



Information and Communication Technologies

Collaborative Project

“V-Charge”

Autonomous Valet Parking and Charging for e-Mobility

Grant Agreement Number 269916



|                     |                                                          |
|---------------------|----------------------------------------------------------|
| Deliverable Number  | 6.7                                                      |
| Deliverable Name    | Second version interaction, communication and management |
| Dissemination Level | RE                                                       |
| WP Number           | 6                                                        |
| Date of Preparation | May, 30 <sup>th</sup> , 2014                             |
| Date of Submission  | June, 2 <sup>nd</sup> , 2014                             |
| Editor/Beneficiary  | Rottmann, Timpner, Wolf / TUBS                           |

**EXECUTIVE SUMMARY:**

This document describes the current state of the V-Charge Server components and the user interfaces.

Section 1 describes the progress on the general server components, Section 2 gives information on new algorithms for high density placement of automated vehicles. In Section 3, the updated version of the HMI device is shown. Section 4 deals with the measurement of the network quality. In Section 5, we are arguing why we stay with IEEE 802.11 a/b/g/n instead of using 802.11p. In Section 6, details on the overhead added by security are given. In the last Section 7 of the document, impressions of the real world deployment of the server components and the complete infrastructure are shown.

## Table of Contents

|                                                     |    |
|-----------------------------------------------------|----|
| Table of Contents .....                             | 3  |
| 1. Extensions of Server components .....            | 4  |
| 1.1. RoadGraph .....                                | 4  |
| 1.2. Debugging features .....                       | 4  |
| 2. Parking spot management .....                    | 5  |
| 2.1. High density parking algorithm .....           | 5  |
| 2.1.1. k-Stacks .....                               | 6  |
| 3. User interfaces .....                            | 7  |
| 4. Network measurements .....                       | 10 |
| 5. IEEE 802.11p considerations .....                | 11 |
| 6. Security overhead .....                          | 13 |
| 7. Real World Demo Setup at Stuttgart Airport ..... | 14 |
| 8. References .....                                 | 18 |

# D6.7: Second Version Interaction, Communication and Management

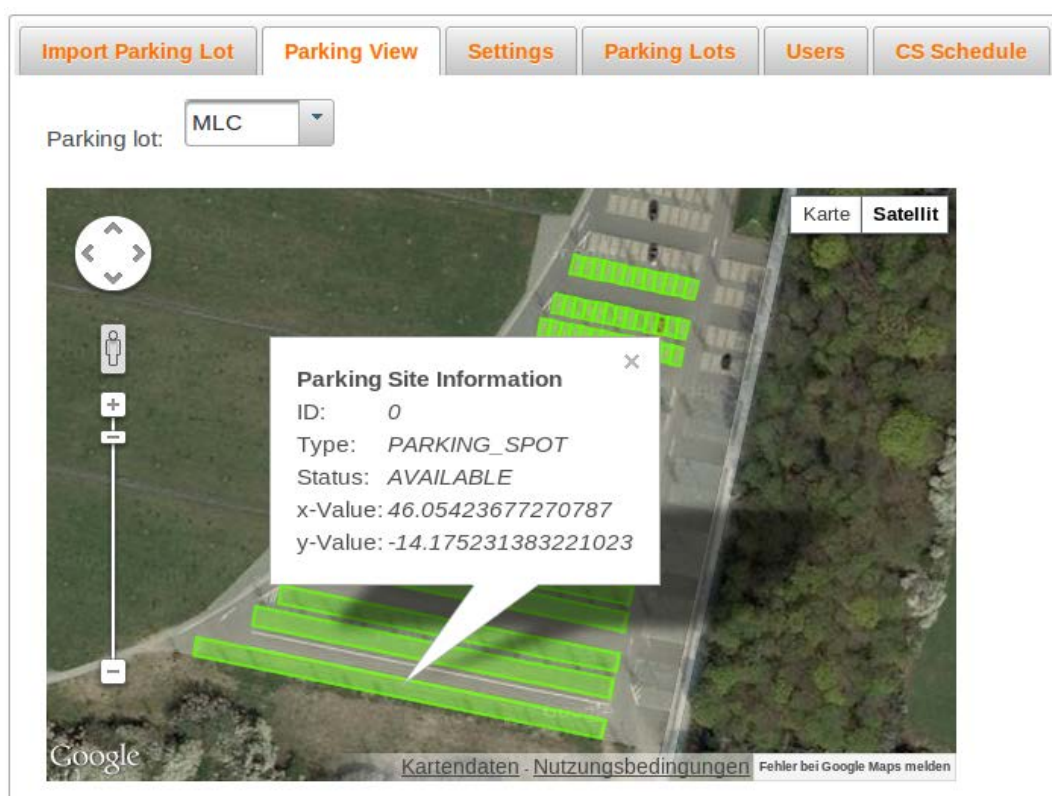
## 1. Extensions of Server components

The JavaEE based server is compiled to a single .war file which has to be deployed to a Glassfish server. The basic setup process has been described in [1].

