GS1 Global Smart Parking System: One Architecture to Unify Them All

Nhat Pham*, Muhammad Hassan†, Hoang Minh Nguyen*, Daeyoung Kim*
* *Auto-ID Labs,* † *Embedded Computing Lab, School of Computing,*
Korea Advanced Institute of Science and Technology,
Daejeon 34141, South Korea
Emails: {nhatphd, m.hassan, minhhoang, kimd}@kaist.ac.kr

***Abstract*—With the increase in the number of cars in big cities, it has become a nightmare for car owners to find available parking spaces. Some studies report that around 25-30% traffic on the roads are cars finding available spaces for parking. While many researches have been conducted to create a smart parking system, most of them only focus on some particular areas and environments without any common standards. As a result, this hurts users' experience as they have to use many different applications to access various APIs when traveling to different places. Thus, in this study, we wish to open the discussion to realize a global and common base for smart parking services by proposing a smart parking systems based on GS1 global standards. By utilizing common and global standards, our proposed architecture could be used globally and also easy to extend with different services. We have also implemented a prototype system which can support parking lots in Busan city and 9 Korean Airports to prove the feasibility of our architecture.**

***Keywords*-GS1, EPCIS, ONS, IoT, Smart Parking System, Smart Cities.**

I. INTRODUCTION

In recent decades, the population and size of our cities have increased significantly and is predicted to steadily increase in the near future[1]. As a result, many urban problems such as communication, mobility, and energy are raised as challenges for our future smart cities to overcome. Finding a parking space in a crowded city is one of the serious challenges with 25-30% of traffic on the roads are estimated to be finding an available parking area to park their cars [2]. This problem is clearly visible in big global cities like Seoul, New York, Los Angles, London, and Beijing. Even though in these cities authorities have built parking plazas to alleviate this issue, during peak hours those plazas are occupied really quickly and people have a lot of problems finding available parking spaces. Thus, a solution is necessary to help people find available parking spaces and authorities to figure out over-utilized spaces so that appropriate maintenance and expansion steps can be taken.

With the advent of the Internet of Things (IoT) and smart cities initiatives all around the world, many research effort and projects in both industry and academia have emerged in recent years to address this issue. While we are able to handle parking problem in some particular cities or areas with the use of various technologies, this heterogeneity also fragments our ecosystem with many different standards being used in the same area. This fact hurts users' experience because when people travel to different areas, cities, or countries, they have to use many different applications to access some particular APIs. Moreover, it also prevents collaborations among diverse companies and stakeholders to share their data with each other.

Thus, having an open and unified community providing a seamlessly integrated service for users anywhere they go has encouraged us to devise an architecture with common global standards to unify all the data and services together. To achieve this vision and push the barrier of our smart parking system research one step further, we propose an idea of a global architecture based on GS1 standards [3] including the uses of Object Name Service (ONS) [4] and Electronic Product Code Information Services (EPCIS) systems [5]. Furthermore, this study also aims to open the discussion on development of unified APIs and data formats in further researches to realize a global and common base for smart parking services.

In this study, we assume that in the future smart cities, every parking lot is capable of monitoring its current available spaces in real-time by using either intrusive devices (e.g. active infrared sensors, inductive loops, magnetometers, etc.) or non-intrusive ones (e.g. cameras, RFID, ultrasonic sensors, etc.) to monitor when cars arrive and depart. These data together with information about parking lots such as name, address, GPS coordinates, and ticket prices should be publicly available and accessible from users' applications. By utilizing GS1 global standards, our study of smart parking system focuses on universality, and extensibility. While a global service could be achieved through GS1's globally unique identification system, extensibility requires not only to employ GS1 standards such as EPCIS and ONS but also a global architecture for them to meet particular requirements of smart parking services. It is up to for application developers to decide which services to be included and how to get the best performance.

