Open architecture for internet-based C-ITS services

Meng Lu
Dynniq Nederland B.V.
Dynniq Group
Amersfoort, The Netherlands
wklm@xs4all.nl

Robbin Blokpoel
Dynniq Nederland B.V.
Dynniq Group
Amersfoort, The Netherlands
robbin.blokpoel@dynniq.com

Manuel FünFrocken
Hochschule für Technik und Wirtschaft des Saarlandes
Saarbrücken, Germany
manuel.fuenfrocken@htwsaar.de

Jacint Castells
Dept. Electronics
Applus+ IDIADA
Tarragona, Spain
jacint.castells@idiada.com

***Abstract*—Efforts in Europe for the deployment of Cooperative Intelligent Transport Systems (C-ITS) have seen a substantial increase in recent years, with various large-scale initiatives having been launched. For these systems two main communication methods are available: short-range ad-hoc local network direct vehicular (and infrastructure) communication based on the IEEE 802.11p standard, and communication via the cellular network. Substantial standardisation efforts have taken place for the first method, while cellular solutions for vehicular communication are still very fragmented. This paper presents a reference architecture for connecting local systems using short-range communication to an infrastructure server, which uses an information broker. For the client side, a topic structure inside the broker facilitates an efficient method for geocasting. On top of this functionality, a combination of certification, web tokens and transport layer security is developed to ensure security. A link of the architecture to the relevant business models is made to demonstrate that the two are compatible.**

Keywords—C-ITS, architecture, interoperability, security, privacy, road transport

I. Introduction

The aim of the development and deployment of Cooperative Intelligent Transport Systems (C-ITS) is to increase road transport safety and efficiency, and to reduce its environmental impact. Two main communication technologies are available for C-ITS: short-range direct communication, between vehicles and with road-side units (infrastructure), using ad-hoc and transient local networking based on the IEEE 802.11p standard, a variant for vehicular applications of the WLAN (Wireless Local Area Network) set of standards; and communication via the 3G/4G cellular network. Vehicular communication based on 802.11p is standardised in the US as WAVE (Wireless Access for the Vehicular Environment), and in Europe as ITS-G5. Standardisation of short-range vehicular communication has been ongoing for more than a decade, with the first version of SAE J2735 published in 2006 [1]. Vehicular short-range communication relies heavily on broadcasts. All stations in the vicinity will receive the same message. Therefore, having two systems in parallel in the same geographic area is challenging. The main problem with the standardized messages used in vehicular short-range communication is the amount of optional elements, and the risk of differences in interpretation of some elements. A number of projects, especially in Europe, are targeting this problem with so-called message profiling. The message profile contains extensive descriptions for interpretation of message elements and clear choices for optional fields. By harmonizing these profiles, true interoperability can be achieved.

For the use of cellular technology for vehicular communication, on the other hand, there are no efforts for standardization. For example, the Dutch Cooperative ITS Corridor project [2] states that it will not cover cellular-based systems, but acknowledges the need and stimulates service providers to work on this. The issue of interoperability for cellular-based communication seems less urgent due to the peer-to-peer nature. Two different systems (service applications using cellular communication) can easily exist in parallel in the same geographic area from a technical point of view. However, it is not practical and efficient from an end-user perspective to have many different services in parallel. Fragmentation of services can cause that end-users are required to run a different app on their cellular device in each city. Therefore, interoperability for cellular-based solutions is also within the R&D scope of the authors. Fragmentation of services is considered an important deployment barrier, next to unclear cost and benefits of the C-ITS services. Another important reason to give more attention to cellular-based solutions is the high investment costs for short-range communication devices, while most road users already own a cellular device.

Following an overview of C-ITS development and deployment efforts in Europe, the paper presents an open reference architecture for cellular-based vehicular communication with dedicated C-ITS servers on the internet. Subsequent sections focus on some important technical elements, such as the connections with local infrastructure, the geocasting facilities, and the security and authorization mechanisms. An approach for implementation of the architecture in view of relevant business models is proposed. Finally a conclusion is drawn.

