link to promotional/marketing pages, competitions etc. SmartLabel also takes a similar approach, although it does not currently include GS1 identification keys within the URI encoded in a QR code. By encoding the GS1 Web URI directly within a QR Code or NFC tag, just as if it were any other URL, we can link it to information and services as well as retaining the ability to extract the GS1 identification keys. This means that a GS1 Web URI encoded within a QR code or NFC tag can either be used by consumers to access product information, promotional/marketing pages, competitions etc. *and* could in future also be used in the supply chain and at point of sale *because* it still encodes the GS1 identification keys within the GS1 Web URI syntax. Translation mechanisms will support translation between element strings and GS1 Web URI syntax. Note that currently, neither NFC tags nor ISO/IEC 18004 QR Codes are approved as GS1 data carriers for use within the supply chain or for point of sale applications.

[Table 3-1](#) provides a summary of the discussion above as it relates to objects identified by GTIN or GTIN plus key qualifiers. Similar considerations apply to other GS1 identification keys.

Table 3-1 Comparison of capabilities of various data carriers

				
--	---	---	---	---

Type of barcode symbol	EAN/UPC barcode containing GTIN	GS1 barcode e.g. GS1 DataMatrix, GS1 DataBar, GS1 QR Code or GS1-128 containing GTIN + other GS1 Application Identifiers	Data carrier containing proprietary URL (e.g. link to marketing page)	Data carrier containing a GS1 Web URI
Can it be used at point of sale and in the supply chain?	Yes	Yes - in some sectors (e.g. healthcare)	No Does not work at point of sale or supply chain	Yes, if specified in GS1 AIDC Application Standards - by translation to element strings
Can it be scanned by consumer smartphones to link to information and services?	Yes - by translation to GS1 Web URI syntax and use of resolvers to redirect	Yes - by translation to GS1 Web URI syntax and use of resolvers to redirect	Yes	Yes. Reference GS1 Web URIs are redirected to corresponding Web URIs specified by each brand owner.
Access to information and services defined at GTIN granularity, e.g. class-level product master data	Yes	Yes	Either no access or vendor-specific proprietary access to information or services about the product	Yes
Access to information and services defined at finer granularity (e.g. traceability/provenance data defined for a GTIN+Lot/Batch or GTIN+Serial), warranty registration for a specific product instance, etc.	No	Yes	Either no access or vendor-specific proprietary access to information or services about the product, possibly at finer granularity than GTIN.	Yes

4 Working assumptions and technical design principles

In this section, the GS1 Web URI MSWG collected working assumptions as well as technical design principles.

4.1 Do no harm to the existing GS1 System Architecture and existing identification practices

The work on GS1 Web URIs is intended to bring additional benefits to more efficiently connect consumers with relevant information and services on the Web about products, locations, assets etc., while also opening up possibilities for additional uses in many industries such as retail and healthcare within their respective supply chains, retailing points, hospitals, pharmacies and the broader area of patient safety. Although element strings already provide the capability to access information at various levels of granularity (e.g. class-level master data accessed via GDSN, lot-level or serial-level traceability data accessed via EPCIS), there was previously no standardised way to link every kind of GS1 identification key to information and services on the Web. The GS1 Web URI syntax defined in this standard provides the ability to express every GS1 identification key at any level of granularity in a Web-native format that functions like a Web address or Web URI and can link to online information on the Web. Such information includes master data but potentially also includes dynamic data such as traceability information. It can be accessed via the use of translation functions and a resolver service for Reference GS1 Web URIs, element strings read from an existing GS1 barcode can be translated into the GS1 Web URI syntax, then used to access

information and services on the Web, using an existing straightforward mechanism, the HTTP / HTTPS Web protocol used to exchange data over the World Wide Web.

Section 3.3 provided some examples of how GS1 Web URI can be used with various data carriers and how it can link to information and services related to identifiers at various levels of granularity, although it introduces no new mechanisms for the generation of such information.

The Web URI shall do no harm to the existing GS1 Architecture, including current practices of encoding 1D/2D GS1 codes (e.g. a GS1 DataMatrix on a pharmaceutical product package encoding AIs 01,21,10,17) as well as operations in GS1 interface standards and data models, especially EPCIS and ALE.

Some elements of the GS1 System (particularly EPCIS and ALE) make extensive use of the EPC URN syntax defined in the GS1 Tag Data Standard. There shall be no ambiguity and confusion in the market for encoding 1D/2D codes or for capturing/querying for ALE/EPCIS messages, in order to eliminate data inconsistencies and integration issues; these could otherwise arise when joining visibility events captured by different stakeholders within a supply chain that used different registered domain names. Although EPC URNs are URIs, they are not Web URIs and they provide no straightforward intuitive way to link to information and services on the Web; if you type an EPC URN into the address bar of a Web browser, it will not resolve to anything useful on the Web.

4.2 Use of barcodes containing GS1 Web URI and GS1 barcodes

The use of 2D barcodes is increasing on consumer trade items, but also logistics units, assets, locations, for patient safety and more.

For consumer trade items, packaging increasingly carries two barcodes:

1. GS1 barcode with GS1 Application Identifiers (or implied Application Identifiers (AI) as is the case with EAN/UPC barcodes).
2. Non-GS1 barcode with a URL, which is not backward compatible with GS1 identification keys.

This scenario makes consumer experiences inconsistent, eliminates the possible use of the non-GS1 barcode by all supply chain parties (as is the case with EAN/UPC), eliminates the possible use of a single barcode to support both supply chain and mobile applications. Use of GS1 identification keys in a barcode containing a GS1 Web URI can provide backward compatible functionality with the URLs used today while also providing for potential interoperability of supply chain and mobile applications, for all trading partners, in the future.

With the GS1 barcode (carrying GS1 identification keys within the element string syntax) and a barcode (carrying GS1 identification keys within the GS1 Web URI syntax) on consumer trade items, several assumptions apply for the solutions phase:

1. The use of GS1 Application Identifiers (whether explicit or implicit as is true of EAN/UPC) will continue to be the mandatory, minimum requirement for all consumer product packaging. This is aligned with the principle of "do no harm".
2. The use of GS1 identification keys within all barcodes on consumer packaging should gradually become the new norm. This is also aligned with the principle of "do no harm" as today's situation makes clear (e.g., consumer confusion, drag on supply chain scanning, inability for ubiquitous use).
3. The co-existence of both syntaxes on consumer packaging means the industry would benefit from translation software (a translator) between them. The translation utility would allow:
 - a. Mobile applications to use either syntax and simplify mobile scanning for consumers.
 - b. Supply chain applications to use either syntax, therefore mitigating or reducing the performance drag on scanning systems (which are increasingly encountering a barcode they cannot process... today's URL in a non-GS1 barcode).
 - c. All trading partners and governments to leverage the GS1 Web URI barcode.
4. GS1 Web URI syntax will have no effect on data carrier selections within GS1 application standards.

For trade item packaging above the consumer packaging level, logistics units, assets, locations, and other entities, 2D use is also increasing. This impacts supply chain professionals (e.g., forklift

operators, store inventory control staff) and automated scanning systems (e.g. conveyor mounted, conveyor tunnel) in terms of imaging-based scanning, but based on continuing to use element string syntax within 2D carriers. Unlike consumer-facing applications, use of Web URIs within a 2D data carrier is not trending in the same way above the consumer packaging level. That said, this does not mean that it will not in the future, but the requirements within this document do not preclude its future application in a consistent and seamless manner. In the meantime, mechanisms will be provided to permit translation of existing element strings into an equivalent reference GS1 Web URI syntax, which can be automatically redirected to relevant online information by a suitable resolver or 'GS1 Digital Bridge,' thus supporting the additional benefits of GS1 Web URI for such objects to link to relevant information, without causing any disruption to existing practices for identifying and scanning such objects, i.e. continued use of element strings in existing GS1 barcodes.

Independent of whether the translation capability were ever to become pervasive for either supply chain or mobile, the identification baseline is still established and the additional value for translation (detailed previously) will be enjoyed incrementally by those who implement it.

If translator functions become pervasive...

- For supply chain application areas, then one barcode containing GS1 Web URI could support supply chain and mobile ID.
- For mobile, then one GS1 barcode (containing element strings) could support supply chain and mobile ID.
- For both, then one GS1 barcode or one barcode containing a GS1 Web URI could support supply chain and mobile ID.

Until they become pervasive, it will remain important to orient consumers and supply chain professionals as to which bar code to use for what purpose and to follow GS1 standards rules for barcode placement.

Also, it is important to remember that consumer trade items are scanned in distribution centres, warehouses, point of care, at home patient settings, field clinics, etc. This means that the same principles will relate, even if on different timelines for these entities.

In summary, the existence of a translator (point 3) does not ensure the pervasive use of translators, hence the reason for point 1. However, the use of GS1 identification in either barcode provides substantial benefits today and leaves open a future where one barcode "could" (if the translator were pervasively present) meet supply chain as well as mobile application requirements. For consumer packaging, if this were to occur, this would permit one barcode instead of two, therefore increasing supply chain scanning performance and making consumer experiences easier and more beneficial than today. If pervasive adoption of a translator does not occur, all the benefits that will be realised without it will remain relevant and still justifies moving from a non-GS1 URL to a GS1 Web URI syntax.

4.3 A note on terminology

Discussion of encoding GS1 Web URIs within barcodes, QR Codes®, DataMatrix codes and more, quickly leads to confusion, especially when colloquial terminology is used to address a wider audience that may have specific meaning within a GS1 audience. Table 4-1 attempts to clear up this confusion.

Table 4-1 Examples of identifier formats and corresponding terminology

Identifier expressed as:	Example	When encoded within any barcode GS1 calls this a ...	When encoded within any barcode the public calls this a...
Element string	(01)05000127163096	GS1 barcode	1D / 2D barcode
Reference GS1 Web URI	https://id.gs1.org/gtin/5000127163096	barcode containing GS1 Web URI on the id.gs1.org domain	1D / 2D barcode

Identifier expressed as:	Example	When encoded within any barcode GS1 calls this a ...	When encoded within any barcode the public calls this a...
GS1 Web URI	http://example.com/gtin/5000127163096	barcode containing GS1 Web URI on any domain	1D / 2D barcode
Non-GS1 Web URI	http://example.com/random/string	barcode according to the relevant ISO/IEC standard or non-GS1 barcode	2D barcode

4.4 Security and privacy of information

Not all information on the Web is openly available to everyone. It is also possible to use a Web URI to link to information that requires a user or client to authenticate and where an authorisation decision on whether or not to provide access to the information depends on those authentication credentials, possibly in combination with other parameters. It is also possible for the same Web URI to provide different views or representations of information with different amounts of detail depending on whether a user or client authenticates – and also depending on the specific access rights of any specific authenticated user or client. This is similar to existing practices on the Web today where a Web page might provide anyone with an abstract or summary of a technical article, academic paper or standards document but require users to register or login in order to access the full document described in the abstract or summary.

Existing Web security technology will be used to restrict access to data that requires authentication and authorisation of the client requesting the data.

The HTTP protocol already provides response codes 401 "Unauthorized" and 407 "Proxy authentication required" to indicate that authorization is required. For further details on Web authentication, see [RFC 7235].

4.5 Contextual information

Contextual information enables a client to express a preferred data format, encoding, human language and character set for the information they would like to retrieve, so that the Web resource can consider that contextual information and attempt to provide information in the requested data format and human language, if they are supported, in order to provide the most appropriate information to each user.

Contextual information can also include location information, in order to request or provide the user with information that is most relevant to a specific geographic location (typically their current location), such as the nearest retail store selling a specific product or the nearest provider of a particular service.

Existing Web protocols enable some contextual information to be provided at the time of making a Web request (for preferences about data format, human language, character set and data encoding). For personal privacy reasons, fine-grained location information is not transmitted by default when making an initial Web request but only via an opt-in basis per request; the initial response may contain a script that subsequently requests the client to allow location information to be shared.

The provision of contextual information about preferred data format and human language is known as Content Negotiation, as described in section 3.4 of [RFC 7231]. This is used to indicate whether the server should return a Web page (text/html) or data in a specified format' (e.g. application/json, text/xml, application/ld+json). Access to geolocation context information is specified in the W3C Geolocation API Recommendation [Geolocation API].

4.6 Access to traceability data by various supply chain stakeholders

Much interest in GS1 Web URI for consumer-facing mobile scanning is related to provision of more detailed information to consumers about the provenance of a product as well as the accreditations

and provision of transparent factual information regarding its ingredients, components, the ethical and environmentally-friendly /sustainable sourcing of these, as well as ethical and environmentally-friendly processes involved in the manufacture of the product. Such information can vary between different instances of a product, whether identified at serial-level or batch-level and where there is variability in such information, it cannot be marked on packaging that is designed for all instances of a product's GTIN. Consumer-facing mobile scanning and the use of a GS1 Web URI with a sufficiently fine-grained identifier (potentially including indications of batch/lot and serial number) provides a mechanism to provide accurate fine-grained information to a consumer (or prospective consumer) at a level of detail beyond the information appearing on the product packaging. The purpose of this section of the document is to remind everyone that the brand owner will typically only be able to provide product provenance data for ingredients, components and processing that is upstream of the brand owner; the brand owner typically will not have access to downstream supply chain traceability data but the retailer may have access to such information and may be able to provide a consumer-relevant subset of this to consumers, either while they browse within a store or after they have purchased the product.

Each party shipping or selling a product is expected to be able to provide some information on its upstream traceability, although typically they might not be entitled to have any downstream traceability information beyond their immediate 1-down customer. This means that the Brand Owner typically will have access to upstream traceability information about the provenance of a product (and its ingredients, components, sourcing and production) up to the point where it is shipped from the manufacturer.

The retailer is more likely to have access to upstream supply chain traceability information and in some regulated industries may be required (together with other intermediate stakeholders such as distributors) to prove an unbroken chain of custody or ownership back to the genuine manufacturer. If the default behaviour is to redirect a reference GS1 Web URI for a product to a corresponding Web URI specified by the Brand Owner, the Brand Owner may only be able to provide product provenance information that is upstream of the manufacturer and may not be able to provide supply chain tracking information because of its lack of privileges to downstream tracking information beyond its 1-down customer and the emergent nature of supply chain paths for mass-produced products that are built to stock, rather than custom built to order.

