HTTP Uniform Resource Identifiers to associate a web resource with a GS1 key and optional Application Identifiers

Background Context Document

Mark Harrison (Auto-ID Labs, Cambridge, UK)
Robert Beideman (GS1 Global Office)
Henri Barthel (GS1 Global Office)
Scott Gray (GS1 Global Office)
Ken Traub (Ken Traub Consulting LLC)

7th March 2014

Auto-ID Labs White Paper WP-SWNET-032

SWNet

1. Introduction

Across a number of active GS1 initiatives, requirements have emerged for representing data in a robust, web-friendly format that can better connect data with consumers that are searching for that data.

Specifically, the Next Generation Product Identification (NGPI) project is considering making use of a URI prefix, a GS1 Key, and a small number of standard attributes to enable consumers and other parties to access trusted detailed product information. Concurrently, GS1 is working with the Open Mobile Alliance (OMA) to develop a universal bar code scanning specification for inclusion in the firmware of future smartphones that will include a standard Application Program Interface to connect consumers with accurate, trusted information. Also, the GS1 Digital program is active and includes two major components: 'GTIN+ on the Web' and GS1 Source. The 'GTIN on the Web' project aims to promote the use of structured data including standard GS1 product identifiers and attributes across the open Web.

To address all of these expressed needs, GS1 is working to assure the creation of a single, robust solution that serves the needs of all of these drivers. We're also looking ahead to ensure that the solutions that we develop with our Community will remain relevant for many years. The Web is a swiftly-evolving landscape, and GS1 is focused on ensuring that consistent, standardised representation of the GS1 System on the Web is adopted around the world. This paper is intended to provide context and background to the recent GS1 technical proposal titled: "HTTP URIs to associate a web resource with a GS1 key and optional AI's."

2. GTIN+ on the Web

"GTIN+ on the Web" is one of the first projects within the GS1 Digital portfolio. It has a particular Business-to-Consumer (B2C) focus on making information about products more readily accessible on the Web.

As search engine technology is becoming smarter and relying increasingly on the ability to extract the meaning of concepts and relationships in data that "lives" in web pages, this same technology is evolving away from merely indexing keywords and the topology of hyperlinks found within and among a set of web pages. To stay current with this massive shift in the realm of search and discovery, standard practices of embedding text-based (or simple XML) information, images, and formatting about products or offers in web pages must also evolve to leverage technologies that allow search engines (and, ultimately, every person that uses the Internet) to more accurately and completely understand what companies are trying to communicate. It's no longer sufficient to create web pages that are only able to be understood by humans… we must all work to assure that web pages can also be understood, or "consumed", by machines. We must add context to the information embedded in web pages to allow search engines and other information consumers to efficiently disambiguate (to resolve conflicts that arise when a term may have more than one meaning).

Because of this emerging reality, the GTIN+ on the Web project makes significant use of Semantic Web technology (also known as Linked Data), to make it easier for mobile applications ('apps') on smart phones and search engines to extract the *intended* meaning of the information embedded within web pages. This technology will ultimately allow Brand Owners and Retailers of any size to have *more control* over how their web pages and products and offerings are understood by the search engines and apps of the world.

The GTIN+ on the Web project aims to make it easier to *discover* products on the Web. Currently, it is quite difficult for end-consumers or search engines to determine whether two web pages about a product are actually referring to the same product. The fundamental reason for this is that today very few websites actually include the Global Trade Item Number (GTIN) within the web page, so it is very difficult to know that two product-specific pages are actually referring to the same real-world product; instead, search engines and end-consumers must rely on comparisons of names and descriptions of products, which are generally not consistent across all web pages about the same product. The net result of this current reality is lower-than-possible visibility of your products across the Web… and this is true for companies of all sizes.

Similarly, the GTIN+ on the Web project aims to make it easier to *understand* information about products on the Web. Currently, apps and search engines struggle to find meaning and connection between products and the accurate, trusted data about those products. Without a structure and ontology that is designed to accurately define and represent contextual information about products, apps and search engines need to search for context… and often don't even "see" what publishers are trying to tell them. The result of this current reality is often erroneous, outdated, or incomplete information about products. Consumers are quickly evolving to have zero tolerance for inaccuracy or missing information.

This is our reality, and the GTIN+ on the Web project has the goal of ensuring that assertions and claims about the products that companies manufacture or sell are understood as they're intended.

