

# **Pflichtenheft PSE - Automotive Remote Sensor Data Management (aRSDM)**

Vladimir Bykovski      Jonas Haas      David Grajzel  
Nicolas Schreiber      Valentin Springsklee      Yimeng Zhu

25. November 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aufgabenstellung . . . . .	1
<b>2</b>	<b>Zielbestimmung</b>	<b>2</b>
2.1	Musskriterien . . . . .	2
2.1.1	/MK1010/Anzeigeelemente . . . . .	2
2.1.2	/MK1020/Remote Acces . . . . .	2
2.1.3	/MK1030/Protokollierung . . . . .	2
2.1.4	/MK1040/Statistiken . . . . .	2
2.1.5	/MK1050/Erweiterbarkeit . . . . .	2
2.1.6	/MK1060/Fahrinformation . . . . .	2
2.1.7	/MK1070/Durschnittsverbrauch . . . . .	3
2.1.8	/MK1080/Restkilometer . . . . .	3
2.1.9	/MK1090/Einparkhile . . . . .	3
2.1.10	/MK10100/POI . . . . .	3
2.2	Wunschkriterien . . . . .	3
2.2.1	/MK2010/Smartphone Utilization . . . . .	3
2.2.2	/MK2020/Personalisierung . . . . .	3
2.2.3	/MK2030/Fahrer . . . . .	3
2.3	Abgrenzungskriterien . . . . .	3
2.3.1	/MK3010/Erweiterung . . . . .	3
2.3.2	/MK3020/Unidirektional . . . . .	3
2.3.3	/MK3030/Bildauswertung . . . . .	3
2.3.4	/MK3040/Android . . . . .	4
2.3.5	/MK3050/Flotte . . . . .	4
2.3.6	/MK3060/Navigation . . . . .	4
2.3.7	/MK3070/Diagnose . . . . .	4
2.3.8	/MK3080/RemoteAcces . . . . .	4
<b>3</b>	<b>Produkteinsatz</b>	<b>5</b>
3.1	Anwendungsbereiche . . . . .	5
3.2	Zielgruppen . . . . .	5
3.3	Betriebsbedingungen . . . . .	5
<b>4</b>	<b>Produktumgebung</b>	<b>6</b>
4.1	Software . . . . .	6
4.2	Hardware . . . . .	7

<b>5</b>	<b>Produktfunktionen</b>	<b>8</b>
5.1	Allgemeine Funktionen . . . . .	8
5.1.1	/FA0010/Statistiken . . . . .	12
5.1.2	/FA0020/Erweiterbarkeit . . . . .	12
<b>6</b>	<b>Produktdaten</b>	<b>13</b>
6.1	/PD1000/physische Sensordaten . . . . .	13
6.2	/PD2000/aggregierte Sensordaten . . . . .	14
6.3	/PD3000/Personenbezogene Daten . . . . .	14
6.4	/PD4000/Laufzeitdaten . . . . .	14
<b>7</b>	<b>Produktleistungen</b>	<b>15</b>
7.0.1	/PL0010/Einparkhilfe-Verzögerung . . . . .	15
7.0.2	/PL0020/Kamerabildaktualisierung . . . . .	15
7.0.3	/PL0030/Multiple Acces . . . . .	15
<b>8</b>	<b>Benutzerschnittstelle</b>	<b>16</b>
8.1	GUI . . . . .	16
8.2	Schnittstelle für Netzwerk . . . . .	16
<b>9</b>	<b>Globale Testfälle</b>	<b>17</b>
9.0.1	/T0010/Verbindung im Auto . . . . .	17
9.0.2	/T0020/RemoteAcces . . . . .	17
9.0.3	/T0030/Dateninput . . . . .	17
9.0.4	/T0040/Protokollierung . . . . .	17
9.0.5	/T0050/Vergangene-Daten . . . . .	17
9.1	Netzwerk . . . . .	17
9.2	Pi im Auto . . . . .	17
9.3	Endgeräte . . . . .	17
<b>10</b>	<b>Qualitätsbestimmung</b>	<b>18</b>
<b>11</b>	<b>Glossar</b>	<b>19</b>

# 1 Einleitung

In der modernen Informationsgesellschaft fallen immer mehr nutzbare Daten an. Seien es Daten von sozialen Netzwerken und Daten von Personen über Nutzung und Gewohnheiten allgemein, oder Daten technischer Art, wie Sensordaten in Anlagen oder Automobilen. Mehr und mehr Sensorik findet Einsatz. Oft ist es sehr interessant diese Daten auszuwerten und visuell zu repräsentieren. Tablets und Smartphones sind omnipräsent. Jeder hat dauerhaft Zugriff auf einen leistungsfähigen Rechner mit hochauflösendem Display. Diese Geräte sind also optimal zur Anzeige von Daten geeignet. Unabhängig von der Art der Daten müssen diese gesammelt, gegebenenfalls vorverarbeitet und bereitgestellt werden. Dazu bedarf es einer generischen Schnittstelle. In dem Rahmen von aRSDM sollen Sensordaten gesammelt, gespeichert und weiterverteilt werden. Die Quelle dieser Daten ist der mittlerweile in fast jedem PKW verfügbare OBD2 Anschluss. Es ist eine (abstrakte) Komponente zwischen Sensoren und Endgerät gefragt, die sich für beliebige, nicht unbedingt schon bekannte Anwendungsszenarien erweitern lässt. Ein Rahmenwerk, das Daten sammelt, speichert und verteilt.