4.7 Supporting machine-to-machine communication and expressing facts and relationships between things and about things

The World Wide Web is not only a collection of Web pages but also a huge decentralised database of (machine-readable) facts. For example, the facts and figures behind every Wikipedia page are extracted and available via Wikidata and DBpedia. However, there are multiple sources of facts that are being embedded within Web pages using Linked Data technology (also known as Semantic Web technology) using W3C Linked Data standards [Linked Data] to make those facts available in a machine-interpretable format that can be processed automatically by software, search engines, smartphone apps etc., also using logical rules to derive additional facts and relationships. Of course, we need a language for describing such relationships. One very popular vocabulary for doing so is schema.org, which provides a broad, high-level collection of terms to describe products, places, people, things, organisations, music, events, etc. GS1 has developed a Web vocabulary [GS1Voc] that serves as an external extension to schema.org and allows products to be described in much richer detail, e.g. to provide detailed nutritional information, ingredients, details of product certifications, allergens, as well as other product features and specifications.

Within schema.org, there are a number of properties that can be used to describe relationships between things. These include:

```
schema:isRelatedTo, schema:isSimilarTo, schema:isConsumableFor,  
schema:isAccessoryOrSparePartFor
```

Within the GS1 Web vocabulary we also define some further properties for expressing relationships between things. These include:

```
gs1:dependentProprietaryProduct, gs1:equivalentProduct,  
gs1:primaryAlternateProduct, gs1:replacedProduct, gs1:replacedByProduct
```

In addition to the schema.org and GS1 Web vocabulary, a number of other foundational Linked Data vocabularies also provide terms for expressing high-level relationships between things. These include Dublin Core [DC] from which the following terms are particularly relevant:

`dcterms:isPartOf`, `dcterms:hasPart`

Schema.org and the GS1 Web vocabulary [GS1Voc] provide many Linked Data terms to describe products in detail. At present, most of these terms can be used to express product class-level master data defined on a per GTIN basis. In principle, the GS1 Web vocabulary can also be extended to support instance/lot master data, to express properties whose values are specific to a particular batch/lot or even serial number and potentially even to express consumer-facing summary information of a subset of critical tracking events from traceability data. This means that if a GS1 Web URI makes use of a GTIN in combination with key qualifiers such as Batch/Lot identifier or Serial Number, we can provide data defined for that GTIN as well as data that are specific to that batch/lot or serial number, all using the same Linked Data technology, all embedded within the same Web page and all retrieved through a single Web request.

At present, neither the GS1 Web vocabulary [GS1Voc] nor schema.org defines an expiration date property for a `schema:Product` – although the `schema:expires` property exists for a `schema:CreativeWork`.

The GS1 Web vocabulary does include some terms for describing production variants – but not for consumer package variants, although this could be addressed in future updates. A Web URI can be used with Linked Data technology to express facts in a machine-readable format. It does this using the W3C Resource Description Framework [RDF] in which facts are written as triples consisting of a Subject, a Property (or Predicate) and Value (or Object) and these are themselves expressed as URIs. Figure 4-1 shows two examples of these Subject-Property-Value triples.

Figure 4-1 Examples of Subject-Property-Value triples



The first example simply says that the resource identified by `https://id.gs1.org/gtin/9507000009060/` is of type 'Product' as defined by schema.org. The second example formally states that the same resource has a specific GTIN identifier with value "09507000009060". (Note that the property `https://gs1.org/voc/gtin` always expects a GTIN in 14 digit representation.)

GS1 keys included within Web URIs are then not only mechanisms to retrieve information about a specific GS1 identification key. They are also first-class citizens in the Web of Linked Data [Linked Data] so that we can link all of these related facts to a GS1 Web URI.

In GS1 we recommend using the schema.org and GS1 Web vocabulary, and a semantic markup format such as JSON-LD [JSON-LD].

4.8 Explicit link to the GS1 identification key(s)

By standardising a Web URI structure under the gs1.org domain and recommending that brand owners also adopt a similar structure (even when constructing Web URIs for products using their own registered domain name), we can provide a consistent pattern from which a GTIN or other GS1 key can be conveniently extracted. This does not restrict a retailer who may wish to do an initial lookup of the Web URI for each product they stock in their store, to confirm that it is being used in accordance with these guidelines, by checking for a `<link rel="http://schema.org/sameAs" ... >` whose href attribute value points to the corresponding reference Web URI under the gs1.org domain.

4.9 Validation principles for GS1 Web URIs

In order to promote consistency with the GS1 General Specifications [GENSPECS], the combinations of structural elements within the GS1 Web URI syntax should comply with all restrictions that apply to permitted or required combinations of GS1 Application Identifiers as expressed within the GS1 General Specifications [GENSPECS] and specifically in section 4.14.1 (Invalid pairs of element strings) and section 4.14.2 (Mandatory associations of element strings)

Software implementations that process GS1 Web URI for the purposes of translation or redirection should perform appropriate validation at least for the GS1 identification keys with which they are concerned, together with generation of helpful error messages when such validation rules indicate invalid or nonsensical combinations of identifiers and data elements or the absence of an Application Identifier or structural element that is required.

5 GS1 Web URI syntax

This chapter represents the core normative content for this standard.

5.1 Formal specification

This section specifies the structure of GS1 Web URIs using a notation called Augmented Backus-Naur Form (ABNF), which is defined in [RFC 5234] and updated by [RFC 7405].

ABNF formally expresses how strings of characters (including URIs) are constructed by concatenating smaller components in a sequential order.

Those smaller components may be defined in terms of further sub-components and/or in terms of sequences of character sets that are also defined by rules.

ABNF also supports repeating components and optional components.

Optional components are enclosed within square brackets.

A sequential group of one more components may be enclosed within round brackets.

Repeating components use the $m*n(\text{component})$ notation to indicate that the component within the round brackets may appear at least m times and at most n times. Default values are $m=0$, $n=\text{infinity}$. If either or m or n are omitted, their default values are assumed.

Everything following a semicolon on a line is considered to be an explanatory comment.

The notation $n(\text{component})$ or $n\text{component}$ where n is one or more digit characters is equivalent to $n*(\text{component})$, indicating that the component must appear exactly n times.

A number of comments are provided to explain the meaning of rules.

5.1.1 Character sets

Firstly, a number of character sets are defined for later re-use in subsequent ABNF rules.

DIGIT = "0" / "1" / "2" / "3" / "4" /
"5" / "6" / "7" / "8" / "9"

UPPERALPHA = %x41-5A ; A-Z (ASCII characters 65-90 decimal, 41-5A hex)

LOWERALPHA = %x61-7A ; a-z (ASCII characters 97-122 decimal, 61-7A hex)

ALPHA = UPPERALPHA / LOWERALPHA ; A-Z or a-z

HEXDIG = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"

DoubleQuote = '"' ; the double-quote character "

The following characters must be represented using percent-encoding (see Section 2.1 of RFC 3986 [PercentEncoding]) when used as literal characters within URIs, since many of these have special meanings within Web URIs:

Octothorpe = "%x23" ; percent-encoding of the # character

ForwardSlash = "%x2F" ; percent-encoding of the / character

Percent = "%x25" ; percent-encoding of the % character

Ampersand = "%x26" ; percent-encoding of the & character

Plus = "%x2B" ; percent-encoding of the + character

Comma = "%x2C" ; percent-encoding of the , character

Exclamation = "%x21" ; percent-encoding of the ! character

LeftBracket = "%x28" ; percent-encoding of the (character

RightBracket = "%x29" ; percent-encoding of the) character

Asterisk = "%x2A" ; percent-encoding of the * character

Apostrophe = "%x27" ; percent-encoding of the ' character

Colon = "%x3A" ; percent-encoding of the : character

Semicolon = "%x3B" ; percent-encoding of the ; character

LeftAngleBracket = "%x3C" ; percent-encoding of the < character

Equals = "%x3D" ; percent-encoding of the = character

RightAngleBracket = "%x3E" ; percent-encoding of the > character

QuestionMark = "%x3F" ; percent-encoding of the ? character

The following group of symbol characters is permitted within the 82-character subset of ISO/IEC 646, indicated in Figure 7.11-1 of the GS1 General Specifications [GENSPECS].

XSYMBOL = DoubleQuote / "-" / "." / "_" / Exclamation / Percent /
Ampersand / Plus / Comma / ForwardSlash / Asterisk /
LeftBracket / RightBracket / Apostrophe / Semicolon /
Colon / LeftAngleBracket / RightAngleBracket / Equals /
QuestionMark

The following group of symbol characters is permitted within the 39-character subset of ISO/IEC 646, indicated in Figure 7.11-2 of the GS1 General Specifications [GENSPECS].

YSYMBOL = "-" / Octothorpe / ForwardSlash

The following character set corresponds to all permitted characters within the 82-character subset of ISO/IEC 646, indicated in Figure 7.11-1 of the GS1 General Specifications [GENSPECS].

XCHAR = DIGIT / UPPERALPHA / LOWERALPHA / XSYMBOL

The following character set corresponds to all permitted characters within the 39-character subset of ISO/IEC 646, indicated in Figure 7.11-2 of the GS1 General Specifications [GENSPECS]. It is currently only used within the value of the Components and Parts Identifier (CPID).

YCHAR = DIGIT / UPPERALPHA / YSYMBOL

5.1.2 Primary identification keys

The following rules indicate which GS1 Application Identifiers (AI) are considered as primary identification keys for GS1 web URI. Note that for each of these (and the rules in the next section), the numeric AI value may be used or alternatively, a corresponding lower-case short name may be used if it is more friendly to software developers. The numeric AI value may be more suitable for use when encoding a GS1 Web URI within a 2D barcode, since this can be encoded more efficiently, resulting in a lower total module count and improved readability.

The %s prefix notation was introduced in [RFC 7405] and simply indicates that the following string value is case-sensitive. For example, in the rule below, gtin-code may be either "01" or "gtin" but not "GTIN" nor "Gtin".

gtin-code	= "01" / %s"gtin"	; GTIN
itip-code	= "8006" / %s"itip"	; ITIP
gmn-code	= "8013" / %s"gmn"	; Global Model Number
cpid-code	= "8010" / %s"cpid"	; CPID
shipTo-code	= "410" / %s"shipTo"	; ship-to
billTo-code	= "411" / %s"billTo"	; bill-to
purchasedFrom-code	= "412" / %s"purchasedFrom"	; purchased from GLN
shipFor-code	= "413" / %s"shipFor"	; ship-for
gln-code	= "414" / %s"gln"	; Physical Location GLN
payTo-code	= "415" / %s"payTo"	; GLN of invoicing party
glnProd-code	= "416" / %s"glnProd"	; GLN of production/service loc.
gsrnp-code	= "8017" / %s"gsrnp"	; GSRN of the Provider
gsrn-code	= "8018" / %s"gsrn"	; GSRN of the Recipient
gcn-code	= "255" / %s"gcn"	; GCN
sscc-code	= "00" / %s"sscc"	; SSCC
gdti-code	= "253" / %s"gdti"	; GDTI
ginc-code	= "401" / %s"ginc"	; GINC
gsin-code	= "402" / %s"gsin"	; GSIN
grai-code	= "8003" / %s"grai"	; GRAI
giai-code	= "8004" / %s"giai"	; GIAI

5.1.3 Key qualifiers

The following rules which GS1 Application Identifiers (AI) are considered as key qualifiers for a GS1 web URI.

cpv-code = "22" / %s"cpv" ; Consumer Product Variant

lot-code	= "10"	/ %s"lot"	; Batch/Lot identifier
ser-code	= "21"	/ %s"ser"	; GTIN Serial Number
cpsn-code	= "8011"	/ %s"cpsn"	; CPID Serial Number
glnx-code	= "254"	/ %s"glnx"	; GLN extension
refno-code	= "8020"	/ %s"refno"	; Payment Reference Number
srin-code	= "8019"	/ %s"srin"	; Service Relation Instance Number

5.1.4 Primary key formats

The following rules express the format of the values of the primary GS1 identification keys.

Note that the GS1 General Specifications [GENSPECS] define further restrictions on some of these values, particularly for those which include a GS1 Check Digit, Indicator Digit or Extension Digit. Please refer to the GS1 General Specifications [GENSPECS] for further details.

gtin-value	= 8DIGIT / 12DIGIT / 13DIGIT / 14DIGIT
itip-value	= 14DIGIT 2DIGIT 2DIGIT ; 14 digits then 2 digits then 2 digits
gmn-value	= 1*30XCHAR ; 1-30 characters from 82-chr subset
cpid-value	= 1*30YCHAR ; 1-30 characters from 39-chr subset
shipTo-value	= 13DIGIT ; exactly 13 digits
billTo-value	= 13DIGIT ; exactly 13 digits
purchasedFrom-value	= 13DIGIT ; exactly 13 digits
shipFor-value	= 13DIGIT ; exactly 13 digits
gln-value	= 13DIGIT ; exactly 13 digits
payTo-value	= 13DIGIT ; exactly 13 digits
glnProd-value	= 13DIGIT ; exactly 13 digits
gsrnp-value	= 18DIGIT ; exactly 18 digits
gsrn-value	= 18DIGIT ; exactly 18 digits
gcn-value	= 13DIGIT 1*12DIGIT ; 13 digits then 1-12 digits
sscc-value	= 18DIGIT ; exactly 18 digits
gdti-value	= 13DIGIT 1*17XCHAR ; 13 digits then 1-17 characters ; from the 82-character subset
ginc-value	= 1*30XCHAR ; 1-30 characters from the 82-character subset
gsin-value	= 17DIGIT ; exactly 17 digits
grai-value	= 14DIGIT 1*16XCHAR ; 14 digits then 1-16 characters ; from the 82-character subset of ISO/IEC 646
giai-value	= 1*30XCHAR ; 1-30 characters from 82-chr subset

5.1.5 Key qualifier formats

The following rules express the format of the values of the key qualifiers of primary GS1 identification keys:

cpv-value	= 1*20XCHAR	; 1-20 characters from 82-chr subset
lot-value	= 1*20XCHAR	; 1-20 characters from 82-chr subset
ser-value	= 1*20XCHAR	; 1-20 characters from 82-chr subset
cpsn-value	= 1*12DIGIT	; 1-12 digits
glnx-value	= 1*20XCHAR	; 1-20 characters from 82-chr subset
refno-value	= 1*25XCHAR	; 1-25 characters from 82-chr subset
srin-value	= 1*10DIGIT	; 1-10 digits