### 1.1. RoadGraph

In order to simplify the exchange of data structures, we have written a parser which reads the RoadGraph used by the other partners. The .xml file can be compressed with World Files<sup>1</sup> and an overhead image of the parking site. This archive is read by the server software. All contained structures (roads, parking spots, ...) will be automatically parsed and extracted by the server software. Relevant structures (parking/charging, drop-off, pickup-zones) will be imported to the server's database and used for scheduling.

The current state of the overall site is displayed in the server's Web Interface. The current state of parking spots (free, occupied) is shown via a color on the image. The image may be sourced from Google Maps or an additional image file provided with the parking spot data. The Google Maps version however, only works if geo-referenced data is available.



### 1.2. Debugging features

During the preparations for the demo in April 2014, we realized that with the fully automated process of vehicle placement etc., the system is hard to debug and use when testing. Until now, it was not possible to assign a specific parking spot for a vehicle when making a reservation. During the real

<sup>1</sup> [http://en.wikipedia.org/wiki/World\\_file](http://en.wikipedia.org/wiki/World_file)

world testing in a partly occupied garage, this becomes a problem. For simplifying the tests of the whole system (Android app, server, vehicle), we added a feature to the server software which allows overriding the automatically chosen destination for the vehicle.

#### Administration

Parking View
Import Parking Lot
Users
Settings
CS Schedule

☒ Disable auto-resend mission on failure  
☐ Send ID (instead of pose)  
☒ Forward parking (vs. backward)  
☒ Enable automatic parking view update  
☒ Simple drop-off

---

☐ Override parking pose  

X

Y

Yaw

Submit

Figure 1: Settings for debugging and development purposes

## 2. Parking spot management

In order to evaluate and optimize the parking algorithms, we have developed a simulation tool which is able to take arrival processes for vehicles as an input and move them on the geographical layout of the parking site, respecting collisions. This tool is being used to evaluate high density parking algorithms like the k-Stacks.

### 2.1. High density parking algorithm

In contrast to regular (human only) (Figure 2) or mixed-mode (human and robot drivers share the same area) parking scenarios, there is a significant potential for space (and, consequently, cost) savings in automated-only parking. Without loss of generality, such a scenario can be implemented very cost-effectively even in existing parking lots by closing off a dedicated area to human drivers. Since autonomous cars do not require ambient space for opening doors or even access for human drivers at all, they can be parked much closer than in regular scenarios (Figure 3).

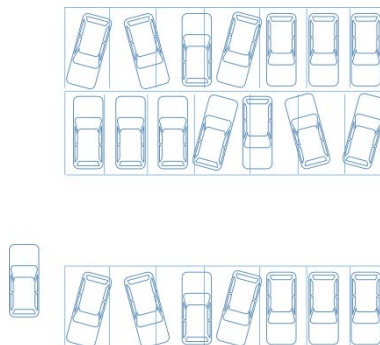


Figure 2: Regular parking scenario

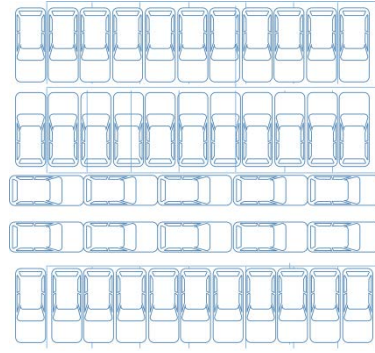


Figure 3: Parking scenario with blocked automated cars

A coordinated parking management controlling autonomous vehicles via V2I communications, as implemented by the V-Charge project, offers the possibility to increase the parking density even further by allowing vehicles to actually block other cars. Obviously, this is not an option in regular or mixed-mode scenarios, where drivers always have to be able to access and remove their car from the parking lot. In the V-Charge scenario, however, blocking vehicles can be tasked to autonomously move out of the way when necessary. This requires a sophisticated parking management, such that the required parking area can be used as efficiently as possible while maintaining reasonable pick-up times and limiting vehicle movements.

#### 2.1.1. k-Stacks

k-Stacks are the basic building block of our high-density parking lot for autonomous vehicles. k-stacks generalize a typical parking lot layout where cars are parked in boxes perpendicular to the road. Assuming the width of the road to be equal to the length of a parking box, this pattern allows a maximum parking density of  $2/3$ .

On a conventional parking lot, the density cannot be increased, because additional cars would block other cars' way out. But when dealing with a parking lot exclusively available to autonomous vehicles as described in, blocking cars can be moved out of the way at any time, allowing us to block cars. We show how to exploit that feature to increase parking lot density.