If we have a convenient and universal way of embedding the GTIN in web pages, the task of correlating information about product specifications, reviews, offers, etc. becomes much simpler, since they can refer to a consistent identifier. In addition, reference back to a single, globally unique identifier helps to mitigate current challenges in complex inventory, picking, and fulfilment models that are trying to work efficiently across both physical and digital paths to purchase.

Future projects within the GS1 Digital portfolio may investigate how such technologies can be used in Business-to-Business (B2B) scenarios, but our first priority is to assure that the existing GS1 System of Standards has a consistent and reliable method of representation on the open Web for existing B2C business requirements. This B2C focus is the particular mission of the "GTIN+ on the Web" project. Eventually, we may look to investigate the potential of the GS1 Digital program to develop an alternative method of exchanging data within and across supply chains if this can help SMEs and micro-businesses align with GS1 global standards at low cost, especially when they lack significant in-house IT infrastructure or expertise, but the priority of such a future effort will be governed by the needs of our users. In any case, we see the GTIN+ on the Web project as the most foundational element of our overall GS1 Digital strategy.

3. Linked Data / Semantic Web Technology

One of the fundamental technologies we plan to use in the GS1 Digital | GTIN+ on the Web project is called Linked Data technology (also known as Semantic Web technology). Linked Data / Semantic Web is founded on technical standards such as Resource Description Framework (RDF) from the World Wide Web Consortium (W3C), who also developed a much more familiar standard (XML) that is currently used for exchanging structured data across the Web today.

Linked Data enables factual claims ('facts') to be expressed in a machine-readable manner and makes it easy to combine such facts from multiple sources (i.e. from multiple organizations that make factual assertions). In this way, this technology allows Brand Owners and Retailers to state critical product information on their web pages that can be more accurately and reliably understood by search engines and apps. Each product has a number of intrinsic properties / attributes that are characteristic of it. These properties or attributes are typically concerned with geometry (width, depth, height), weight or mass, technical specifications, ingredients lists, nutritional information, etc.

A company might want to publish a fact such as 'GTIN 0614141112345 identifies a product with weight 600 grams' - and would like that to be machine-readable, with unambiguous semantics, so that computer software can interpret that information correctly, instead of just displaying that as a text string of characters that is meaningful to a human being but rather meaningless to most computer software. In fact, the company may want to state many such facts about a particular Trade Item. Using Linked Data technology to structure these facts will allow the publisher to state, with complete clarity, all of the relevant information about the products it manufactures or sells.

And, by using Linked Data technology, a number of facts about a product can be accessed using the same mechanism that our web browsers already use when requesting a web page (HTTP GET). Further, web-servers can be configured to serve not only web pages for human consumption (as with the web pages that you view on any given day)… but also to serve machine-readable data containing sets of facts in response to similar HTTP GET requests for other representations of the factual data. And, Linked Data technology provides a common framework for making it easy to extract and combine the essential facts from any data source, whether it be from a relational database, spreadsheet, XML data or web pages, as well as being able to 'click' on any HTTP URI to find out more information about its meaning.

But, as might be imagined, there's a good bit of structure needed to represent pieces of information in such an understandable manner, and there's a need to have a method for the accurate locating and combining of pieces of information to form a holistic picture of any particular product. Fortunately, both the structure and the needed method already exist. The W3C SPARQL query language standard is designed specifically for combining facts from multiple sources and can even be used to create new facts that are inferred from existing facts using rules that a user defines as the conditions ('WHERE' clause) of a SPARQL query. This query language works hand-in-hand with Linked Data / Semantic Web technology to create a robust and very accessible foundation on which the GTIN+ on the Web project can be built.

Before we talk about actually structuring data using Linked Data technology, we must first discuss the most fundamental elements of identification on the Web. On the Web, it is possible to identify virtually anything that can be identified in the physical world, whether it's an object, a place, a feature, a category, or a value. Much work has been done over the years to define types of identifiers (and their proper usage). In general, we will concern ourselves with the type of identifier called a Uniform Resource Identifier.