5.1.6 Primary identifier and value concatenation

The following rules express how each primary identifier code and its value should be concatenated (for use within the URI path information) :

gtin-comp	= "/"	gtin-code	/"	gtin-value
itip-comp	= "/"	itip-code	/"	itip-value
gmn-comp	= "/"	gmn-code	/"	gmn-value
cpid-comp	= "/"	cpid-code	/"	cpid-value
shipTo-comp	= "/"	shipTo-code	/"	shipTo-value
billTo-comp	= "/"	billTo-code	/"	billTo-value
purchasedFrom-comp	= "/"	purchasedFrom-code	/"	purchasedFrom-value
shipFor-comp	= "/"	shipFor-code	/"	shipFor-value
gln-comp	= "/"	gln-code	/"	gln-value
payTo-comp	= "/"	payTo-code	/"	payTo-value
glnProd-comp	= "/"	glnProd-code	/"	glnProd-value
gsrnp-comp	= "/"	gsrnp-code	/"	gsrnp-value
gsrn-comp	= "/"	gsrn-code	/"	gsrn-value
gcn-comp	= "/"	gcn-code	/"	gcn-value
sscc-comp	= "/"	sscc-code	/"	sscc-value
gdti-comp	= "/"	gdti-code	/"	gdti-value
ginc-comp	= "/"	ginc-code	/"	ginc-value
gsin-comp	= "/"	gsin-code	/"	gsin-value
grai-comp	= "/"	grai-code	/"	grai-value
giai-comp	= "/"	giai-code	/"	giai-value

5.1.7 Key qualifier concatenation

The following rules express how each key qualifier and its value should be concatenated (for use within the URI path information) :

cpv-comp	= "/"	cpv-code	/"	cpv-value
lot-comp	= "/"	lot-code	/"	lot-value
ser-comp	= "/"	ser-code	/"	ser-value

```

cpsn-comp      = "/" cpsn-code "/" cpsn-value
glnx-comp      = "/" glnx-code "/" glnx-value
refno-comp     = "/" refno-code "/" refno-value
srin-comp      = "/" srin-code "/" srin-value
  
```

5.1.8 Path element order

The following rules express how the URI path information should be structured for each primary GS1 identification key. Note that some primary identifiers such as SSCC do not have any associated key qualifier. Other primary identifiers such as GTIN may have multiple key qualifiers. The square bracket notation indicates that the enclosed key qualifier component may be omitted but the sequence in which they appear is important and must be preserved. For example, the rule for gtin-path would permit any of these:

```

/gtin/01234567890128/cpv/2A/lot/ABC123/ser/12345XYZ
/gtin/01234567890128/lot/ABC123/
/gtin/01234567890128/lot/ABC123/ser/12345XYZ
/gtin/01234567890128/ser/12345XYZ
  
```

but does not permit strings such as:

```

/gtin/01234567890128/ser/12345XYZ/lot/ABC123
  
```

in which the sequential ordering of the key qualifier components is not preserved.

```

gtin-path      = gtin-comp [cpv-comp] [lot-comp] [ser-comp]
itip-path      = itip-comp [cpv-comp] [lot-comp] [ser-comp]
gmn-path       = gmn-comp
cpid-path      = cpid-comp [cpsn-comp]
shipTo-path    = shipTo-comp
billTo-path    = billTo-comp
purchasedFrom-path = purchasedFrom-comp
shipFor-path   = shipFor-comp
gln-path       = gln-comp [glnx-comp]
payTo-path     = payTo-comp
glnProd-path   = glnProd-comp
gsrnp-path     = gsrnp-comp [srin-comp]
gsrn-path      = gsrn-comp [srin-comp]
gcn-path       = gcn-comp
sscc-path      = sccc-comp
gdti-path      = gdti-comp
ginc-path      = ginc-comp
gsin-path      = gsin-comp
grai-path      = grai-comp
giai-path      = giai-comp
  
```

The following rule simply states that any of the above is considered as a gs1path (which will be referenced in a later rule).

```

gs1path = gtin-path / itip-path / gmn-path / cpid-path / shipTo-path /
billTo-path / purchasedFrom-path / shipFor-path / gln-path /
payTo-path / glnProd-path / gsrnp-path / gsrn-path / gcn-path /
sscc-path / gdti-path / ginc-path / gsin-path / grai-path /
giai-path
  
```

5.1.9 Data attributes

The following rules are concerned with GS1 Application Identifiers that are considered to be data attributes rather than primary identifier keys or key qualifiers. Data attributes and their values

SHALL be expressed via the URI query string as key=value pairs. For most of these, the key is always the numeric AI value, rather than a more human-friendly short name.

```

netWeightVMTICode      = "3100" / "3101" / "3102" / "3103" / "3104" / "3105" /
                        "3200" / "3201" / "3202" / "3203" / "3204" / "3205" /
                        "3560" / "3561" / "3562" / "3563" / "3564" / "3565" /
                        "3570" / "3571" / "3572" / "3573" / "3574" / "3575"
netWeightVMTIValue     = 6DIGIT
netWeightVMTIPParameter = netWeightVMTICode "=" netWeightVMTIValue

lengthVMTICode         = "3110" / "3111" / "3112" / "3113" / "3114" / "3115" /
                        "3210" / "3211" / "3212" / "3213" / "3214" / "3215" /
                        "3220" / "3221" / "3222" / "3223" / "3224" / "3225" /
                        "3230" / "3231" / "3232" / "3233" / "3234" / "3235"
lengthVMTIValue        = 6DIGIT
lengthVMTIPParameter   = lengthVMTICode "=" lengthVMTIValue

widthVMTICode          = "3120" / "3121" / "3122" / "3123" / "3124" / "3125" /
                        "3240" / "3241" / "3242" / "3243" / "3244" / "3245" /
                        "3250" / "3251" / "3252" / "3253" / "3254" / "3255" /
                        "3260" / "3261" / "3262" / "3263" / "3264" / "3265"
widthVMTIValue         = 6DIGIT
widthVMTIPParameter    = widthVMTICode "=" widthVMTIValue

depthVMTICode          = "3130" / "3131" / "3132" / "3133" / "3134" / "3135" /
                        "3270" / "3271" / "3272" / "3273" / "3274" / "3275" /
                        "3280" / "3281" / "3282" / "3283" / "3284" / "3285" /
                        "3290" / "3291" / "3292" / "3293" / "3294" / "3295"
depthVMTIValue         = 6DIGIT
depthVMTIPParameter    = depthVMTICode "=" depthVMTIValue

areaVMTICode           = "3140" / "3141" / "3142" / "3143" / "3144" / "3145" /
                        "3500" / "3501" / "3502" / "3503" / "3504" / "3505" /
                        "3510" / "3511" / "3512" / "3513" / "3514" / "3515" /
                        "3520" / "3521" / "3522" / "3523" / "3524" / "3525"
areaVMTIValue          = 6DIGIT
areaVMTIPParameter     = areaVMTICode "=" areaVMTIValue

netVolumeVMTICode      = "3150" / "3151" / "3152" / "3153" / "3154" / "3155" /
                        "3160" / "3161" / "3162" / "3163" / "3164" / "3165" /
                        "3600" / "3601" / "3602" / "3603" / "3604" / "3605" /
                        "3610" / "3611" / "3612" / "3613" / "3614" / "3615" /
                        "3640" / "3641" / "3642" / "3643" / "3644" / "3645" /
                        "3650" / "3651" / "3652" / "3653" / "3654" / "3655" /
                        "3660" / "3661" / "3662" / "3663" / "3664" / "3665"
netVolumeVMTIValue     = 6DIGIT
netVolumeVMTIPParameter = netVolumeVMTICode "=" netVolumeVMTIValue

massPerUnitAreaVMTICode = "3370" / "3371" / "3372" / "3373" / "3374" /
                        "3375"
massPerUnitAreaVMTIValue = 6DIGIT
massPerUnitAreaVMTIPParameter = massPerUnitAreaVMTICode "="
                                massPerUnitAreaVMTIValue

grossWeightCode        = "3300" / "3301" / "3302" / "3303" / "3304" / "3305" /
                        "3400" / "3401" / "3402" / "3403" / "3404" / "3405"
grossWeightValue       = 6DIGIT
grossWeightParameter   = grossWeightCode "=" grossWeightValue
  
```

```

logisticLengthCode    = "3310" / "3311" / "3312" / "3313" / "3314" / "3315" /
                        "3410" / "3411" / "3412" / "3413" / "3414" / "3415" /
                        "3420" / "3421" / "3422" / "3423" / "3424" / "3425" /
                        "3430" / "3431" / "3432" / "3433" / "3434" / "3435"
logisticLengthValue   = 6DIGIT
logisticLengthParameter = logisticLengthCode "=" logisticLengthValue

logisticWidthCode     = "3320" / "3321" / "3322" / "3323" / "3324" / "3325" /
                        "3440" / "3441" / "3442" / "3443" / "3444" / "3445" /
                        "3450" / "3451" / "3452" / "3453" / "3454" / "3455" /
                        "3460" / "3461" / "3462" / "3463" / "3464" / "3465"
logisticWidthValue    = 6DIGIT
logisticWidthParameter = logisticWidthCode "=" logisticWidthValue

logisticDepthCode     = "3330" / "3331" / "3332" / "3333" / "3334" / "3335" /
                        "3470" / "3471" / "3472" / "3473" / "3474" / "3475" /
                        "3480" / "3481" / "3482" / "3483" / "3484" / "3485" /
                        "3490" / "3491" / "3492" / "3493" / "3494" / "3495"
logisticDepthValue    = 6DIGIT
logisticDepthParameter = logisticDepthCode "=" logisticDepthValue

logisticAreaCode      = "3340" / "3341" / "3342" / "3343" / "3344" / "3345" /
                        "3530" / "3531" / "3532" / "3533" / "3534" / "3535" /
                        "3540" / "3541" / "3542" / "3543" / "3544" / "3545" /
                        "3550" / "3551" / "3552" / "3553" / "3554" / "3555"
logisticAreaValue     = 6DIGIT
logisticAreaParameter = logisticAreaCode "=" logisticAreaValue

logisticVolumeCode    = "3350" / "3351" / "3352" / "3353" / "3354" / "3355" /
                        "3360" / "3361" / "3362" / "3363" / "3364" / "3365" /
                        "3620" / "3621" / "3622" / "3623" / "3624" / "3625" /
                        "3630" / "3631" / "3632" / "3633" / "3634" / "3635" /
                        "3670" / "3671" / "3672" / "3673" / "3674" / "3675" /
                        "3680" / "3681" / "3682" / "3683" / "3684" / "3685" /
                        "3690" / "3691" / "3692" / "3693" / "3694" / "3695"
logisticVolumeValue   = 6DIGIT
logisticVolumeParameter = logisticVolumeCode "=" logisticVolumeValue

processorCode         = "7030" / "7031" / "7032" / "7033" / "7034" / "7035" /
                        "7036" / "7037" / "7038" / "7039"
processorValue        = 3DIGIT 1*27XCHAR
processorParameter    = processorCode "=" processorValue

contentCode          = "02"
contentValue         = 14DIGIT
contentParameter     = contentCode "=" contentValue

prodDateCode         = "11"
prodDateValue        = 6DIGIT
prodDateParameter    = prodDateCode "=" prodDateValue

dueDateCode          = "12"
dueDateValue         = 6DIGIT
dueDateParameter     = dueDateCode "=" dueDateValue

packDateCode         = "13"
packDateValue        = 6DIGIT
packDateParameter    = packDateCode "=" packDateValue
  
```

```

bestBeforeDateCode      = "15"
bestBeforeDateValue     = 6DIGIT
bestBeforeDateParameter = bestBeforeDateCode "=" bestBeforeDateValue

sellByDateCode          = "16"
sellByDateValue         = 6DIGIT
sellByDateParameter     = sellByDateCode "=" sellByDateValue

firstFreezeDateCode     = "7006"
firstFreezeDateValue    = 6DIGIT
firstFreezeDateParameter = firstFreezeDateCode "=" firstFreezeDateValue

harvestDateCode         = "7007"
harvestDateValue        = 6*12DIGIT
harvestDateParameter    = harvestDateCode "=" harvestDateValue

pricePerUnitCode        = "8005"
pricePerUnitValue       = 6DIGIT
pricePerUnitParameter   = pricePerUnitCode "=" pricePerUnitValue

variantCode             = "20"
variantValue            = 2DIGIT
variantParameter        = variantCode "=" variantValue

varCountCode            = "30"
varCountValue           = 1*8DIGIT
varCountParameter       = varCountCode "=" varCountValue

countCode               = "37"
countValue              = 1*8DIGIT
countParameter          = countCode "=" countValue

mutualCode              = "90"
mutualValue             = 1*30DIGIT
mutualParameter         = mutualCode "=" mutualValue

additionalIdCode         = "240"
additionalIdValue        = 1*30DIGIT
additionalIdParameter    = additionalIdCode "=" additionalIdValue

custPartNoCode          = "241"
custPartNoValue         = 1*30DIGIT
custPartNoParameter     = custPartNoCode "=" custPartNoValue

mtoVariantCode          = "242"
mtoVariantValue         = 6DIGIT
mtoVariantParameter     = mtoVariantCode "=" mtoVariantValue

pcnCode                 = "243"
pcnValue                = 1*20DIGIT
pcnParameter            = pcnCode "=" pcnValue

secondarySerialCode     = "250"
secondarySerialValue    = 1*30DIGIT
secondarySerialParameter = secondarySerialCode "=" secondarySerialValue

refToSourceCode         = "251"
refToSourceValue        = 1*30DIGIT
refToSourceParameter    = refToSourceCode "=" refToSourceValue
  
