The Global Language of Business

The How and Why of GS1 Digital Link

A discussion of why GS1 Digital Link was developed and how it meets the needs of industry.

Version 1.1 November 2021

Document Summary

Document Item	Current Value
Document Name	The How and Why of GS1 Digital Link
Document Date	3 November 2021
Document Version	1.1
Document Issue	
Document Status	Informal

Contributors

Name	Organisation
Phil Archer	GS1 Global Office
Mark Harrison	Milecastle Media

Abstract

The fundamental aim of GS1 Digital Link is to enable anyone to find answers to their questions about the thing in front of them. This is a very significant departure from the traditional practice of managing a large but discrete database covering as many items as possible and then using an item identifier to look up the relevant information. The change is motivated by changes in society where, thanks to the Internet and, in particular, the ubiquity of smartphones in many parts of the world, there is an expectation that all facts are known and immediately knowable. Given that powerful motivation, this document sets out how the GS1 Digital Link standard was developed to enable stakeholders throughout the supply chain and beyond to meet that demand. Where choices of technology are available, these are reviewed and the reasons for making each choice stated. In most cases, choices are very limited and/or the correct choice obvious, given the existing state of the art and the massive implementation of Web technologies.

Table of Contents

1 Introduction and purpose of this document

It was on 31st March, 1971, that a small group of industry players met in New York to discuss the problem of how to automate the process of distinguishing between different products on the shelf and doing away with the need to apply price stickers to everything. 3 years and 3 months later, at 08:01 on 26 June 1974, at Marsh Stores, Troy, Ohio, the first checkout went beep. As if by magic, checkout operator Sharon Buchanan's cash register 'knew' that customer Clyde Dawson's purchase was a 10 pack of Wrigley's Juicy Fruit Gum and the price was $0.69 [Marsh]

Members of the ad-hoc committee that met in New York, 31 March 1971	
R. Burt Gookin	H.J. Heinz Company
John F Hayes	H.J. Heinz Company
Robert B Wegman	Wegman Food Markets, Inc.
J.P. McFarland	General Mills, Inc.
Thomas P. Nelson	General Mills, Inc.
Robert O. Aders	The Kroger Company
John L. Strubbe	The Kroger Company
John C. Suerth	Gerber Products Company
Alan L. Haberman	First National Stores
Robert A. Stringer	General Foods Corporation
Bert L. Thomas	Winn Dixie Stores
Fred Butler	Bristol Myers Company
James T. Wyman	Super Valu Stores, Inc.
Raymond Wolfe	The Oshawa Group Ltd.
Earl W. Madsen	Madsen Enterprises, Inc.

How?

Even among people old enough to remember, it's hard now to conceive of the technology of the early 70s. Computers were the preserve of very large companies and had big reels of tape whizzing back and forth inside gun-metal grey cabinets the size of wardrobes. As a reference point, the first floppy disk, which was a full 8 inches (20cm) in diameter, didn't arrive until 1972, the 5¼" version in 1976 and the 3.5" version, that actually wasn't floppy at all but survives to this day as the ubiquitous 'save' icon, arrived in 1981 [FDD].

You would use one of those 'mainframe' computers via a terminal that was physically connected to it. The technology has, of course, changed dramatically. Devices of all sizes communicate wirelessly as well as physically using a common protocol (the Internet) for which, as it happens, 1974 is also a pivotal year [RFC 765].

PFC Patricia Barbeau operates a tape-drive on the IBM 729 at Camp Smith, Hawaii, October 1969

1.1 Why GS1 Digital Link is a turning point

Although there has been a huge change in information technology since the 1970s, the mindset and business practice have remained remarkably static throughout. Start with a big database held in an expensive computer and use a key, such as a number read from a barcode, to look up an item *in that database*.

This is exactly the situation today with manufacturer and retailer IT systems, supply chain and back of store operations, point of sale scanners, and the thousands of apps available for mobile devices that allow you scan a barcode and look up an item. These are simply user interfaces for whichever big database your device is connected to or your app communicates with.

GS1 Digital Link turns that on its head.

Rather than starting with a database and looking up and item, GS1 Digital Link starts with the item and points to one or more places where there is information about it.

This change is motivated and made necessary by the modern world in which all facts are known and all facts are instantly knowable by anyone. Which film won the Best Picture Oscar in 1974? What's the highest mountain in Ghana? What colours can my dog see? What's the atomic number of molybdenum? Who won the 100m freestyle in the 1968 Olympic Games? Give me a recipe for spaghetti puttanesca!

All you have to do is take out your phone and ask.

The fundamental aim of GS1 Digital Link is to enable anyone to find answers to their questions about the thing in front of them. Where did this come from? What's in it? Where can I buy spares? How do I use it? Has it been recalled? How many of these are on the shelf already? Is this the oldest one in stock and therefore the one to sell first? How can I recycle this? How can I dispose of this safely?

In this document, we'll return to that fundamental aim time and again. In the modern world, being able to find this information is not novel or unusual. It's simply expected.

That's the *why* of GS1 Digital Link.

What about the how?

How can we make the item – the product, the medical device, the shipment, the location, the asset, the *thing* – the starting point on a journey to finding whatever it is we want to know about it?

2 Connecting barcodes online

If GS1 identifiers in barcodes and RFID tags are to be the starting point for finding information then one way or another, they need to connect to the Internet. There are several ways this could be done.

1. The EPCglobal/GS1 Object Name Service [ONS] uses the Internet's Domain Name System infrastructure while implementing the Dynamic Delegation Discovery System algorithm to locate authoritative metadata and services associated with a given GS1 Identification Key. First standardised in 2004, the standard and services that implement it, while technically sound, require specialist applications. Implementations of the standard are very few in number.
2. GS1 could create and promote its own app. This could be offered as 'the universal barcode reader' that would link to, for example, a service like Verified by GS1. In this sense, it would be continuing the traditional method of starting with a database and looking up a particular item, rather than the desired outcome of starting with the item and being led from there to the data. Furthermore, it would put GS1 in the position of competing with all the other apps already in the market, which is not a goal of GS1 (which is a global, not-for-profit, neutral Federation).
3. GS1 could define its own URI scheme, in the same way that, for example, ISBN defined its own URN namespace in 2001 [RFC 3187]. While this eliminates any possible ambiguity concerning a GS1 identifier such as the 8, 12, 13 or 14 digits of a GTIN, it does not obviate the need for specialist software to interpret and act on it.

4. GS1 could connect its identifiers specifically to the World Wide Web. That is, the world of URLs, hyperlinks and APIs, leveraging existing standards, practices and infrastructure wherever possible.

The desire to meet society's expectation of being able to ask for any information about anything points firmly to the fourth option. After all, the Web is the world's database.

In order to describe how GS1 Digital Link uses the Web, it is helpful to break up the standard and the ecosystem it defines into its component parts. The five-part 'layer cake' model has proved useful in discussing this with multiple audiences. For each layer we will consider the specific problem, the available technologies that could solve it and, where more than one such technology exists, why a particular choice was made.

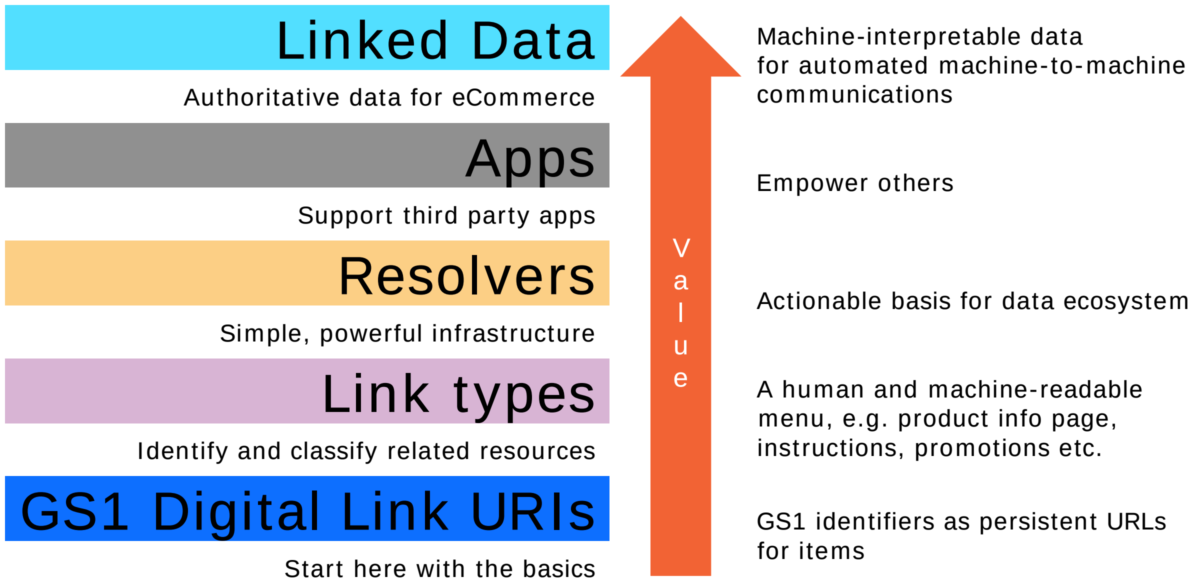

Linked Data
Authoritative data for eCommerce
Apps
Support third party apps
Resolvers
Simple, powerful infrastructure
Link types
Identify and classify related resources
GS1 Digital Link URIs
Start here with the basics
Value
Machine-interpretable data for automated machine-to-machine communications
Empower others
Actionable basis for data ecosystem
A human and machine-readable menu, e.g. product info page, instructions, promotions etc.
GS1 identifiers as persistent URLs for items

Figure 1 The GS1 Digital Link layer cake

3 Layer 1: The GS1 Digital Link URI

After nearly 50 years, the GS1 system of identifiers is very mature and widely implemented. In the world of barcodes and RFID tags, the group of technologies collectively known as AIDC – Automatic Identification and Capture – GS1 is a recognised authority. It is axiomatic that GS1 Digital Link cannot affect, for example, the structure of GTINs or SSCCs, or require any change to, say, the way dates and weights are represented.