4. Uniform Resource Identifiers

The most common and familiar type of a Uniform Resource Identifier (URI) is a web address or Uniform Resource Locator (URL). Any web page that you've ever visited has one, and it's most commonly shown in a box at the top of a web browser's window. This string of characters is uniquely associated with the particular web page viewed in the browser. That string of characters has two major purposes: it tells the web browser *where* to point to request information, and it tells the web browser *how* to request the information that is located at a particular address. Here are some examples of URLs:

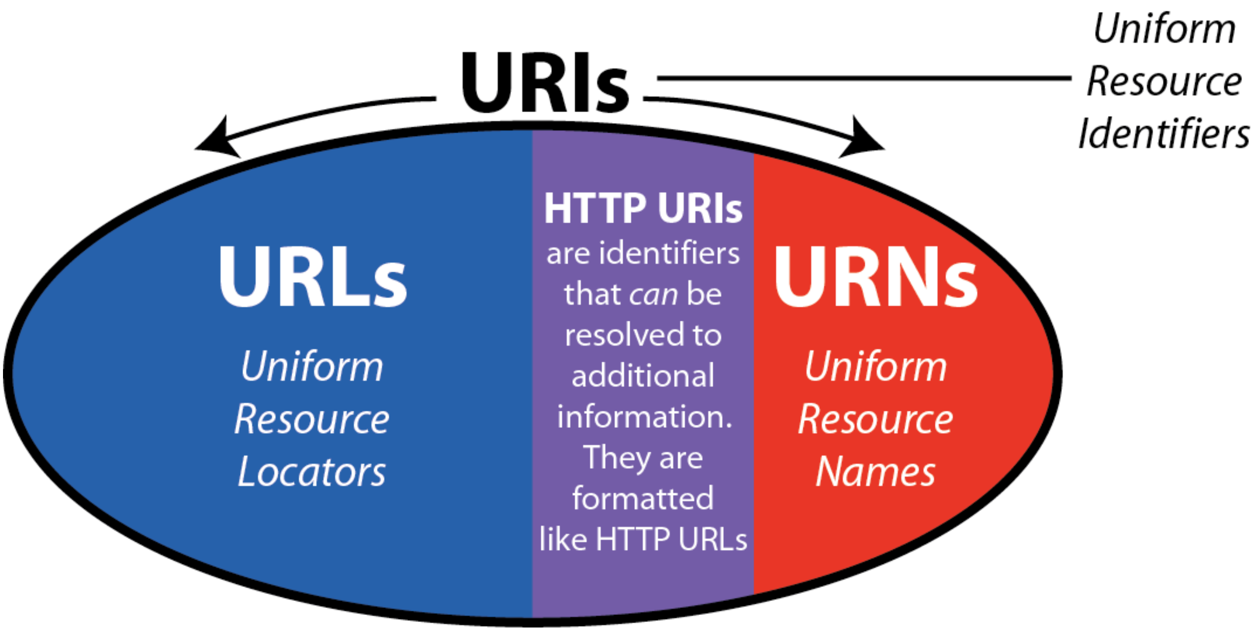
- http://www.gs1.org
- http://www.google.com
- http://www.nestle.com

A Global Trade Item Number (GTIN) or and Electronic Product Code (EPC), when expressed in its canonical pure identity format, is a Uniform Resource Name (URN), which is also a type of URI. A URN is a globally unique representation of a particular thing.

Here are some examples of URN's:

- urn:epc:id:sgtin:0614141.112345.400
- urn:epc:idpat:sgtin:0614141.112345.*

Identifiers called URIs include all URLs and all URNs. So, all URNs are URIs and all URLs are URIs, but the converse is not true, as shown in the diagram below.



URIs also include HTTP URIs, which can serve ***either*** as names ***or*** as locators, ***either*** as a name for identifying things ***or*** as a locator, for retrieving factual information about things. Things that are identified with URIs are called resources. Resources might be non-information resources, such as real-world objects - or they can be information resources that describe non-information resources.

It is very common practice to see HTTP URIs being used in Linked Data because in addition to providing a globally unambiguous name for something or for a particular kind of relationship that connects two things, it is possible to retrieve information about the thing or relationship by treating the HTTP URI like a URL and performing a web request (HTTP GET request) for the thing.