```



```
amountCode           = "3900" / "3901" / "3902" / "3903" / "3904" / "3905"  
amountValue         = 1*15DIGIT  
amountParameter     = amountCode "=" amountValue  
  
amountISOCODE       = "3910" / "3911" / "3912" / "3913" / "3914" / "3915"  
amountISOValue     = 3DIGIT 1*15DIGIT  
amountISOPARAMETER = amountISOCODE "=" amountISOValue  
  
priceCode           = "3920" / "3921" / "3922" / "3923" / "3924" / "3925"  
priceValue         = 1*15DIGIT  
priceParameter     = priceCode "=" priceValue  
  
priceISOCODE       = "3930" / "3931" / "3932" / "3933" / "3934" / "3935"  
priceISOValue     = 3DIGIT 1*15DIGIT  
priceISOPARAMETER = priceISOCODE "=" priceISOValue  
  
percentOffCode     = "3940" / "3941" / "3942" / "3943" / "3944" / "3945"  
percentOffValue   = 4DIGIT  
percentOffParameter = percentOffCode "=" percentOffValue  
  
orderNumberCode    = "400"  
orderNumberValue   = 1*30DIGIT  
orderNumberParameter = orderNumberCode "=" orderNumberValue  
  
routeCode          = "403"  
routeValue         = 1*30DIGIT  
routeParameter     = routeCode "=" routeValue  
  
shipToLocCode      = "410"  
shipToLocValue     = 13DIGIT  
shipToLocParameter = shipToLocCode "=" shipToLocValue  
  
billToCode         = "411"  
billToValue        = 13DIGIT  
billToParameter    = billToCode "=" billToValue  
  
purchaseFromCode   = "412"  
purchaseFromValue  = 13DIGIT  
purchaseFromParameter = purchaseFromCode "=" purchaseFromValue  
  
shipForLocCode     = "413"  
shipForLocValue    = 13DIGIT  
shipForLocParameter = shipForLocCode "=" shipForLocValue  
  
locNoCode          = "414"  
locNoValue         = 13DIGIT  
locNoParameter     = locNoCode "=" locNoValue  
  
payToCode          = "415"  
payToValue         = 13DIGIT  
payToParameter     = payToCode "=" payToValue  
  
prodServLocCode    = "416"  
prodServLocValue   = 13DIGIT  
prodServLocParameter = prodServLocCode "=" prodServLocValue  
  
shipToPostCode     = "420"  
shipToPostValue    = 1*20XCHAR  
shipToPostParameter = shipToPostCode "=" shipToPostValue
```

```
shipToPostISOCode      = "421"
shipToPostISOValue     = 3DIGIT 1*9XCHAR
shipToPostISOParameter = shipToPostISOCode "=" shipToPostISOValue

originCode              = "422"
originValue             = 3DIGIT
originParameter        = originCode "=" originValue

countryProcessCode     = "424"
countryProcessValue    = 3DIGIT
countryProcessParameter = countryProcessCode "=" countryProcessValue

countryFullProcessCode = "426"
countryFullProcessValue = 3DIGIT
countryFullProcessParameter =
    countryFullProcessCode "=" countryFullProcessValue

countryInitialProcessCode = "423"
countryInitialProcessValue = 3DIGIT 1*12DIGIT
countryInitialProcessParameter =
    countryInitialProcessCode "=" countryInitialProcessValue

countryDisassemblyCode = "425"
countryDisassemblyValue = 3DIGIT 1*12DIGIT
countryDisassemblyParameter =
    countryDisassemblyCode "=" countryDisassemblyValue

originSubdivisionCode  = "427"
originSubdivisionValue = 1*3XCHAR
originSubdivisionParameter =
    originSubdivisionCode "=" originSubdivisionValue

nhrnPZNCode           = "710"
nhrnPZNValue          = 1*20XCHAR
nhrnPZNParameter      = nhrnPZNCode "=" nhrnPZNValue

nhrnCIPCode           = "711"
nhrnCIPValue          = 1*20XCHAR
nhrnCIPPParameter     = nhrnCIPCode "=" nhrnCIPValue

nhrnCNCode            = "712"
nhrnCNValue           = 1*20XCHAR
nhrnCNParameter       = nhrnCNCode "=" nhrnCNValue

nhrnDRNCode           = "713"
nhrnDRNValue          = 1*20XCHAR
nhrnDRNParameter      = nhrnDRNCode "=" nhrnDRNValue

nhrnAIMCode           = "714"
nhrnAIMValue          = 1*20XCHAR
nhrnAIMParameter      = nhrnAIMCode "=" nhrnAIMValue

nsnCode                = "7001"
nsnValue               = 13DIGIT
nsnParameter           = nsnCode "=" nsnValue

meatCutCode           = "7002"
meatCutValue          = 1*30XCHAR
meatCutParameter      = meatCutCode "=" meatCutValue
```

activePotencyCode	= "7004"
activePotencyValue	= 1*4DIGIT
activePotencyParameter	= activePotencyCode "=" activePotencyValue
catchAreaCode	= "7005"
catchAreaValue	= 1*12XCHAR
catchAreaParameter	= catchAreaCode "=" catchAreaValue
fishingGearTypeCode	= "7009"
fishingGearTypeValue	= 1*10XCHAR
fishingGearTypeParameter	= fishingGearTypeCode "=" fishingGearTypeValue
prodMethodCode	= "7010"
prodMethodValue	= 1*2XCHAR
prodMethodParameter	= prodMethodCode "=" prodMethodValue
refurbLotCode	= "7020"
refurbLotValue	= 1*20XCHAR
refurbLotParameter	= refurbLotCode "=" refurbLotValue
funcStatCode	= "7021"
funcStatValue	= 1*20XCHAR
funcStatParameter	= funcStatCode "=" funcStatValue
revStatCode	= "7022"
revStatValue	= 1*20XCHAR
revStatParameter	= revStatCode "=" revStatValue
giaiAssemblyCode	= "7023"
giaiAssemblyValue	= 1*30XCHAR
giaiAssemblyParameter	= giaiAssemblyCode "=" giaiAssemblyValue
dimensionsCode	= "8001"
dimensionsValue	= 14DIGIT
dimensionsParameter	= dimensionsCode "=" dimensionsValue
cmtNoCode	= "8002"
cmtNoValue	= 1*20XCHAR
cmtNoParameter	= cmtNoCode "=" cmtNoValue
ibanCode	= "8007"
ibanValue	= 1*34XCHAR
ibanParameter	= ibanCode "=" ibanValue
prodTimeCode	= "8008"
prodTimeValue	= 8DIGIT 1*4DIGIT
prodTimeParameter	= prodTimeCode "=" prodTimeValue
versionCode	= "8012"
versionValue	= 4DIGIT 1*20XCHAR
versionParameter	= versionCode "=" versionValue
refNoCode	= "8020"
refNoValue	= 1*25XCHAR
refNoParameter	= refNoCode "=" refNoValue
couponIDNACode	= "8110"
couponIDNAValue	= 1*70XCHAR
couponIDNAParameter	= couponIDNACode "=" couponIDNAValue

```

pointsCode           = "8111"
pointsValue          = 4DIGIT
pointsParameter      = pointsCode "=" pointsValue

paperlessCouponIDNACode = "8112"
paperlessCouponIDNAValue = 1*70XCHAR
paperlessCouponIDNAParameter =
    paperlessCouponIDNACode "=" paperlessCouponIDNAValue
  
```

```

internalCode         = "91" / "92" / "93" / "94" / "95" /
    "96" / "97" / "98" / "99"
internalValue        = 1*90XCHAR
internalParameter    = internalCode "=" internalValue
  
```

Batch/Lot may also be used as a data attribute in conjunction with an SSCC [AI (00)] and a CONTENT [AI (02)] in order to indicate that the SSCC contains GTINs of a specific batch/lot. For this reason, LotParameter is defined for use within the URI query string.

```

LotParameter         = lot-code "=" lot-value
  
```

Expiry Date [AI (17)] and Expiry Date/Time [AI (7003)] are data attributes. However, because of their importance in managing stock rotation and checking for expired products, the following rules also define a lower-case short name, "exp" and "expdt" that may be used in place of numeric AIs "17" and "7003" respectively.

```

expiryDateCode       = "17" / %s"exp"
expiryDateValue      = 6DIGIT
expiryDateParameter  = expiryDateCode "=" expiryDateValue
  
```

```

expiryTimeCode       = "7003" / %s"expdt"
expiryTimeValue      = 10DIGIT
expiryTimeParameter  = expiryTimeCode "=" expiryTimeValue
  
```

The URI query string is a natural extension point within the syntax that can accommodate additional key=value pairs to express data attribute parameters that cannot be expressed using GS1 Application Identifiers. Examples of such usage may be to express a specific role, action, activity or type of service to be accessed. The following extensionParameter is based on the ABNF rule for query appearing in [RFC 3986] and serves as the main extension point for the GS1 Web URI syntax. It permits multiple arbitrary key=value pairs to be included within the query string of a GS1 Web URI. Any key=value pairs used for extension data SHALL NOT conflict with keys used for GS1 Application Identifiers either in terms of semantics or syntax; a key for extension data SHALL NOT use a numeric GS1 Application Identifier NOR the values 'lot', 'exp' or 'expdt' nor should they be used to express a value (such as a value for net weight) if that value can be expressed using GS1 Application Identifiers as data attributes.

```

extensionParameter   = *( pchar / "/" / "?" )
    ; any other query string parameter permitted by RFC 3986
    ; including additional arbitrary key=value pairs
  
```

The following rule states that any of the above parameters for data attributes may appear as a query string parameter (queryStringParam), referenced later.

```

queryStringParam     = netWeightVMTIPParameter / lengthVMTIPParameter /
    widthVMTIPParameter / depthVMTIPParameter / areaVMTIPParameter /
    netVolumeVMTIPParameter / massPerUnitAreaVMTIPParameter /
    grossWeightParameter / logisticLengthParameter /
    logisticWidthParameter / logisticDepthParameter /
    logisticAreaParameter / logisticVolumeParameter /
    processorParameter / LotParameter / expiryDateParameter /
    expiryTimeParameter / contentParameter / prodDateParameter /
    dueDateParameter / packDateParameter / bestBeforeDateParameter /
    sellByDateParameter / firstFreezeDateParameter /
    harvestDateParameter / pricePerUnitParameter / variantParameter /
  
```

```

varCountParameter / countParameter / internalParameter /
additionalIdParameter / custPartNoParameter /
mtoVariantParameter / pcnParameter / secondarySerialParameter /
refToSourceParameter / amountParameter / amountISOParameter /
priceParameter / priceISOParameter / percentOffParameter /
orderNumberParameter / routeParameter / shipToLocParameter /
billToParameter / purchaseFromParameter / shipForLocParameter /
locNoParameter / prodServLocParameter / shipToPostParameter /
shipToPostISOParameter / originParameter /
countryProcessParameter / countryFullProcessParameter /
countryInitialProcessParameter / countryDisassemblyParameter /
originSubdivisionParameter / nhrnPZNParameter / nhrnCIPPParameter /
nhrnCNParameter / nhrnDRNParameter / nsnParameter /
meatCutParameter / activePotencyParameter / catchAreaParameter /
fishingGearTypeParameter / prodMethodParameter /
refurbLotParameter / funcStatParameter / revStatParameter /
giaiAssemblyParameter / dimensionsParameter / cmtNoParameter /
ibanParameter / prodTimeParameter / versionParameter /
refNoParameter / couponIDNAParameter / pointsParameter /
paperlessCouponIDNAParameter /
internalParameter / mutualParameter / extensionParameter
  
```

5.1.10 Constructing the URI

The following rules are derived from rules appearing in [RFC 3986] and are used for defining the general structure of a Web URI or URL. These are particularly relevant for GS1 Web URIs that are not under the id.gs1.org domain.

```

scheme                = "http" / "https"

unreserved            = ALPHA / DIGIT / "-" / "." / "_" / "~"

reserved              = gen-delims / sub-delims

pct-encoded           = "%" HEXDIG HEXDIG

gen-delims            = ":" / "/" / "?" / "#" / "[" / "]" / "@"

sub-delims            = "!" / "$" / "&" / "'" / "(" / ")" / "*" /
                        "+" / "," / ";" / "="

pchar                 = unreserved / pct-encoded / sub-delims / ":" / "@"

segment               = *pchar

reg-name              = *( unreserved / pct-encoded / sub-delims )

dec-octet             = DIGIT                ; 0-9
                      / %x31-39 DIGIT        ; 10-99
                      / "1" 2DIGIT           ; 100-199
                      / "2" %x30-34 DIGIT     ; 200-249
                      / "25" %x30-35         ; 250-255

IPv4address           = dec-octet "." dec-octet "." dec-octet "." dec-octet

IPv6address           = 6( h16 ":" ) ls32
                      / "::" 5( h16 ":" ) ls32
                      / [ h16 ] "::" 4( h16 ":" ) ls32
                      / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
                      / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
                      / [ *3( h16 ":" ) h16 ] "::"   h16 ":"   ls32
                      / [ *4( h16 ":" ) h16 ] "::"   ls32
  
```

```

      / [ *5( h16 ":" ) h16 ] "::"          h16
      / [ *6( h16 ":" ) h16 ] "::"

ls32      = ( h16 ":" h16 ) / IPv4address
           ; least-significant 32 bits of address

h16       = 1*4HEXDIG
           ; 16 bits of address represented in hexadecimal

IP-literal = "[" ( IPv6address / IPvFuture  ) "]"

IPvFuture = "v" 1*HEXDIG "." 1*( unreserved / sub-delims / ":" )

host      = IP-literal / IPv4address / reg-name

port     = *DIGIT

hostname  = host [ ":" port ]
  
```

Finally, the following four rules define the syntax of a reference GS1 Web URI from the concatenation of previous defined components:

```

queryStringDelim = "&" / ";"

queryStringComp  =
    "?" queryStringParam *( queryStringDelim queryStringParam )

gsluriPattern    = gslpath [queryStringComp]

referenceGS1webURI = "https://id.gs1.org" gsluriPattern
  
```

The following rules define the syntax of a non-reference GS1 Web URI from the concatenation of previous defined components. An example of usage of a non-reference GS1 Web URI is when a company chooses to use their own registered Internet domain name to construct the Web URI but aligns with this specification for the format of the final part of the URI path information and query string. Note that zero or more path segments are permitted to appear after the hostname or domain name and before the start of the `gsluriPattern` defined in this specification.

```

customURISTem    = scheme "://" hostname *( "/" segment )

customGS1webURI  = customURISTem gsluriPattern
  
```

The formal ABNF syntax for the Web URI should be read in combination with the GS1 General Specifications [GENSPECS] to ensure appropriate usage of Application Identifiers that represent data attributes of identified things. In particular, section 4.14 of the GS1 General Specifications [GENSPECS] provides guidance about data relationships, including invalid pairs of element strings (see section 4.14.1) and mandatory associations of element strings (see section 4.14.2). In the GS1 General Specifications [GENSPECS], Section 2 specifies which identifiers are used for an application, Section 3 provides definitions for each Application Identifier, while Section 4 explains the management rules for each GS1 identification key.

As previously mentioned, some GS1 primary identifier keys include GS1 check digits and some also include indicator digits or extension digits that are to be used for specific purposes. Section 7 of the GS1 General Specifications [GENSPECS] provides details of AIDC validation rules and Section 7.2.7 explains the GS1 check digit calculation. Nothing in this GS1 specification changes the existing validation rules that apply to the values of GS1 Application Identifiers; this document only specifies how valid GS1 AI values shall be expressed in a Web URI structure.

