Paper ID #

Open cloud architecture for connected C-ITS services

Robbin Blokpoel[1*], Manuel FünFrocken[2], Meng Lu[3]

1. Lead Researcher, Dynniq, The Netherlands, robbin.blokpoel@dynniq.com

2. Team leader, HTW Saarbrucken, Germany

3. Strategic innovation manager, Dynniq, The Netherlands

Abstract

Deployment effort of Cooperative Intelligent Transport Systems (C-ITS) is increasing rapidly in the last years with various large-scale initiatives being launched. These systems can use two main methods of communication: Standardization for Dedicated Short Range Communication (DSRC) and cloud based cellular communication. The first has seen a lot of effort for standardization, while the latter is still very fragmented. This paper presents a reference architecture how local systems with DSRC can be connected to a cloud server, which uses an information broker. For the client side, a topic structure inside the broker is presented that 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. Lastly, a link of the architecture to the business models is made to demonstrate the compatibility between the two.

Keywords:

C-ITS, Architecture, cloud-based connectivity.

Introduction

Deployment effort of Cooperative Intelligent Transport Systems (C-ITS) is increasing rapidly in the last years with various large-scale initiatives being launched. An example is the Cooperative ITS-Corridor project [1], which started in 2015 and will be operational in 2018 in The Netherlands, Germany and Austria. This project has formed a cooperation with SCOOP@F [2], the UK Department of Transport and the Flanders government in Belgium to form the InterCor project [3], covering a total of six European countries. In parallel the C-Roads platform [4] has 12 core and associated member states at the moment and works on cross-border harmonization and interoperability.

Two main communication technologies are used for C-ITS: Dedicated Short Range Communication (DSRC) using IEEE 802.11p standards (also known as G5) and cloud-based 3G/4G cellular communication. Standardization for DSRC has been going on for more than 10 years, with the first version of SAE J2735 published in 2006 [5]. The nature of DSRC, which relies heavily on broadcasts, requires standardization. All stations in the vicinity will receive the same message, so having two systems in parallel in the same geographic area is challenging. The main problem with the standardized messages used on DSRC is the amount of optional elements and possibilities of different interpretation of some elements. The previously introduced large-scale projects are targeting this problem with so-called profiling of these messages. The message profile contains extensive descriptions about interpretation of message elements and clear choices on optional fields. By harmonizing these profiles true interoperability can be achieved.

For cloud-based communication, on the other hand, there are no efforts for standardization. For example, the Dutch Cooperative ITS Corridor project, which is related to InterCor, states in [6] that it will not cover cloud-based systems, but acknowledges the need and stimulates other services providers to work on this. This issue of interoperability for cloud-based communication seemed less urgent due to the peer-to-peer nature. Two different systems 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 end-users to have to run a different app on their cellular device for each city. Therefore, the C-MobILE project has decided to work on interoperability for cloud-based solutions as well. This objective of the project was confirmed at stakeholder workshop in November 2017. Fragmentation of services was identified as the second most important deployment barrier by the audience and was only defeated by unclear cost and benefits of the services. Another important reason to give more attention to cloud-based solutions is the high investment costs for DSRC devices, while most road users already own a cellular device.

This paper will present the open architecture for cloud-based communications that resulted from the work in C-MobILE. The paper is organized in different sections that first cover an overall architecture, followed by dedicated sections for important elements. These include the connections with local infrastructure, the geocasting facilities, and the security and authorization mechanisms. The last sections are dedicated to the business models supported by the architecture and a conclusion.

Local infrastructure

Offering services through the cloud 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 this data, which means the algorithm has to be replaced and a new configuration is required. In other cases, an interface has to be implemented to access the data.

Another consideration is the security of the local systems. TLCs need to be well protected against attacks and are therefore connected with a (Virtual) Private Network (VPN). Direct access from the cloud 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 Figure 1.