For example, the following are HTTP URIs that serve as names of non-information resources:

http://dbpedia.org/resource/Brussels
http://data.ordnancesurvey.co.uk/id/postcodeunit/CB30FS

The following are the corresponding HTTP URIs that serve as locators for corresponding information resources:

http://dbpedia.org/page/Brussels
http://data.ordnancesurvey.co.uk/doc/postcodeunit/CB30FS
http://data.ordnancesurvey.co.uk/doc/postcodeunit/CB30FS.ttl

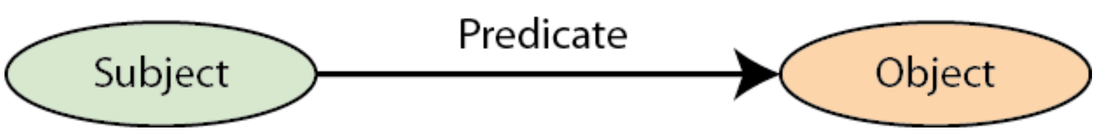
Note that we did not say that an HTTP URI is a globally unique name for something or some relationship. One of the principles of Linked Data is the Non-Unique Naming Assumption, i.e. two individuals or companies might use distinct HTTP URIs to refer to the same thing. An HTTP URI is globally unambiguous because it points in a well-defined way to one thing or one collection of things - but an HTTP URI created by someone else can do exactly the same thing.

Lastly, it should be noted that names and descriptions of things can be represented in all local languages through the use of Unicode strings and language tags, e.g. @de, @fr.

5. Expressing facts about products and product offerings using RDF triples

Linked Data allows us to write such facts using simple logical sentences called RDF triples. RDF stands for Resource Description Framework, a W3C standard.

An RDF triple consists of three elements (hence the word 'triple') - these are the Subject, Predicate and Object and these represent a relationship that connects the Subject to the Object via some property known as the Predicate.



This structure is not much different than the traditional sentence structure that we all learned to create as children. Some example triples, when spoken in English, could be:

- GS1 defines standards.
- This product has a GTIN.
- This GTIN is encoded in a UPC Code.

consi
a dire

- GS1 owns the domain GS1.org.
- This EPC is encoded in an RFID tag.
- This candy bar has 8g of sugar.

represents one of these linkages in a mind-map of facts (also known as a 'graph') and even complicated mind-maps can be decomposed into a collection of triples.

Linked Data technology uses Uniform Resource Identifiers (URIs) to provide globally unambiguous names for things and relationships. In the process of defining such relationships through the creation of RDF triples, there is a common need to refer to words that, while important to the context and the meaning of the triple, may not make specific reference to a globally unambiguous thing. We call such things "blank nodes". As example, consider the triple-formatted version of the sentence "Mark has a friend who lives in Edinburgh." To write this properly in triple notation (and using the FOAF vocabulary), one may write:

ex:Mark	foaf:knows	_:p1 .
_:p1	foaf:based_near	http://dbpedia.org/resource/Edinburgh .

In this example, the reference to _:p1 is a reference to Mark's anonymous friend who, by way of association through the above triples, we know lives in Edinburgh. Note that, in this case, p1 is a reference to a "blank node".

The **Subject** of an RDF triple is expected to be a URI (representing a named thing) or a blank node (representing an un-named thing). We use a blank node to refer to 'something', without giving that 'something' a name (see above example). We use a URI to provide a globally unambiguous reference to a specific thing. Human-readable names for that thing in multiple human languages can be accessed through triples that link the Subject URI to such labels via an **rdfs:label** predicate.

The **Predicate** is expected to be a URI. Linked Data vocabularies such as schema.org and GoodRelations define lists of predicates and types, which can be used for expressing relationships. It is expected that GS1 will publish its own GS1 vocabulary / vocabularies to express the attributes and terms that are defined in GDSN data formats and in the GS1 Global Data Dictionary (GDD).

The **Object** of an RDF triple may be a literal data type such as a text string, date/time, integer, floating point number - or it can be another resource (representing a complex data object), in which case it is expected to be a URI or a blank node that represents a collection of attributes and values but is not assigned a URI.

It should be noted that an HTTP URI that serves as a **name** can be used in the Subject or Predicate of an RDF triple. It can also be used as the Object of an RDF triple, although an Object can instead by a literal value, such as a date-time timestamp, a string, an integer, floating point number etc.

An HTTP URI that serves as a **locator** is not used within the Subject, Predicate or Object positions of an RDF triple but is instead used for retrieving multiple facts or a collection of related RDF triples.

6. Putting it all together

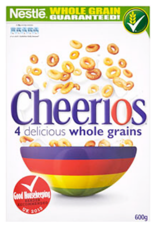
Returning to information about products, a brand owner could construct an HTTP URI such as

http://id.nestle.com/gtin/05011476100885

as an **HTTP URI name** for their product with GTIN 05011476100885. They can use such an HTTP URI as part of the language that they need to author RDF triples that define facts about the thing called http://id.nestle.com/gtin/05011476100885, but this string is not a **locator** and does not point to a set of information.