5.2 Canonical GS1 Web URIs

The rules in section [5.1](#) provide the formal specification of GS1 Web URIs in which the most commonly used identification keys can be represented as either their numeric value (their AI) or as

a more developer-friendly string. As shown in section [5.4](#), the following URIs are both 'reference GS1 Web URIs' and both identify the same resource:

```
https://id.gs1.org/gtin/614141123452
https://id.gs1.org/01/614141123452
```

This flexibility allows any combination of numeric and string identifiers so that:

```
https://id.gs1.org/gtin/614141123452/cpv/2A
https://id.gs1.org/01/614141123452/22/2A
https://id.gs1.org/01/614141123452/cpv/2A
https://id.gs1.org/gtin/614141123452/22/2A
```

are all acceptable.

This can be extended even further since a GS1 Web URI can be constructed in any domain name, may contain additional key/value pairs in the query string and so on. This flexibility is a deliberate feature of the standard to support its use in as many scenarios as possible.

However, in some contexts it is necessary to identify a *single* preferred version of the URI. This is defined in [RFC 6596] as the *canonical URI*. Since the GS1 Web URI encodes element strings as defined in the GS1 General Specifications [GENSPECS] such that `https://example.com/gtin/614141123452` and `(01)614141123452` are equivalent, and this document only defines short string alternatives for a subset of GS1 keys, we define the canonical URI as follows:

- the domain name SHALL be `id.gs1.org`;
- convenience string equivalents for AIs SHALL NOT be used;
- non-GS1 key=value pairs SHALL NOT appear in the query string.

5.3 An explanation of Web URI syntax and options for extensions

A Web URI is constructed from the concatenation of the protocol, the hostname, the path information and optionally a URI query string.

The protocol may be either "http" or "https", although "https" is recommended because it provides a secure communication channel over the Web.

Depending on usage, the hostname may be defined under the registered internet domain name of the brand owner, retailer / e-retailer or the designated hostname defined by GS1 as a reference Web URI prefix: `id.gs1.org`.

The path information consists of pairs of keys and values delimited by the forward slash ("/") or solidus character, such that each key precedes the corresponding value.

The query string typically consists of key=value pairs concatenated using either the ampersand("&") or semicolon(";") character. Within a GS1 Web URI, the query string accommodates data attributes in which the key is a GS1 Application Identifier corresponding to a data attribute, as well as serving as an extension point for additional arbitrary key=value pairs that may be used for example to indicate a specific role, action, activity or type of service to be accessed. Any key=value pairs used for extension data SHALL NOT conflict with keys used for GS1 Application Identifiers either in terms of semantics or syntax; a key for extension data SHALL NOT use a numeric GS1 Application Identifier NOR the values 'lot', 'exp' or 'expdt' nor should they be used to express a value (such as a value for net weight) if that value can be expressed using GS1 Application Identifiers as data attributes.

Section [5.1](#) defines the formal ABNF grammar for a Web URI structure for GS1 identification keys and the corresponding key qualifiers identified for each.

Note also that the formal ABNF grammar does indicate a preferred ordering of the key qualifiers for GTIN and ITIP, with CPV first, then lot/batch, then serial number - although any of these may be omitted, since the square brackets in ABNF notation indicate that an element is optional.

Note that there is no preferred ordering for key=value pairs appearing within the URI query string.

Section [5.4](#) provides a small number of examples of GS1 Web URIs to support many typical use cases.

5.4 Examples of GS1 Web URIs

5.4.1 GTIN

`https://id.gs1.org/gtin/614141123452`

`https://id.gs1.org/01/614141123452`

are equivalent reference GS1 Web URIs for GTIN 614141123452; the second version is canonical. They are both equivalent to the following element string:

`(01)00614141123452`

The following are all further valid GS1 Web URIs for GTIN 614141123452 using a custom domain name e.g. `example.com` instead of `id.gs1.org`

`https://brand.example.com/gtin/614141123452`

`https://brand.example.com/some-extra/pathinfo/gtin/614141123452`

`https://retailer.example.com/some-extra/pathinfo/gtin/614141123452`

If a product carries an EAN/UPC barcode encoding a GTIN, any software can construct a reference GS1 Web URI for that GTIN by appending the GTIN value to `"https://id.gs1.org/gtin/"`.

If redirection information has been specified to GS1 by the corresponding licensee of that GTIN or the GS1 Company Prefix (for GTINs constructed from GS1 Company Prefixes), a GS1 Resolver that supports GS1 Web URIs will be able to effectively redirect any requests for that GS1 Web URI to a corresponding Web URI or URL specified by the licensee.

5.4.2 GTIN + CPV

`https://id.gs1.org/gtin/614141123452/cpv/2A`

`https://id.gs1.org/01/614141123452/22/2A`

are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with Consumer Product Variant '2A'; the second example is canonical. They are both equivalent to the following element strings:

`(01)00614141123452(22)2A`

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with Consumer Product Variant '2A'.

`https://brand.example.com/gtin/614141123452/cpv/2A`

`https://brand.example.com/01/614141123452/22/2A`

`https://retailer.example.com/01/614141123452/22/2A`

5.4.3 GTIN + Batch/Lot

`https://id.gs1.org/gtin/614141123452/lot/ABC123`

`https://id.gs1.org/01/614141123452/10/ABC123`

are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC123'; the second version is canonical. They are both equivalent to the following element strings:

(01)00614141123452(10)ABC123

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC123'

<https://brand.example.com/gtin/614141123452/lot/ABC123>

<https://brand.example.com/01/614141123452/10/ABC123>

<https://retailer.example.com/01/614141123452/10/ABC123>

5.4.4 GTIN + Serial Number (also known as SGTIN)

<https://id.gs1.org/gtin/614141123452/ser/12345>

<https://id.gs1.org/01/614141123452/21/12345>

are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with Serial Number '12345'; the second version is canonical. They are both equivalent to the following element strings:

(01)00614141123452(21)12345

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with Serial Number '12345'

<https://brand.example.com/gtin/614141123452/ser/1234>

<https://brand.example.com/01/614141123452/21/12345>

<https://retailer.example.com/01/614141123452/21/12345>

5.4.5 GTIN + Batch/Lot + Serial Number + Expiry Date

<https://id.gs1.org/gtin/614141123452/lot/ABC1/ser/12345?exp=180426>

<https://id.gs1.org/01/614141123452/10/ABC1/21/12345?17=180426>

are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC1' and Serial Number '12345' and with an expiry date of 26th April 2018; the second version is canonical. They are both equivalent to the following element strings:

(01)00614141123452(17)180426(10)ABC1(21)12345

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC1' and Serial Number '12345' and with an expiry date of 26th April 2018.

<https://example.com/gtin/614141123452/lot/ABC1/ser/12345?exp=180426>

<https://example.com/01/614141123452/10/ABC1/21/12345?17=180426>

5.4.6 GTIN + Net Weight

<https://id.gs1.org/gtin/614141123452?3103=000195>

<https://id.gs1.org/01/614141123452?3103=000195>

are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with a net weight of 0.195 kg; the second version is canonical. They are both equivalent to the following element strings:

(01)00614141123452(3103)000195

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with a net weight of 0.195 kg :

<https://example.com/gtin/614141123452?3103=000195>

<https://example.com/01/614141123452?3103=000195>

5.4.7 GTIN + Typed Link

These examples are very similar to those of section [5.4.1](#) above but include an additional key=value pair in the URI query string.

In the first two examples, the key=value pair may be something like `typedLink=gs1_instructionsForUse` to provide a convenient declarative way of requesting the instruction manual for the specified product.

This additional key=value pair can be used to redirect to a different Web resource from the redirection that would otherwise happen, so whereas the example in section [5.4.1](#) might redirect to the brand owner's Web page about the product, these examples might be configured to redirect directly to the instruction manual for the product, providing a convenient direct link to specific types of information or services related to the product and greater convenience to the user by reducing or eliminating the need to navigate through multiple Web pages to find a particular kind of information or service.

```
https://id.gs1.org/gtin/614141123452?typedLink=gs1_instructionsForUse
```

```
https://id.gs1.org/01/614141123452?typedLink=gs1_instructionsForUse
```

Both of these are equivalent reference GS1 Web URIs for GTIN 614141123452 but they also include an extra key=value pair within the URI query string, which represents information that cannot be expressed unambiguously using GS1 Application Identifiers in element strings. The additional key=value pairs are lost when translating GS1 Web URIs to element strings and are not part of the canonical GS1 Web URI which would be `https://id.gs1.org/01/614141123452`. All three are equivalent to the following element string:

```
(01)00614141123452
```

The following are further valid GS1 Web URIs for GTIN 614141123452, also including the same additional key=value pair in the URI query string.

```
https://example.com/gtin/614141123452?typedLink=gs1_instructionsForUse
```

```
https://example.com/01/614141123452&typedLink=gs1_instructionsForUse
```

Note that at this stage, `typedLink` is only an example of a key. Its values have not yet been standardised by GS1. Brand owners, retailers and others are at liberty to use any key=value pairs but should take care to use percent encoding where appropriate for symbol characters, including colon. Future work on GS1 Web URI may define standardised values for various common typed links. There may be opportunities to align well with terms already defined in the GS1 Web vocabulary, as well as opportunities to extend that vocabulary, as well as contributing candidates for potential inclusion in [IANA link relations].

5.4.8 GTIN + Batch/Lot + Serial Number + Expiry Date + Typed Link

These examples are very similar to those of section [5.4.5](#) above but include an additional key=value pair in the URI query string.

In these two examples, the key=value pair may be something like `typedLink=hcpPortal` to provide a convenient declarative way of requesting information about the identified product that is provided via a portal for healthcare providers. This additional key=value pair can be used to redirect to a different Web resource from the redirection that would otherwise happen (e.g. in section [5.4.5](#)). The alternative Web resource may also check for an appropriate security token that asserts that the requestor is a healthcare provider.

```
https://id.gs1.org/gtin/614141123452/lot/ABC1/ser/12345?exp=180426&typedLink=hcpPortal
```

```
https://id.gs1.org/01/614141123452/10/ABC1/21/12345?17=180426&typedLink=hcpPortal
```

Both of these are equivalent reference GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC1' and Serial Number '12345' and with an expiry date of 26th April 2018 but they also include an extra key=value pair within the URI query string, which represents information that cannot be expressed unambiguously using GS1 Application Identifiers in element strings. The additional key=value pairs are lost when translating GS1 Web URIs to element strings and do not form part of the canonical URI:

<https://id.gs1.org/01/614141123452/10/ABC1/21/12345?17=180426>

All three are equivalent to the following element strings:

(01)00614141123452(17)180426(10)ABC1(21)12345

The following are further valid GS1 Web URIs for GTIN 614141123452 combined with Batch/Lot 'ABC1' and Serial Number '12345' and with an expiry date of 26th April 2018, also including the same additional key=value pair in the URI query string.

<https://example.com/gtin/614141123452/lot/ABC1/ser/12345?exp=180426&typedLink=hcpPortal>

<https://example.com/01/614141123452/10/ABC1/21/12345?17=180426&typedLink=hcpPortal>

5.4.9 SSCC

<https://id.gs1.org/sscc/106141412345678908>

<https://id.gs1.org/00/106141412345678908>

are equivalent reference GS1 Web URIs for SSCC 106141412345678908; the second version is canonical. They are both equivalent to the following element strings:

(00)106141412345678908

The following are further valid GS1 Web URIs for SSCC 106141412345678908 :

<https://example.com/sscc/106141412345678908>

<https://example.com/00/106141412345678908>

5.4.10 SSCC with specified Content, Count and Batch/Lot

<https://id.gs1.org/sscc/106141412345678908?02=00614141123452&37=25&10=ABC123>

<https://id.gs1.org/00/106141412345678908?02=00614141123452&37=25&10=ABC123>

are equivalent reference GS1 Web URIs for SSCC 106141412345678908 containing a count [AI (37)] of 25 instances of Content [AI (02)] 00614141123452 having Batch/Lot identifier [AI (10)] 'ABC123'; the second version is canonical. They are both equivalent to the following element strings:

(00)106141412345678908(02)00614141123452(37)25(10)ABC123

The following are further valid GS1 Web URIs for SSCC 106141412345678908 containing a count [AI (37)] of 25 instances of Content [AI (02)] 00614141123452 having Batch/Lot identifier [AI (10)] 'ABC123':

<https://example.com/sscc/106141412345678908?02=00614141123452&37=25&10=ABC123>

<https://example.com/00/106141412345678908?02=00614141123452&37=25&10=ABC123>

5.4.11 Physical location represented by a GLN or GLN + GLN Extension

<https://id.gs1.org/gln/0614141123452>

<https://id.gs1.org/414/0614141123452>

are equivalent reference GS1 Web URIs for GLN 0614141123452; the second version is canonical. They are both equivalent to the following element string:

(414)0614141123452

<https://id.gs1.org/gln/0614141123452/glnx/32a%2Fb>

<https://id.gs1.org/414/0614141123452/254/32a%2Fb>

are equivalent reference GS1 Web URIs for GLN 0614141123452 combined with a GLN extension '32a/b'; the second version is canonical. Note that because the forward slash character has a special meaning within Web URIs, it is replaced with %2F, its percent encoding when it is being used as a literal value, rather than as a URI path separator.

They are both equivalent to the following element strings:

```
(414)0614141123452(254)32a/b
```

The following are valid GS1 Web URIs for GLN 0614141123452 :

```
https://example.com/gln/0614141123452
```

```
https://example.com/414/0614141123452
```

The following are further valid GS1 Web URIs for GLN 0614141123452 combined with a GLN extension '32a/b' :

```
https://example.com/gln/0614141123452/glnx/32a%2Fb
```

```
https://example.com/414/0614141123452/glnx/32a%2Fb
```

6 Resolving GS1 Web URIs

This chapter is intended to be informative to provide background information about how Web URIs will be resolved, together with a number of examples.

Resolving a Web URI is simply the process of handling a Web request or looking up that Web URI and delivering a corresponding service. The service could:

- provide information in a human-readable format, such as a Web page;
- provide data in a machine readable format such as JSON or JSON-LD;
- provide any combination of the previous two, either as separate entities via [RFC 7231] or as a combined Web page with embedded machine readable data;
- redirect to another resolver capable of delivering a service.

That process might involve multiple organisations and multiple resolver services redirecting from one to another.

Resolvers will leverage existing normative standards such as those that define the HTTP Protocol and associated HTTP response codes. Any GS1-specific normative content regarding behaviour of resolvers will be provided in a separate document to be developed following pilot activities.