## 1.1 Aufgabenstellung

Smartphones und Tablets des Users sind die sogenannten Endgeräte. Auf ihnen erfolgt eine Grafische Darstellung von Sensordaten. Die Quelle der Daten befindet sich in lokaler Distanz, die Daten gelangen über ein Netzwerk auf das Endgerät. aRSDM bietet eine API, die von Dritten für weitere Anwendungsfälle genutzt werden kann. Als Anwendung liefern wir ein Automotive-Modul, welches die Sensordaten, die über den CAN-Bus eines PKW verfügbar sind, verarbeitet und darstellt.

## 2 Zielbestimmung

Die Software aRSDM soll diejenigen Sensordaten von einem Auto sammeln, die über den OBD2-Adapter verfügbar sind. aRSDM soll diese Daten speichern und Endgeräten zur Darstellung zur Verfügung stellen. Dabei soll eine API implementiert und genutzt werden, die auch auf weitere Anwendungen erweiterbar ist.

Die Darstellung soll in verschiedenen Internet-Browsern auf Endgeräten erfolgen.

### 2.1 Musskriterien

#### Allgemeine Funktionen

##### 2.1.1 /MK1010/Anzeigeelemente

Der Nutzer kann Anzeigeelemente ein- und ausblenden.

##### 2.1.2 /MK1020/Remote Acces

Falls Netzwerkverbindung besteht, können entfernte Nutzer die Sensordaten einsehen.

##### 2.1.3 /MK1030/Protokollierung

Der Nutzer soll jederzeit in der Lage sein, Daten von vergangenen Zeitpunkten abzurufen.

##### 2.1.4 /MK1040/Statistiken

Der Benutzer kann sich Statistiken der gespeicherten Sensorwerte anzeigen lassen.

##### 2.1.5 /MK1050/Erweiterbarkeit

Drittentwickler können Module für die Verarbeitung und Auswertung weiterer Sensordaten schreiben.

#### Automotive Kontext

##### 2.1.6 /MK1060/Fahrinformation

Der Benutzer bekommt Fahrinformationen angezeigt.

### **2.1.7 /MK1070/Durschnittsverbrauch**

Die Software zeigt den Durchschnittsverbrauch des Fahrzeugs an.

### **2.1.8 /MK1080/Restkilometer**

Die Software zeigt die verfügbaren Restkilometer bis zur nächsten Tankfüllung an. Diese werden anhand des Fahrverhaltens berechnet.

### **2.1.9 /MK1090/Einparkhile**

Dem Fahrer steht eine Einparkhilfe mit Augmented Reality-Features zur Verfügung.

### **2.1.10 /MK10100/POI**

Bei kritischen Sensorwerten kann der Nutzer die nächsten Tankstellen / Werstätten einblenden.

## **2.2 Wunschkriterien**

### **2.2.1 /MK2010/Smartphone Utilization**

Die Software verwendet auch Sensordaten aus dem Endgerät.

### **2.2.2 /MK2020/Personalisierung**

Aggregierte Funktionen sind personalisiert verfügbar.

### **2.2.3 /MK2030/Fahrer**

Ein Nutzer, der mit dem Server verbunden ist, kann sich mit der Software auf seinem Endgerät als Fahrer identifizieren.

## **2.3 Abgrenzungskriterien**

### **2.3.1 /MK3010/Erweiterung**

Für andere Sensorquellen werden keine Treiber bereitgestellt.

### **2.3.2 /MK3020/Unidirektional**

Die Software liest und interpretiert die Daten nur und sendet keine Daten an Quellen zurück.

### **2.3.3 /MK3030/Bildauswertung**

Kamerabilder werden von der Software nicht interpretiert.

#### **2.3.4 /MK3040/Android**

Die Visualisierung der Daten erfolgt nur im Browser, wir stellen keine gesonderte App zur Verfügung.

#### **2.3.5 /MK3050/Flotte**

Der Zugriff auf verschiedene Fahrzeuge wird nicht verwaltet.

#### **2.3.6 /MK3060/Navigation**

Die Anzeige der aktuellen Position und der umgebenden POI erfolgt über Drittsoftware und kann nicht als Navigationssystem benutzt werden.

#### **2.3.7 /MK3070/Diagnose**

Die Datenspeicherung ersetzt nicht den Fehlerspeicher des Steuergeräts im Auto.