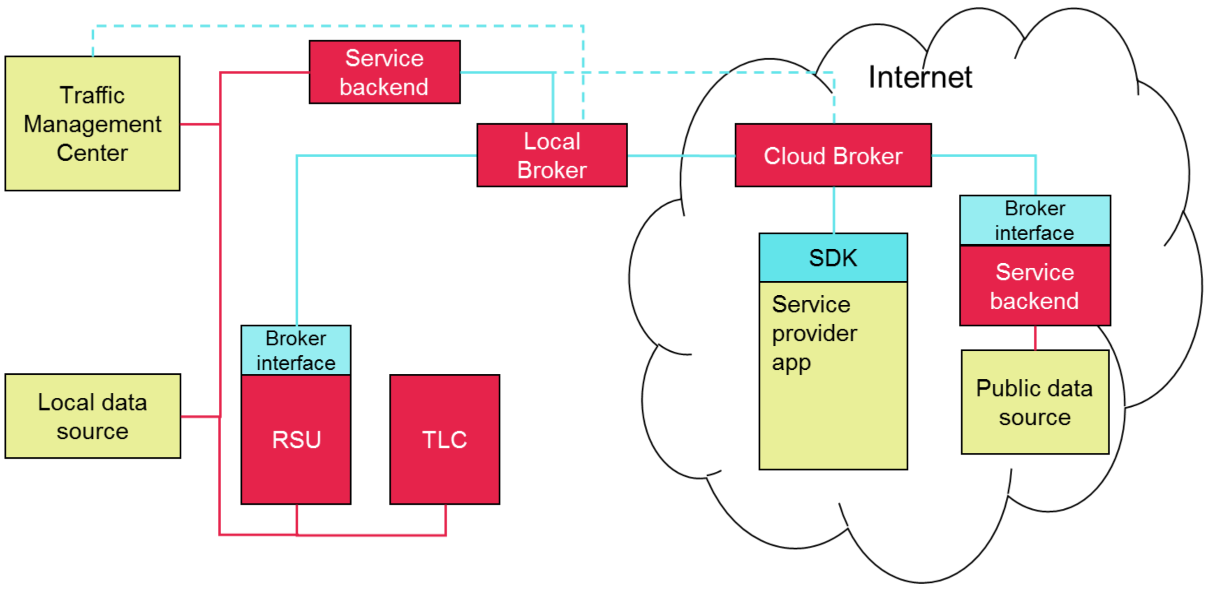

Figure 1 - Connection to local data sources

The red lines indicate local protocols. Preferably, these should be open protocols and a service backend should translate between the local protocol and the standardized C-MobILE protocol (light blue lines). This standardized interface should be the same at every deployment location. It uses the MQTT protocol [8], which is a lightweight protocol developed for Internet of Things (IoT) applications. It has a central information broker, to which clients can publish or subscribe to messages of certain topics. The topic structure is designed for geocasting facilities and is further discussed in the next section. The body of the MQTT messages exchanged is based on SAE standards [9]. Depending on the service, these can be DENM, CAM, MAP, SPaT, etc. On top of these standards, the profiling efforts of aforementioned InterCor and C-Roads initiatives should also be used again.

The interfacing with the MQTT Cloud broker 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. When there is only one service backend accessing the cloud 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 cloud broker using standard MQTT broker synchronization methods, which also allow using multiple cloud brokers for scaling up the solution to handle more users. In a scenario where multiple services use the same data from the cloud, the local broker can significantly reduce the traffic between the VPN and the cloud 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 them. In this case, both local services can subscribe at 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 go a step further by directly connecting the RSU 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 also offered via DSRC communication. This means the RSU already has the facilities to carry out mapmatching, encoding and decoding of standardized DSRC messages. Since the message body of the MQTT messages also contains these same messages, it is very easy for the RSU to publish or receive them directly 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 mapmatches 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 DSRC. 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 cloud broker using a Service Development Kit (SDK), that provides similar facilities to DSRC On Board Units (OBUs).

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 the C-MobILE cloud on the other side. Since information from the cloud is available in the broker, the TMC is also encouraged to retrieve its information there, but this may take time due to the often proprietary nature of these systems.

Geocasting facilities

As already mentioned the topic structure in the MQTT broker is a key element for the design of the geocasting solution. Broadcasting on DSRC is a form of geocasting due to the limited reception range between 300 and 1000 meters depending on the local circumstances. Using cloud based communication, a different solution is required 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 broadcasted, as is the case with DSRC. Geocasting should ensure only messages that 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 GeoNet [10] project and implemented by SCOOP@F, which started as a multihop protocol for DSRC. 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 5G 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 [10]. 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 usage 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 [11]. 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 that only want information from the system, i.e. they do not need 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 to scale up.

A better solution was found by the CONVERGE project [12], which introduced a tiling concept, illustrated in Figure 2. Using this method, users do not need to send CAM data with their location repeatedly. A vehicle simply registers once and receives the edges of its current map-tile. Once it leaved 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 usage of pseudonyms, for example 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.