6.1 Resolving Protocol

The resolution of a GS1 Web URI may be done in part by one or more resolvers. The initial resolver contacted for a particular Web URI is determined by the domain name or hostname used in this URI.

For example, the GS1 Web URI:

```
https://id.gs1.org/gtin/9507000000001/ser/9090909090
```

will result in invoking the resolver bound to the: `https://id.gs1.org` hostname.

Similarly, the following GS1 Web URI:

```
https://example.com/gtin/9507000000001/ser/9090909090
```

will result in invoking the resolver bound to the: `https://example.com` hostname.

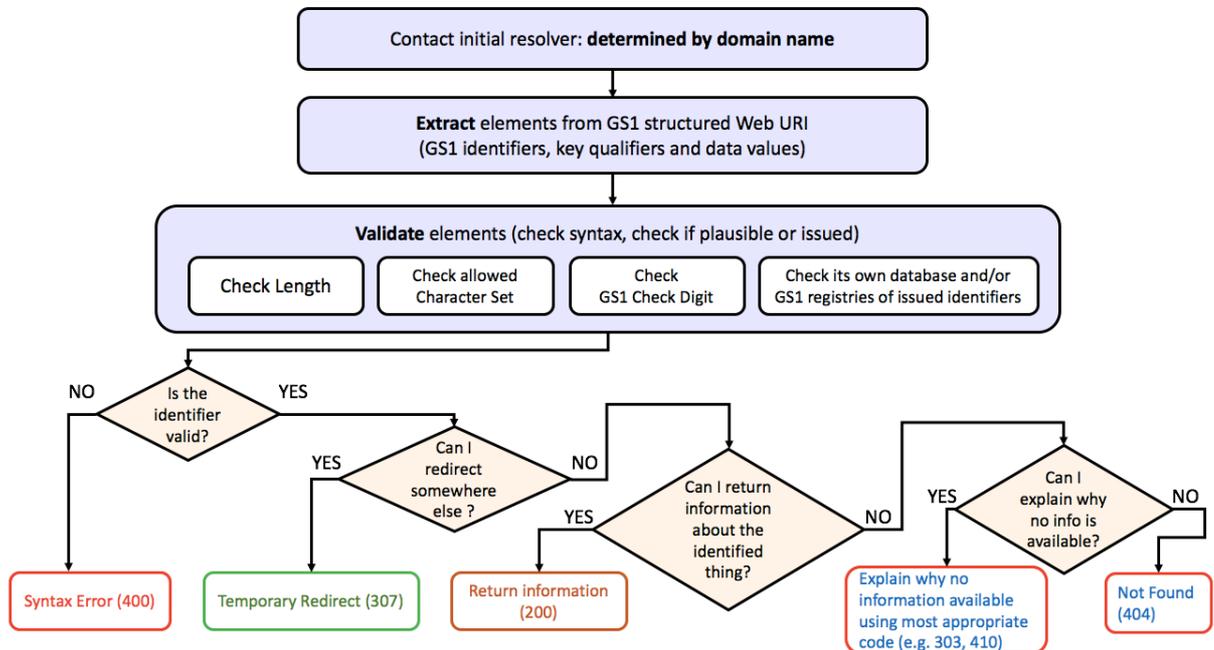
A resolver then performs 3 steps:

1. Extraction of elements from the GS1 Web URI
2. Validation of the extracted elements

3. Resolution of the GS1 Web URI to information or a redirection

The process is summarised in the flowchart in [Figure 6-1](#):

Figure 6-1 Summary flowchart of typical behaviour of a resolver for GS1 Web URIs



For redirection, use of HTTP response 302 (Found) is not recommended because of a number of inconsistent implementations in clients. It is safer and more flexible to use HTTP response 307 (Temporary Redirect) since this has been implemented more consistently and also provides greater flexibility in changing redirections, since it is not cached by the client or web browser.

6.2 Extraction step

For extraction of elements from a GS1 Web URI, to extract GS1 identification keys, key qualifiers and data values, a resolver may use a dedicated Web service or server-side script to parse the Web URI or it might make use of extraction rules with regular expressions. In situations where a resolver redirects one URI to another URI, it should be possible to pass any additional key=value pairs from the query string of the first URI to the query string of the second URI, without loss of information. Furthermore, there may be some situations in which specific key=value pairs are used to assist with Content Negotiation, as a declarative way of requesting a redirection to a specific type of information resource of service, e.g. to say "I'd like the instruction manual for this product" (see section [5.4.7](#)).

6.3 Validation step

Validation should be done for each extracted value. The actual validation steps to be performed depend on whether the extracted value is a GS1 identification key, a key qualifier or a data attribute.

Before performing these validation tests, it is essential to convert any percent-encoded characters to their ASCII equivalent, as shown in [Table 6-1](#), because some symbol characters have a special meaning when used within Web URIs and URLs and therefore need to be 'escaped' via percent encoding (see section 2.1 of RFC 3986 [PercentEncoding]) when they are used literally.

Table 6-1 Percent-encoding of symbol characters within Web URIs

Percent-encoded	Character	ASCII code (decimal)	Name of character
%21	!	33	Exclamation mark

Percent-encoded	Character	ASCII code (decimal)	Name of character
%23	#	35	Octothorpe (also known as Hash or US pound symbol)
%25	%	37	Percent
%26	&	38	Ampersand
%28	(40	Left bracket
%29)	41	Right bracket
%2A	*	42	Asterisk
%2B	+	43	Plus
%2C	,	44	Comma
%2F	/	47	Solidus or Forward slash

6.3.1 Check the length

For any value, check that its total number of digits or characters is within the limits in the ABNF grammar specified in Section 5.1 of this document or within Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

For example, if validating a GTIN value, the ABNF grammar states that the value should be exactly either 8, 12, 13 or 14 digits (`gtin-value = 8DIGIT / 12DIGIT / 13DIGIT / 14DIGIT`).

If validating a Batch/Lot number, the ABNF grammar states that the value should be up to 20 alphanumeric characters (`lot-value = 1*20XCHAR`) or `N2+X..20` in Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

If validating a GTIN Serial number, the ABNF grammar states that the value should be up to 20 alphanumeric characters (`ser-value = 1*20XCHAR`) or `N2+X..20` in Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

Similar checks should be made for other key qualifiers and data attribute values.

6.3.2 Check the character set

For any value, check that each character is within the character set specified in the ABNF grammar defined in Section 5.1 of this document or within Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

For example, if validating a GTIN value, the ABNF grammar states that the value should only consist of digits (`gtin-value = 8DIGIT / 12DIGIT / 13DIGIT / 14DIGIT`) or `N2+N14` in Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

If validating a Batch/Lot number, the ABNF grammar states that the value should consist of alphanumeric characters in the 82-character subset of ISO/IEC 646 (see Figure 7.11-1 of the GS1 General Specifications [GENSPECS]) (`lot-value = 1*20XCHAR`) or `N2+X..20` in Figure 3.1-1 of the GS1 General Specifications [GENSPECS].

The alphanumeric character set corresponding to Figure 7.11-1 corresponds to the following regular expression [`\x21-\x22\x25-\x39\x41-\x5A\x5F\x61-\x7A`]

The alphanumeric character set corresponding to Figure 7.11-2 corresponds to the following regular expression [`\x23\x2D\x2F\x30-\x39\x41-\x5A`]

6.3.3 Check the GS1 check digit (where appropriate)

For any value that corresponds to a GS1 identification key that has a GS1 check digit, perform the GS1 check digit calculation appropriate to that identifier as specified in section 7.9 of the GS1

General Specifications [GENSPECS] and check that it matches the value of the digit at the position where the check digit should appear.

Note that some GS1 identification keys do not include a check digit. Also note that the check digit is not always the final digit of a GS1 identification key. [Table 6-2](#) indicates which GS1 identification keys have a GS1 check digit - and its position within the identifier.

Table 6-2 Presence and position of GS1 check digit in various GS1 identification keys

GS1 identification key	Check Digit present?	Position of Check Digit
GTIN	Yes	Final digit of GTIN-8, GTIN-12, GTIN-13 or GTIN-14
ITIP	Yes	14 (calculated over digits 1-13)
SSCC	Yes	18 (calculated over digits 1-17)
GDTI	Yes	13 (calculated over digits 1-12)
GLN	Yes	13 (calculated over digits 1-12)
GRAI	Yes	13 (calculated over digits 1-12)
GSRN	Yes	18 (calculated over digits 1-17)
GSRN-P	Yes	18 (calculated over digits 1-17)
GSIN	Yes	17 (calculated over digits 1-16)
GCN	Yes	13 (calculated over digits 1-12)
GIAI	No	
CPID	No	
GMN	No	

Additionally, for values of the following data attributes, check digits should be checked, as shown in [Table 6-3](#):

Table 6-3 Position of GS1 check digit for various data attributes

AI for data attribute	Gen Specs data title	Position of Check Digit
02	CONTENT	Final digit of GTIN value
410	SHIP TO LOC	13 (calculated over digits 1-12)
411	BILL TO	13 (calculated over digits 1-12)
412	PURCHASE FROM	13 (calculated over digits 1-12)
413	SHIP FOR LOC	13 (calculated over digits 1-12)
414	LOC No	13 (calculated over digits 1-12)
415	PAY TO	13 (calculated over digits 1-12)
416	PROD/SERV LOC	13 (calculated over digits 1-12)



Note: many of the validation checks on the length, character set and GS1 check digit (if any) can be performed at the time of inserting a redirection rule into the resolver for that specific identifier; the redirection will only happen for a specified identifier that is valid.

6.3.4 Check the internal database of identifiers supported by this resolver

Some resolvers (especially those operated by brand owners, manufacturers, retailers or service providers) may only support a limited set of values for a limited range of GS1 identification keys. This internal database may be implemented as a database that is accessed by a validation script or it may be implemented as an explicit literal string pattern appearing within a URI rewriting tool, e.g. within the configuration file of the company's Web site. It may also be possible to perform validation using GS1 registries of issued identifiers.

6.4 Resolution step

Once the GS1 Web URI has been validated, a resolver may match specific identifiers, including extension identifiers of which it is aware, and redirect to Web URIs or URLs that were specified by the respective licensee of a specific GS1 identification key or GS1 Company Prefix.

- If no redirection was specified for that GS1 identification key or GS1 Company Prefix by the respective licensee, a final stage of the process may do further checks of the Web URI syntax and also attempt to retrieve information from other sources.
- If a syntax error is found, it should return an HTTP 400 response and an error page that explains the details of the syntax error.
- If no information nor redirection was found for the GS1 Web URI a 404 'Not Found' response code should be returned to the client.
- If the syntax is valid, it may attempt to retrieve some useful information from other services, including some information about the product (potentially retrieved from GS1 Cloud) or about the licensee (potentially retrieved from GEPIR.org)

6.4.1 A note on HTTP response codes:

302 (Found) or 307 (Temporary Redirect) is more appropriate than 301 (Moved Permanently) as in typical situations, flexibility is needed to update the redirection at any time; a 301 (Moved Permanently) response is typically stored within a Web browser cache, which will generally mean that the Web browser will not discover any new redirection until the browser cache is cleared, whereas 302 and 307 (Temporary Redirect) are not cached.

6.4.2 When a resolver has only partial information

A resolver should make a *best effort* to redirect the request to a source of information or to return information directly. There may be circumstances where the resolver is not able to complete the request, which can be for any number of reasons. There are a number of HTTP response codes that may be useful in providing an explanation as to *why* a request cannot be completed. These include 401 (Unauthorised) meaning that the client must be logged in, 402 (Payment Required) and 410 (Gone) meaning that the identifier is no longer in use and the request should not be made again. 303 (See Other) can be used to redirect to a resource that explains why the original request cannot be completed.

The use of such response codes is optional but is encouraged to minimise the number of occasions when a request results in a 404 (Not Found) response.

Implementations should use the most appropriate code for each circumstance [HTTPcodes].

6.5 Examples of use

This section contains examples of use of resolvers. It describes two ways of using resolvers: using the provided domain and replacing the domain to use alternative resolvers.

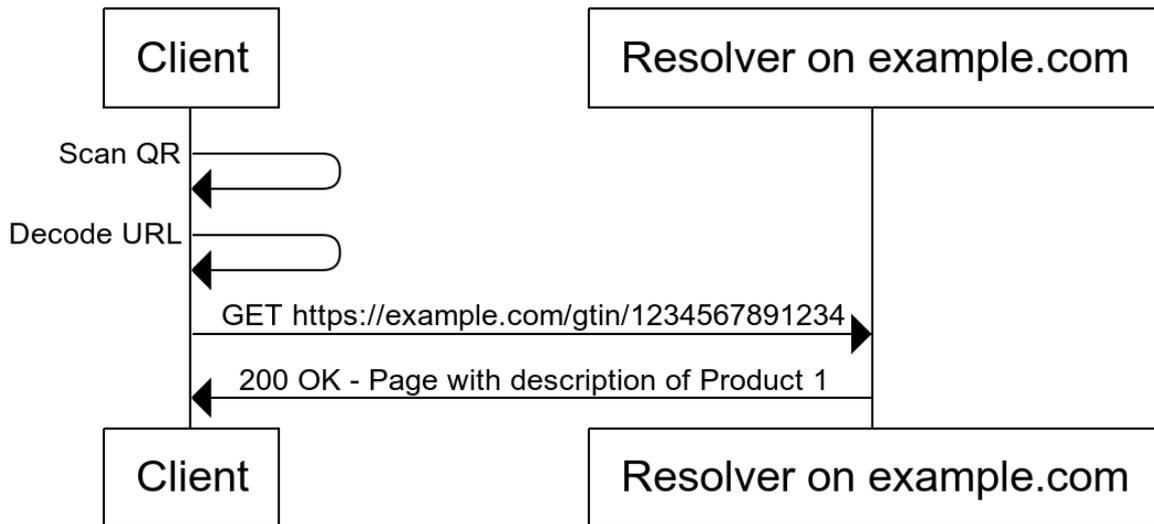
6.5.1 Using the provided domain to call the resolver

In this first case the resolver is identified by the domain name contained in the GS1 Web URI. To call this resolver the client simply needs to "go to this Web address".