#### **2.3.8 /MK3080/RemoteAcces**

Für entferntes Monitoring wird keine maximale Latenz garantiert

## **3 Produkteinsatz**

Das Produkt dient zur Speicherung und Distribution von Daten mehrerer verteilter Sensoren eines Autos. Dies beinhaltet Informationen wie die Geschwindigkeit, Motortemperatur, Raddrehzahl und ähnliches. Damit bietet es die Möglichkeit diese Daten auf einem Mobiltelefon, Laptop oder anderem Computer mit Bildschirm anzuzeigen.

### **3.1 Anwendungsbereiche**

- Dieses Produkt dient zum Speichern und Anzeigen von in einem PKW verfügbaren Sensordaten.

### **3.2 Zielgruppen**

- Private Autofahrer mit oder ohne tiefere technische Kenntnisse sowie Firmen mit Automobilflotten.

### **3.3 Betriebsbedingungen**

- Der PKW muss über einen OBD2 Anschluss verfügen.
- Das Produkt muss Strom aus dem Auto beziehen können.
- Für Remote-Zugriff muss eine Netzwerkverbindung vorhanden sein.



## 4 Produktumgebung

Das Rahmenwerk und die Module werden in TypeScript programmiert. Der Server wird in node.js laufen. Als Datenbanksystem wird LevelUp verwendet.

- Eine Client-Server Architektur nach dem Thin Client-Konzept
- Auf dem Server läuft der Teil der Software, der die Sensordaten empfängt, einpflegt und über WebRTC an die Endgeräte versendet.
- Auf der Userseite laufen verschiedene Applikationen neben aRSDM. Diese Anwendungen sollen die Software aber nicht beeinflussen und auch nicht von ihr beeinflusst werden.

### Testumgebung

- Testfahrzeug: Opel Astra H 1.6 Bj 2005
- Testfahrzeug: ITK-Engineering Modellauto  
Ein Modellauto der Firma ITK-Engineering AG, angetrieben über einen Elektromotor. Alle Daten die hier verfügbar sind liegen über CAN an.

#### 4.1 Software

- **Serverseite:**  
Ein Webserver mit einer Datenbank  
Aufgebaut auf einer Linux-Distribution
- **Clientseite:**  
Web-Browser:
  - Google Chrome Version 46
  - Safari Version 9
  - Firefox Version 42
  - Google Chrome Android Version 14.10Getestet auf:
  - Mac OS X
  - Fedora 22
  - Windows 7

## 4.2 Hardware

- **Serverseite:**

Kleincomputer mit:

- Broadcom BCM2836 Arm7 Quad Core Processor (900MHz)
- 1GB RAM
- 4 x USB 2 ports

Can-Modul

- **Clientseite:**

Standardrechner (min. 1 GHz und 2 GB RAM)

Android Smartphone (min. 1 GHz und 1 GB RAM)

# 5 Produktfunktionen

## 5.1 Allgemeine Funktionen

### 010/ GUI-Instrumente auswählen

#### 011/ Instrumente anzeigen:

Vorbedingung: Server läuft. Es sind Endgeräte mit ihm verbunden.

Auslösendes Ereignis: Der Nutzer wählt in den Einstellungen auf dem Endgerät ein Instrument aus.

Nachbedingung: Das Instrument wird auf dem Screen angezeigt und zeigt die aktuellen Daten an. Falls keine aktuellen Daten vorhanden sind, steht es in neutraler Position (0).

### 021/ Datenhaltung

#### 021/ Protokoll führen

Ziel: Es werden eingehende Daten gespeichert.

Vorbedingung: Server läuft. Bluetooth-Verbindung mit OBD2 Adapter ist hergestellt.

Auslösendes Ereignis: Von der Bluetooth-Schnittstelle treffen Daten ein. Nachbedingung: Die Daten, die von der Bluetooth-Schnittstelle kommen, befinden sich in der Datenbank und kommen über das Netzwerk auf Endgeräten an.

#### 022/ Vergangene Daten aufrufen

Ziel: Es werden Daten von vergangenen Zeitpunkten dargestellt.

Vorbedingung: Es wurde bereits eine Fahrt getätigt. Der Server läuft.

Auslösendes Ereignis: Der Nutzer fordert Daten aus der Vergangenheit an.

Nachbedingung: Die Daten werden auf dem Endgerät angezeigt.

#### 023/ Maximale Kapazität einstellen

Ziel: Es soll eine neue maximale Kapazität der Datenbank gesetzt werden.

Vorbedingung: Der Server läuft.

Auslösendes Ereignis: Der Benutzer versucht, eine neue maximale Kapazität der Datenbank zu setzen.

Nachbedingung: Die neue gespeicherte maximale Kapazität der Datenbank ist der vom Nutzer eingegebene Wert.

#### 024/ Daten automatisch löschen

Ziel: Um Überfüllung der Datenbank zu vermeiden werden Daten gelöscht.

Vorbedingung: Der Server läuft.