II. Review of C-ITS Developments in Europe

C-ITS has been developed more than one decade in Europe. In 2005 the EC, under the FP6-IST funding scheme, launched three so-called Integrated Projects, targeting cooperative systems: SAFESPOT (Co-operative Systems for Road Safety "Smart Vehicles on Smart Roads"; focusing on the in-vehicle side and traffic safety) [3], CVIS (Cooperative Vehicle Infrastructure Systems; focusing on the infrastructure side and traffic efficiency) [4], and COOPERS (CO-OPerative SystEms for Intelligent Road Safety;

focusing on the domain of the road operator) [5]. Work was continued in two follow-up projects: PRE-DRIVE C2X (2008-2010) developed a detailed system specification and a functionally-verified prototype for I2V (infrastructure-to-vehicle) systems [6]; and DRIVE C2X (2011-2014) carried out a comprehensive assessment of cooperative systems through Field Operational Tests in Europe [7].

In 2012, the MOBiNET project [8] was launched, with the aim to deploy an open platform for offering a solution for a one-stop shop for Europe-wide (roaming and virtual ticketing) mobility services. In 2013-2017, the Compass4D project implemented three (IEEE-802.11p-based) cooperative services (Energy Efficient Intersections, Road Hazard Warning and Red Light Violation Warning) in seven European cities, based on a consolidated and interoperable architecture [9]. The German research project CONVERGE (2012-2015) created an ITS architecture (called Car2X Systems Network), which mainly focused on interoperability, economic viability, scalability, decentralisation and security [10]. DITCM was a Dutch program (2014-2015), which aimed to accelerate the deployment at large scale of C-ITS and Connected-Automated Driving, and developed a reference architecture [11]. Talking Traffic is an initiative addressing C-ITS deployment, set up as a collaboration between the Dutch Ministry of Infrastructure and the Environment, regional and local authorities, and national and international companies [12]. It explores new business models and focuses on the following use cases: In-vehicle signage, Road hazard warning, Priority at traffic lights, Traffic lights information, Flow optimization and In-vehicle parking information.

The Cooperative ITS-Corridor project [2], a collaboration between The Netherlands, Germany and Austria, was launched in 2015 and will be operational in 2018. This project has formed a cooperation with the French project SCOOP@F [13], the UK Department of Transport and the Flanders government in Belgium to form the InterCor project [14], in which ITS-G5 and/or 3G/4G communication solutions will be implemented for operation and evaluation of C-ITS services in The Netherlands, France, the UK and Belgium. The ongoing NordicWay project aims to implement C-ITS services in Finland, Sweden, Norway and Denmark using cellular communication (3G and LTE/4G). [15]. The NordicWay architecture uses a message queuing approach to transfer messages between Service Providers, Automotive Industry and Traffic Message Centres.

The C-MobILE project (2017-2020), funded by the European Union under the Horizon2020 Programme, aims to stimulate large-scale, real-life and interoperable C-ITS deployments across Europe, and in particular targets complex urban areas for all road users, including VRUs (Vulnerable Road Users). [16-18]

In parallel with these national and European C-ITS projects, the C-Roads platform [19] was launched in 2016. It currently has eight Member States as core members, and additional Member States as associated members, and it works on cross-border harmonization and interoperability.

It should be emphasised that C-ITS activities are, at the same time, also carried out outside Europe. For instance, in the US, the CVRIA (Connected Vehicle Implementation Architecture) Team, led by the ITS Joint Program Office, comprises the National ITS Architecture Team, the Standards Program Technical Support Services Team and the Policy Team (ITS JPO Policy Program and the Volpe National Transportation Systems Center). CVRIA is developed as the basis for identifying the key interfaces across the connected-vehicle environment, supporting in this way further analysis for the identification and prioritization of activities concerning the development of standards. The approach taken to develop the CVRIA involves operational concepts and the core system architecture developed for connected vehicle applications, existing national and international standards, and the existing national ITS architecture. The development of the system architecture was based on the fundamentals of the ISO/IEC/IEEE 42010:2011 standard, including steps to define data, messages, and the full environment in which the concerns of involved parties are satisfied. "CVRIA aims to become a framework for developers, standards organizations, and implementers to all use as a common frame of reference for developing the eventual systems"[20].

III. Local Infrastructure

Offering services via cellular from dedicated C-ITS servers on the internet starts with data acquisition from local sources. This can be challenging, because local infrastructure does not always offer interfaces to access relevant data. A good example for this is the Green Light Optimal Speed Advice (GLOSA) service. Traffic Light Controllers (TLC) have to send data about signal status and predictions of future status to the service provider. Some TLC control algorithms do not provide such data. To solve this, requires the algorithm to be replaced, reconfiguration of the system, and implementation of an interface. In other cases, just an interface needs to be implemented to access the data.