However, those identifiers can already be expressed in multiple syntaxes, depending on the data carrier to be used. Figure 2 shows the full variety of syntaxes in which, for example, a GTIN and serial number for an item can be expressed. To explain this further, we'll look in more detail at one of the available syntaxes, the Element String syntax.

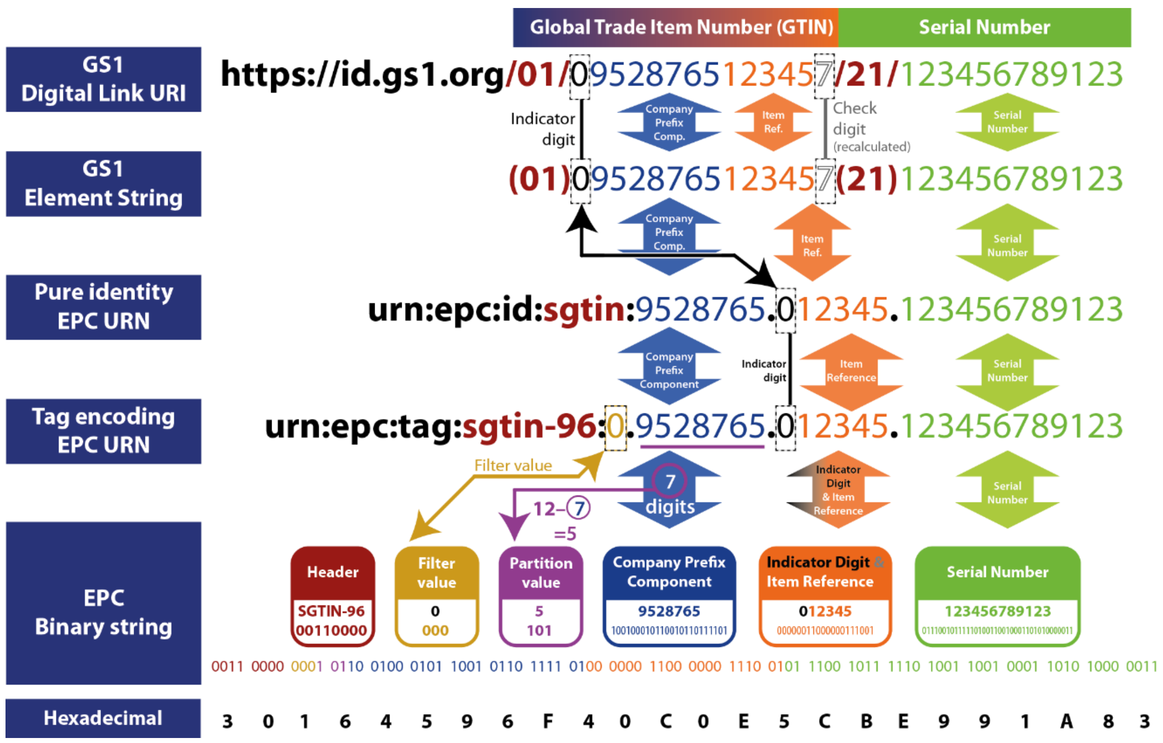


Figure 2 The different syntaxes in which GS1 identifiers can be encoded

3.1 Element String syntax

The GS1 DataMatrix shown here contains 4 pieces of data:

Label	AI	Value
GTIN	01	09506000134376
Expiry date	17	2022-12-25 (encoded as 221225)
Batch/lot	10	ABCDEF
Serial number	21	1234

Each piece of data has a text label, such as GTIN, and an *application identifier* such as '01'. GS1 maintains a full list of application identifiers which are all numeric [AI Table]. This is because barcodes have a limited capacity to carry data and using numbers is the most efficient method available. Documents such as the GS1 General Specifications [GenSpecs] define exactly how multiple data elements should be concatenated into a string and then encoded in a barcode or other symbol. This is known as *Element String syntax.* Again, the driving force here is efficiency. The 4 data elements in the example are encoded in a string as shown immediately below. The AIs are shown in parentheses here for legibility but the parentheses themselves are not included in the string encoded in the barcode.

<FNC1>(01)09506000134376(17)221225(10)ABCDEF<FNC1>(21)1234

In brief, the structure is:

1. The 'Function 1 character' indicating that what follows is one or more GS1 elements.
2. If the primary identifier is of fixed length, as in the case of a GTIN (01), it is placed first..
3. As this specific primary identifier is of fixed length, there is no need to mark the end of the GTIN and the AI of the next element.
4. The expiry date (17) is also of fixed length so, again, there's no need to include any kind of separator before the next element.
5. The batch/lot number is of variable length and so a separator *is* required before the final element, the serial number (21), which is also of variable length. The Function 1 character can be used for this or ASCII character 29.

Neither Function 1 nor ASCII character 29 are printable, they exist in barcodes as barcode characters with no direct textual equivalent. Using Element String syntax like this, our data carrier contains four separate application identifiers and their values with just two extra characters needed.

3.2 No online lookup necessary

Although widely implemented in systems all over the world, Element String syntax clearly has no meaning in the online world. You can't usefully put (01)09506000134376(17)221225(10)ABCDEF<FNC1>(21)1234 in a Web browser's address bar, for example. So maybe one option is to encode a URL that a consumer can scan to get information about the item and a business partner can query to find the GS1 identifiers? In other words, the URl could be something without any meaning like https://example.com/iwv15 that when looked up online returned either a page of information about the item or, if using some sort of app for industry, an element string like (01)09506000134376(17)221225(10)ABCDEF<FNC1>(21)1234.

Doing this would create multiple dependencies and so would be very unsafe for day to day business practices. Firstly the local internet connection would have to work every day without fail. Secondly, the look up service would have to work without fail. Add in the potential security problems and it's quickly apparent that this is not an option.

Maybe the need for an online lookup could be eliminated by simply using that opaque URL https://example.com/iwv15 as the identifier string? For that to work, every scanner and system to which scanners are connected would need to be re-programmed, and it would be impossible to distinguish between a URL intended as identifiers and any other URL that may appear on an item. It *could* work if every URL intended for use as an identifier used a specific domain name but that again creates a single point of failure and is not business-friendly. See *Decentralised control and the network effect* below for more on this topic.

What is needed is a way to encode GS1 identifiers in a Web address such that they can be extracted as easily as they are using Element String syntax, *without* needing any online lookup.

The only way to do that is to define and use structured URLs.

3.3 URL History

URLs have a hierarchy because they access information on computers. You probably have files on your computer organised into top level topics, with sub divisions beneath that and more sub divisions below that. Almost all computer operating systems will write out the 'path' to an individual file as

/top/sub-division/sub-sub-division/file.ext

Windows uses the backslash character \ instead of the forward slash / but the principle is exactly the same [Windows]. In the earliest days of networked computers, the way to access another computer on the network would be to begin the path with a double slash, followed by the remote computer name, thus:

//{computerName}/top/sub-division/sub-sub-division/file.ext

Move forward to the creation of HTTP, the protocol for accessing those files over the Internet, and you get the familiar structure [RFC1738]

http://{computerName}/top/sub-division/sub-sub-division/file.ext

Further details are available in an article by Zack Bloom [Bloom] who suggests that the history of the hierarchical structure of paths in URLs actually goes back to a 2 hour conversation with Albert Einstein.