Auslösendes Ereignis: Die definierte maximale Kapazität der Datenbank ist

erreicht.

Nachbedingung: Die Datenbank hat weniger Einträge als ihre maximale Kapazität. Die ältesten Einträge wurden gelöscht.

034/ **Remote Access erzeugen**

Ziel: Der Benutzer kann die Software auf einem entfernten Endsystem benutzen.

Vorbedingungen: Das System läuft und es kommen Daten über den OBD2-Adapter an. Das System hat eine Verbindung zum Internet.

Ablauf: Der Benutzer verbindet ein entferntes Endgerät über Eingabe einer URL im Browser mit dem Server.

Nachbedingung (Erfolg): Die Funktionen GUI-Konfiguration, Fahrinformation, vergangene Daten, Karte und Einparkhilfe stehen auf dem Endsystem zur Verfügung.

Nachbedingung (Fehl Schlag): Das Endsystem zeigt an, dass die Verbindung zum Server fehlgeschlagen ist.

044/ **Fahrinformationen darstellen**

Ziel: Darstellung der aktuellen Sensorwerte auf dem Endgerät

Vorbedingung: Das System läuft und es werden durchgängig neue Daten zur Datenbank hinzugefügt. Es existiert eine Verbindung mit mindestens einem Endgerät. Dieses Gerät besitzt eine Konfiguration des GUI.

Auslösendes Ereignis: Das Endgerät empfängt Sensordaten vom Server.

Nachbedingung: Das Endgerät zeigt diese Daten entsprechend der aktuellen GUI-Konfiguration an. Die Datenbank bleibt unverändert.

054/ **Aggregierte Funktionen berechnen**

Ziel: Berechnung der aggregierten Funktionen.

Vorbedingung: FA/Fahrinformationen funktioniert. Gewünschte aggregierte Funktionen sind angegeben.

Auslösendes Ereignis: Es kommen zur Berechnung der aggregierten Funktionen benötigte Sensorwerte am Server an.

Ablauf: Die Software berechnet die gewünschten aggregierten Funktionen. Sie schickt die Informationen an das Endgerät. Das Endgerät zeigt die berechneten Daten entsprechend der GUI-Konfiguration an.

Nachbedingungen: Das Endgerät zeigt die Daten im UI an.

Aggregierte Funktionen sind:

- Treibstoffverbrauch: Die Software berechnet über den Tankfüllstand und die Zeit den aktuellen Treibstoffverbrauch.
- Durchschnittsverbrauch: Die Software berechnet über den Tankfüllstand und die Zeit den Durchschnittsverbrauch über die letzten 100km
- Restkilometer: Die Software berechnet eine Schätzung der Restkilometer abhängig vom aktuellen Treibstoffverbrauch.
- Strecke: Die Software berechnet die zurückgelegte Strecke, über die Geschwindigkeit und die Zeit.

- Durchschnittsdauer/strecke: Die Software berechnet die durchschnittliche Strecke einer Fahrt, entweder über die Zeit oder über die Strecke.

#### 064/ **Smartphone Utilization**

Vorbedingung: Es existiert eine Verbindung mit einem Endgerät. Die Sensoren des Endgeräts sind aktiviert.

Auslösendes Ereignis: Auf dem Endgerät sind neue Sensordaten verfügbar.

Nachbedingung (Erfolg): Die Sensordaten vom Endgerät werden auf den Server übertragen. Die Daten werden auf dem Server gespeichert und von Software bearbeitet.

Nachbedingung (Fehl Schlag): Die Sensoraten werden nicht übertragen.

#### 074/ **Einparkhilfe**

##### 071/ **Einparken vereinfachen**

Ziel: Vereinfachen des rückwärts beziehungsweise rückwärts-Seitwärts Einparkens durch Anzeige einer Rückfahrkamera und Einblenden einer Visualisierung der möglichen Fahrstrecke.

Vorbedingung: Server läuft.

Auslösendes Ereignis: Der Fahrer schaltet ins Rückwärtsgang. Alternativ: Der Benutzer fordert das System auf die Kamera einzuschalten.

Nachbedingung (Erfolg): Das Bild der Rückfahrkamera wird angezeigt und regelmäßig aktualisiert. Die Visualisierung der Fahrbahn wird korrekt dargestellt und regelmäßig aktualisiert.

Nachbedingung (Fehl Schlag): Das Bild der Rückfahrkamera wird nicht angezeigt, ist fehlerbehaftet oder wird nicht aktualisiert. Die Visualisierung der Fahrbahn wird nicht dargestellt, ist nicht korrekt oder wird nicht aktualisiert.

##### 072/ **Ultraschallsensoren visualisieren**

Ziel: Die Distanzwerte der Ultraschallsensoren visualisieren.

Vorbedingung: Eine Kamera ist vorhanden und eingeschaltet. Der Server läuft.

Auslösendes Ereignis: Der Fahrer schaltet in den Rückwärtsgang. Alternativ: Der Benutzer möchte die Messwerte anzeigen.