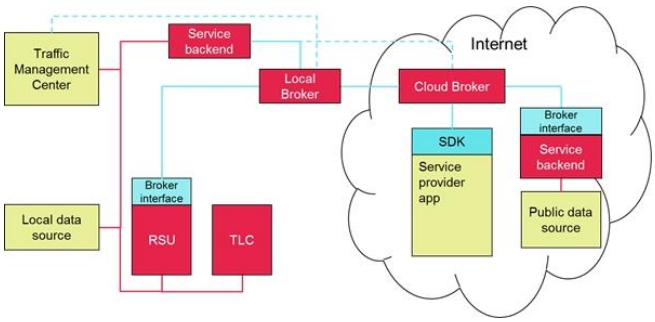

Fig. 1. Connection to local data sources.

Another consideration is the security of the local systems. TLCs need to be well protected against attacks and are therefore connected through a (Virtual) Private Network (VPN). Direct access from the internet to TLCs is therefore rarely possible and not recommended. This introduces the need to have a service backend inside the VPN to form a secure bridge between the open internet and the protected environment of the local infrastructure as is shown in Fig. 1.

The red lines indicate local protocols. Preferably, these

should be open protocols and a service backend should translate between a local protocol and the standardized protocol as is developed in the C-MobILE project for this purpose (light blue lines). This standardized interface should be the same at every deployment location. It uses the Message Queuing Telemetry Transport (MQTT) protocol [21], which is a lightweight protocol developed for Internet of Things (IoT) applications. It requires a central (server-based) information broker. Clients can publish messages to the broker on a topic, and subscribe to messages from the broker on certain topics. The topic structure is designed for geocasting facilities and is further discussed in the next section. The body of the MQTT messages is based on SAE standards [22]. Depending on the service, these can, inter alia, be of type DENM (Decentralized Environmental Notification Message), CAM (Cooperative Awareness Message), MAP or SPaT (Signal Phase and Timing). On top of these standards, the profiling efforts as conducted by the InterCor and C-Roads initiatives should also be used again.

The interfacing with the central MQTT broker on the internet can be implemented using different methods, which can be used in parallel. One is to publish data directly from the service into the broker. This is useful for a service backend that does not need to be inside the VPN of the TLCs. When there is only one service backend accessing the central broker, it is also most efficient to have the service backend to publish directly. On the other hand, when there are multiple services in the same VPN, it can be useful to have an additional local broker. From the service backend point of view, the interface is the same; the only difference is the configuration of the address of the broker. The local broker can synchronize with a central broker using standard MQTT broker synchronization methods, which also allow using multiple central brokers for scaling up the solution to handle more users. In a scenario where multiple services use the same data from the internet, the local broker can significantly reduce the traffic between the VPN and the central broker. For example, a service that extends the green light for disabled pedestrians needs to receive CAM messages, while a green-light-priority-for-trucks service also requires these. In this case, both local services can subscribe to the local broker, which only needs to receive the information from the internet once.

The same example about two services requiring CAM messages can even be extended a step further, by directly connecting the RSU (Road Side Unit) to the local broker. Equipment in the field often has limited bandwidth connections to the backend, so efficiency is of key importance. Another consideration is that services provided by an RSU are in the first place offered via short-range communication. This means that the RSU already has the facilities to carry out map matching, encoding and decoding of standardized short-range communication messages. Since the MQTT exchange carries the same messages, it is very easy for the RSU to publish or receive these directly to or from the local broker. This means the extra block of the broker interface on the RSU can be very simple. In the example of the CAM message, the RSU receives the message and map matches the sender in the Local Dynamic Map (LDM). The local services for the pedestrian green light and the truck priority are both connected to the LDM and respond the same as when the message was received through short-range communication. The only extra requirement is that the LDM should verify the age of the message in case the same message arrives through multiple channels.

End-users connect to the system with a service provider app that would usually combine multiple services in attractive bundles for consumers. This app connects to the central internet broker using a Service Development Kit (SDK), that provides similar facilities to on-board units (OBUs) for short-range communication.