3.4 Application with GS1 Digital link

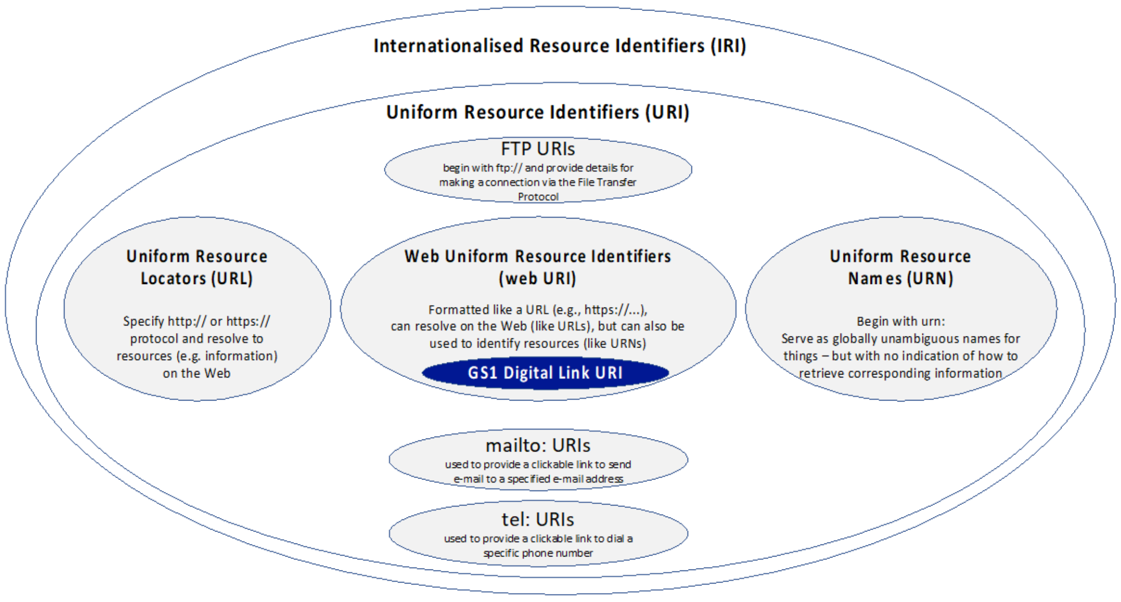

Internationalised Resource Identifiers (IRI)
Uniform Resource Identifiers (URI)
FTP URIs
begin with ftp:// and provide details for making a connection via the File Transfer Protocol
Uniform Resource Locators (URL)
Specify http:// or https:// protocol and resolve to resources (e.g. information) on the Web
Web Uniform Resource Identifiers (web URI)
Formatted like a URL (e.g., https://...), can resolve on the Web (like URLs), but can also be used to identify resources (like URNs)
GS1 Digital Link URI
Uniform Resource Names (URN)
Begin with urn:
Serve as globally unambiguous names for things – but with no indication of how to retrieve corresponding information
mailto: URIs
used to provide a clickable link to send e-mail to a specified e-mail address
tel: URIs
used to provide a clickable link to dial a specific phone number

Figure 3 Relationship between URIs, URLs, URNs and more

Referring back to our example above, GS1 Digital Link defines how to write those same four identifiers in a Web URI:

https://example.com/01/09506000134376/10/ABCDEF/21/1234?17=251225

Where does this structure come from?

GS1 application identifiers [AI Table] fall into two broad categories:

- *item* identifiers, such as GTINs to identify trade items, GLNs for locations and companies, SSCCs for shipments, and more;
- *attribute* identifiers, such as measured weight, expiry date, ship to address etc.

For several identifiers, you can add more specific identifiers. The GS1 Digital Link standard talks about primary keys and key qualifiers. In the example given, the GTIN is the primary key and that is refined by the batch/lot which comes second. In other words, it matters which order things are in. In contrast, it doesn't matter at all which order one provides attribute data like expiry date and measured weight.

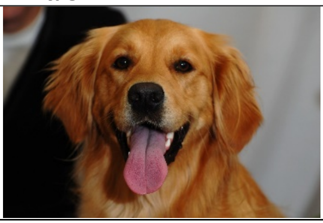
This distinction is very familiar in information science: they're classes, sub classes and properties. Figure 4 provides an everyday illustration of this. Mammals can be divided into placental mammals and marsupials. Within that classification we can place whales, dogs and koalas. So we might write:

mammal/placental-mammal/whale

mammal/placental-mammal/dog

mammal/marsupial/koala

It's this idea of classification and sub-classification that gives us the general structure and order of a GS1 Digital Link URI. The usefulness of this becomes more obvious in *Layer 5: Linked Data*.

Mammals		
Placental mammals		Marsupials
		

Whales by *Isaac Kohane* *cc-by* *Dog* by *David Locke* *cc-by* *Koala* by *Taz* *cc-by*

Figure 4 Classes and sub classes. Whales and dogs are in the subclass placental mammals, koalas are in the subclass marsupials. All three animals are mammals

Serialised items are known as instances of a class. For example, you are an individual instance of the class of human which is a sub class of mammal. This fact is not affected by your name or age.

So, to return to our example, we have an instance, 1234, of the class of products with batch/lot number ABCDEF which is a sub class of the product with GTIN 9506000134352. The fact that its expiry date is 25 December 2022 does not change that identification. Expiry dates, like prices, ship to addresses and measured weights are properties (attributes) of the item, not identifiers.

This structure alone meets many of the goals of GS1 Digital Link.

1. It is a formal syntax for providing one or more GS1 identifiers in a string in a way that can be extracted *without* an online lookup.
2. It is a URL, meaning it can be an entry point to the Web.
3. Software that can make use of the URL – notably Web browsers – is massively implemented around the world in smartphones, laptops and other devices.

The GS1 Digital Link standard has several pages of detail on this, expressed using ABNF [RFC 5234]. Naturally it talks specifically about how to encode GS1 identifiers. It would be perfectly possible to define a more general standard for any identification scheme that followed the same underlying pattern.

3.5 Why not use Element String syntax with a domain name 'stub'?

Given the preceding discussion, one might ask why GS1 Digital Link doesn't retain the Element String syntax and simply append it to a domain name. Something like:

https://example.com/01095060001343761722122510ABCDEF/211234

That has all the same information as the previous versions with only a single / character required to separate the variable length batch/lot number and the item serial number. Although, yes this would be possible, it would be much more awkward to use with Linked Data (layer 5 below). It would require broad dissemination among developers of parsing rules and details of the GS1 system that are irrelevant in the majority of cases. Something like it *is* used, however, in the compression algorithm discussed in the appendix.

4 Layer 2: Link types

The definition of the GS1 Digital Link URI – a syntax for GS1 identifiers in the form of a URL – starts us on the journey from the identified thing to information about it (rather than starting with, and being limited to, a big database in which we can look the thing up). But recall that **the fundamental aim of GS1 Digital Link is to enable anyone to find answers to their questions about the thing in front of them.** Where did this come from? What's in it? Where can I buy spares? How do I use it? Has it been recalled? How many of these are on the shelf already? Is this the oldest one in stock and therefore the one to sell first? How can I recycle this? How can I dispose of it safely?

For any given item, it's possible that answers to those questions exist, but they may not exist in the same place and under the same organisation's control. For example, information for patients about a given pharmaceutical might be in one repository, pharmacological information in another and 'how to use' instructions in yet another. This emphasises the point about *starting* with the item and going from there.

There really is only one way to achieve this.

4.1 History

Figure 5 is taken from Tim Berners-Lee's original proposal for the Web, dated 12 March 1989 [IMP].

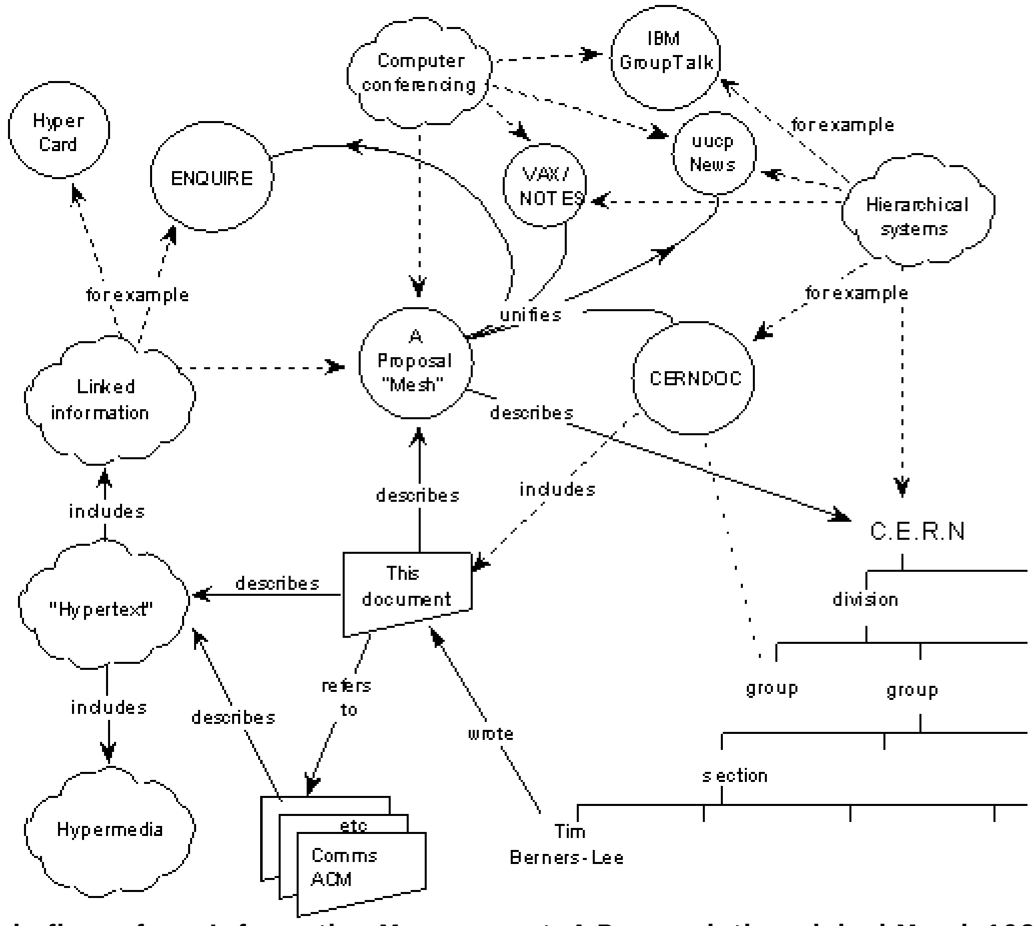


Figure 5 Main figure from Information Management: A Proposal, the original March 1989 proposal for what became the World Wide Web

Notice that each entity — be it a person, a concept, a document or whatever — is linked to other things via lines that are annotated with labels like 'describes', 'unifies', 'refers to' and so on. Those are known as relationships. The idea of an Entity-Relationship model goes back to well before the Web but the model was first formalised by Peter Chen in 1976 [ER].