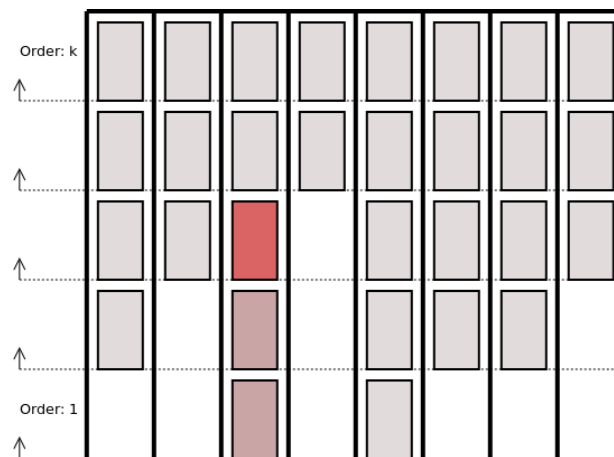


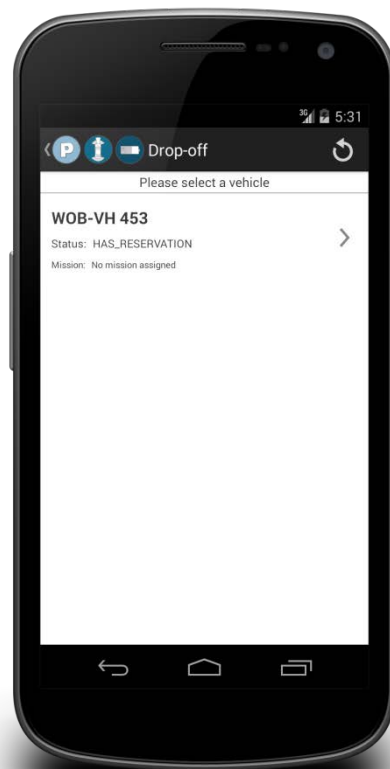
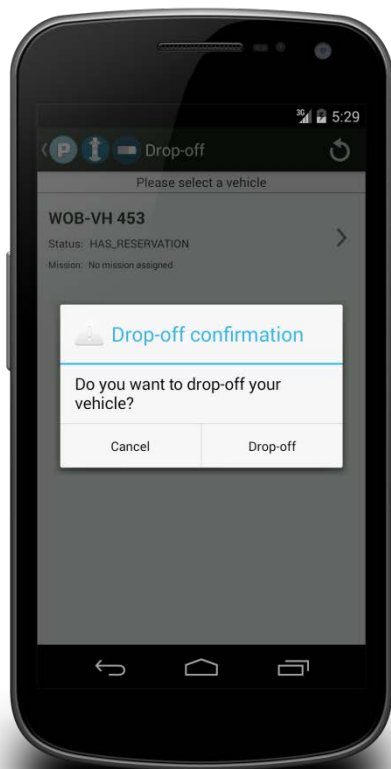
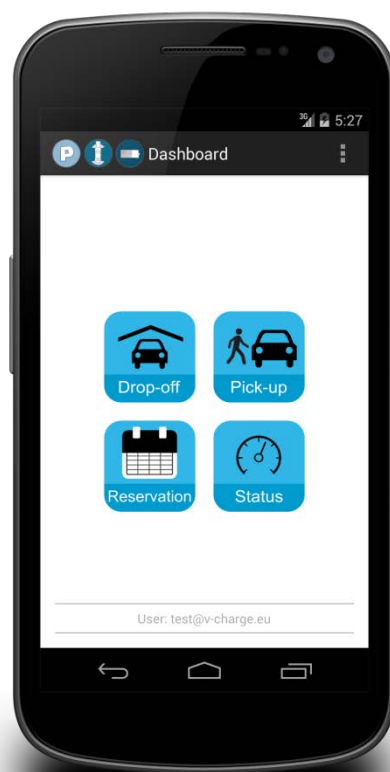
Figure 4: k-Stacks