The last two blocks in the architecture are the Traffic Light Controller (TLC) and Traffic Management Centre (TMC). These have locally standardized protocols, and either a service backend or an RSU can retrieve their data and connect to a central-broker on the internet. Since information from internet-based C-ITS servers is available in the broker, the TMC is also encouraged to retrieve its information there, but to realise this may take time and effort, due to the often proprietary nature of TMC systems.

IV. Geocasting Facilities

The topic structure in the central broker is a key element for the design of the geocasting solution. Broadcasting via short-range wireless communication is a form of intrisic geocasting, due to the limited reception range between 300 and 1000 meter, depending on local circumstances. Using internet-based communication via cellular requires a specific geocasting solution to ensure scalability. A system with data from all major European cities would result in an unmanageably large data stream when all SPaT messages are broadcast, as is the case for short-range communication. Geocasting should ensure that only messages which are relevant to the location of the receiver are forwarded.

Several solutions have been developed for geocasting. A common solution is based on the results of the EU-funded GeoNet project (2008/2009, FP7) [23] and implemented by SCOOP@F. The GeoNet protocol was initially developed as a multihop protocol for short-range communication. In this protocol the destination location is encoded in the message header. Routers in the backbone network check these messages and determine where to forward based on the destination. A major disadvantage of this method is that changes to the routers of the backbone network are required, which in this case is the internet and the access points of the cellular network. If in the future geocast is offered for example as part of a next generation cellular network, then it could be used. A small disadvantage is that some processing is required at every router in the network, but this overhead can be minimized with the use of extended Domain Name Service (eDNS) as described in [23]. An advantage is that the network load on the backbone network can be reduced when there are many subscribers to the same streams.

A second possible solution is use of CAM messages of vehicle nodes in combination with Geographic Information Systems (GIS) databases like PostGIS. This was for example the first implementation of the geocasting solution in MOBiNET [8]. There are, however, two disadvantages to this solution. The first is that by uploading the CAM messages, it introduces an unnecessary privacy risk for users

who only want information from the system, and shall not be required to upload information to a service. The second is that the continuous exchange of CAM information results in many queries to the GIS database, which makes the system less efficient for scaling up.

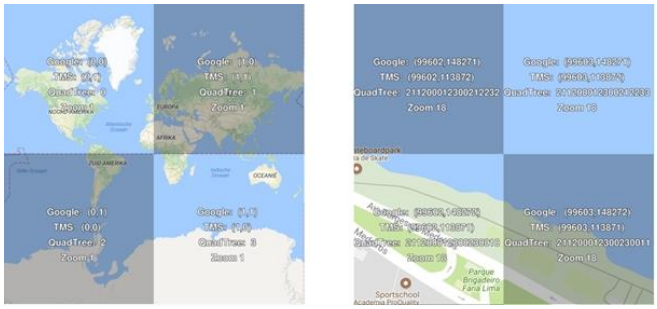

Fig. 2. Tiling concept, with zoom level 1 example on the left and zoom level 18 on the right. Image acquired using [25].

A better solution was found by the CONVERGE project [24], which introduced a tiling concept, illustrated in Fig. 2. Using this method, users do not need to send CAM data with their location. A vehicle simply registers once (not CAM based) and receives the edges of its current map-tile. Once it leaves the tile, it contacts the server again for new coordinates. In this case, a broker can still try to track an end-user, but with smart use of pseudonyms, for example changed every time a new tile is requested, this task becomes impossible on a system with many users. Additionally, not using CAM data has the added benefit of not having extra data that may identify the user, like vehicle length and width. Users automatically receive the data relevant for their tile. The solution is also very scalable; the broker only has to look at the list of subscribed clients when forwarding a message. No computationally intensive geographic calculations are required for each message. Data sources that have a fixed dissemination area, like SPaT data, can simply have a table to which tiles they should publish. The only drawback is that the dissemination is limited to a combination of square tiles, which can lead to a slightly larger dissemination area than initially intended.

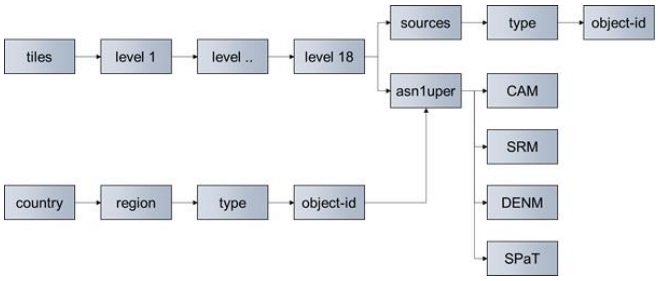

Fig. 3. Schematic representation of topic structure.