Since the publication of the Atom Syndication standard in 2005 [RFC 4287], IANA has maintained a registry of well-known link relation types [IANA-REL] but we can go back further than that. Link relation types are discussed by name in HTML 4.0 [HTML4.0] but they're used in HTML as values of the rel and rev attributes which go back to [HTML3] from March 1995 (see Dave Raggett's History of HTML [DSR]). The link relation types in the IANA registry are not suited to the GS1 use cases which are themselves not suited to be added to the registry for more general use. The Web Linking standard [RFC 5988] details how new link relation types may be defined and used (essentially by giving each one its own URL) and this is what the GS1 Digital Link standard does. New 'link types' are managed as part of the GS1 Web Vocabulary [WebVoc].

4.2 Application within GS1 Digital Link

GS1 Digital Link's 'link type' is simply a shorthand for link relation type. It is the label for the relationships between related entities, such as between a product and its information page, a medicine and its information leaflet or a company and its entry in a company register.

The idea of being able to programmatically discover a set of typed links, and to act on one or more of them, is, again, not a new concept developed for GS1 Digital Link. As well as the Atom Syndication and Web Linking standards referred to previously, Hypermedia as the Engine of Application State [HATEOAS] explores this idea and is based on the Hypertext Application Language [HAL] that dates back to at least June 2012. That in turn is based on Roy Fielding's original work on Representational State Transfer [REST] from 2000. Roy Fielding is one of the leading architects of the Web, responsible for, among other things, the original definition of a relative URL [RFC1808] in 1995, and working with Tim Berners-Lee and Larry Masinter on the URI specification in 1998 [RFC2396].

By using defined link relation types, links between items and information about them are machine-discoverable. The machine in question – the one that makes links of different types discoverable and actionable – is called a "resolver."

5 Layer 3: Resolvers

As repeated already: the fundamental aim of GS1 Digital Link is to enable anyone to find answers to their questions about the thing in front of them. To do that, it is likely to be necessary to consult multiple resources. Furthermore, a user will want those resources to be directly relevant to them. That implies that GS1 Digital Link cannot just think in terms of URLs and link types, but must also consider the context of the request. What language does the user speak? For which country do they need information? The number of possible links to related information for a single item can quickly multiply.

How can this complexity be handled?

Hand over to a search engine

One way, that is almost a non-answer, is simply to look up the identifier in a search engine. That will return a set of links that can be followed by a user and that may or may not lead to accurate or even relevant information.

Maintain a single webpage

Another way to provide an answer directly is to show a webpage with clickable links to those sources of information, in effect creating a mini-website for each item. This can work well when presenting information to consumers about products and is the approach taken by, for example, *SmartLabel*®. Like any Web page, a product information page can take account of things like the user's language and cookies set on previous visits so that the experience can be personalised. However, it is less appropriate for machine-machine exchanges that, ideally, can take place without any need for human intervention.

Return a set of links as structured data

A similar approach is that looking up a GS1 Digital Link URI returns the set of all available links as a dataset and leaves it entirely to the client application to decide what to do with them. If each link is described with consistent metadata, links of a desired type can be discovered and followed automatically in machine-to-machine communications or displayed to end users as above.

Both of the latter approaches have significant merit and both are supported in GS1 Digital Link. But we can go further. GS1's core competence is as licensor of identifiers for products and related items in the supply chain. With that starting point, it's fortunate that the solution to the problem of how to relate a given identifier to multiple sources of information about the item it identifies, and to take account of the context in which a given query is made, has been solved more than once already.

5.1 History

Citations are an important aspect of scholoarly publishing. This is not surprising since the motivation is strong: the number of scholarly citations accrued is a significant metric by which academic success is measured. As a result, there is a rich collection of technologies, standards and practices around linking from one publication to others.

The OpenURL Framework for Context-Sensitive Services, a NISO standard from 2005 [openURL] that was based on the success of work from the previous decade, provides a mechanism through which links to related sources of information can be obtained. The links are compiled dynamically based on the user's context. The example given is that a service should only offer links to paywalled journals if the user is a member of an institution that has a subscription to the journal. Following the NISO standard's references quickly leads to a paper from the 1995 World Wide Web Conference that describes an early system called a Distributed Link Service [DLS]. Again, this has the familiar components of a set of links that is compiled dynamically to meet the user's context. Further work in 2001 explicitly expanded the scope of OpenURL "*beyond the Web-based scholarly information environment into the realm of references to published works in general (CDs, CD-ROMs, audio files, videos, etc.), objects (cities, cars, people, companies, etc.) and abstract concepts referenced in web-pages*" [vdSompel]. It also introduced the concept of a default link and the provision of alternative pathways to other services described by their service type that "*describes the nature of the resolution requested*". This maps directly to the concept of a link relation type (link type) used in GS1 Digital Link.

Although these efforts added some useful detail, the core ideas were already well-established.

One of the most widely used 'resolvable identifiers' online today, particularly in the field of academic publishing, is the Digital Object Identifier (DOI), which became an ISO standard in 2012 [ISO-DOI] having been introduced more than a decade earlier. The definition of 'resolution' offered by the International DOI Foundation is: [DOI-HB]

Resolution is the process in which an identifier is the input — a request — to a network service to receive in return a specific output of one or more pieces of current information (state data) related to the identified entity: e.g., a location (URL). Multiple resolution, that is made possible by the Handle System used as the DOI resolution component, is the return as output of several pieces of current information related to a DOI-identified entity — specifically at least one URL plus defined data structures allowing management.

Bob Kahn in Geneva, May 2013

Credit: Вени Марковски | Veni Markovski, *CC BY-SA 3.0*, via Wikimedia Commons

When DOIs were introduced in 2000, one of the first resolver services it used was the Handle System which itself first went live in autumn 1994, having been developed by one of the co-inventors of the Internet, *Bob Khan* [HDL]. Back even further - the concept of resolving an identifier online goes back at least to the introduction of the Domain Name System in 1985 [DNS].

The term 'state data' used in the DOI Handbook quoted above refers to the fact that when a client, such as a browser or mobile app, looks up a URL on the Web, the request is always made in context, and context can have a big impact on the answer given. The single question "what can you tell me about this tomato?" will have different answers on a farm, in a store or in a kitchen. The question “what’s special about this toy?” will have a different answer if the questioner is a child or an adult. If you ask a visitor to your home at 10 in the morning if they’d like a drink, you and they will likely be thinking of either tea or coffee. Ask the same visitor the same question at 6pm and you might both be thinking wine or gin & tonic.

In the case of DOI resolution and openURL, ‘State data’ or ‘context’ is recognised as being crucial when providing meaningful results to the end user. Other work from the early days of the Web captures and uses the context provided as part of the client’s HTTP request, the so called HTTP Request Headers. There are several online services that *show you what your browser sends* to every website you visit. One of the pieces of information is the ‘user agent string’ that identifies the type of client and the device’s operating system. Every website you visit knows whether you’re using Firefox, Edge, Chrome, Safari, Brave, Opera, Vivaldi etc. The user’s language preferences and any cookies are also part of that request. Servers typically use that data to tailor their response. The W3C Recommendation in this area, Composite Capability/Preference Profiles [CC/PP] from 2004, built on earlier User Agent Profile work at the WAP Forum from 1999 [UAProf].

5.2 Application within GS1 Digital Link

A GS1 Digital Link resolver has a great deal in common with its predecessors. It takes one or more identifiers and ‘state data’ as input and *resolves* to information about the identified object. As with the Handle system, it can resolve to multiple sources of data. The term ‘state data’ is not used in the GS1 Digital Link standard but, as in the Handle system, it simply means ‘all the information about the request that isn’t the actual identifier.’

In this way, a resolver can make use of features of the Web that people use every day but that aren't immediately obvious to those users. You don’t need to know how the Web works in order to use it any more than you need to know how to fly a plane to catch a flight. One of its most powerful features is a Web server’s ability to show different things to different people at the same time. For example:

- when you login to social media, you see information from your social network – which is different from everyone else’s.
- If someone shares their calendar with you, you might simply see when they’re busy, whereas *they* see all the details of their appointments.
- Take a look at the *GS1 contact page*. You should see contact details for your local GS1 Member Organisation – which will be different depending in your location.
- Visit a multilingual website and you should automatically see the website in your language.

These are all examples of the end user seeing something different depending on factors such as:

- who they are;
- whether they're authorised to access content or not;
- where they are;
- what language they speak;
- the time of the request.

Some of these factors are part of the original HTTP specification [RFC2068], notably language and authorisation.

What about location?