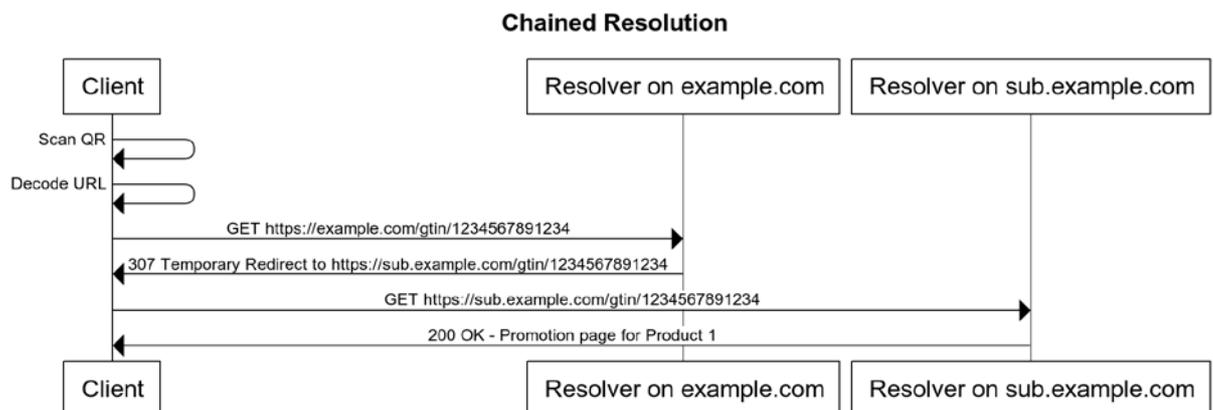
Let's assume an on pack QR code on product 1 containing a brand example.com managed GS1 Web URI such as `https://example.com/gtin/1234567891234`. A customer scanning this QR code with a native QR code application (e.g., an iOS camera application) will reach the resolver at `https://example.com`. This resolver then chooses what experience should be served to the customer, for instance offering a page with a description of product 1.

Figure 6-2 Simple resolution of a Web URI

Simple Resolution



However, resolvers can also be chained. For instance, brand `example.com` might have sub-brands (e.g., `sub.example.com`) managing products with their own resolvers. In this case, brand `example.com` might decide to redirect the client calling `https://example.com/gtin/1234567891234` to a more specific sub-brand resolver at `https://sub.example.com/gtin/1234567891234`. The website at `sub.example.com` then decides to serve a page with a promotion for product 1.

Figure 6-3 Chained resolution for a Web URI


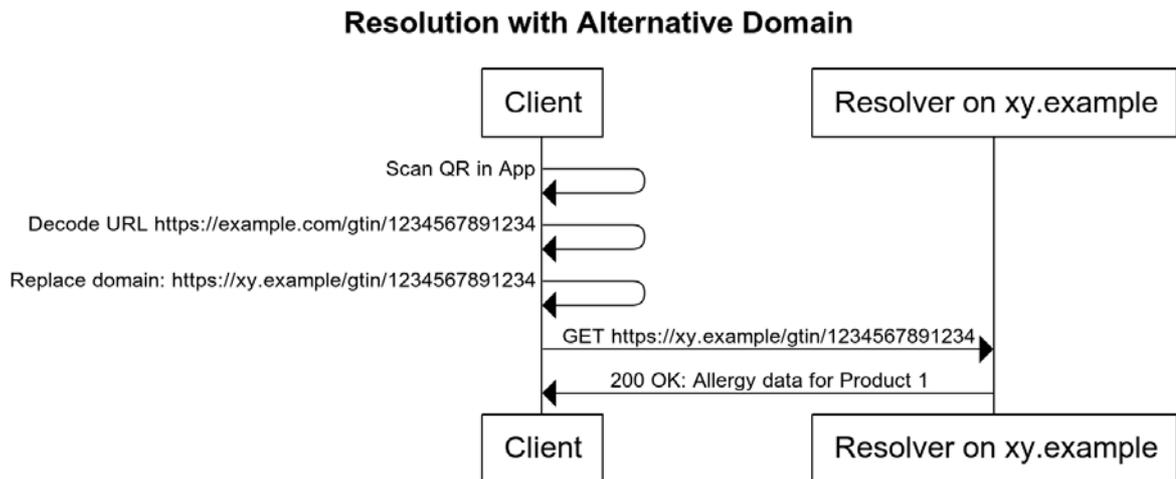
In this example, the second resolver (`sub.example.com`) is capable of serving the request. However, further redirection (e.g., to a third or fourth resolver) is possible. In practice a large number of redirections should be avoided as each redirection needs to be called by the client leading to a slow response time and a poor consumer experience.

6.5.2 Replacing the domain to call an alternative resolver

The following use cases make use of alternative resolvers rather than the resolvers provided by the brand owner or by GS1. They do so in order to be able to provide additional information from other parties, including information about price comparison, availability and promotions, as well as comparisons with related products that may be more suitable in some way. They may also link to independent accreditation agencies who certify claims about the product (such as organic status, fair trade status, sustainable sourcing etc.). At a technical level, these all behave in a similar way,

typically by extracting the path information and query string of the original Web URI and appending it to their own domain name or Web URI or passing these as input parameters to a Web resource that they operate, which may draw upon data from multiple sources and may use these to present data about the scanned product and related products in a format that is more convenient to the user or which provides additional benefits or additional insights for the benefit of the consumer.

Figure 6-4 Resolution with an alternative domain



6.5.3 Retailer operates their own resolver

A retailer may wish to configure its in-store WiFi or in-store handheld self-scan terminals or the retailer's own smartphone app to make use of the retailer's own resolver so that when a product is scanned, in order that the retailer can provide their information about the offer for the product, including price, availability, promotional offers etc., as well as including (embedding) or linking to product master data provided by the brand owner and potentially also by others (such as independent accreditation agencies).

Figure 6-5 Example in which a retailer operates their own resolver

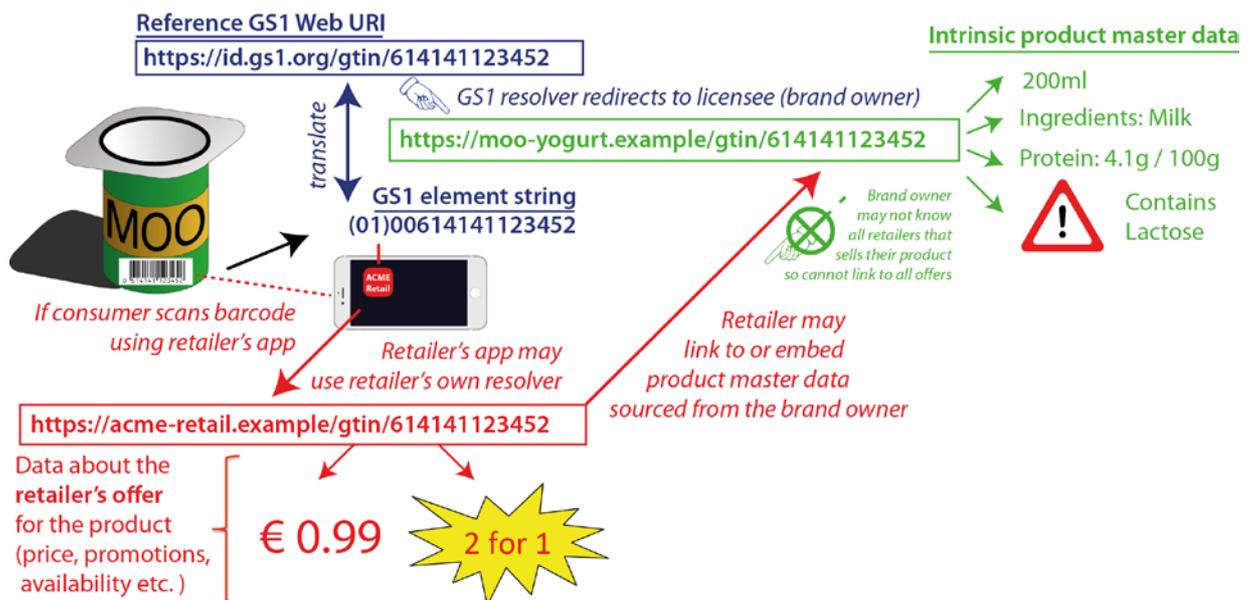


Figure 6-5 shows a GTIN in plain syntax being read from an existing UPC or EAN barcode but the same principles apply for other GS1 barcodes such as GS1 DataMatrix, GS1 QR Code or GS1 DataBar that carry one or more element strings. For any existing GS1 data carrier that encodes a

primary GS1 identification key, it is possible to translate the corresponding element strings or plain syntax into a reference GS1 Web URI. A GS1 Resolver can redirect requests for the reference GS1 Web URI to an alternative Web URI, such as one specified by the licensee of the respective GS1 identification key or GS1 Company Prefix. Alternatively, a GS1 Web URI might be encoded in any data carrier that can accommodate a URL or Web URI. In that situation, it is possible to translate from the GS1 Web URI to the corresponding element string, to support supply chain or retail applications that need that.

The diagram also shows how a retailer may use their own resolver or proxy server to redirect requests to their own corresponding product pages, so that they can provide information about their offers for the product, including price, availability and promotions as well as opportunities for cross-selling, up-selling or alerting the consumer to related products that may be of interest. Note that there are a limited number of opportunities through which a retailer can achieve such redirection. One scenario is when a consumer connects to in-store WiFi provided by the retailer. Another scenario is when a consumer uses a particular smartphone app provided by the retailer. This limited ability for the retailer to redirect to their product offer page is independent of the kind of data carrier used. In other situations where a consumer uses a generic scanning app and uses their own data connectivity, they will normally directly access information resources and services specified by the respective brand owner, manufacturer or licensee of the GS1 identification key or GS1 Company Prefix, rather than any information provided by any particular retailer.

This is a particularly important use case to consider because the manufacturer or brand owner will not know the details of every potentially seller of their product and will generally not be able to provide an outward link to each retailer and the retailer's details about the product offer. However, each retailer can publish their own information about their offer for the product and can embed or link back to the product page and associated master data published by the brand owner.

6.5.3.1 Semantic implementation

When considering machine-readable access to semantic structured data, it is worth noting that the schema.org vocabulary includes a data class that represents the retailer's offer for a product (<http://schema.org/Offer>). It has an 'itemOffered' property (<http://schema.org/itemOffered>) that provides a semantic typed link to the corresponding Web URI that represents the product itself and the associated intrinsic product master data, as published by the brand owner.

6.5.4 Smartphone app for nutrition / dietary / allergen information

A smartphone app that is focused on provision of customised nutrition and dietary information to a consumer may also be configured to use its own resolver to provide a different response to the consumer, rather than simply redirecting to the brand owner's product page and associated master data.

The provider of an app may provide a Web resource that either makes a dynamic synchronous query to the brand owner's site (and possibly to other Web resources) to retrieve relevant data or it may have previously cached a recent copy of such data in its own internal database and provide that in response.

Such an app may display such information to the user either in a similar format to how it appears on the product packaging or it may provide additional benefits, such as:

- Translation of the names of ingredients into the user's local language (particularly useful if the user is shopping for food in a country where they don't understand the local language).
- Alerting the user to the health benefits/risks of the product (e.g. whether the product is high or low in fat, sugar, calories etc.).
- Alerting the user to potential allergens within the product. This could also be custom configured by the user for allergens that are particularly relevant to the user. Note that the filtering by allergen may be performed entirely client-side within the app for privacy reasons, without any real need to share such details with the back-end service that supports the app.
- Using the product's GTIN and GPC to find related products that are more nutritious / healthier or more compatible with the user's specific dietary needs.

6.5.5 Product comparison apps

A smartphone app that is focused on comparison of products may be configured to use its own resolver to provide a different response to the consumer, rather than simply redirecting to the brand owner's product page and associated master data.

The provider of an app may provide a Web resource that either makes a dynamic synchronous query to the brand owner's site (and possibly to other Web resources) to retrieve relevant data or it may have previously cached a recent copy of such data in its own internal database and provide that in response.

Such an app may display such information to the user either in a similar format to how it appears on the product packaging or it may provide additional benefits, such as:

- Details about current and historic online price comparison information for offers of the product, so that the user can find the best deals either online or in local physical stores and understand objectively how the current price compares with historic price fluctuations for that product
- Information (product master data, price, availability etc.) about related products that are broadly similar (typically having the same GPC brick value) but which may be preferable in some way, such as:
 - Healthier / more nutritious
 - Locally sourced
 - Source or manufactured in a way that is more:
 - Environmentally friendly / sustainable
 - Ethical / fair trade / cooperative
 - Suitable for a particular diet

Such product comparison apps may make use of data from multiple parties, including the brand owner, multiple retailers making offers for the product, as well as independent accreditation agencies (e.g. Marine Stewardship Council, Soil Association etc.)