For pan-European C-ITS applications, the standard will go one step further in the efficiency of the tile system. In this solution not the coordinates of tile edges are exchanged between the vehicle and the broker, but the Google XYZ standard as described in [25] is used. This enables any system to calculate the relevant geographic tiles for its current location. Fig. 2 shows the level 1 tiles on the left, the first level divides the world in 4 squares. For level 2 each level 1 tile is again divided into 4 squares. The right side of the figure shows level 18 tiles. Because of the Mercator projection used for the map, the size of a tile differs with the latitude. The closer to the poles, the smaller the tiles. For ITS applications, the level 18 is a good trade-off between having enough granularity to avoid receiving irrelevant messages and not having to update the tile subscriptions too often. This results in a quadtree structure (tile repeatedly divided in four equal parts) with a lowest-level tile size of 65m in Oulu in northern Finland, which is at 65 degrees latitude. At the equator, the tile size is 154m.

There are two ways of referencing the map tiles, an (x,y) coordinate or the quadtree. The latter is chosen for the C-MobILE system, because it allows structures that span multiple zoom levels. In Fig. 2, both methods are printed inside the tile. As can be seen, all tiles on the right start with a "2" in the quadtree, which indicates that the tile is inside the quadtree tile "2" in the level 1 view on the left. This is correct because the right side shows a part of Rio de Janeiro in Brazil, which is in the southwest quadrant of the earth (22 degrees south and 43 west).

The schematic topic structure is presented in Fig. 3. As an example if a road user wants to receive all data for the bottom-right level 18 tile of Fig. 2, a subscription to the following topic should be made:

/tiles/2/1/1/2/0/0/0/1/2/3/0/0/2/3/0/0/1/1/#

By using single-level wildcards a user can effectively subscribe to a level 15 tile as follows:

/tiles/2/1/1/2/0/0/0/1/2/3/0/0/2/3/0/#

The publisher, however, still has to publish on all individual tiles unless it knows all clients are listening on a specific level. When a message has to be published on a level 15 tile, this does not mean that the data will be copied to 64 tiles. This is where the "sources" element of the topic structure comes in. This enables the data source to publish a reference on those 64 tiles and not the actual data. The reference can in theory contain any topic structure for the actual data, but it is recommended to follow the following example:

Topic:
/tiles/2/1/1/2/0/0/0/1/2/3/0/0/2/3/0/0/1/1/sources/intersections/RSU701

Message body: /nl/helmond/intersections/RSU701/

This structure allows for easy identification of the data source when managing the broker. Since the actual data follows the same standardized messages as are used for short-range communication, the actual data is published with a topic containing "asn1uper", as this is the encoding mechanism used for those messages (ASN.1 UPER). When the user receives the message of the data source from its tile subscription, it should then add a subscription to the following data topic:

nl/helmond/intersections/RSU701/#

Through which short-range communicationmessages will be received on these two topics for a GLOSA service:

nl/helmond/intersections/RSU701/asn1uper/map

nl/helmond/intersections/RSU701/asn1uper/spat

The other way around, vehicles can publish for example CAM and SRM messages to the broker for services that

require actions at the infrastructure side.

V. Security and Authorization

Security and authorization play a major role in any communication system, but is even more important in the field of C-ITS due to the high number of participants and the critical role that transportation plays in our society. Even more important, security is a necessity for safety of road users. To ensure security for C-ITS, the European Commission mandated the C-ITS Platform with the definition of a security policy [26]. This policy regulates the use of certificates mainly associated with short-range communication of ITS Stations (short-range-communication nodes with C-ITS functionality, like vehicles, road-side units and pedestrians). Certificates are an essential part of state-of-the-art security mechanisms. The concept defined in the respective certificate policy is applicable for communication to and between traffic participants, mainly using short-range communication.