Nachbedingung (Erfolg): Die korrekten Messwerte werden angezeigt.

Nachbedingung (Fehl Schlag): Es werden keine oder falsche Messwerte angezeigt.

##### 073/ **Einparkhilfe beenden**

Ziel: Das System soll die Darstellung der Einputrkhilfe automatisch vom Bildschirm entfernen, wenn die Daten nicht mehr sinnvoll dargestellt werden können.

Vorbedingung: Auf dem Bildschirm wird die Einputrkhilfe angezeigt.

Nachbedingung (Erfolg): Die Einputrkhilfe wird nicht mehr angezeigt.

Auslösendes Ereignis: Der Benutzer fordert das System auf, die Einputrkhilfe

zu beenden. Alternativ: Das Fahrzeug fährt schneller als 5 km/h vorwärts.  
Nachbedingung (Fehlschlag): Die Einparkhilfe verschwindet nicht von dem Bildschirm.

## 083/ **Karte und POI**

### 081/ **Karte anzeigen**

- i. Vorbedingung: Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk. Die Lokalisierungsinformationen stehen zur Verfügung.  
Auslösendes Ereignis: Der Benutzer fordert die Karte an.  
Nachbedingung: Eine Anfrage nach der Karte in Umgebung mit Lokalisierungsinformationen wird am Drittanbietern geschickt, der die Kartendienst anbieten kann.
- ii. Vorbedingung: Eine Anfrage nach der Karte in Umgebung mit Lokalisierungsinformationen ist am Kartendiensteanbietern geschickt. Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk.  
Auslösendes Ereignis: Der Drittanbieter antwortet die Anfrage.  
Nachbedingung: Die Karte von Umgebung wird angezeigt.

### 082/ **Tankstellen auflisten**

- i. Vorbedingung: Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk. Die Lokalisierungsinformationen stehen zur Verfügung.  
Auslösendes Ereignis: Das OBD-system detektiert einen niedrigen Tankfüllstand  
Nachbedingung: Eine Anfrage nach den Tankstellen in der Nähe mit Lokalisierungsinformationen wird am Drittanbietern geschickt, der die Kartendienst anbieten kann.
- ii. Vorbedingung: Eine Anfrage nach den Tankstellen in der Nähe ist am Kartendiensteanbietern geschickt. Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk.  
Auslösendes Ereignis: Der Drittanbieter antwortet die Anfrage.  
Nachbedingung: Eine Liste, auf der die Tankstellen in der Nähe mit jeweiligen Preis bzw. Entfernung stehen, werden erzeugt und angezeigt. Auf der Karte werden die Tankstellen in der Nähe kenngezeichnet.

### 083/ **Route zur gewünschten Tankstelle berechnen**

- i. Vorbedingung: Die Tankstellen in der Nähe sind angezeigt. Die Endgeräte vom Fahrer haben die Verbindung zur Netzwerk.  
Auslösendes Ereignis: Der Fahrer wählt eine Tankstelle aus.  
Nachbedingung: Eine Anfrage nach der Route zur vom Fahrer ausgewählten Tankstelle wird am Kartendiensteanbietern geschickt.
- ii. Vorbedingung: Eine Anfrage nach der Route zur Tankstelle ist am Drittanbieter von Kartendienst geschickt. Die Endgeräte vom Fahrer haben die Verbindung zur Netzwerk.  
Auslösendes Ereignis: Der Drittanbieter antwortet die Anfrage.

Nachbedingung: Die Route zur gewünschten Tankstelle wird auf Karte angezeigt.

#### 084/ **Werkstätten anzeigen**

- i. Vorbedingung: Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk. Die Lokalisierungsinformationen stehen zur Verfügung.  
Auslösendes Ereignis: Das OBD-system detektiert einen kritischen Sensorwert.  
Nachbedingung: Eine Anfrage nach der nächsten Werkstatt mit Lokalisierungsinformationen wird am Drittanbietern geschickt, der die Kartendienst anbieten kann.
- ii. Vorbedingung: Eine Anfrage nach der Route zur nächsten Werkstatt ist am Kartendienstanietern geschickt. Die Endgeräte vom Fahrer haben eine Verbindung zur Netzwerk.  
Auslösundes Ereignis: Der Drittanbieter antwortet die Anfrage.  
Nachbedingung: Auf der Karte wird die Route zur nächsten Werkstatt angezeigt.

### **5.1.1 /FA0010/Statistiken**

Der Benutzer kann sich Statistiken der gespeicherten Sensorwerte anzeigen lassen.

### **5.1.2 /FA0020/Erweiterbarkeit**

Drittentwickler können Module für die Verarbeitung und Auswertung weiterer Sensordaten schreiben.

#### **/FA0020W/Fahrer**

Ein Nutzer, der mit dem Server verbunden ist, kann sich mit der Software auf seinem Endgerät als Fahrer identifizieren (/FA0070/, /FA1020W/).