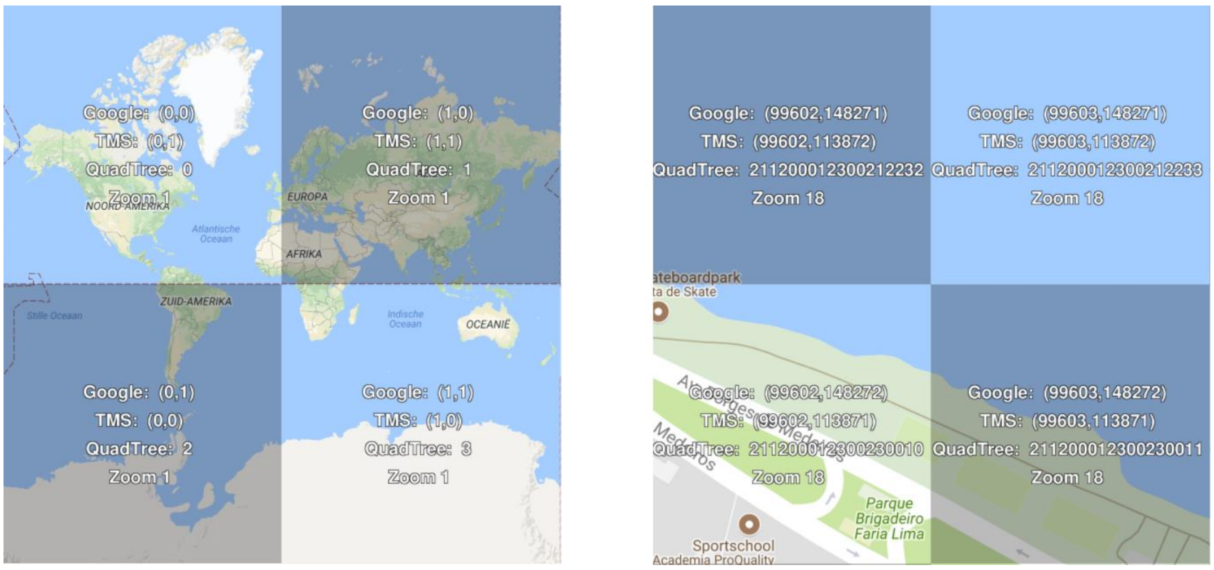

Figure 2 - Tiling concept, with zoom level 1 example on the left and zoom level 18 on the right. Image acquired using [13].

For C-MobILE the standard will go one step further in the efficiency of the tile system. In this solution the tile edges are no longer exchanged between the vehicle and the broker, but the Google XYZ standard as described in [13] is used. This enables any system to calculate the relevant geographic tiles for its current location. Figure 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 tile size of 65m in Oulu, 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 Figure 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 indicate 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).

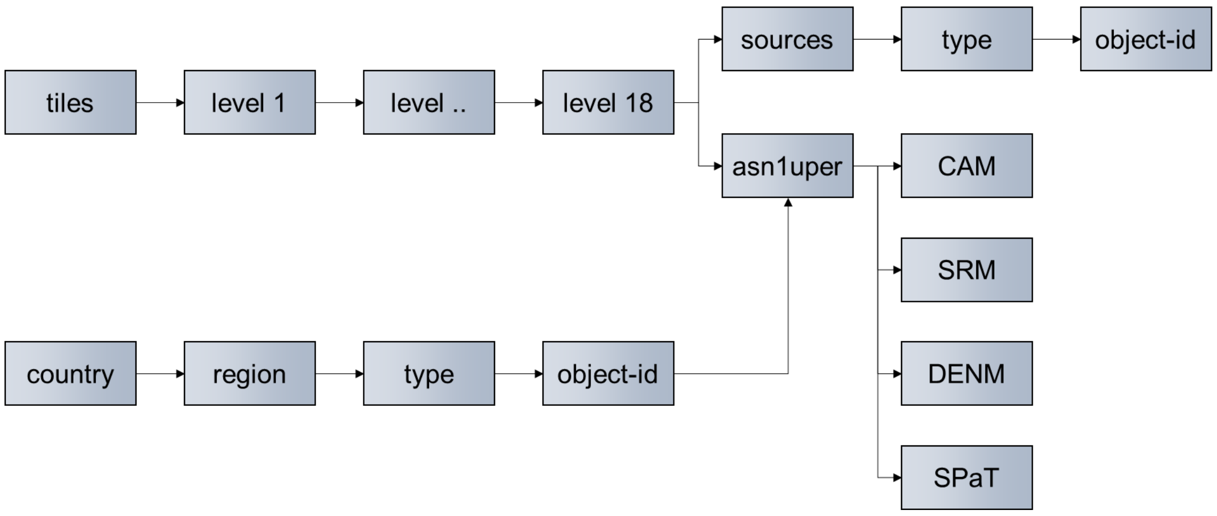

Figure 3 - Schematic representation of topic structure

The schematic topic structure is presented in Figure 3. As an example if a road user wants to receive all data for the bottom-right level 18 tile of Figure 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 $4^3 = 64$ 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 for DSRC, the actual data is published with a topic containing “asn1uper”, as this is the encoding mechanism used for those messages. 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 DSRC messages 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.