The EPCIS and ONS systems we use in this paper were developed as parts of our Oliot project. Oliot provides a complete implementation of GS1/EPCglobal standards and also extends the code system of GS1 and their standard ar-

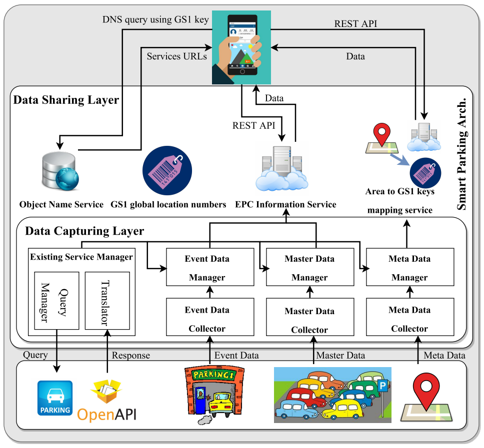


Figure 1. High level concept of GS1 Smart Parking System

chitecture to support various IoT connectivity and protocols. It's open-source and freely available at [6].

The rest of the paper is organized as follows: our proposed architecture are discussed in detail in section II. Section III presents a prototype implementation for parking lots in Korean airports and Busan city. Finally, we conclude our study in section IV.

II. GS1 Smart Parking System Architecture

Fig. 1 illustrates the high level concept of our GS1 Smart Parking System. As the object in this system is parking lot, we only need to use GS1 Global Location Number (GLN)[7] as identification key. To acquire GLNs, parking lot owners or government officers could register with GS1 to get company prefix digits. The rest of the first 12 digits (in total of 13 digits) of a GLN could be used to identify parking lots belonging to that owner. In our system, static information of a parking lot such as: name, address, GPS coordinates, Global Location Number etc. could be stored as master data while real time data such such as current available spaces monitored by sensors could be defined as event data.

Our smart parking system is divided into three parts: Data Capturing Layer, Data Sharing Layer and Mobile application. In Data Capturing Layer, we can collect data about parking lots either from existing services (e.g. by querying from open APIs) or directly from parking lot itself. If we capture data from existing services, these data have to be translated into master data and event data before forwarding to EPCIS system. On the other hand, if we collect master and event data directly from parking lots, we just need to forward these data to EPCIS system. In the Data Sharing Layer, there are three main components are needed in our system, namely, Area-2-GLNs mapping service, EPC Information Service (EPCIS), and Object Name Service (ONS). From the application developer's perspective, when the application is opened, we only know user's current location or a desired area which we want to find parking lots' information without knowing which parking lots are inside that area.

Thus, we propose and implement an Area-2-GLNs mapping service based on EPCIS to provide mapping between an area and parking lots' GLNs which are inside. For Area-2-GLNs mapping service to work, we need to collect additional meta data holding information about GPS coordinates and global location number of each parking lot. This meta data can also be extracted from master data if possible. After we get parking lots' GLNs in our desired area, we will need to query ONS system to find which services (e.g. retrieve current available parking spaces, etc.) are available for a GLN and URLs to corresponding EPCIS server for each service. With the URLs to EPCIS system, we can now request master data and event data and display information of parking lots on smart phone screen. The detail architecture of Area-2-GLNs mapping, ONS, and EPCIS services in our smart parking system will be discussed in the next sections.

The interactions among mobile applications, Area-2-GLNs service, ONS, and EPCIS could be summarized as follows: When user clicks on Find Parking Lots button, application will get the designated area boundaries by capturing GPS coordinates of North-East and South-West corners of the current map view on screen and use these coordinates to find nearby parking lots GLNs from our Area-2-GLNs service. After having the list of GLNs from Area-2-GLNs service, application then use these GLNs to get EPCIS URLs of corresponding EPCIS servers from ONS system. Then, it can query information including static information in master data and real-time information in the latest event data from EPCIS server. After that, application can repeatedly query latest event data after each period of time (e.g. several minutes) from EPCIS server to update information on its screen.

A. Area-2-GLNs service

To handle our Smart Parking system in global scale, we propose a hierarchical architecture for Area-2-GLNs service in Fig.2. User's application starts by querying at a fixed root server with its current GPS coordinates (1). Using a third-party service such as Google Map API's reverse geocoding, we could convert GPS coordinate to corresponding address. E.g: *36.367056, 127.363965* $\Rightarrow$ *Daehak-ro, Yuseong-gu, Daejeon, South Korea*. By using the country name in resolved address, root server redirects the application to a matching Area-2-GLNs server belonging to that country (2). The application repeats the same process again until it get desired data from a Area-2-GLNs server. The URLs of each Area-2-GLNs servers queried by the application will be cached for further queries. When user moves to another location and application could not get data from the same Area-2-GLNs server, it will use cached URLs to trace upwards until it find another sever holding needed data.