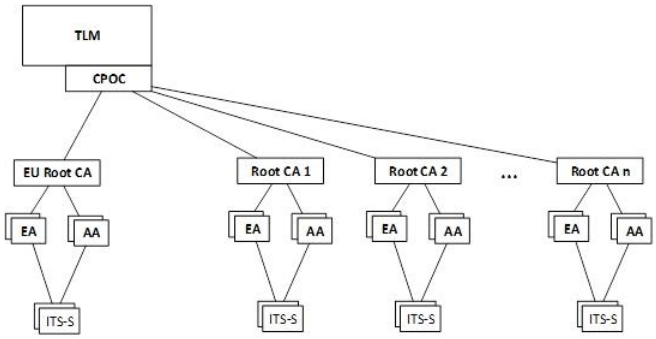

Fig. 4. C-ITS trust model. (Source: [26])

As shown in Fig. 4, the new certificate policy enhances the trust model, which has been defined by ETSI [27]. In the original trust model, the root Certificate Authority (CA) was the trust anchor at the top. As this was identified as hard to realize for the whole of Europe, an additional entity was introduced, the Trust List Manager (TLM). This entity manages the European Certificate Trust List (ECTL), which contains the certificates of all trusted root CAs of Europe. By doing this, an ITS Station (ITS-S) which receives information signed with a certificate originating from another Root-CA can verify the trust by checking if this root CA is present in the ECTL. Additionally, the TLM defines a policy, which includes certain requirements for the various entities below it, to ensure security. These requirements define the algorithms to be used and the processes to be followed, in accordance with specific requirements for a CA, an Authorization Authority (AA) or an Enrollment Authority (EA).

For direct communication, the trust model is used for all messages that are standardized for short-range communication. However, the security profiles associated with these short-range communication mechanisms are insufficient to use with internet-based services. This can be the case because those services require other message formats than already defined, do not use message-based communication at all, or require restriction of access to broadcast information to enable a certain business model. Therefore, an additional type of security mechanism is needed, which is suited to secure those communication types, especially between different entities in the backend network, where Denial of Service (DoS) attacks are likely to occur. For this, we foresee the use of JSON Web Tokens (JWT) [28], as a means of authorization, and the use of (mutual) Transport Layer Security (TLS) [29] to secure communication against eavesdropping, malicious modification, and man-in-the-middle-attacks.

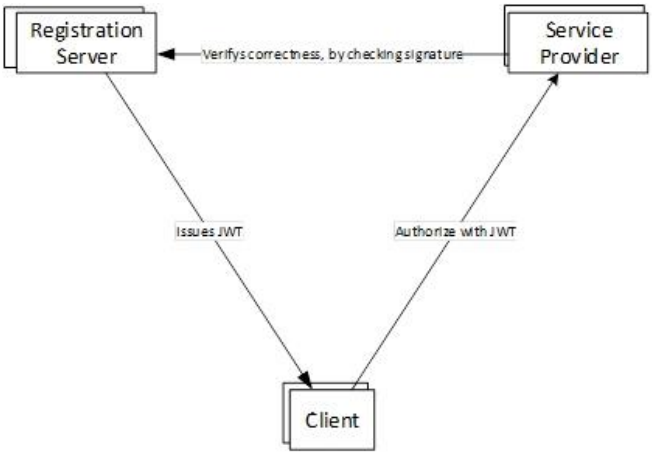

Fig. 5. Outline of the JWT authorization process.

Fig. 5 shows the basic principle of the JWT authorization process. A client is associated to a Registration Server. There can be multiple Registration Servers, e.g. for different manufacturers, organizations and states. From this Registration Server, the client can request tokens, which prove, that the client indeed is registered with this registration server. Those tokens can also contain additional claims, e.g. that the client has deposited a certain amount of money and a list of services it is allowed to use. This token, can then be used by the client to authorize itself with a service provider. The service provider can use the public key of the registration server to verify that the signature of the JWT is correct, thus ensuring that the claims in the token are backed by the Registration Server. This has the additional advantage that the Service Provider does not need to know the identity of the client to decide if it has permission to use the service. In this process, it is necessary that the communication links between all entities are secured, to prevent malicious third parties from obtaining the token.