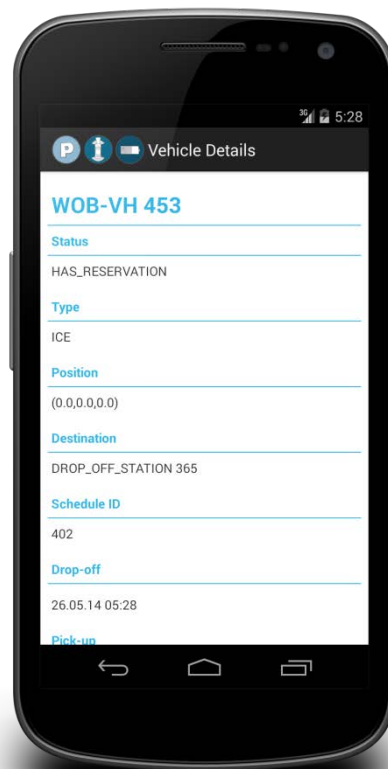
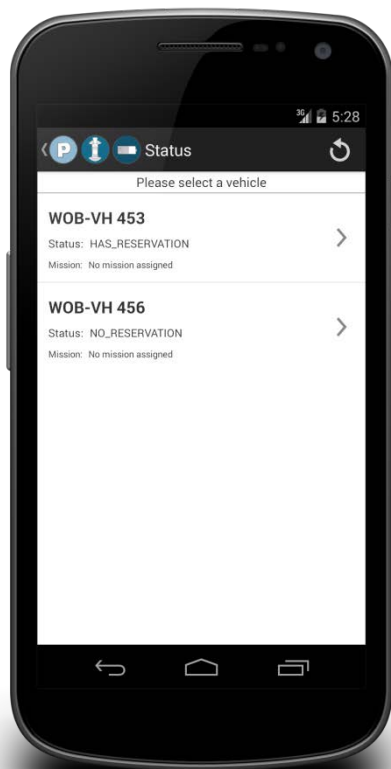
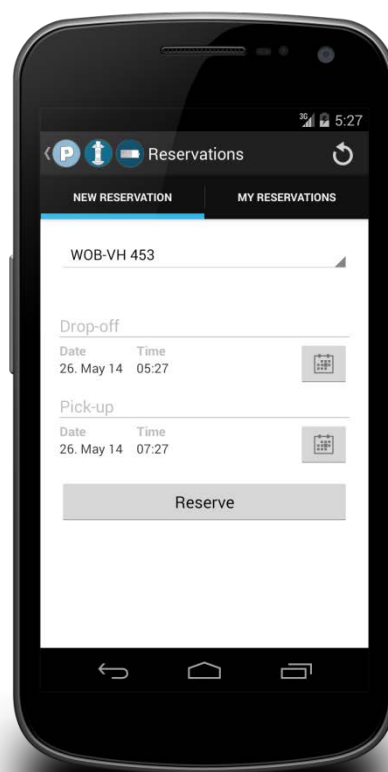
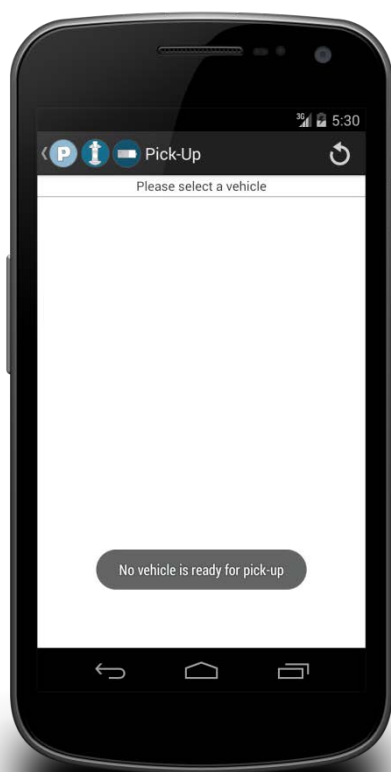
Single spots are merged to stacks for  $k$  vehicles (see Figure 4). Each vehicle is assigned an order. Order  $k$  is the “most blocked spot” on the stack, order 1 is the most easily accessible. If all stacks are completely filled, we can achieve a maximum density of  $2k/(2k+1)$ . To release a vehicle at order  $i$ , all other vehicles with order less than  $i$  from that stack have to be moved to make room. When pulling back in after the vehicle has left, their order is increased by 1. This means that any parked vehicle has to move at most  $k-1$  times for letting other cars pass.

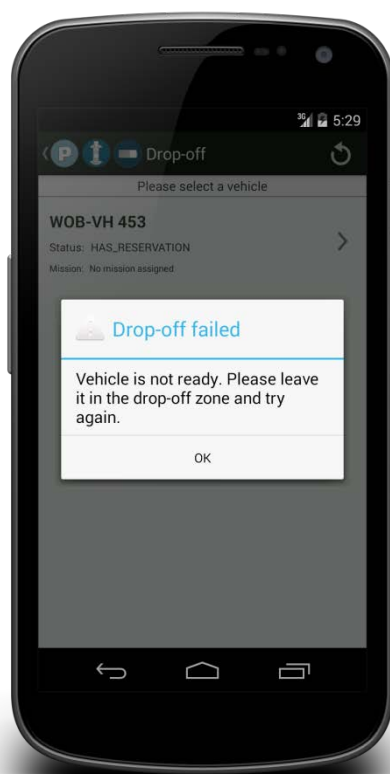
The complexity of inserting a vehicle into a stack is  $O(1)$ , it just has to park at the end of the least populated stack; removing a vehicle has complexity  $O(k)$ .

### **3. User interfaces**

The interface design of the mobile Android devices has undergone a slightly redesign to look cleaner. New development features have been added which simplify the use in a real world testing scenario, when a mission has to be reset after a failure.