There are two methods of detecting a user's location. One is based on their IP address, the other via an API to whatever system the user's device has for detecting its location. The latter — the Geolocation API [GeoLoc] — is only accessible by an application running on a device and resolvers have no access to that (at least, it’s not part of the GS1 Digital Link specification). In contrast, the IP address of the user is always known to a resolver as that is the point on the internet to which the response must be sent. But … the allocation of IP address blocks is administered entirely separately from the HTTP specifications and can change. Although there are any number of services that allow you to look up the approximate geolocation of a given IP address, to make this part of HTTP would require a commonality of approach that is not deemed necessary or practical.

As a consequence, GS1 Digital Link does not include a specific field for location that would require resolvers to act accordingly. However it does include the optional context variable. Resolver implementations are free to use the context variable for whatever they want but one possibility is to use it for locations, as in "give me the product information page relevant to Jordan" or "give me the product information page relevant to Singapore".

Directions *by* Joohander
Some rights reserved

5.3 Redirection

GS1 Digital Link resolvers are envisaged as way finders, not providers of information. This is done through redirection. That is, redirecting a request from one server to another. This has always been part of the Web — it's part of the original HTTP specification from January 1997 [RFC2068]. But, like so many things in computing, that basic idea goes back well before then. For example, even on the pre-Web Internet, the Internet Control Messaging Protocol [RFC777] from 1981 includes redirection - 8 years before the Web was conceived.

5.4 The linkType parameter

There is one feature of a GS1 Digital Link resolver that is not part of any existing standard: the linkType parameter. However, its definition is an obvious step.

Again we come back to the motivation behind the standard: **we want to be able to find the answer to any question about the thing in front of us**. Any solution to this must, of course, meet the broader needs of the businesses that GS1 exists to serve, in particular the need to communicate with the consumers, patients and business partners. That high level description of 'the need to communicate' might take many different forms in the real world and GS1 Digital Link needs to be able to handle as many as possible. For example:

1. A consumer uses their phone to scan a QR code on a product and is presented with a page of information about that product. There is no need for the consumer to use a particular app to do this, just the phone's camera.
2. A consumer with a specific dietary need uses an app designed to find allergen information about a product. Using that app, they scan the same QR code on a product and are shown the allergy information directly.
3. A clinician and a patient scan the same barcode. One sees pharmacological information aimed at professionals, the other sees general information for patients.
4. A general purpose app simply wants to present users with whatever information is available.

Supporting these different use cases requires that client applications can have some control over the resolver's response. This is achieved by passing information to the resolver in the query string.

We are now so familiar with the idea of the query string in a URL that it's hard to imagine that this wasn't the norm when the Web was first introduced in 1991, although it followed shortly afterwards. The now very familiar ?param1=value1&param2=value2 syntax was developed by NCSA, University of Illinois, as the Common Gateway Interface (CGI) in 1993 [CGI] and formalised in 2004 [RFC3875].

Alternatives to the query string have been suggested. Tim Berners-Lee wrote about Matrix URIs in December 1996 [Matrix]. In that system, parameters appear within path segments and are separated by semicolons. Here's a key section:

"The analogy with procedure call holds still when looking at combined forms: The hierarchical part of the URL is [parsed] first, and then the semi-colon separated qualifiers are [parsed] as indicating positions in some matrix. As an example let's imagine the URL of an automatically generated map in which the parameters for latitude, longitude and scale are given separately. Each may be named, and each if omitted may take a default. So, for example,

//moremaps.com/map/color;lat=50;long=20;scale=32000

might be the URL of an automatically generated map."

However, matrix parameters did not take off. In 2020 their use was considered in the development of the W3C Decentralized Identifier standard but in the end they were not used [DID159].

In the context of GS1 Digital Link, a resolver has one or more links associated with an identified item. Each link is labelled with a link relation type. It is therefore logical that in order to control which link is followed when querying the resolver for a specific type of information about an identified item, a client application should pass the desired type of link in the query string. Hence the linkType parameter.

There is one further parameter that a conformant resolver may use: the context parameter. The GS1 Digital Link standard does not specify what 'context' means. It is simply offered as a further method of differentiating one link from another in the query string which is more easily manipulated by client applications than the HTTP Request Headers.

In terms of the history of ideas, note the example used in Tim Berners-Lee's document about Matrix URIs from 1996 [Matrix] - it's a URL with a query about a location in it, and that's one use that GS1's own resolver service supports via the context parameter. That is, a client can ask for a link to a page tailored for one country rather than another. This is independent of language.

5.5 Decentralised control and the network effect

A final consideration when defining how a GS1 Digital Link conformant resolver should operate is whether there should be one single online service or whether the resolution functions should be distributed.

A single, central resolver or resolver-like service has a number of advantages:

1. There is no doubt where an application should seek information about items identified using the GS1 system.
2. GS1 can, on behalf of its Member Organisations and their Member Companies, effectively ensure that all data discovered through a GS1 identifier leads to brand authorised data plus any other policies it may wish to adopt.

This is the approach being taken by colleagues at GS1 China with their 2dcode.org service.

The alternative is to take decentralised approach. This has its own advantages:

1. Anyone can run a GS1 conformant resolver. This includes brand owners and solution providers, as well as GS1 MOs, who are free to operate the service under whatever policy they wish. This has the business-friendly effect of leaving control in the hands of the service operator.
2. From an engineering point of view, the existence of multiple resolvers eliminates any single point of failure. Furthermore, if a resolver cannot answer a given query, it can pass the request on to another service.
3. The decentralised approach offers the benefits of *the network effect*. This is commonly explained in relation to telephones. The value of the first telephone was zero until the second telephone was connected to the network. Each of those two telephones increased in value by the addition of the third and so on. Again, this reflects the business-led use cases behind GS1 Digital Link.

6 Layer 4: Apps

Although apps are part of the story, there is no 'standard GS1 Digital Link app' and the standard itself does not define an app. This is because the potential applications of the standard are many and varied. The needs of an apparel manufacturer will differ significantly from a clinical practitioner's needs. A logistics operator will want to interact with an identified object for very different reasons than, say, a consumer of beauty care products.

So rather than offer a standardised app that would inevitably be compared with the many existing apps that include barcode scanners, GS1 took the decision to make *open source code* available under a permissive licence that implements the core of the standard. This is so that developers can create whatever app they choose and include GS1 Digital Link-based functionality as easily as possible.

This approach was inspired in particular by the development and success of the Web itself. The concepts were first formalised by Tim Berners-Lee in March 1989 [IMP]. On 6 August 1991, the first publicly available website was published at *http://info.cern.ch*. However, it is often argued that the key date in the history of the Web is 30 April 1993 – the day CERN put all the code for the first server and the first browser in the public domain [BW].

7 Layer 5: Linked Data

Again: the fundamental aim of GS1 Digital Link is to enable anyone to find answers to their questions about the thing in front of them. To achieve this, GS1 Digital Link defines how to connect identified items to the Web (layer 1). Having done that, it uses the well-established concepts of typed links (layer 2) and resolvers (layer 3) to lead from the item to wherever the answer to the user's question might be, perhaps using free, open source software (layer 4).

This creates a machine-readable entity-relationship network – a simple knowledge graph – that can improve business processes, enhance search engine visibility and more, all based on GS1 identifiers.

It does this because GS1 Digital Link has a number of features:

1. The use of GS1 identifiers expressed as HTTP URIs.
2. Links being labelled with the type of information available from that link.
3. Services that make those links discoverable and actionable.

These features map directly to those of Linked Data. Formally defined in 2006, Linked Data has its origins in the late 1990s as 'PICS-Next generation' [PICS-NG], later named RDF, and the Semantic Web as described in 2001 [SciAm].

More fundamentally, these technologies and ideas have their roots in that original entity-relationship diagram for the Web (Figure 5) which, in turn, is an expression of the Memex machine conceived by Vannevar Bush in 1945 [Memex]. As an aside, *Vannevar Bush* was born 11 March 1890 and died on 28 June 1974, just 2 days after that first checkout beep [Marsh].

By following Linked Data principles, GS1 Digital Link maximises the value of the identification system. It makes GTINs, SSCCs, GLNs, GIAIs etc. as functional and powerful as possible. It's what allows that journey *from* the identified thing, *to* the information about it, wherever that might be. The alternative is Google, Bing, Yandex, Baidu etc.

7.1 History

Tim Berners-Lee defined Linked Data in four bullet points [LinkedData]:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
- Include links to other URIs so that they can discover more things.

Those bullet points refer to RDF which first appeared in 1997 [RDF-1] and today is a technology defined in a set of W3C standards [RDF]. The terms RDF, Linked Data and Semantic Web are often used informally as synonyms and there is a large body of literature about the topic. Of particular relevance in the current discussion are the documents that talk about URI structure and how this can be used to reflect classes, subclasses and instances.

The best known document on this topic is Cool URIs for the Semantic Web [Cool-SW], a W3C Note from December 2008 that distils ideas that had been around for many previous years. Notice the first example:

http://www.example.com/

the homepage of Example Inc.

http://www.example.com/people/alice

the homepage of Alice

http://www.example.com/people/bob

the homepage of Bob

And that's in a section talking about 'the old Web' The URIs are structured so that, from left to right, you go from the organisation to its people to a specific person. A more formal approach to hierarchical URIs was set out by Leigh Dodds and Ian Davies in 2012 [LDP]. All that work was inspired in part at least by Tim Berners-Lee's note in 1998 called Cool URIs Don't Change [Cool-URI]. That document was about URI persistence but the key to that persistence is URI structure.