# 6 Produktdaten

## 6.1 /PD1000/physische Sensordaten

Es sind zeitgestempelte und fahrergestempelte Messwerte der folgenden Sensoren zu speichern:

- Temperatur der Motorkühlflüssigkeit
- Kraftstoffdruck
- Motordrehzahl
- Geschwindigkeit des Fahrzeugs
- Lenkraddrehung
- Ansauglufttemperatur
- Laufzeit des Motors seit letztem Start
- Tankfüllstand
- Status von
  - Abgasrückführung
  - Kontrollsystem für Einspritzdruck
  - Kontrollsystem für Kraftstoffdruck
- Gasdruck des Verdampfers
- Katalysatortemperatur
- Position der Drosselklappe
- Position des Gaspedals
- Außentemperatur
- Motordrehmoment
- Abgastemperatur
- Abgasdruck
- Distanzwerte der Ultraschallsensoren



## **6.2 /PD2000/aggregierte Sensordaten**

Es sind Werte der folgenden virtuellen Sensoren zu speichern:

- Anzahl der Einzelfahrten
- Aktueller Treibstoffverbrauch

## **6.3 /PD3000/Personenbezogene Daten**

Es sind Profilinformationen der Fahrer zu speichern:

- Endgerät-Signatur
- Anzeigeeinstellungen

## **6.4 /PD4000/Laufzeitdaten**

Zur Laufzeit sind weiterhin folgende Daten temporär zu speichern:

- Bluetooth-Verbindungsdaten mit dem OBD2-Adapter
- Netzwerk-Verbindungsdaten aller verbundenen Geräte
- Aktuell anzuzeigende Sensordaten auf den Endgeräten
- Aktuell als Fahrer eingetragener Benutzer
- Aktuelle Fahrtnummer
- Durchschnittsverbrauch
- Zurückgelegte Strecke
- Durchschnittsdauer/strecke

# 7 Produktleistungen

## 7.0.1 /PL0010/Einparkhilfe-Verzögerung

Die Verzögerung des Informationsflusses darf maximal 100 ms betragen. Dabei sind die Netzwerkverzögerungen (Bluetooth, LAN ,Internet...) zwischen den verschiedenen Geräten nicht berücksichtigt.

## 7.0.2 /PL0020/Kamerabildaktualisierung

Das Kamerabild sowie die dargestellte Fahrbahn sollen mindestens 10-mal pro Sekunde aktualisiert werden.

## 7.0.3 /PL0030/Multiple Acces

Die Verbindung von bis zu vier Personen gleichzeitig ist möglich.

## 7.0.4 /PL0040/Cache

Oft abgefragte Daten werden auf dem Server zwischengespeichert.

## 8 Benutzerschnittstelle

### 8.1 GUI

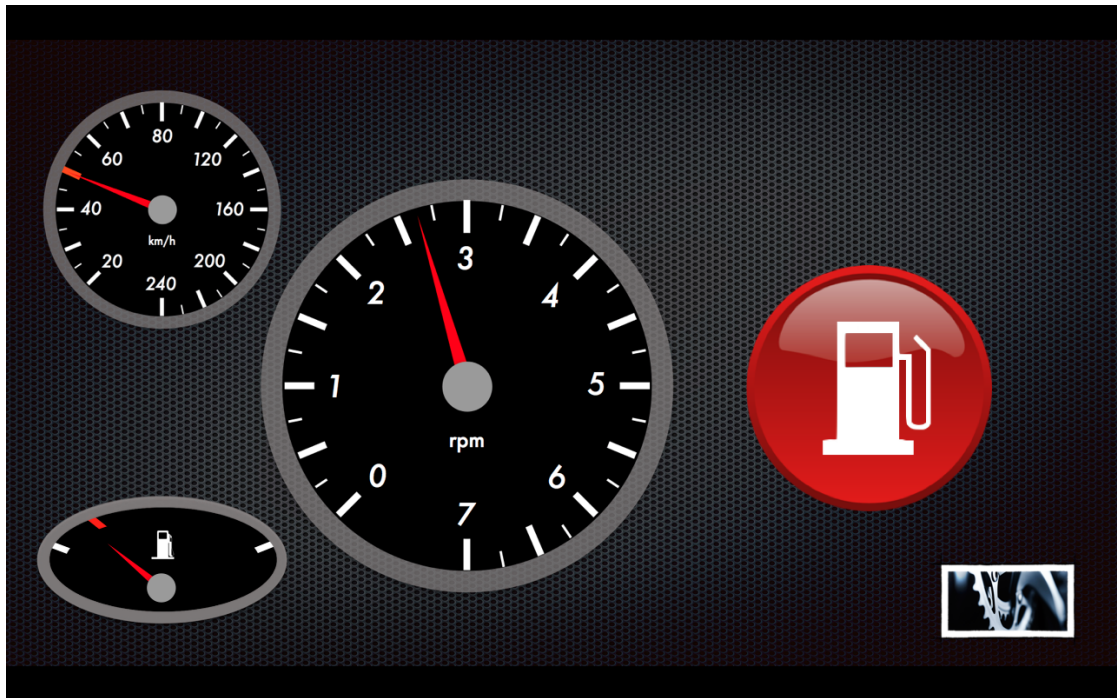


Abbildung 8.1: Das normale Dashboard.