#### 4. Network measurements

To be able to continuously monitor the signal quality over the parking site, we have developed capture tools which are running on the vehicles wherever they are. For every received Wi-Fi packet, it is possible, to extract network quality information. These data are stored in a local GIS database containing a timestamp of recording and the vehicle's position read from DDS topics.

Stationary monitoring nodes consisting of a Rasberry Pi SBC exist as well. From a central communication hub (cf. Section 7), they are powered using PoE. This simplifies the installation since only a single CAT 5/6 cable is necessary and no additional power outlets are required. Their position need to be configured in the software since no DDS topics as in the vehicles are available. Those stations are used to recognize bad network quality early.

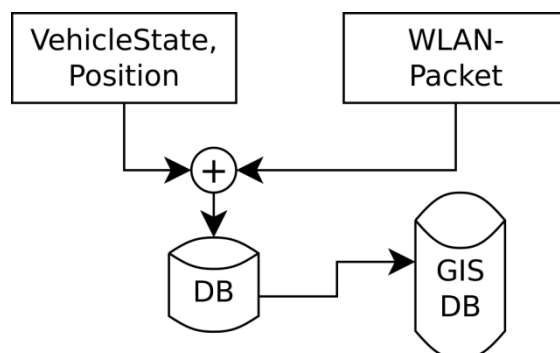


Figure 5: NQL Architecture

For gathering information on the quality, we are using the Radiotap<sup>2</sup> headers provided by the Linux kernel. After enabling a so-called monitoring network interface, it is possible to get this information using libpcap<sup>3</sup>. Depending on the loaded network driver, several fields like channel bandwidth, actual transmission speed of packet, RSSI value and many more are available.

Besides the network information headers, the complete IEEE 802.11 header is recorded as well. This is used to obtain information on the communication partners. They can be found in form of MAC addresses in the data. Since many types of packets are sent by the Wi-Fi stations (other vehicles, every single AP, ...) frequently, a very detailed map will be generated by just capturing the packets.

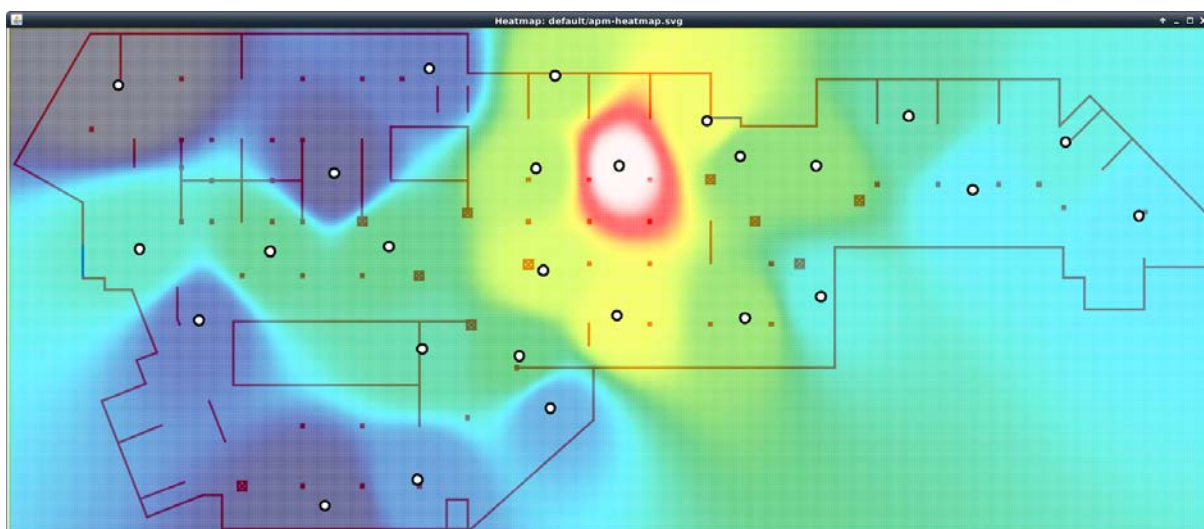


Figure 6: Heatmap of Wi-Fi RSSI

Data is stored in a GIS database which means that the network quality of geographical regions can easily be queried. From this data, heat maps can be generated for a quick visual inspection of the on-site quality. Figure 6 shows an example of a generated heat map of the RSSI. Additionally, the acquired data can also be used in combination with the algorithms for the assignment of parking spots [2]. Depending on the requirements of the customer, the vehicle may be placed accordingly. If a great amount of data has to be transferred to the vehicle, it should be parked in a region with a high speed Wi-Fi coverage. After the transfer is completed, it can be moved to a regular parking spot in a greater distance to the next access point.

## 5. IEEE 802.11p considerations

As described in the second periodic report of the V-Charge project, we investigated on benefits from using IEEE 802.11p for the V2X communications.