A final topic that was tackled by the original developers of Linked Data was how a URI can identify a physical object. If a URI identifies an electronic document, or *Information Resource* in the jargon, then there's no problem looking it up on the Web and receiving that item. But what if it identifies a building, or a mountain? Neither will pop out of a computer screen. This issue has been at the heart of so many arguments over so many years it even has its own name: HTTP-Range 14 [HR14].

7.2 Application within GS1 Digital Link

Let's return to the example we used when discussing the GS1 Digital Link URI syntax:

https://example.com/01/09506000134376/10/ABCDEF/21/1234

As it's an HTTP URI, you can look the item up on the World Wide Web. It has structure, showing increasingly granular identification from left to right, expressing the class – subclass – instance relationship. If it takes you to a resolver, you'll find linksets expressed in standardised formats and links to other sources of data. If the data that is linked to is itself in the form of Linked Data, such as a block of JSON-LD embedded in a product's Web page, then the GS1 identifier is a node in the network of Linked Data. If that data is open, it becomes part of the Linked Open Data Cloud [LOD].

It is many years since a meaningful visualisation of the Linked Open Data Cloud could be shown in a single page. It's easier to compare its initial state in 2007, Figure 6, with the huge interlinked datasets it includes today, Figure 7.

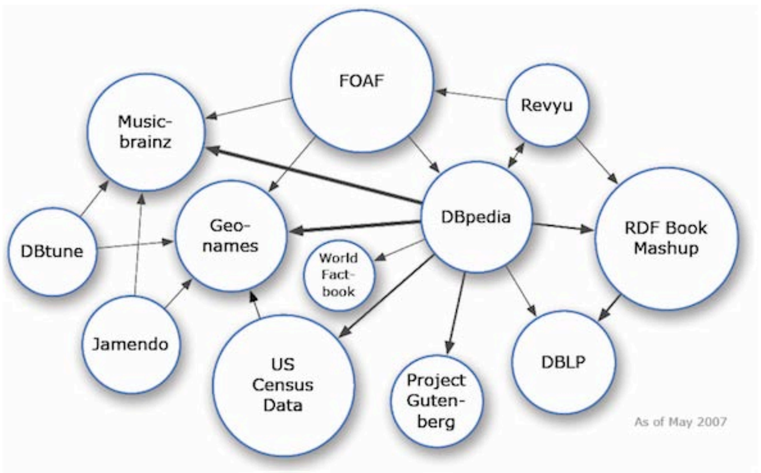


Figure 6 The Linked Open Data Cloud diagram from May 2007

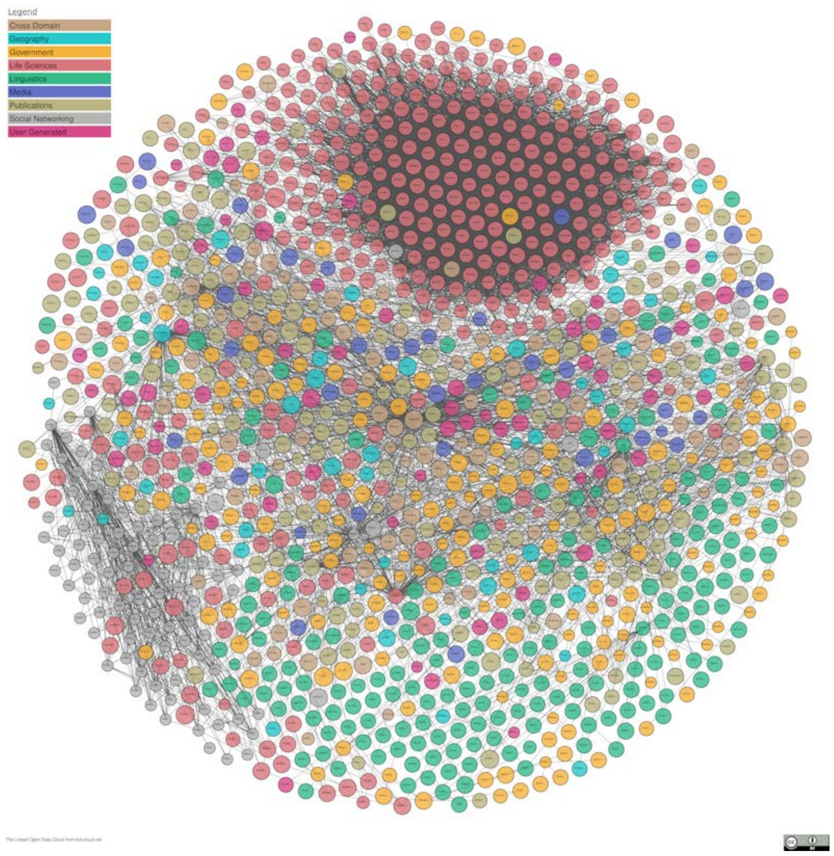


Figure 7 The Linked Open Data Cloud, May 2021

Following Linked Data principles, a GS1 conformant resolver supports queries at any level in the hierarchy within a GS1 Digital Link URI. A query for all available links about item 1234 will return links associated with that specific item, but it will also return links associated with the sub class ABCDEF of which it is an instance and links associated with the class identified by the GTIN of which the batch/lot is itself a subclass.

7.3 Broader implications

Working with Linked Data concepts increases the power of the GS1 identification system beyond the confines of the GS1 Digital Link standard.

If an item has a GTIN then, by definition, it is a trade item, also known as a product. It will be offered for sale, there will be associated stock levels, a price, imagery, condition, ratings and more. If it is in the subclass of products that are food and beverages, there will also be ingredients and allergen statements. If it's apparel, there won't be ingredients and allergen statements but there will be size and colour. Likewise, if an item is identified by a Global Location Number, using an AI of 414, it is a location and will have geospatial coordinates and perhaps things like dock gate entry requirements. These are the *semantics* of the GS1 system and can be expressed using terms defined in the GS1 Web Vocabulary [WebVoc].

For example , we can express the semantics of our running example in natural language thus:

There is a class of product with GTIN 09506000134376.

There is a subclass of product 09506000134376 that has a batch/lot number of ABCDEF.

An instance of the class ABCDEF has a serial number of 1234 and an expiry date of 2022-12-25

The details of this are set out in the GS1 Digital Link standard which makes full use of the relevant RDF standards [RDF]. What is important in the current discussion is that GS1 application identifiers have precise meanings that are machine-readable and can be interpreted by information systems *outside* the GS1 ecosystem. This is a necessary condition if we are to meet **the fundamental aim of enabling anyone to find answers to their questions about the thing in front of them.** The structure of a GS1 Digital Link URI encodes a lot of this meaning and, following the standard, the machine-readable interpretation can be derived just from, for example, https://example.com/01/09506000134376/10/ABCDEF/21/1234?17=251225.

One of the consequences of this is that it is recognised across GS1 that is it increasingly important to ensure that the meanings of GS1 Application Identifiers are expressed using terms defined within the GS1 Web vocabulary. This has already resulted in some work requests and additions of extra terms (classes, properties etc.) but some further work is required in this area in order to fully support the majority of GS1 Application Identifiers as far as possible.

Finally, on the issue of whether a GS1 Digital Link URI identifies the physical item itself, such as a product or location, or information about the item, the standard knowingly glosses over the point by stating:

“This is a very old and well known problem [URIDoc] [HR14] but it is one that presents few practical problems in the real world where the assumption that the identifier is for the physical object is usually safe.”

GS1 Digital Link version 1.1, section 9.5 [DL1.1]

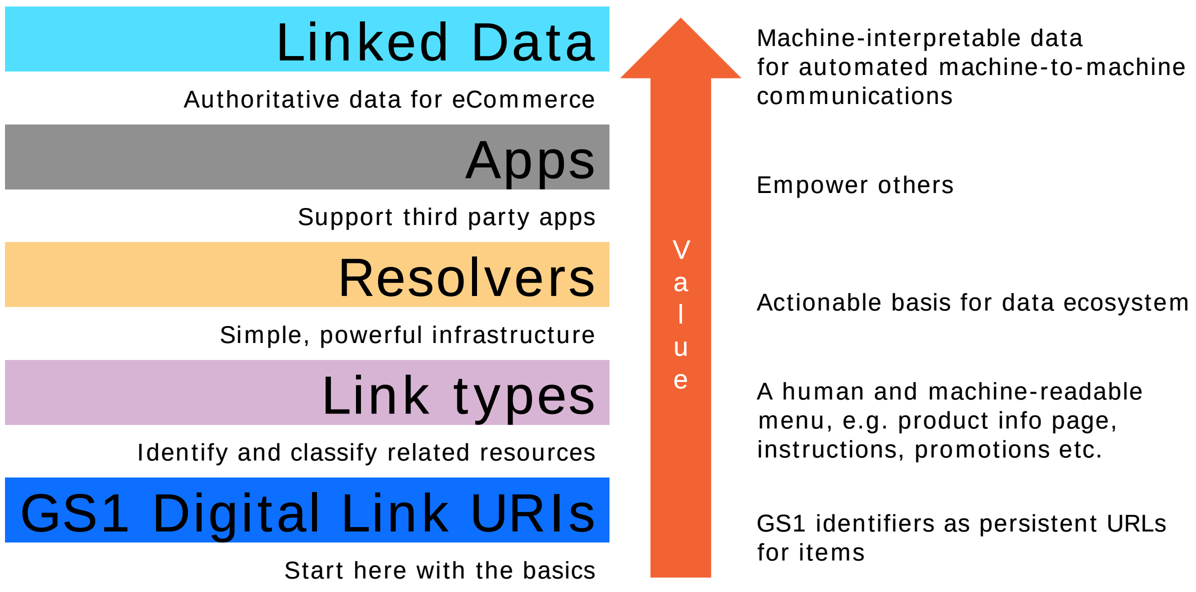
8 Conclusion

In many of today's societies, it is seen as normal that all facts can be found immediately. This is true for consumers, patients, business partners and more, all of whom care little about *how* this is possible. Industry came together at GS1 to work out the how.