### 8.2 Schnittstelle für Netzwerk








	Zeigt die Geschwindigkeit, die über OBD2 gesendet wird in Kilometer pro Stunde. Darstellung als Rundinstrument.	✓	▲
	<b>Instrument zur Geschwindigkeitsanzeige [mph]</b> Zeigt die Geschwindigkeit, die über OBD2 gesendet wird in US Survey Miles pro Stunde. Darstellung als Rundinstrument.		
	<b>Intsrumen für Motordrehzahl [rpm]</b> Zeigt die Geschwindigkeit, die über OBD2 gesendet wird. Darstellung im Rundinstrument	✓	
	<b>Tankfüllstand [Liter]</b> Zeigt den Tankfüllstand im klassischen Rundinstrument-Design (Oval)	✓	
	<b>Kühlmitteltemperatur [°C]</b> Zeigt die Kühlmitteltemperatur, die über den OBD2 Anschluss verfügbar ist in °C. Im klassischen Rundinstrument-Design (Oval)		
	<b>Kühlmitteltemperatur [°F]</b> Zeigt die Kühlmitteltemperatur im klassischen Rundinstrument-Design (Halbkreis)		
	<b>Öldruckanzeige</b> Zeigt den Öldruck sim klassischen Rundinstrument-Design (Halbkreis)		▼