At the beginning of the project, it was planned to use IEEE 802.11p hardware like the NEC LinkBird. Due to power budget and space constraints inside the trunk of the vehicles, we were forced to stay with conventional PCI Express hardware. No 11p radios have been available at that time. During the project, the DDS middleware has been intensively used by the partners for offering a simple, universal interface.

We decided to stay with IEEE 802.11a/b/g/n for several reasons.

First, IEEE 802.11p describes only the physical layer which is optimized for vehicular communications. For example, it uses half the channel bandwidth of 802.11a, on which it is based. In order to be compliant to the standard, all C2C protocols have to be used on 11p. TUBS has been in contact with

<sup>2</sup> <http://www.radiotap.org/>

<sup>3</sup> <http://www.tcpdump.org/>

the according department of VW group research. Being able to use the 11p channels for our (special) use case would require lots of work on defining new messages for the protocols. Basically, 11p relies mostly on Ad-Hoc communication which means that we would not have been able to use the DDS middleware at the beginning of the project; a lot of work would have been required to adapt to 11p compliant messages. Using IBR-DTN as an underlying communication layer is no problem since we are completely free regarding the format of the messages which means that the use of IBR-DTN can be switched on or off transparently.

Second, the “CAR 2 CAR Communication Consortium Manifesto” [3] defines exemplarily several scenarios for C2C communication:

- *Safety*
  - *Cooperative Forward Collision Warning*
  - *Pre-Crash Sensing/Warning*
  - *Hazardous Location V2V Notification*
- *Traffic Efficiency*
  - *Enhanced Route Guidance and Navigation*
  - *Green Light Optimal Speed Advisory*
  - *V2V Merging Assistance*
- *Infotainment and Others*
  - *Internet Access in Vehicle*
  - *Point of Interest Notification*
  - *Remote Diagnostics*

Reading the descriptions of the scenarios, it becomes clear that the V-Charge application would fall into the third section, “*Infotainment and Others*”. Figure 7 shows the protocol layer stack of the C2C Communication System. Following the path from the *Infotainment Application* to the lower layers, it can be seen that those applications should use TCP/UDP or any other transport protocol.

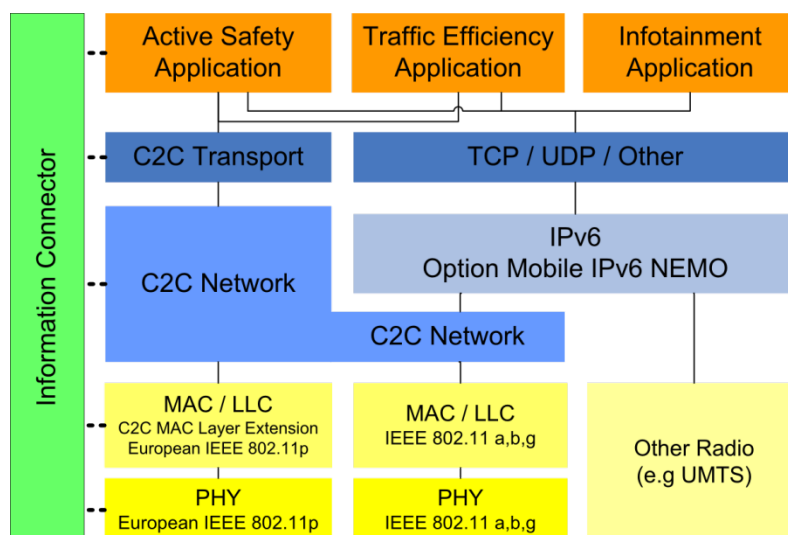


Figure 7: Protocol architecture of the C2C Communication System [3]

The description of this graphic says:

*On top of the radio layers MAC and PHY of the specific wireless technologies, the network layer provides wireless multi-hop communications based on geographical addressing and routing, and executes functions specific to vehicular communications like congestion control and vehicles' movement dissemination (beaconing). [3]*

The Car 2 Car Manifesto explicitly states that any data which is not safety relevant may bypass the communication stack and use conventional wireless links like IEEE 802.11a/b/g.

The communication stack that we are using, IBR-DTN, also includes the same features such as multi-hop communications and even geographical addressing as defined similarly by the C2C Network layer. This framework offers the transparent use of several network interfaces, even those, for which the C2C network layer is not defined (e.g. UMTS).

## 6. Security overhead

For communication and distributing the missions and status messages, we are using IBR-DTN. The actual DDS messages are tunneled via this software framework, which enables us the transparent use of several routing or security algorithms.

We have developed and implemented a comprehensive security architecture for the project which is based on cryptographic algorithms provided by the IBR-DTN software.