The Object Name Service, ONS has been tried and found to be inadequate, however unfair that may be from a technological point of view. Existing apps and other systems in use today are, in effect, user interfaces for a single, usually proprietary, database.

The only realistic method of meeting the demand for facts is to harness the global database known as the World Wide Web. Its ubiquity, its technological maturity, and its plethora of existing standards makes it the natural choice.

Having made that choice, the rest is surprisingly obvious.


Linked Data
Authoritative data for eCommerce
Apps
Support third party apps
Resolvers
Simple, powerful infrastructure
Link types
Identify and classify related resources
GS1 Digital Link URIs
Start here with the basics
Value
Machine-interpretable data for automated machine-to-machine communications
Empower others
Actionable basis for data ecosystem
A human and machine-readable menu, e.g. product info page, instructions, promotions etc.
GS1 identifiers as persistent URLs for items

Layer 1:	GS1 identifiers need to be expressed in a syntax that is also a URL [RFC2396, 1998]
Layer 2:	The relationships between items typically identified using GS1 identifiers need to be defined [HTML3, 1995]
Layer 3:	Those relationships between identified items and relevant information need to be actionable. Resolver technologies are an obvious choice [HDL, 1994]. Maximise value for industry by harnessing the network effect.
Layer 4:	Define the protocols as broadly as possible, provide free open source software wherever possible [BW, 1993]
Layer 5:	Use Linked Data principles to create a simple knowledge graph based on GS1 identifiers [LinkedData, 2006]

A final word arising from section 1.1: The Sting, won *Best Picture* in the 1974 Oscars. Mount Afadjato is the *highest peak* in Ghana. Dogs *can see* blue and yellow. The *atomic number* of molybdenum is 42 and Australia's Mike Wendon won the *100m freestyle* in the 1968 Olympic Games. Any good *recipe for puttanesca* includes anchovies.

9 Changes since version 1.0

Small addition to section 5.1 to highlight OpenURL work and its antecedents.

Small changes to the text concerning the term 'state data' as used in DOI resolution.

10 References

[AI Table] GS1 Application Identifiers

[Bloom] The History of the URL The Cloudflare blog, 5 March 2020. See, in particular, the section on the origin of the path.

[BW] The Birth of the Web, CERN

[CC/PP] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. Graham Klyne, Franklin Reynolds, Chris Woodrow, Hidetaka Ohto, Johan Hjelm, Mark H. Butler, Luu Tran. W3C Recommendation 15 January 2004.

[CGI] The original documentation for CGI was originally published at http://hoohoo.ncsa.uiuc.edu/cgi/ but is no longer available from that site. The content can, however, be accessed using the Memento Web

[Cool-SW] Cool URIs for the Semantic Web, Leo Sauermann, Richard Cyganiak. W3C Note, 3 December 2008

[Cool-URI] Cool URIs don't change. Tim Berners-Lee, 1998.

[DID159] W3C Decentralized Identifier Working Group, DID Core Specification issue tracker. Issue 159 Remove matrix parameters from the DID specification. January 2020.

[DL1.1] GS1 Digital Link 1.1. Mark Harrison, Phil Archer et al. GS1, 19 February 2020 (PDF)

[DLS] The Distributed Link Service: A Tool for Publishers, Authors and Readers. Les Carr, David De Roure, Wendy Hall, Gary Hill. Proceedings of the 4th International World Wide Web Conference, December 1995. https://www.w3.org/Conferences/WWW4/Papers/178/

[DNS] Domain Name System, Wikipedia entry.

[DOI-HB] DOI® Handbook, section 3.1. DOI Foundation, last updated 16 August 2018

[DSR] A history of HTML, Dave Raggett. Published by Addison Wesley Longman, 1998.

[ER] The entity-relationship model—toward a unified view of data. Peter Pin-Shan Chen. Published in: ACM Transactions on Database Systems (TODS) - Special issue: papers from the international conference on very large data bases: September 22–24, 1975, Framingham, MA. Volume 1 Issue 1, March 1976, Pages 9-36

[FDD] https://en.wikipedia.org/wiki/History_of_the_floppy_disk History of the Floppy Disk. Wikipedia.

[GenSpecs] GS1 General Specifications. See https://www.gs1.org/standards/barcodes-epcrfid-id-keys/gs1-general-specifications for the latest version.

[GeoLoc] Geolocation API Specification 2nd Edition. Andrei Popescu. W3C Recommendation 8 November 2016.

[HAL] HAL - Hypertext Application Language. Mike Kelly. 13 June 2011, updated 18 September 2013.

[HATEOAS] HATEOAS on Wikipedia

[HDL] The Handle System went online autumn 1994 and was formally written up in A Framework for Distributed Digital Object Services, Robert Kahn, Robert Wilensky, CNRI, 13 May 1995

[Held, Marshall] Data compression : techniques and applications : hardware and software considerations. Gilbert Held and Thomas R. Marshall. 3rd Edition, Chichester, West Sussex, England ; New York : Wiley, c1991.

[HTML3] HyperText Markup Language Specification Version 3.0, Hypertext links. Dave Raggett. W3C/IETF draft, March 1995

[HTML4.0] HTML 4.0 Specification section 13.3.2: Links. Dave Raggett, Arnaud Le Hors, Ian Jacobs. W3C Working Draft 17 September 1997

[HR14] HttpRange14Webographyv. The chronology of a permathread, Sandro Hawke, W3C Wiki, 2003

[IANA-REL] Link Relations registry, IANA, created 2005-08-26. Specified by RFC8288

[IIP] Industrial Image Processing: Visual Quality Control in Manufacturing. Christian Demant, Bernd Streicher-Abel and Peter Waszkewitz. Springer 1999. See, in particular, chapter 5.

[IMP] Information Management: A Proposal, Tim Berners-Lee, 12 March 1989.

[ISO-DOI] ISO 26324:2012 Information and documentation — Digital object identifier system. ISO/TC 46/SC 9, May 2012

[LDP] Linked Data Patterns, A pattern catalogue for modelling, publishing, and consuming Linked Data. Leigh Dodds and Ian Davies. 31 May 2012 (see the section on hierarchical URIs)

[LinkedData] Linked Data, Tim Berners-Lee, 2006-07-27

[LOD] The Linked Open Data Cloud. Currently maintained by John P. McCrae for the Insight Centre for Data Analytics. The original was created by Richard Cyganiak and Anja Jentzsh in 2007.

[Marsh] Marsh holds place of honour in history of GS1 barcode. GS1, 2013. See also The First Barcode Scan in History and What It Tells Us About How Some Stories Make Us Care Jim Wildman, 9 November 2017 and The History of Barcodes by the History Guy on YouTube.

[Matrix] Matrix URIs, Matrix spaces and Semicolons. Tim Berners-Lee, 19 December 1996

[Memex] As We May Think. Vannevar Bush. The Atlantic Magazine, July 1945.

[Lynch] Data compression: techniques and applications. Thomas J. Lynch. New York : Van Nostrand Reinhold, c1985.

[ONS] Object Name Service GS1 Ratified Standard 2004 – 2013. See https://www.gs1.org/standards/epcis/epcis-ons/2-0-1

[openURL] The OpenURL Framework for Context-Sensitive Services. Herbert Van de Sompel, Patrick Hochstenbach, Oren Beit-Arie. ANSI/NISO Z39.88-2004 (R2010), National Information Standards Organization, 15 April 2005. The standard refers to previous workdating back to 1999, for example https://doi.org/10.1045/april99-van_de_sompel-pt1

[PICS-NG] PICS-NG Metadata Model and Label Syntax. Ora Lassila, W3C Note 14 May 1997

[RDF] RDF 1.1 Primer, Guus Schreiber, Yves Raimond, W3C Working Group Note, 24 June 2014

[RDF-1] Resource Description Framework (RDF) Model and Syntax, Ora Lassila, Ralph Swick, W3C Working Draft, 1 August 1997