Abbildung 8.2: Eine Tabelle

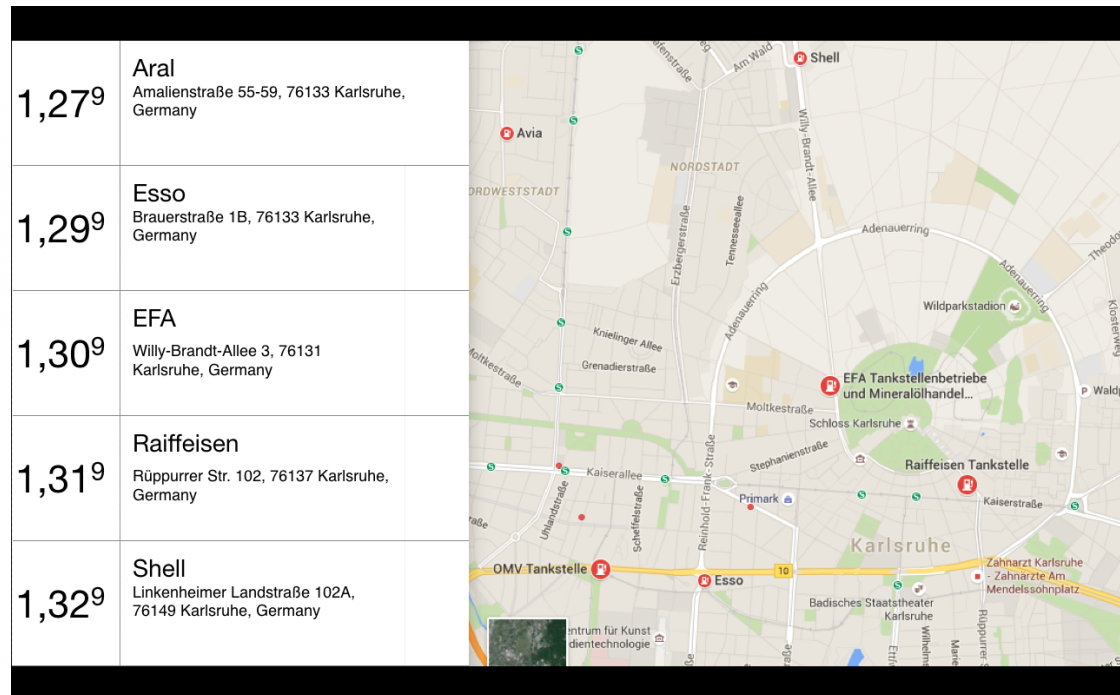


Abbildung 8.3: Die Kartenanzeige mit einer Tabelle

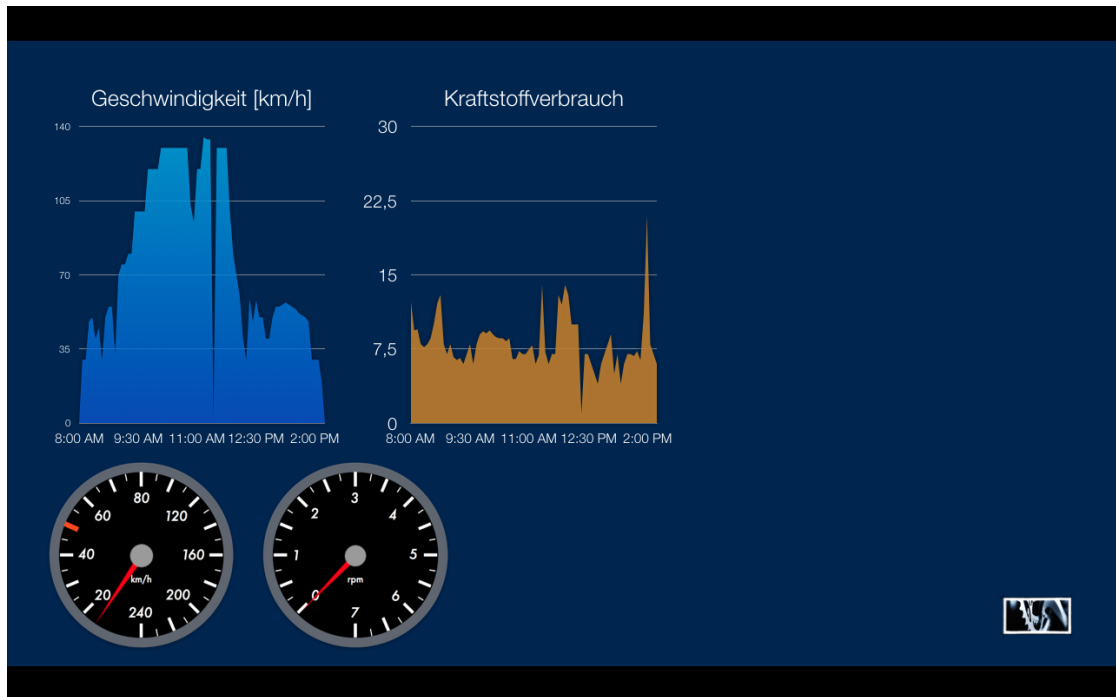


Abbildung 8.4: Das normale Dashboard.

## 9 Globale Testfälle

### 9.0.1 /T0010/Verbindung im Auto

Testfall

### 9.0.2 /T0020/RemoteAcces

Der Server ist im Betrieb. Der Benutzer verbindet sich vom entfernten Endgerät mit dem Server mit einer URL im Browser. Darauf bekommt er das Userinterface angezeigt und empfängt Sensordaten.

### 9.0.3 /T0030/Dateninput

Es wird auf Differenzen zwischen vom Fahrzeug angezeigten und auf der Software dargestellten Werten geprüft.

### 9.0.4 /T0040/Protokollierung

Es wird überprüft, ob Fahrdaten in die interne Datenbank eingepflegt werden. Die Daten werden mit manuell genommenen Vergleichswerten auf Korrektheit geprüft.

### 9.0.5 /T0050/Vergangene-Daten

Es wird die korrekte Anzeige vergangener Daten aus der Datenbank überprüft. Die Anzeige wird mit den separat aus der Datenbank entnommenen Daten verglichen.

## 9.1 Netzwerk

## 9.2 Pi im Auto

## 9.3 Endgeräte

## 10 Qualitätsbestimmung

Produktqualität	Sehr gut	Gut	Normal	Irrelevant
Funktionalität				
Angemessenheit				
Richtigkeit				
Interoperabilität				
Ordnungsmäßigkeit				
Sicherheit				
Zuverlässigkeit				
Reife				
Fehlertoleranz				
Wiederherstellbarkeit				
Benutzbarkeit				
Verständlichkeit				
Erlernbarkeit				
Bedienbarkeit				
Effizienz				
Zeitverhalten				
Verbrauchsverhalten				
Änderbarkeit				
Analysierbarkeit				
Modifizierbarkeit				
Stabilität				
Prüfbarkeit				
Übertragbarkeit				
Installierbarkeit				
Konformität				
Austauschbarkeit				



# 11 Glossar

- **Benutzer, Nutzer, User:** Jeder Nutzer der Software
- **Fahrinformationen:** Die Gesamtheit aller für den aktuellen Fahrer während der Fahrt interessanten Sensordaten.
- **Durchschnittsverbrauch:** Errechner durchschnittlicher Verbrauch an Kraftstoff pro 100 Kilometer.
- **Fahrer:** Der Benutzer, der das Fahrzeug zur Zeit fährt.
- **Endgerät:** Computer mit Bildschirm, der von einem Benutzer zur Anzeige der Sensordaten benutzt wird.
- **kritischer Sensorwert:** Sensorwert, der schwere Auswirkungen auf das zukünftige Verhalten oder die Funktion des Fahrzeugs haben kann.
- **POI:** Point of Interest; Ort in der Nähe, der (Aufgrund bestimmter Ereignisse) für den Benutzer interessant ist.
- **GUI, UI:** (Graphical) User Interface: Die grafische Benutzerschnittstelle der Software.