In the physical world, this step is similar to having this: 5 011476 100885

without having affixed it onto this:



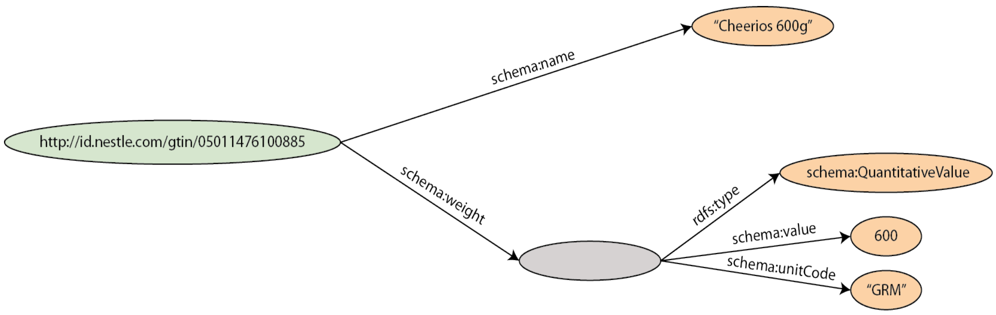
In this case, an HTTP URI for a product has been created, but it has not yet been associated with the actual product (or with relevant information about the product). At this point, it's like seeing the name of a person, but not knowing which person has that name or knowing anything about that person.

Now, to associate this unique identifier with information about the product (the physical-world equivalent of placing the barcode on the Cheerios packaging or "putting a name to a face"), a brand owner could describe the product as completely as they wish to by authoring clear, unambiguous information about the product on the Web. For example, a brand owner could use terms from existing vocabularies, and they could write facts such as:

<http://id.nestle.com/gtin/05011476100885> schema:name "Cheerios 600g"^^xsd:string .
<http://id.nestle.com/gtin/05011476100885> schema:weight _:1 .

_:1 rdf:type schema:QuantitativeValue .
_:1 schema:value "600"^^xsd:float .
_:1 schema:unitCode "GRM"^^xsd:string .

This corresponds to the following mind-map of facts:



While the above mind-map only asserts two facts, it should be clear that this example can easily be extended to include dozens of relevant facts about a product (in fact, the amount of information that could be represented is limitless). And, when represented in Linked Data format, these facts can be accessed in a variety of ways by different requestors.

Once the needed facts are defined as RDF Triples, those RDF triples can then be accessed in a variety of ways. As was said before, this is where the power of this particular form of expression lies.

7. Interpretations of RDF Triples and Automatic redirection by Web Servers

If we make a web request for http://id.nestle.com/gtin/05011476100885

using a web browser, and if we remember that this string HTTP URI that is behaving as a **name** and not as a **locator**, we would expect to get nothing in return to our request.

However, web servers are often configured to interpret a request for http://id.nestle.com/gtin/05011476100885 as a human-driven request, and they may automatically redirect such a request to a web page such as http://id.nestle.com/gtin/05011476100885.html
that presents the facts about the product with GTIN = 05011476100885 in a tabular format, easy for humans to read.

Alternatively, computer software (such as the crawler of a search engine or an 'app' on a mobile phone or tablet) might prefer to obtain direct access to the essential facts and figures, without having to parse a human-readable web page to extract this information.

Such software could make a request for http://id.nestle.com/gtin/05011476100885.ttl to obtain a set of RDF triples in Terse Triple (Turtle) [.ttl] notation, as above. The data that would be returned to this request would be much more suitable to the needs of an app or a web crawler.

Alternatively, it might make a request for http://id.nestle.com/gtin/05011476100885.json to obtain the RDF triples encoded using JSON-LD (JavaScript Object Notation for Linking Data), which might make it more convenient for application software (e.g. mobile 'apps') to iterate through tables or lists of facts and figures.

The most important thing to take away from the last few paragraphs is the idea that using the right technologies to represent information about your products online can open up the door to data sharing in multiple important ways. GS1 is actively investigating the details of what needs to be done to ensure standard representation of the GS1 System on the web, so that you can begin to take advantage of this technology as quickly as possible.