We evaluated the IBR-DTN daemon regarding its overhead resulting from cryptography as required by the Description of Work [4]. The evaluation is based on unit tests calculating the overhead in bytes for different cryptographic schemes. We differentiate between the schemes PIB-only (sign-only), PCB-only (encrypt-only), PCB-then-PIB (encrypt-then-sign), and PIB-then-PCB (sign-then-encrypt). IBR-DTN is configured to use ciphersuite PIB\_RSA\_SHA256 for PIBs, and PCB\_RSA\_AES128\_PAYLOAD\_PIB\_PCB for PIBs, as defined in RFC 6257 [5]. V-Charge's configuration favors PCB-then-PIB to allow verification of bundles' security source by nodes forwarding this bundle. While we evaluated the overhead for RSA with 1024, 2048, and 4096 bit, currently V-Charge's Android application generates 4096 bit keys, for long-term security (cf. Section 2.5). Our numerical results are presented in Table 1.

| Key type | PIB | PCB | PCB-then-PIB | PIB-then-PCB  |
|----------|-----|-----|--------------|---------------|
| RSA-1024 | 142 | 174 | 316          | 365 ... 367   |
| RSA-2048 | 270 | 302 | 572          | 621 ... 623   |
| RSA-4096 | 526 | 558 | 1084         | 1132 ... 1135 |

**Table 1: Overhead of cryptographic schemes in IBR-DTN; measurements in bytes**

The results are of fixed size, because every test is configured to use the same ciphersuites, as previously introduced. Insignificant deviations in the PIB-then-PCB scheme result from non-deterministic output from the surrounding PCB due the underlying padding scheme. Figure 2.8 depicts the overhead corresponding to the bundle size for relevant schemes with both authenticity and confidentiality, (Figure 8a) PCB-then-PIB and (Figure 8b) PIB-then-PCB. As defined, cryptographic schemes should not introduce more than 20 % overhead. This boundary is depicted in Figure 8 by a boundary line. Conclusively, to comply with this requirement when utilizing V-Charge's favored scheme, PCB-then-PIB with 4096 bit RSA key pairs, the initial bundle size should be at least 1084 byte.

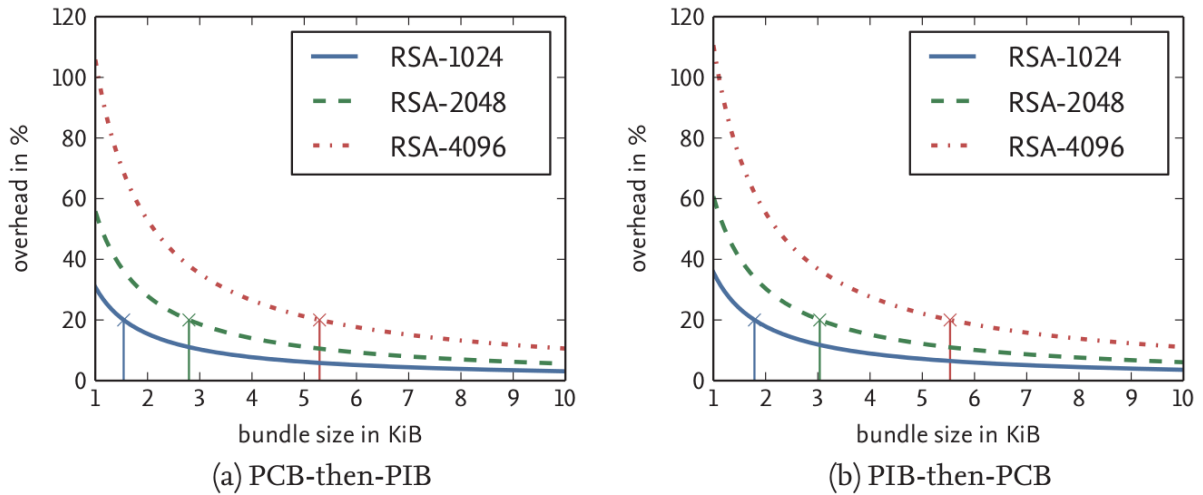


Figure 8: Overhead in percent depending on bundle size; 20 % boundary marked

More details on the implemented architecture can be found in [6].

## 7. Real World Demo Setup at Stuttgart Airport

In line with the demonstration of the project's status in early May '14, we created a real world deployment of the communication infrastructure. We had to ensure a network coverage for the open air drop-off zone in front of the Stuttgart Messe. The planned path of the vehicle leads through a kind of street canyon between the two garages, of which one will be entered. Inside the building, Wi-Fi was needed, too.