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 stakeholders and the critical role that transportation plays in our society. Even more important, security is a necessity to ensure safety of road users. To ensure security for C-IT'S the European Commission mandated the C-ITS Platform with the definition of a security policy [14]. This policy regulates the use of certificates mainly associated with DSRC and the communication of ITS Stations, like vehicles or pedestrians. Certificates are an essential part of state-of-the-art security mechanisms. The concept defined in the certificate policy is applicable for communication to and between traffic participants, mainly using DSRC.

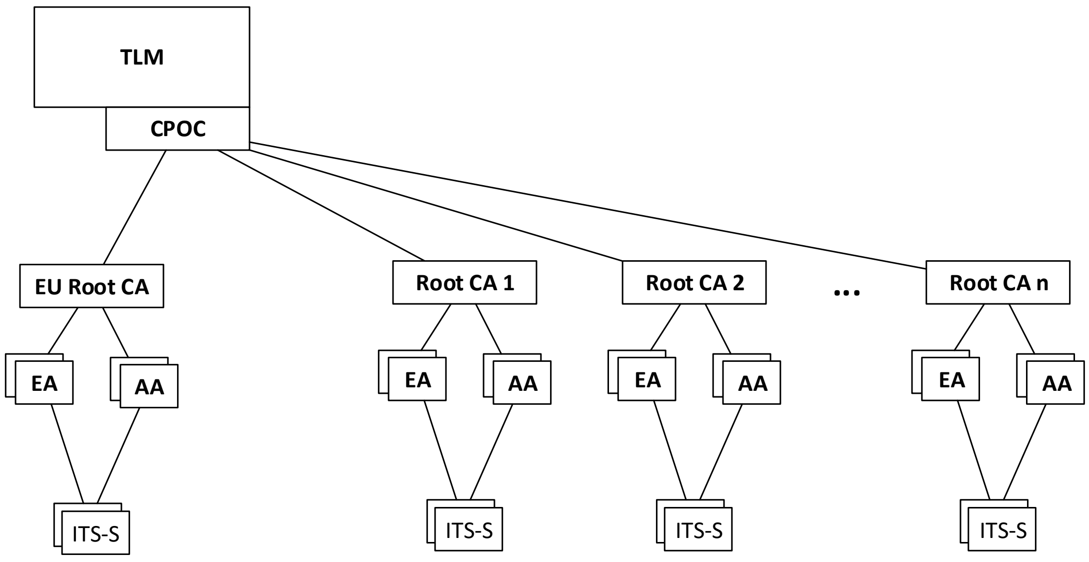

Figure 4: C-ITS trust model. (Source: [14])

As shown in Figure 4, the new certificate policy enhances the trust model, which has already been defined by ETSI [**Error! Reference source not found.**]. In the original trust model, the root Certificate Authority (CA) was the trust anchor at the top. As this has been identified as hard to realize for whole Europe, an additional entity has been introduced, the Trust List Manager (TLM). This entity manages the European Certificate Trust List (ECTL), which contains the certificates of all trusted root CA's 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 defines certain requirements for the various entities below it, to ensure security. These requirements define which algorithms to use, define the processes, which need to be followed up to specific requirements for the facilities of a CA, Authorization Authority (AA) or Enrollment Authority (EA).

For connected communication, the trust model is used for all messages that are standardized for DSRC. However, the security profiles associated with those DSRC mechanisms are insufficient to use with internet-based services. This can be because those services require other message formats as already defined, do not use message-based communication at all or require restriction of access to broadcasted 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) [0], as a means of authorization and the use of (mutual) Transport Layer Security (TLS) [0] to secure communication against eavesdropping, malicious modification, and man-in-the-middle-attacks.