[REST]	Representational State Transfer (REST) from Architectural Styles and the Design of Network-based Software Architectures. Roy T Fielding's PhD. thesis from 2000
[RFC20]	ASCII format for Network Interchangev. V. Cerf. IETF 16 October 1969.
[RFC 765]	Specification Of Internet Transmission Control Program, Vinton Cerf, Carl Sunshine, IETF December 1974.
[RFC777]	Internet Control Message Protocol. J. Postel. IETF, April 1981.
[RFC1738]	Uniform Resource Locators (URL). T Berners-Lee, L Masinter, M McCahill, IETF December 1994. See in particular section 3.3.
[RFC1808]	Relative Uniform Resource Locators, R. Fielding. June 1995
[RFC2068]	Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee. IETF January 1997. See sections on language, authorisation, content negotiation and redirection.
[RFC2396]	Uniform Resource Identifiers (URI): Generic Syntax. T. Berners-Lee, R. Fielding L. Masinter. August 1998
[RFC 3187]	Using International Standard Book Numbers as Uniform Resource Names. J Hakala, H Walravens, IETF October 2001.
[RFC3548]	The Base16, Base32, and Base64 Data Encodings. S. Josefsson, Ed. IETF July 2003
[RFC3875]	The Common Gateway Interface (CGI) Version 1.1. D Robinson, K. Coar.IETF October 2004
[RFC 4287]	The Atom Syndication Format. M Nottingham, R Sayre, IETF December 2005.
[RFC 5234]	Augmented BNF for Syntax Specifications: ABNF, D. Crocker, Ed, P. Overell, IETF January 2008
[RFC 5988]	Web Linking. M. Nottingham. IETF October 2010. Since updated by RFC 8288, October 2017.
[Salomon]	Data Compression: The Complete Reference; David Salomon; Published by Springer-Verlag (1998); ISBN 10: 0387982809 ISBN 13: 9780387982809. The fourth edition from 2007 is available online.
[SciAm]	The Semantic Web, Tim Berners-Lee, James Hendler, Ora Lassila. Scientific American, 1 May 2001 (PDF available)
[TDS]	EPC Tag Data Standard. GS1 Ratified Standard. The first full vesion of this standard was numbered as Version 1.1, published February 2004. See https://web.archive.org/web/20040228171448/http://www.epcglobalinc.com/standards_technology/specifications.html
[UAProf]	User Agent Profiling Specification (1989) as amended by WAP-174_100 User Agent Profiling Specification Information Note (2001) Wireless Application Protocol Forum
[Unicode]	ISO/IEC 10646:2017 Information technology — Universal Coded Character Set (UCS) ISO/IEC JTC 1/SC 2. ISO/IEC Standard December 2017
[URIDoc]	Providing and Discovering URI Documentation, W3C TAG Finding (draft)v 2 February 2002. Jonathan Rees
[vdSompel]	Generalizing the OpenURL Framework beyond References to Scholarly Works, The Bison-Futé Model. Herbert Van de Sompel, Oren Beit-Arie. D-Lib Magazine, July/August 2001 http://www.dlib.org/dlib/july01/vandesompel/07vandesompel.html
[WebVoc]	The GS1 Web Vocabulary. Eric Kauz, Mark Harrison, Phil Archer. GS1 ratified standard 2013 and subsequent revisions.
[Windows]	Why does Windows use backslashes for paths and Unix forward slashes? Superuser.com, September 2011

Some of the content of this document is taken from A history of the ideas in GS1 Digital Link, Phil Archer, with contributions from Jeanne Duckett (Avery Dennison), Keith Jeffery (independent), Simon Dobson (University of St Andrews), Herbert Van de Sompel (then at DANS), Mark Harrison and Henri Barthel (GS1). 25 November 2019.

11 Appendix: Lossless compression / decompression on and offline.

The issue of compression came up very early in the development of GS1 Digital Link. Some data carriers have very restricted capacity, or rather, the amount of space available on a product might restrict the size and therefore the capacity of the data carrier. A GS1 Digital Link URI containing anything more than a GTIN and with anything other than a short domain name quickly exceeds such a restriction so there was a need for a means of compressing a GS1 Digital Link URI as much as possible given the other constraints.

Those constraints are the same ones that led to the development of the GS1 Digital Link URI structure:

- it must be possible to decompress the URI and extract the GS1 application identifiers and their values without needing an online look up;
- the compressed GS1 Digital Link URI must still function as a URL without any processing.

There is only one way to meet those requirements and constraints. And it’s one that GS1 already used in its EPC binary format (see Figure 2) as defined in the EPC Tag Data Standard [TDS] since 2004

11.1 History

Compression – the process of reducing the amount of data used to encode a given message – has a long history.

In order to be able to write any number and any Latin character in both upper and lower case letters (0-9, A-Z and a-z) you need an 'alphabet' of 10+26+26=62 characters. But punctuation is also necessary so the alphabet must expand to include things like a space character plus .,/:;!"'+-=_ and more. This is the kind of thinking that led to the definition of the ASCII character set [RFC20] which has 127 characters, each of which is encoded using 7 bits (2^7 = 128). But, of course, there are so many more characters than that. There are accented characters, for example, like éèöô plus things like ©®™↑←↓→‘'~#[]{}*&^%$£. And beyond this there are many more non-Latin characters.

In reality, there are tens of thousands of characters (there are over 50,000 Chinese characters alone). The 2017 version of the Unicode standard [Unicode] lists 137,994 characters, and details how these can be encoded using one or more bytes per character, that is, at least 8 bits per character. That means that to store the lower case letter a following the Unicode standard, what's actually stored is 01100001 (8 bits). But if you restrict yourself to only 64 characters (that is, don't use Unicode) then you can store the *same* letter using just 6 bits (2^6 = 64) meaning that the letter 'a' can be stored as 011010 (6 bits). So the basic method of compression is:

- write out your string in binary using the most efficient number of bits per character / value (possibly using some bits to indicate which encoding method is used);
- now chop that binary string up every 6 bits;
- convert each set of 6 bits into a character from your restricted 64 character alphabet.

This is a simplification of the actual compression algorithm defined in the TDS and GS1 Digital Link standards, in which the compression of digits is much more efficient, but conveys the general idea.

This process is reversible with no loss of information.

A number of foundational restricted alphabets were standardised in July 2003 [RFC3548] and if you want to go back further, Louis Braille (1809 - 1852) is credited with inventing "the first small binary form of writing developed in the modern era." David Salomon's book 'Data Compression, the Complete Reference', which includes this restricted alphabet-based technique, was first published in 1998 [Salomon]. More than a decade earlier, "Data compression: techniques and applications" by Thomas J. Lynch [Lynch], included a discussion of what the author calls "Compact notation", "logical compression" or "minimum-bit compression". This:

"takes advantage of the fact that in a given field, each n-bit character word may not have meaningful values over the entire 2n possible combinations. Thus, some characters can be coded with shorter fixed-length words, thereby providing a compression."

Lynch goes on to provide an example of an uncompressed date MMDDYY usually requiring 6 x 8 bits = 48 bits. However, he notes that

- a 4-bit word (16 possible values) is sufficient to encode 12 months;
- a 5-bit word (32 possible values) is sufficient to encode 31 days per month;
- a 7-bit word (128 possible values) is sufficient to encode 100 distinct year values.

Therefore, only 4+5+7 bits = 16 bits is required, achieving a compression ratio of 3:1.

A 1993 publication by Held and Marshall [Held, Marshall] uses the same DDMMYY example, referring to the technique as "Logical compression", and includes information about character sets using less than 8 bits per character, including 5-bit Baudot code, which dates from the 1870s, and 4-bit Binary Coded Decimal (BCD), which played an important part in determining the patentability of software and algorithms in 1972. Held and Marshall also discuss the use of 'header' bits that precede the compressed data payload and which indicate the type of compression used as well as the number of compressed characters that follow. Both of these are highly relevant to the approach taken in GS1 Digital Link, but that wasn't the first GS1 standard to use the technique. The GS1 Tag Data Standard, which dates from the early 2000s, includes the USDOD-64 scheme which used 6-bit compaction for alphanumeric sequences (for the CAGE/NCAGE or DODAAC value), and most EPC schemes use integer encoding for numeric strings, to achieve approximately 3.32 bits per digit, rather than 7 or 8 bits per numeric character.

11.2 Application within GS1 Digital Link

GS1 Digital Link uses exactly the techniques discussed above to define a compression/decompression algorithm. It includes a number of enhancements that apply specifically to GS1 identifiers, for example, using some compression headers to state that what follows is a GTIN-14 followed by a serial number. This saves a few bytes by eliminating the need to include the application identifier for 'serial number.'

11.3 The GS1 China compromise

The Mission Specific Working Group recognised that colleagues at GS1 China have been tackling many of the issues addressed by GS1 Digital Link, including compression, in their 2D barcode project. Their solution to the problem of a data carrier's limited capacity and the need to be able to extract identifiers without an online lookup is to use a compromise that accepts the need, which GS1 Digital Link does not, for an online lookup for everything other than the primary identifier, thus:

http://2dcode.org/0106901234567892OXjVB3

In this example, the GTIN (01)06901234567892, which is of fixed length, is followed by a generated string. If looked up online (at 2dcode.org) that extra string of OXjVB3 can provide batch/lot or any other set of GS1 application identifiers and their values. In this way, the primary identifier is always available without an online lookup (subject to following parsing rules); only the ancillary information requires an internet connection and a data service.

This approach was taken into account when developing the compression algorithm. One option within the GS1 Digital Link compression algorithm is to leave the primary identifier uncompressed and only compress any key qualifiers and the attributes in the query string. Unlike the Chinese version, all extra information can be decompressed from a GS1 Digital Link URI at the point of scan without any need for an online service.