VI. Business Model

The proposed open architecture provides a secure, pragmatic, cost-effective, and easy to be operated approach for authorities to implement C-ITS services. The open topic structure (see Fig. 3) allows operation by neutral brokers, and avoids vendor-lock-in situations. If Company A supplies a local or an internet broker, which is publicly accessible, then Company B only needs authorization to publish messages to be able to use the broker for a new service, and no extra effort is required from Company A. In case open protocols wold not be used, implementation of an interface between Company A and Company B would be

required. When referring back to Fig. 1, it is even possible to have multiple brokers of different vendors in parallel as long as the service provider apps are configured to connect to them through the SDK. Since the service providers for the apps need to make contracts with internet broker providers, there is also the possibility to let the internet brokers compete for these contracts through the free market.

The security model presented in Fig. 5 also enables business models where users have to pay for certain data that would otherwise be broadcast through short-range communication. The JWT token can indicate a user has the right to receive this data for a number of services. In this case, the service provider that supplies the app to the end user (app provider) has a contract with the provider of the central internet broker. This contract can have several forms, but it is recommended to charge per message (or block of thousand messages) sent to an end-user. This way the broker provider only has to check whether an end-user is allowed to use the service and do accounting based on volume of messages. The app provider is not involved in the real-time data exchange, but only receives an account of the use. However, it is the responsibility of the app provider to market the services in an attractive way to end-users, for instance making bundles for unlimited use of a set of services paid per month.

The architecture with JWT also allows an app provider to buy services at different internet-broker providers. This is especially interesting for roaming, as different countries could have different providers. The JWT token does not only contain the authorization details but also the address of the internet-broker provider. Therefore, when a user is about to cross the border, it can request a new JWT token and already set up the new connection. Having these two connections open in parallel, allows for an uninterrupted service when crossing the border.

CONCLUSION

The paper demonstrates a scalable architecture to connect local IEEE-802.11p-based short-range communication systems to an internet-based service. An important aspect is the use of the MQTT broker, which separates information providers from users of their information. This adds security by having only one connection to the outside world, shared by many services. On the other hand, the services providing information do not need to know the location of the end-users, which helps ensuring privacy. The broker-centric architecture ensures scalability, as the amount of end-users has no impact on the services providing information. More brokers can be added and synchronized to each other to distribute the load.

The presented tile-based topic structure shows a clear improvement over previous work. The tiles reduce communication load and eliminate the need for regular position updates of end-users and location comparison calculations. Using a standardized method for tile edge calculation even eliminated the need to communicate tile locations. Lastly, this open topic structure also allows for easy extension of the amount of services without requiring effort of the broker supplier. This prevents vendor-lock-in situations, in which the broker provider might exploit its position of being at the central point of the ITS system.

The proposed security model makes optimal use of the standardization efforts already made for short-range communication messages and security. It also enables to use the same mechanisms for validating identity and authorization at local Road Side Units independent of the communication channel used for the message. However, the paper also demonstrated that for internet communication, additional security is required. JWT and TLS protect against DoS, man-in-the-middle, eavesdropping and malicious modification attacks. At the same time, JWT enables various business models, in which app providers can create attractive bundles of C-ITS services, while relieving the broker provider of the specific concerns associated with recruiting end-users. Lastly, roaming support is also very straightforward using the JWT concept.

The architecture elements presented in this paper should enable service providers to both compete and complement each other in an efficient way at an international scale for providing attractive and seamless services to end users.

ACKNOWLEDGMENT

The paper presents some preliminary results of the EU-funded project C-MobILE (Accelerating C-ITS Mobility Innovation and depLoyment in Europe). C-MobILE is funded by the European Union’s Horizon 2020 Research and Innovation Programme under grant agreement No. 723311. The authors especially thank C-MobILE partners that have contributed to the C-ITS architecture development.

REFERENCES

[1] SAE International, Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE J2735, 2016.

[2] A. Paier, "The end-to-end intelligent transport system (ITS) concept in the context of the European cooperative ITS corridor", IEEE MTT-S International Conference on ICMIM, Heidelberg, Germany, 2015.

[3] SAFESPOT Consortium, "Annex I - Description of Work", SAFESPOT Consortium, Brussels, 2005. (restricted)

[4] CVIS Consortium, "Annex I - Description of Work", CVIS Consortium, Brussels, 2005. (restricted)

[5] COOPERS Consortium, "Annex I - Description of Work", COOPERS Consortium, Brussels, 2005. (restricted)

[6] PRE-DRIVE consortium, "Annex I - Description of Work, PRE-DRIVE C2X (Preparation for driving implementation and evaluation of C2X communication technology)", PRE-DRIVE Consortium, Brussels, 2008. (restricted)

[7] DRIVE C2X consortium, "Annex I - Description of Work", DRIVE C2X Consortium, Brussels, 2011. (restricted)

[8] U. Noyer, T. Schlaug, P. Cercato, and L. Mikkelsen, "MOBiNET – architecture overview of an innovative platform for European mobility services", World Congress on Intelligent Transport Systens, Bordeaux, 2015.