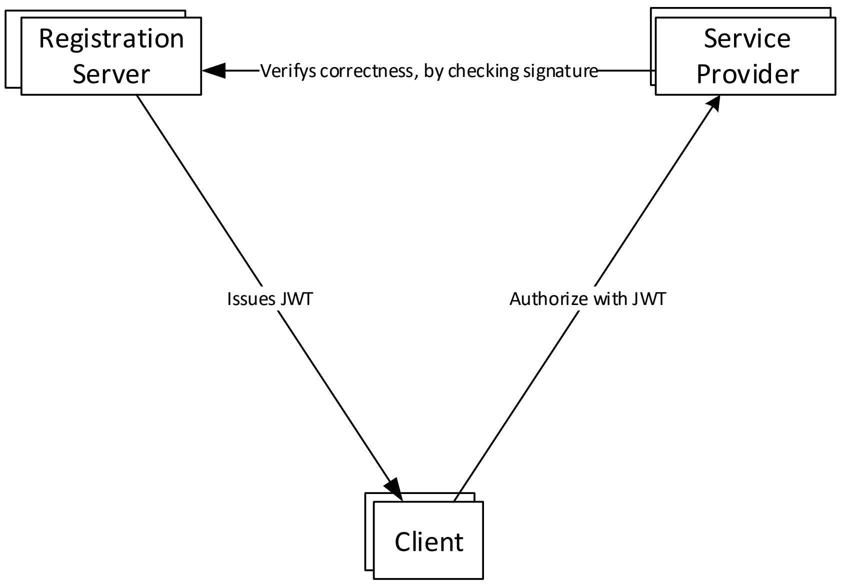

Figure 5: JWT for authorization

Figure 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 is allowed to use the service. In this process, it is necessary that the communication links between all entities be secured, to prevent malicious third parties from obtaining the token.

Business model

The open topic structure of Figure 3 allows for neutral brokers and avoids vendor-locked situations. If company A supplies a local or cloud 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. No extra effort is required from company A, as would be the case without open protocols, because that would require implementation of an interface between A and B. When referring back to Figure 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 cloud broker providers, there is also the possibility to let the cloud brokers compete for these contracts through the free market.

The security model presented in Figure 5 also enables business models where users have to pay for certain data that would otherwise be broadcasted through DSRC. 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 cloud provider. 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 cloud 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 usage. However, it is their responsibility 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 cloud 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 cloud 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 uninterrupted service experience when crossing the border.

Conclusion

The paper has demonstrated a scalable architecture to connect local DSRC systems to the cloud. An important aspect is the use of the MQTT broker, which separates information providers from their receivers. 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 the 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 tile-based topic structure presented 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 about the 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-locked situations where the broker provider can exploit its position of being at the central point of the ITS system.

The security model presented in the paper makes optimal use of the efforts already made for DSRC message and security standardization. This also enables to use the same mechanisms for validating identity and authorization at local RSUs independent of the communication channel used for the message. However, the paper also demonstrated that for cloud 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 where app providers can create attractive bundles of C-ITS services, while relieving the cloud 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 an efficient international ecosystem where service providers can compete and complement each other to give a seamless and attractive experience to end-users.

Acknowledgement

The C-MobILE project has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 723311

References

1. 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.
2. H. Aniss, Overview of an ITS Project: SCOOP@F, Springer International Publishing, pages 131–135, 2016.
3. http://intercor-project.eu/, last visited 23-11-2017.
4. C-Roads project, Harmonised C-ITS specifications for Europe, project publication, 14-09-2017.
5. SAE International, Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE J2735, 2016.
6. E. van der Walle, Description of the System Concept, Cooperative ITS Corridor Project deliverable, 2016.
7. M. Lu, R. Blokpoel, M. Pillado, G. Somma, ICT Infrastructure-based cooperative and connected systems for intelligent European road transport, ITS European congress, Strasbourg, France, 2017.
8. A. Banks, R. Gupta, MQTT Version 3.1.1, OASIS Standard, 29th October 2014.

9. SAE J2735: “Dedicated Short Range Communications (DSRC) Message Set Dictionary”. 2016-03-30.
10. T. Fioreze, G. Heijenk, Extending DNS to Support Geocasting Towards VANETs: A Proposal, IEEE Vehicular Networking Conference, USA, December 2010.
11. U. Noyer, T. Schlaug, P. Cercato, L. Mikkelsen, MOBiNET – architecture overview of an innovative platform for European mobility services.
12. Manuel Fünfrocken, et al., Architecture of the Car2X Systems Network, CONVERGE project deliverable D4.3, 30-01-2015.
13. K. Přidal, http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/, last visited 4-12-2017.
14. C-ITS Platform Phase II, “Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems (C-ITS)”, https://ec.europa.eu/transport/sites/transport/files/c-its_certificate_policy_release_1.pdf, Release 1, 2017-06, last visited 12-12-2017.
15. Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, 2015-05, https://www.rfc-editor.org/info/rfc7519.
16. Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, 2008-08, https://www.rfc-editor.org/info/rfc5246
17. ETSI TS 102 940 V1.2.1, “Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management”, June 2012. http://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.02.01_60/ts_102940v010201p.pdf