Our Area-2-GLNs service is designed as an EPCIS system. Meta data of each parking lot is stored as an EPCIS Event data which is modifiable by using ADD/DELETE actions of EPCIS system. Fig. 3 represents the information stored in XML files of our Area-2-GLNs service. To register a new parking lot to Area-2-GLNs system, an EPCIS event needs to be created with an assigned GLN from GS1. To query information from a Area-2-GLNs server, application needs to modify EPICS SimpleEventQuery to retrieve only parking lots within a desired area using GPS coordinates of northeast and southwest boundaries.

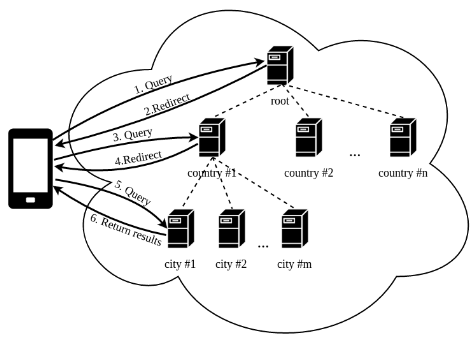


Figure 2. Hierarchical architecture for Area-2-GLNs service

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader">
    <xs:simpleType name="gps_latitude">
        <xs:annotation>
            <xs:documentation>
                parking lot's gps location (latitude)
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
        <xs:simpleType name="gps_longtitude">
        <xs:annotation>
            <xs:documentation>
                parking lot's gps location (longtitude)
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="parkingspace_name">
        <xs:annotation>
            <xs:documentation>
                parking lot's name
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="gpc">
        <xs:annotation>
            <xs:documentation>
                GS1 GLN code
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 3. XSD schema for Area-2-GLNs service

B. ONS systems

ONS system, according to GS1's standard[4], leverages the existing DNS infrastructure and standards for resolving GS1 identification keys and it should be installed alongside our current DNS system. Thus, entries in ONS are also stored as NAPTR (A Name Authority Pointer) records. An example of ONS NAPTR record is shown in Fig. 4. NAPTR record of ONS has some noteworthy fields:

- Service field: in our smart parking system, this field contains an URL which indicates a particular service which will be found by following the URL in RegExp field.
- RegExp field: this field, in our system, holds a corresponding EPCIS URL to a service which has been indicated in Service field. For an example, In Fig. 4, RegExp has the URL to a EPCIS server keeping data for available spaces of parking lots.

When parking lot owners register their EPCIS server with ONS system, they could specify which services they want to provide such as: query current available spaces, reserve parking space in advance, etc. in Service field of ONS NAPTR, and provide matching EPCIS URLs for these services in the RegExp field. On the other hand, when application want to get information about a parking lot, it first converts parking lot's GLN to FQDN (fully qualified domain name) and use this FQDN to query NAPTR records containing information about parking lot's services and corresponding EPICS URLs.

Order	Pref	Flags	Service	RegExp	Replacement
0	0	"U"	"www.parking-space-finder.org/freespace"	"!^.*$!http://143.248.53.173:10024!"	.

Figure 4. An example of ONS NAPTR record.

C. EPCIS systems

To participate in our smart parking system, governments (or organizations) needs to get an GLN numbers from GS1 and install Area-2-GLNs and EPCIS systems. Then, parking lot owners can register with governments and setup their parking lot sensors system to send Master, Event and Meta data to EPCIS and Area-2-GLNs system, respectively. If possible, parking lot owners could also install a EPCIS server by themselves and register their EPCIS server with the ONS system.

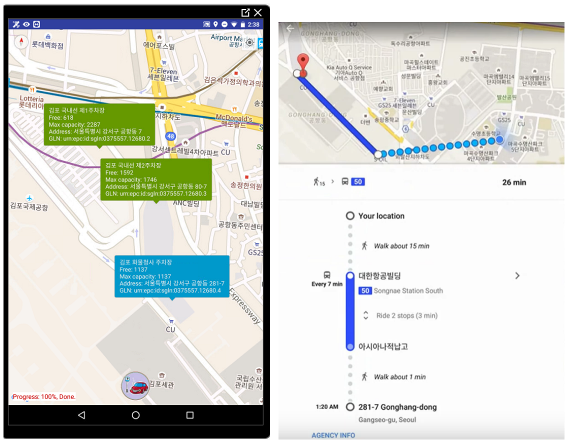
```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader">
    <xs:simpleType name="parkingspace_name">
        <xs:annotation>
            <xs:documentation>
                parking lot's name
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="parkingspace_address">
        <xs:annotation>
            <xs:documentation>
                parking lot's address
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="parkingspace_max_cap">
        <xs:annotation>
            <xs:documentation>
                parking lot's maximum capacity
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="gpc">
        <xs:annotation>
            <xs:documentation>
                GS1 GLN code
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
```

Figure 5. XSD schema for EPCIS Master data.