7 Glossary

The glossary lists the terms and definitions that are applied in this document. Please refer to the www.gs1.org/glossary for the online version

Term	Definition	Source
attribute	An element string that provides additional information about an entity identified with a GS1 identification key, such as batch number associated with a Global Trade Item Number (GTIN).	GENSPECS
brand owner	The organisation that owns the specifications of a trade item, regardless of where and by whom it is manufactured. The brand owner is normally responsible for the management of the Global Trade Item Number (GTIN).	GENSPECS
Canonical GS1 Web URI	The definitive GS1 Web URI for a given resource. See section 5.2	
Certified information agency (also known as an 'accreditation agency')	An organization which certifies that a product meets specific criteria (e.g., Fairtrade, Kosher, Halal, Electrical Rating)	
Consumer	Often considered as the "recipient" of the supply chain in the past, today's consumer is an active part of the supply chain and expects more data, with higher accuracy, and greater ease.	
consumer product variant (CPV)	An alphanumeric attribute of a GTIN assigned to a retail consumer trade item variant for its lifetime.	GENSPECS

Term	Definition	Source
Content Negotiation	Content negotiation is a mechanism for offering information in different data formats and different languages. When a user agent (such as a browser) makes an HTTP request, it sends along some HTTP headers to indicate what data formats and language it prefers. The server then selects the best match from its file system or generates the desired content on demand, and sends it back to the client.	
data field	A field that contains a GS1 identification key, an RCN, or attribute information	GENSPECS
data titles	Data titles are the abbreviated descriptions of element strings which are used to support manual interpretation of barcodes.	GENSPECS
DataMatrix	A standalone, two-dimensional matrix symbology that is made up of square modules arranged within a perimeter finder pattern. DataMatrix ISO version ECC 200 is the only version that supports GS1 system identification numbers, including the Function 1 Symbol Character (FNC1). DataMatrix symbols are read by two-dimensional imaging scanners or vision systems.	
Dereferencing a URI	The use of an appropriate access mechanism (e.g. Web request) to perform an action on the URI's resource (e.g. to retrieve an information representation via HTTP GET or to send data to a resource via an HTTP POST operation). Dereferencing a URI is often considered synonymous with making a Web request or 'looking up' a URI on the Web.	
direct mode	Mobile device information retrieval function when the barcode contains either the address (URL) of the content or service, or the content itself, in-line.	GENSPECS
Domain name	A domain name is an identification string that defines a realm of administrative autonomy, authority or control within the Internet. Domain names are formed by the rules and procedures of the Domain Name System (DNS). Any name registered in the DNS is a domain name. Domain names are used in various networking contexts and application-specific naming and addressing purposes. Domain names provide an abstraction layer that separates a registered name for an organisation or activity from the actual internet addresses (IP addresses) that provide its associated information services such as its Website, its e-mail server etc. The system that connects the domain names with the corresponding IP addresses is the Domain Name System (DNS).	
EAN/UPC barcode symbology	A family of barcodes including EAN-8, EAN-13, UPC-A, and UPC-E barcodes. Although UPC-E barcodes do not have a separate symbology identifier, they act like a separate symbology through the scanning application software. See also EAN-8 barcode, EAN-13 barcode, UPC-A barcode, and UPC-E barcode.	
element string	The combination of a GS1 Application Identifier and GS1 Application Identifier data field.	GENSPECS
GS1 AIDC data carrier	A means to represent data in a machine readable form; used to enable automatic reading of the element strings as specified for use by GS1.	GENSPECS
GS1 Application identifier	The field of two or more digits at the beginning of an element string that uniquely defines its format and meaning.	GENSPECS
GS1 Application identifier data field	The data used in a business application defined by one GS1 Application Identifier.	GENSPECS
GS1 barcode	A data carrier which encodes GS1 Application Identifier element strings.	
GS1 barcode using GS1 Application Identifiers	All GS1 endorsed barcode symbologies that can encode more than a GTIN namely GS1-128, GS1 DataMatrix, GS1 DataBar and Composite and GS1 QR Code.	
GS1 DataBar	The GS1 DataBar family consists of seven types of barcodes. Four of the barcodes are applied in scanning at retail point-of-sale (POS), two of which are able to carry additional information such as a serial number, lot number of expiry date. GS1 DataBar is also applied in general distribution and logistics environments.	
GS1 DataMatrix	GS1 implementation specification for use of DataMatrix	GENSPECS
GS1 Digital Bridge	A synonym for GS1 Resolver	

Term	Definition	Source
GS1 identification key	A unique identifier for a class of objects (e.g. a trade item) or an instance of an object (e.g. a logistic unit).	GENSPECS
GS1 key qualifier	A key qualifier is an additional attribute that is designated for use as part of a compound key (e.g., GTIN + serial number is a compound key, with the serial number being a key qualifier for the GTIN)	
GS1 QR Code	GS1 implementation specification for use of QR Code	GENSPECS
GS1 Resolver	A Web server that is able to understand the GS1 Web URI syntax	
GS1 Web URI	A Web URI conforming to the GS1 Web URI syntax.	
HTTP status codes	The status-code element is a three-digit integer code giving the result of the attempt to understand and satisfy the request.	
human readable interpretation (HRI)	Characters, such as letters and numbers, which can be read by persons and are encoded in GS1 AIDC data carriers confined to a GS1 standard structure and format. The human readable interpretation is a one-to-one illustration of the encoded data...	GENSPECS
identification number	A numeric or alphanumeric field intended to enable the recognition of one entity versus another.	GENSPECS
Indirect mode	Mobile device information retrieval function when the code contains an identifier, which needs to be resolved to obtain the content or service. Resolving an identifier means looking it up, typically at a network service, to determine the corresponding content or service.	
Internet application provider	A company that provides an application or service available to consumers via an Internet-based platform (desktop or mobile), including Websites, smartphone apps etc.	
LGTIN (GTIN + Lot/Batch)	A compound key formed from the combination of GTIN [AI (01)] and Batch/Lot identifier [AI (10)]. LGTIN is defined as an EPC Class URN in the current GS1 Tag Data Standard (v1.11), sections 6.4.1 and 7.14, which describes the mapping between the EPC Class URN format for LGTIN and the corresponding element string.	
Media Type (also known as MIME type or Content type)	A two-part string identifier that indicates a data format as a pair of type and subtype, e.g. image/jpeg, image/gif, image/png, text/html, text/rtf Media types are sometimes also referred to as MIME types (MIME is an acronym of Multipurpose Internet Mail Extensions) or Content Types (after the HTTP header that indicates the Media type)	
Mobile scanning	An approach to giving consumers access to additional information or services about trade items through their mobile device. It is the ability to retrieve additional information about the trade item through mobile devices or in general between link a trade item with virtual information or services.	
Parsing	The process of analysing the structure of a sentence or URI structure in order to extract relevant information from it. Note that within the context of EPC URN structures, parsing refers to the ability to extract structural components within the EPC structure, e.g. for the purpose of matching against EPC URN patterns.	
QR Code®	A two-dimensional matrix symbology consisting of square modules arranged in a square pattern. The symbology is characterised by a unique finder pattern located at three corners of the symbol. QR Code® symbols are read by two-dimensional imaging scanners or vision systems	
Reference GS1 Web URI	A GS1 Web URI that uses the id.gs1.org domain	
Retailer	An organization engaged in the sale and distribution of products to consumers. Also includes online retailers / e-tailers	
SGTIN (Serialised GTIN)	A compound key formed from the combination of a GTIN [AI (01)] with Serial Number [AI (21)] which provides globally unique identification for every instance of a product. The term SGTIN appears in section 6.3.1 and 7.1 of the current GS1 Tag Data Standard, v1.11	

Term	Definition	Source
Subdomain	A subdomain is a domain that is part of a main domain. Although example.com is a subdomain of the top-level domain (.com), we most often think of a subdomain as the part of the hostname that precedes the registered domain name. For example, the registered domain name gs1.org has one subdomain ('www') [as in www.gs1.org] that is used for its Website. It also has a subdomain ('id') [as in id.gs1.org] that is used for Web-based data services for GS1.	
URI	Uniform Resource Identifier. A string of characters used to identify a resource. The resource may be an information resource such as a Web page or a thing in the real world, such as a physical object, person or location. URIs refer to the superset of Uniform Resource Names (URNs), Uniform Resource Locators (URLs) and Web URIs (which can function both as globally unambiguous names, while also behaving like URLs by enabling intuitive retrieval of related information via the Web).	
URI fragment identifier	The fragment identifier component of a URI allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information. The identified secondary resource may be some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations. A fragment identifier component is indicated by the presence of an octothorpe / hash / number sign ("#") character and terminated by the end of the URI. A typical use of a URI fragment identifier is to provide a direct link to a specific section within a very long Web document such as https://www.w3.org/TR/dwbp/#DataIdentifiers	
URI path information	A path consists of a sequence of path segments separated by a slash ("/") character. A path is always defined for a URI, though the defined path may be empty (zero length). The path component contains data, usually organized in hierarchical form, that, along with data in the non-hierarchical query component, serves to identify a resource within the scope of the URI's scheme and naming authority (if any). The path is terminated by the first question mark ("?") or number sign("#") character, or by the end of the URI.	
URI query string	The query component contains non-hierarchical data that, along with data in the path component, serves to identify a resource within the scope of the URI's scheme and naming authority (if any). The query component is indicated by the first question mark ("?") character and terminated by a number sign("#") character or by the end of the URI.	
URI resolution	URI "resolution" is the process of determining an access mechanism and the appropriate parameters necessary to dereference a URI; this resolution may require several iterations.	
URI resolver service	A URI resolver service is any system that performs the resolution of a URI (see URI resolution) in order that it can be dereferenced (in order to support some kind of interaction with the corresponding resource, such as information retrieval).	
URI rewriting	URI rewriting is a mechanism that provides a mapping between a logical address for an information resource (expressed via a URI) and the actual mechanism (such as a server-side script) that is used to retrieve the corresponding information resource.	
URL	Uniform Resource Locator (URL), a specific type of URI colloquially known as Web address. A URL is a URI starting with http or https .	

8 References

ID	Name	Details
[ARCH]	GS1 System Architecture	v7, GS1, February 2018 https://www.gs1.org/docs/architecture/GS1_System_Architecture.pdf
[RFC 7231]	HTTP/1.1 Semantics and Content, section 3.4, "Content Negotiation"	https://tools.ietf.org/html/rfc7231#section-3.4
[DC]	Dublin Core Metadata Initiative	http://dublincore.org/documents/dcmi-terms/
[GS1 identification keys]	GS1 identification keys	https://www.gs1.org/standards/id-keys
[GENSPECS]	GS1 General Specifications	v18, January 2018 https://www.gs1.org/sites/default/files/docs/barcodes/GS1_General_Specifications.pdf
[Geolocation API]	Geolocation API Specification 2nd Edition	https://www.w3.org/TR/geolocation-API/
[GS1History]	How we got here	https://www.gs1.org/about/how-we-got-here
[GS1Voc]	The GS1 Web Vocabulary	https://www.gs1.org/voc/
[HTTPcodes]	Hypertext Transfer Protocol [RFC 2616], Section 10: Status Code Definitions	https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
[IANA Link Relations]	List of defined Link Relations, hosted by IANA.	https://www.iana.org/assignments/link-relations/link-relations.xhtml
[IMP]	Information Management: A Proposal	https://www.w3.org/History/1989/proposal.html
[IRIs]	Internationalized Resource Identifiers (IRIs)	https://tools.ietf.org/html/rfc3987
[JSON-LD]	A JSON-based serialization for Linked Data	https://www.w3.org/TR/json-ld/
[Linked Data]	Linked Data, Tim Berners-Lee, 2006	https://www.w3.org/DesignIssues/LinkedData
[PercentEncoding]	Uniform Resource Identifier (URI): Generic Syntax, section 2.1: Percent-Encoding	https://tools.ietf.org/html/rfc3986#section-2.1
[RDF]	Resource Description Framework Primer	https://www.w3.org/TR/rdf11-primer/
[RDF Intro]	Introduction to RDF Metadata	https://www.w3.org/TR/NOTE-rdf-simple-intro
[RFC 2606]	Reserved Top Level Domain Names	https://tools.ietf.org/html/rfc2606

ID	Name	Details
[RFC 3986]	Uniform Resource Identifier (URI): Generic Syntax	https://tools.ietf.org/html/rfc3986
[RFC 5234]	Augmented BNF for Syntax Specifications: ABNF	https://tools.ietf.org/html/rfc5234
[RFC 6570]	Uri Template	https://tools.ietf.org/html/rfc6570
[RFC 6596]	The Canonical Link Relation	https://tools.ietf.org/html/rfc6596
[RFC 6761]	Special-Use Domain Names	https://tools.ietf.org/html/rfc6761
[RFC 7232]	HTTP/1.1 Conditional Requests	https://tools.ietf.org/html/rfc7232
[RFC 7235]	HTTP/1.1 Authentication	https://tools.ietf.org/html/rfc7235
[RFC 7405]	Case-Sensitive String Support in ABNF	https://tools.ietf.org/html/rfc7405

A Intellectual property

Preliminary note

Readers are reminded of the disclaimer included at the beginning of this document regarding the GS1 IP Policy, its application and scope.

US Patent No. 8,590,776

GS1 has been made aware of US Patent No. 8,590,776 (for the purpose of this specific paragraph, the "Patent"). The Patent creates a system of identifiers and sample code. This sample code, in the section about the owner, contains a URI that may point to additional information concerning properties or rights reading on the physical object the code is attached to. The GS1 Web URI Standard transforms existing GS1 identifiers into the URI syntax. GS1 identifiers have been around for a long time [GS1 History]. The URI syntax to point to physical objects has been around since 1997 [RDF intro]. This means that transforming existing GS1 identifiers to URIs does not correspond to the claims that require additional information to be present at a specific point of the identifier. While the patented system encodes meta-information directly into the sample code, the GS1 Web URI Standard supports requesting more meta-information from an HTTP server and this has been state of the art since 1989 [IMP].

US Patents No. 9,794,321 and 9,582,595

EVERYTHNG Limited of 122 East Road, London N16FB (United Kingdom), the owner of the patents listed in this sub-section (for the purpose of this specific paragraph, the "Patents"), who participated to the Work Group designing the GS1 Web URI Standard (Release 1.0), gave notice that the Patents may have relevance to implementations of the GS1 Web URI Standard (Release 1.0), particularly in relation to implementing resolvers leveraging context to dynamically change redirections (related to section [4.5](#) of this standard). The owners have acknowledged that the Patents do not contain any Necessary Claims (as defined in the GS1 IP Policy) but have offered to grant a royalty free, non-exclusive license to the Patents to users of the GS1 Web URI Standard (Release 1.0), for use with their applications implementing it, and subject to (i) GS1 including a notice of the availability of the license in its documentation of the Standard, (ii) users who wish to benefit of the license registering with the owner to receive the license (via <https://evrythng.com/gs1-license/>) and (iii) such users including an acknowledgement of the use of the Patents in any of their applications that implement the GS1 Web URI Standard. The term of the offered license will be for the life of the Patents. For the avoidance of any doubt, this licensing offer pertains only to implementations of Release 1.0 of the GS1 Web URI Standard

US Patent No. 9,864,889 and pending applications EP3147890 and CN107016430 US Patent applications 20180025195, EP3276503 and CN107065291

MobiLead SAS of 104 avenue de France, 75013 Paris (France), the owner of the granted patent and five patent applications listed in this sub-section (for the purpose of this specific paragraph, the "Patent and Patent Applications") who participated to the Work Group designing the GS1 Web URI Standard (Release 1.0), gave notice in accordance with the GS1 IP Policy of certain Necessary Claims in relation to the GS1 Web URI Standard (Release 1.0). The use case descriptions mentioned in section [6.5.4](#) and [6.5.5](#) of the standard mention several aspects that may be covered by the Patents and/or Patent Applications. The GS1 IP Policy is only applicable to Necessary Claims. Necessary Claims means all present, pending and hereafter acquired patent claims that would be necessarily infringed by implementing the Standard. A claim is necessarily infringed only when it is not possible to avoid such infringement because there is no non-infringing alternative for implementing the Standard. Given that the claims of the Patents and/or Patent Applications only concern non-normative parts of the GS1 Web URI Standard (Release 1.0), the GS1 Web URI Standard (Release 1.0) can be implemented without necessarily infringing the Patent and/or Patent Applications.