[9] G. Alcaraz, S. Tsegay, M. Larsson, G. Vernet, E. Koenders, J. Vreeswijk, F. Ophelders, P. Mathias, F. Couly, J. Fernandez, A. Perpey, M. Feringa, and M. Annoni, "Deliverable D2.2 Overall reference architecture (Ver 2.0)", Compass4D Consortium, Brussels, 2015

[10] CONVERGE Consortium, COmmunication Network VEhicle Road Global Extension, Saarbrücken. [Accessed 12-04-2018] https://converge-online.de

[11] M. van Sambeek, O. Turetken, F. Ophelders, R. Eshuis, T. Bijlsma, K. Traganos, and P. Grefen, "Towards an architecture for cooperative ITS applications in The Netherlands (Ver. 1.0)", DITCM. 17 April 2015.

[12] W. Vandenberghe, "Partnership Talking Traffic", Dutch Ministry of

Infrastructure & Environment, ITS Belgium Congress, 2017 [Accessed 12-04-2018]. www.partnershiptalkingtraffic.com

[13] H. Aniss, "Overview of an ITS Project: SCOOP@F", Springer International Publishing, pp. 131–135, 2016.

[14] European Commission, InterCor 2015-EU-TM-0159-S North Sea - Mediterranean Corridor. [Accessed 12-04-2018] https://ec.europa.eu/inea/sites/inea/files/fiche_2015-eu-tm-0159-s_final.pdf; http://intercor-project.eu/

[15] European Commission, NordicWay 2014-EU-TA-0060-S. [Accessed 12-04-2018] https://ec.europa.eu/inea/sites/inea/files/fiche_2014-eu-ta-0060-s_final.pdf

[16] M. Lu, R. Blokpoel, M. Pillado, and G. Somma, "ICT Infrastructure-based cooperative and connected systems for intelligent European road transport", European Congress on Intelligent Transport Systems, Strasbourg, 2017.

[17] C-MobILE Consortium, "Deliverable D3.1 High-level reference architecture", C-MobILE Consortium, Brussels, 2018. (restricted)

[18] C-MobILE Consortium, "Deliverable D3.2 Medium-level concrete architecture and services definition", C-MobILE Consortium, Brussels, 2018. (restricted)

[19] C-Roads, Harmonised C-ITS specifications for Europe, 14 September 2017. [Accessed 12-04-2018] www.c-roads.eu
The Platform of Harmonized C-ITS Deployment in Europe.

[20] CVRIA - Connected Vehicle Reference Implementation Architecture. [Accessed 12-04-2018]
http://local.iteris.com/cvria/html/about/about.html

[21] A. Banks, and R. Gupta, MQTT Version 3.1.1, OASIS Standard, 29 October 2014.

[22] SAE J2735, Dedicated Short Range Communications (DSRC) Message Set Dictionary, 30 March 2016.

[23] T. Fioreze, and G. Heijenk, "Extending DNS to support geocasting towards VANETs: a proposal", IEEE Vehicular Networking Conference, USA, December 2010.

[24] M. Fünfrocken, et al., "Deliverable D4.3 Architecture of the Car2X systems network", CONVERGE Consortium, 30 January 2015.

[25] K.P. Přidal, Maptiler. [Accessed 12-04-2018] www.maptiler.org/google-maps-coordinates-tile-bounds-projection/

[26] European Commission, "Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems (C-ITS)", C-ITS Platform Phase II, Brussels. [Accessed 12-04-2018] https://ec.europa.eu/transport/sites/transport/files/c-its_certificate_policy_release_1.pdf

[27] ETSI, ETSI TS 102 940 V1.2.1 "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management", June 2012. [Accessed 12-04-2018] www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.02.01_60/ts_102940v010201p.pdf

[28] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, May 2015. DOI 10.17487/RFC7519

[29] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, 2008. DOI 10.17487/RFC5246