XSD schema for EPCIS Master and Event data are presented in Fig. 5 and Fig. 6, respectively. In our current design, Master data consists of parking lot's name, address, maximum capacity and GS1 Global Location Number. We could easily extend information of Master data by adding new blocks in Master data XSD schema. Event data, on

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
    attributeFormDefault="unqualified" elementFormDefault="qualified"
    targetNamespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader">
    <xs:simpleType name="available_space">
        <xs:annotation>
            <xs:documentation>
                current available parking spaces
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Figure 6. XSD schema for EPCIS Event data



(a) Main screen (b) Direction screen

Figure 7. Android application.

the other hand, holds information about current number of available spaces in each parking lot. Similar to Master data, we could extend Event data in the future by modifying Event data XSD schema.

To query information of master data from EPICS system, application can use SimpleMasterDataQuery provided by EPCIS system. The server will return to application a XML file. Application needs to parse returned XML file to get information about parking lot's name, address, etc. Similarly, application can request event data from EPCIS server to get real-time data about current available parking spaces. Because we only need to know the latest information, application just need to query the last event has been pushed to EPCIS server by modifying the query to sort in a decreasing order of event time and only retrieve the first entry.

III. Prototype Implementation

To prove the feasibility of our proposed architecture, we have implemented a prototype system for parking lots in Korea. Our prototype supports 9 international airports such as: Gimpo Intl. Airport, Gimhae Intl. Airport, Jeju Intl. Airport, etc. and Busan city with total 35 parking lots. Real-time data about these parking lots provided by Korean government are publicly available on the Internet which can be queried by open RESTful APIs at [8], and [9]. To comply with our architecture, Existing Services Manager has been implemented in our data capturing layer to capture and generate EPCIS master and event data for our system. An Android application (Fig. 7) also has been implemented for Nexus 7 (Android 6.0) device. By using Google Map API, application could show users' current locations and information of parking lots in a specified area. It will highlight each parking lot with different color based on its current state (e.g. red means parking lot currently doesn't have available parking spaces, green means it has available spaces, and blue is for the nearest one which has available spaces). We could also use application to show us direction to a parking lot by using direction API from Google Map. After each 5 minutes, Android application will query the latest event data from EPCIS servers of the parking lots in current map view area to update their current available spaces.

IV. Conclusion

In this paper, we present a global architecture for smart parking system by using GS1 standards including architectures for Data Capturing and Data Sharing layers. This also provides a new and open business model for both parking lots owners and users everywhere in the world to smoothly integrate their information and services around common standards. As a result, user just needs to use an unified application anywhere they go to find parking areas around them. By utilizing GS1 standards, our proposed architecture is easy to scale and extend to support future services. Lastly, we have made a prototype system to show the feasibility of our proposed architecture. As our current implementation is only a prototype, more efforts to evaluate performance and security are needed to deploy our architecture in practical environment.

V. Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. R7115-16-0002, Wise-IoT).

References

[1] U. Habitat, "From habitat ii to habitat iii: Twenty years of urban development," *World Cities Report*, 2016.
[2] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.
[3] Gs1. [Online]. Available: http://www.gs1.org
[4] Gs1 object name service (ons). [Online]. Available: http://www.gs1.org/epcis/epcis-ons/2-0-1
[5] Epcis and core business vocabulary (cbv) —gs1. [Online]. Available: http://www.gs1.org/epcis
[6] Auto-id labs, kaist, open language for internet of things. [Online]. Available: http://gs1oliot.github.io/oliot/
[7] Global location number. [Online]. Available: http://www.gs1.org/gln
[8] Busan parking garages real-time status. [Online]. Available: https://data.go.kr/dataset/3070682/openapi.do
[9] Airport parking garages real-time status. [Online]. Available: https://data.go.kr/dataset/3050226/openapi.do