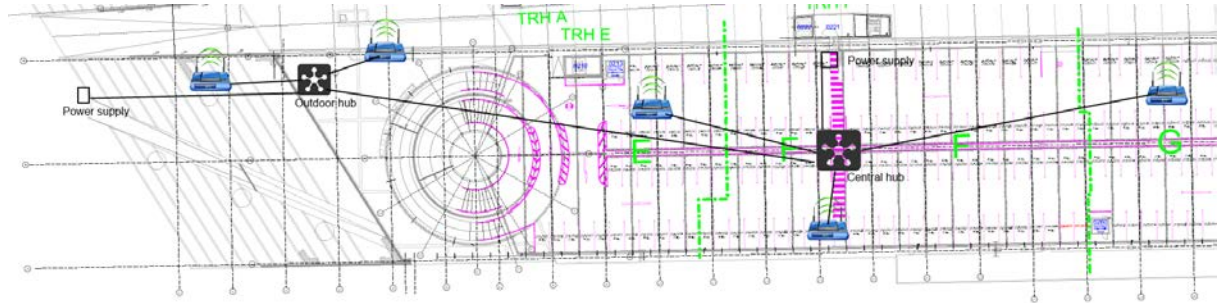


Figure 9: Network components installed in Bosch Parkhaus

Figure 9 shows the installed components. We used five Ubiquiti Access Points of the UniFi outdoor series. Due to the small distance to the Airport and the RADAR system, we decided against using IEEE 802.11a in the 5 GHz band and chose IEEE 802.11 b/g. Two communication hubs connected with a 200 m fiber line supplied the access points via PoE. Thus, only Cat 6 cables were required. One outdoor AP was mounted on a tripod, the other was fixated on the balustrade on the garage's roof.

The hubs consisted of a waterproof box made of aluminum with IP64 rated cable feedthrough.

The indoor APs have been mounted to cable trays found for cabling the lights and other infrastructure. All components have been secured against theft by locks.

The indoor communication hub also contained an industry PC by MSI which was running the actual server software and the management tool for the Wi-Fi. All access points have been offering a single ESSID to which the clients could connect. An UMTS stick was installed to the PC. A script established a reverse SSH tunnel to a computer at TUBS which allowed remote monitoring and maintenance of the system. This enabled the possibility to re-deploy the server software and have an eye on the communication.

Figure 10 through Figure 15 show some impressions of the actual deployment.

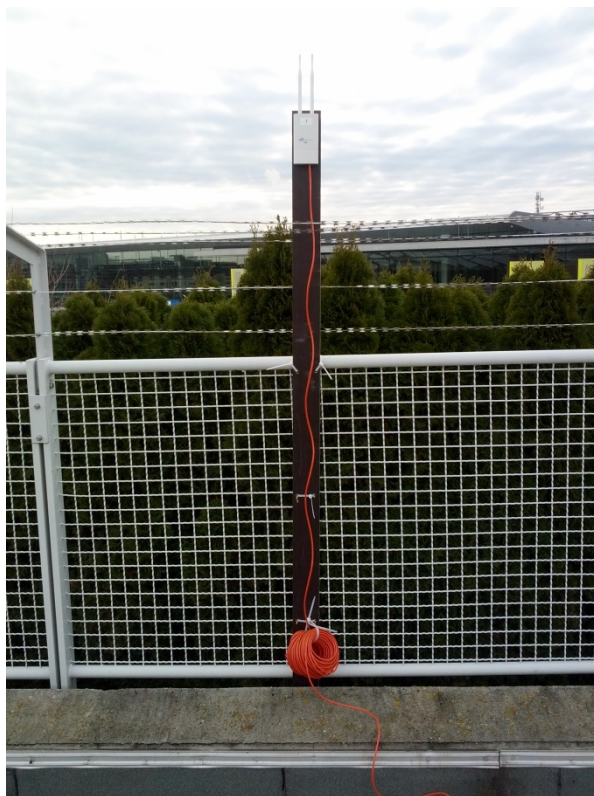


Figure 10: Outdoor mounted access point



Figure 11: Cable feedthrough at indoor communication hub



Figure 12: Components of central communication: Media converter for fiber line, PC running server software, PoE enabled switch



Figure 13: Communication hub placed under the garage's ceiling



Figure 14: Access point mounted on tripod



Figure 15: AP mounted on cable tray

## 8. References

1. **V-Charge-Consortium.** *Deliverable 6.5.* 2012.
2. **Pögel, Tobias, et al., et al.** Estimation of Vehicular Connectivity in Autonomous Parking Scenarios. *PIK - Praxis der Informationsverarbeitung und Kommunikation.* Volume 36, 2013, Issue 4.
3. **Consortium, CAR 2 CAR Communication.** *CAR 2 CAR Communication Consortium Manifesto.* 2007.
4. **V-Charge-Consortium.** Description of Work. *V-Charge - Autonomous Valet Parking and Charging for e-Mobility.* 2010.
5. **Symington, S., et al., et al.** *RFC 6257 (Experimental): Bundle Security Protocol Specification.* s.l. : Internet Engineering Task Force, 2011.
6. **Timpner, Julian, Schürmann, Dominik and Wolf, Lars.** Secure Smartphone-based Registration and Key Secure Smartphone-based Registration and Key. *CyCAR '13 Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles.* Berlin : ACM, 2013.