Instead of using filename suffix notation to request the RDF triples in a particular serialization format, another approach is to use HTTP Content Negotiation. A web browser will typically specify the MIME type 'text/html' in its HTTP Accept: header, e.g. Accept: text/html

However, if an HTTP GET request instead specifies Accept: text/turtle

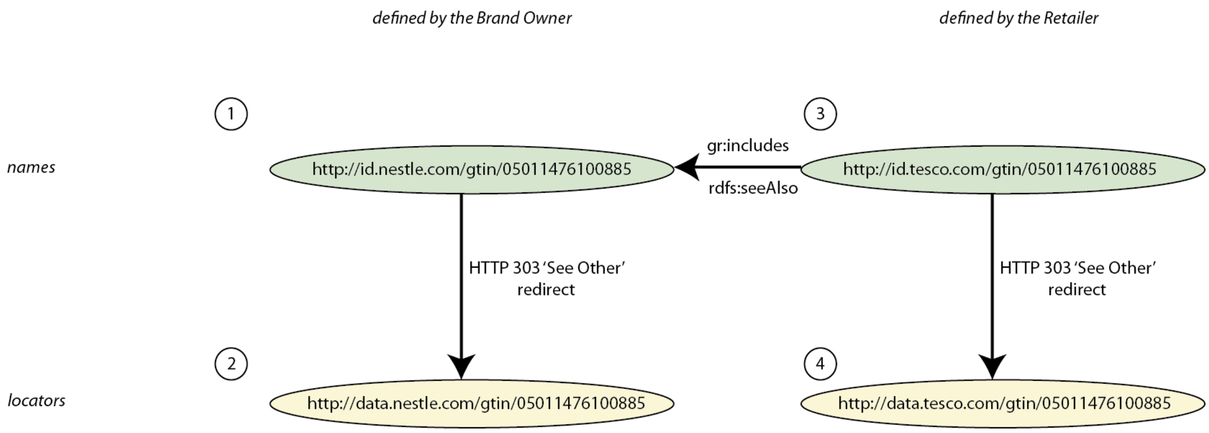
then they are requesting the Terse Triple (Turtle) serialization of the RDF triples.

8. Four kinds of HTTP URIs to consider for GS1 Digital | GTIN+ on the web

In practice, for GS1 Digital | GTIN+ on the web, we need to consider, at most, 4 different kinds of HTTP URIs relating to a product GTIN:

(1) An HTTP URI defined by the brand owner, serving as a **name** and as a Subject in RDF triples about the product
(2) An HTTP URI defined by the brand owner, serving as a **locator** for a collection of RDF triples about the product
(3) An HTTP URI defined by another party such as a retailer or reviewer, serving as a **name** and as a Subject in RDF triples about their offering or review for the product
(4) An HTTP URI defined by another party such as a retailer or reviewer, serving as a **locator** for their collection of RDF triples about their offering or review for the product

All four of these should be considered as part of the 'Share' layer of the GS1 Architecture - not the 'Identify' layer; the only URI format defined in the 'Identify' layer of the GS1 Architecture are EPC pure identity URIs (technically URNs), which are not expected to be used in Linked Data because they cannot be dereferenced in a trivial manner.



9. Possible entry points into 'GTIN+ on the web' HTTP URIs

1. Web Search / Website Search by keyword, product category, refinement with contextual filters (both qualitative and quantitative) to identify candidate GTINs
2. Follow a semantic link (e.g. rdfs:seeAlso, gr:includes) from a retailer's HTTP URI for a GTIN to a brand owner's HTTP URI for a GTIN
3. Scan a Next Generation Product Identifier (NGPI) data carrier containing an HTTP URI locator
4. Scan of an existing 2D data carrier (GS1 QR Code) that may have encoded within it a full HTTP URI serving as a locator.
5. Scan an existing data carrier; construction of the corresponding brand owner's HTTP URI may be enabled by a new or existing GS1 lookup service (e.g. ONS) that maps from a GS1 element string to the brand owner's HTTP URI for that GTIN / SGTIN / GTIN + lot/batch, GTIN + variant etc as previously registered by the brand owner with that lookup service.

10. Next steps

The accompanying technical proposal document is intended as input to the new GS1 GTIN+ on the Web MSWG, where these URI structures will undergo formal technical standardisation. The GS1 GTIN+ on the Web MSWG will work closely with the Next Generation Product Identifier (NGPI) MSWG and also ensure close communication with the Open Mobile Alliance (OMA) to avoid divergence or duplication of effort.